Track Angle Error (TAE) displays and their effect on Pilot Performance during Instrument Approaches.

by

Scott Andrew Mody Rasmussen
B.Sc., Computer Science (1991)
B.E., Aeronautical Engineering (1993)
University of Sydney

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfilment of the Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology
September 1996

Signature of Author ______________________
Department of Aeronautics and Astronautics
August 9, 1996

Certified by ______________________
Dr. Charles M. Oman
Thesis Supervisor

Accepted by ______________________
Professor Jaime Peraire
Chairman, Department Graduate Committee

Track Angle Error (TAE) displays and their effect on Pilot Performance during Instrument Approaches.

by

Scott Andrew Mody Rasmussen

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfilment of the Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

ABSTRACT

According to the FAA's Technical Standard Order (TSO) C129a, Global Positioning System (GPS) based area navigation (RNAV) devices used for non-precision approaches under Instrument Flight Rules (IFR) must be able to display Track Angle Error (TAE). The TSO requires that at least numeric TAE should be available but suggests that analog TAE may be more desirable. TAE has potential to be beneficial to pilots because it provides Cross-Track Error (XTE) derivative information for use in the tracking manual control loop. However, in traditional General Aviation (GA) aircraft, this supplementary TAE information can usually only be displayed on the front panel of the RNAV unit itself, a location which forces the pilot to widen his instrument scan. A previous study (Oman, Huntley and Rasmussen, 1995) showed that analog TAE information improves tracking performance during simulated GPS instrument approaches. The purpose of the present study was to compare tracking performance and workload using four different display formats: one showing TAE in numeric form and three which presented TAE in a variety of analog formats. Twelve pilots each flew 16 non-precision approaches in a modified Frasca 242 flight simulator. During the initial portions of the approach, when the sensitivity of the XTE display was low, use of one of the analog TAE displays (a "track vector" format) produced as much as a 20% decrease in root mean square (RMS) XTE and up to a factor of two improvement in the width of XTE envelopes (estimated 95% limits). The analog TAE formats increased both Bedford workload ratings (up to 18%) and aircraft RMS roll angle. After being artificially displaced 0.25 nm off the final approach course, pilots using the "XTE predictor" analog format re-intercepted at a consistently shallower angle. Independently of which display they were using, the subgroup of pilots with better than average inner loop roll control performance achieved superior outer loop XTE tracking. When using any of the three analog TAE formats this pilot subgroup also improved their performance as they established themselves on final approach. Ranking the displays on both XTE performance and workload ratings the "track vector" format was best. It was followed by the numeric TAE format, the XTE predictor format and finally a dual pointer XTE/TAE format. This project was supported by the Department of Transportation Contract DTRS-57-92-C-0054 TTD#27B to the MIT Center for Transportation Studies.

Thesis Supervisor: Dr. Charles M. Oman

Title: Senior Research Engineer and Senior Lecturer

ACKNOWLEDGMENTS

Formally, I wish to acknowledge the support of the US Department of Transportation and in particular the Volpe National Transportation Systems Center. This project was funded through a collaborative program between these organisations and the Massachusetts Institute of Technology Center for Transportation Studies under the supervision of Steve Huntley and Dan Hannon. These people created the idea and continued to support and nurture it to its completion. Work was conducted within the Center for Transportation Human Factors, directed by Don Sussman. The Volpe center provided not only physical support in the form of an office and the Frasca Simulator but also technical expertise in maintaining the simulator, through the programming efforts of John Bastow and Frank Sheelan, and guidance in matters of statistics, through the help of Peter Mengert and Bob DiSario. And then there was John Turner and Jack Giurleo who kept our wings level about the realities of IFR flying.

Of course, without the pilots that agreed to subject themselves to the rigorous protocol we had designed, based simply on a phone call assurance, nothing could have been done. Much gratitude goes to them for their stamina and genuine interest in the project. Also, thanks must be given to Colleen Donovan who maintained the Volpe Center's database of subjects and provided numerous fresh contacts each time the pool appeared to dry up.

On the MIT side, Chuck Oman proved an outstanding supervisor making me feel as if I really was the expert in this field but all the time knowing what was coming around the next corner and providing plenty of warning. It was Chuck's vision which kept this project alive even when he was distracted by astronauts, other students and things generally Space-y. The numerous drafts and edits he returned to me even while zipping twice around the world were invaluable. Also, Alan Natapoff added his own flavour giving further assistance with matters statistical and just simply listening to the ravings of someone trying desperately to fathom the enormous experiment that was the heart of this thesis.

Personally, it was Beth Henderson that kept me sane, clothed and fed me and generally provided that intangible assistance which no-one can measure. Together with the friends I made in Building 6 and at the Man Vehicle Lab and along with Dava Newman, my academic adviser, these people were my support network substituting for my family who were almost 14000 miles away.

CONTENTS

FIGURES

TABLES

LIST OF ABBREVIATIONS

ADF	Automatic Direction Finder
AGI	Aircraft Ground Instructor
AGII	Aircraft Ground Instructor Instruments
AGL	Above Ground Level
ATC	Air Traffic Control
CDI	Course Deviation Indicator
CFI	Certified Flying Instructor
CFII	Certified Flying Instructor Instruments
CRT	Cathode Ray Tube
CTAF	Central Traffic Advisory Frequency
DDE	Dynamic Data Exchange
DG	Directional Gyroscope
DME	Distance Measuring Equipment
DTK	Desired Track
DTW	Distance to Waypoint
ETA	Estimated Time to Arrival
ETW	Estimated Time to Waypoint
FAA	Federal Aviation Administration
FAF	Final Approach Fix
fpm	feet per minute
GA	General Aviation
GAE	Glide Angle Error
GPS	Global Positioning System
GS	Ground Speed
HSI	Horizontal Situation Indicator
IAF	Initial Approach Fix
IBM	International Business Machines
IF	Initial Fix
IFR	Instrument Flight Regulations
ILS	Instrument Landing System
IMC	Instrument Meteorological Conditions
knots	Nautical Miles per Hour
MAP	Missed Approach Point
MDA	Minimum Descent Altitude
MEI	Multi-Engine Instructor
MEII	Multi-Engine Instructor Instruments
mile	Nautical Miles
MSA	Minimum Sector Altitude
NDB	Non-Directional Beacon
nm	Nautical Miles
OTW	Out The Window
RAIM	Receiver Autonomous Integrity Monitoring
RMI	Remote Magnetic Indicator
RMS	Root Mean Square
RNAV	aRea NAVigation
RTCA	Radio Technical Commission for Aeronautics
SID	Standard Instrument Departure
STAR	Standard Terminal Approach Route
TACAN	Tactical Air Navigation
TAE	Track Angle Error
TCP/IP	Transfer Control Protocol/Internet Protocol
TERPS	Terminal Instrument Procedures
TSO	Technical Standard Order
VFR	Visual Flight Rules
VMC	Visual Meteorological Conditions
VNTSC	Volpe National Transportation Systems Centre
VHF	Very High Frequency
VOR	VHF Omnidirectional Range
WP	Waypoint
XAE	Cross Angle Error
XTE	Cross Track Error

1. INTRODUCTION

Navigating an aircraft is considered by some to be an art in itself. The tools available to a pilot to assist in the navigation task have been increasingly improving over the years. Satellite-based navigation systems, in particular the Global Positioning System (GPS), and a new generation of microprocessor-based cockpit avionics are revolutionising this skill and offering devices which can provided a more sophisticated measurement of the aircraft's position than has ever been available in the past. Previously, a pilot has either relied on a ground based controller equipped with a RADAR system to provide guidance during critical phases of a flight or has used various combinations of radio navigation aids to triangulate a position. Advanced systems now being placed inside modern avionics equipment give a pilot access to accurate area navigation (RNAV) information. These devices can not only tell a pilot the position of an aircraft but also the ground referenced speed and direction in which it is travelling and they are quickly becoming affordable to the average General Aviation (GA) pilot.

New regulations (TSO-C129) introduced by the Federal Aviation Administration (FAA) of the United States have allowed GPS based RNAV devices to be used as primary navigation aids while executing non-precision instrument approaches. The TSO sets forth all of the conditions and requirements that a GPS receiver must satisfy on order to be qualified to be used for navigation. This has opened the door to a whole new phase of aircraft navigation. Simultaneously, though, it introduces a number of new human factors issues which have not been previously studied in detail.

One such issue concerns the new information that these devices are capable of presenting to the pilot. Some questions that arise are: is this information really useful? does it distract the pilot? This study is focused on one particular piece of new information - Track Angle Error (TAE). This is a measure of how quickly the aircraft is flying towards or away from the desired track and could potentially be of significant benefit to a pilot. The writers of TSO-C129 recognised this fact and included paragraphs which required manufacturers to design their devices so that TAE information was available to the pilot. They specified that numeric readouts should be available but that an analog depiction may be advantageous. The question then became: is it really better to display TAE information in an analog form rather than with a numeric readout?

A preliminary study closely associated with this thesis established that TAE information is useful in some form, at least for certain portions of an instrument approach. The current study takes that idea further and, using a larger pilot population, seeks to show that analog presentations do offer advantages over a simple numeric presentation. By comparing three analog with one numeric display in a variety of approach situations it has been possible to gain a good understanding of the issues. The displays were studied in situations that required large scale manoeuvres at low display sensitivities as well as both short scale course corrections and straight-line course following at high display sensitivities.

1.1. Motivation

By using TAE information a pilot should be able to more easily determine how quickly and in what direction the Cross Track Error (XTE) needle is moving and thus more accurately follow the desired course. Doing so, however, may take too much of the pilot's attention and thus adversely increase the workload of the navigation task. The chief motivation is a desire to increase pilot performance during difficult instrument approaches. Making it easier for a pilot to maintain accurate course guidance during all phases of an approach will increase general safety and possibly allow approaches into airports that were previously closed during mildly inclement weather.

The purpose of this research project was to investigate how pilots used explicit TAE indications, both when presented digitally and also in several analog formats. The desire was to answer questions like: what format do pilots prefer and why do they prefer it? Also, to what extent does the addition of TAE allow pilots to quantitatively improve their performance during an instrument approach task, or to what extent does it allow them to reduce their workload? Perhaps there is, in fact, a cost of higher workload to achieve a higher performance.

The study was also concerned with the question of what approach phases most benefited from TAE information. It is most natural that TAE indications should give significant benefit at low display sensitivities. At these times, the TAE indication provides information on the rate of movement of the error needle when it is otherwise difficult to observe such information. At high sensitivities it is possible that TAE indications could provide warning of deviations before they occur, thus preventing dangerous situations that would otherwise develop very rapidly.

1.2. Thesis Organisation

This thesis documents an experiment developed and conducted in 1995 and early 1996. Primarily, it is this experiment that is described. Chapter Two collects together much of the background information necessary to understand the navigation equipment and regulations governing that equipment which was the basis of the experiment. Definitions of most of the terminology and acronyms used throughout this thesis can be found here. Also described is the preliminary work that proceeded the main experiment, in particular studies of the turbulence model used to simulate realistic gusty conditions and a smaller study that was conducted to gauge many of the ideas that contributed to the final display designs and experiment protocol.

Chapter Three develops the details of the TAE displays. Importantly, it is here that the designs of the various analog TAE formats used for the main experiment are discussed. Before doing so, however, this chapter discusses some general theory of aircraft navigation and manual control.

Chapter Four describes in depth the hardware and software used to execute the experiment. Included here are the simulator itself and the support computers surrounding it, the display hardware used to present information to the pilots and the code that was developed to supplement the existing systems.

The experiment, its design and how it was conducted is the subject of Chapter Five including a description of the methods that were employed to analyse the large volume of data collected. Chapter Six presents the results of that data analysis. Finally, the conclusions and overall insight gained through this study are related in Chapter Seven.

The appendices include some supplemental information that supports the major work. Appendix A describes an additional analog TAE display that was developed for this experiment but which was eventually dropped from the protocol. Appendix B provides a short descriptive history of the various advances in aircraft navigation and provides some background information which it is helpful to understand when examining any work in this field. Appendices C through G record the experiment forms, the data collected from the subjects and the computer code used to run the experiment and the analysis.

2. BACKGROUND

In order to understand the details of this experiment it is important to know a little about the process of navigating an aircraft. The terminology and acronyms used in this field are fairly extensive and require some definition and explanation, particularly for the non-pilot reader. They are also not standardised. This chapter endeavours to define some of the terms that will be used in later chapters. It is also helpful to have an understanding of some of the classic cockpit instruments currently available. Appendix B presents as a brief history of aircraft navigation to provide some perspective on the issue and is recommended reading for anyone unfamiliar with radio navigation. Regulations play a significant role in determining aircraft processes and procedures so the pertinent documents are also briefly introduced with a discussion of their impact on modern navigation techniques. Also the preliminary study conducted at the VNTSC prior to this current work under discussion has bearing on the details of the experiment and is presented to complete the supporting information necessary for a full understanding.

2.1. Cross Track Error and Track Angle Error

First it will be expedient to define all of the terms and acronyms that will be used in this thesis to describe an aircraft's position relative to a desired course. These are all pictured in Figure 2.1 which shows an aircraft and two waypoints along the intended route.

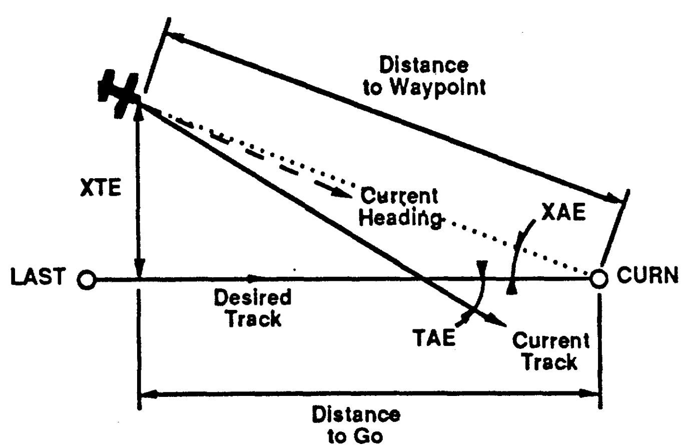


Figure 2.1: Definition of terms used to describe an aircraft's position relative to a desired course.

The route itself is defined by a series of waypoints. Each pair of waypoints gives the coordinates for the ends of a directed line segment. These waypoints are created to establish small sections of the route for which the flying task is basically constant. At each waypoint some change will likely take place in the flying task, for example, a change in direction or the beginning of a descent. The first waypoint shown in the figure is termed the 'last' (or previous) waypoint because it is the one that has just been passed. The second waypoint is the 'current' (or next) waypoint because it is the next targeted position along the route. Waypoints are commonly named by assigning them a five letter identifier and a large database of waypoints has been established by the Federal Aviation Administration of the United States (FAA) to cover the areas in their jurisdiction. (The fictitious waypoints used for this experiment have been given four letter identifiers to help prevent them from being confused with actual waypoints.) The shortest path between these two points defines the 'desired course'. It is by maintaining a position on this desired course that the pilot fundamentally navigates the aircraft. Coordinates for each waypoint are generally given in degrees latitude and longitude thus making the desired course itself referenced to the ground, although, frequently, these waypoints are derived by reference to land based navigation beacons.

Terms defining an aircraft's direction of travel with relation to the ground are identified by describing them with the word 'track'. In comparison, terms defining the direction with respect to the aircraft itself use the word 'heading'. The term Desired Track (DTK) thus refers to the compass direction parallel to the line joining the last and current waypoints. Desired Heading, however, refers to the compass direction that it is desired to point the aircraft along. An aircraft's Current Track defines the compass direction of the aircraft velocity vector relative to the ground, while the Current Heading is the compass direction of the aircraft relative to the air mass around the aircraft. These two can be different because of cross-winds. An aircraft will always travel in the direction it is pointing relative to the volume of air it is enclosed within but this air mass will move in the direction of the prevailing wind taking the aircraft with it[1]. The difference between the track and the heading caused by the wind is called the 'Crab Angle'.

Using the desired course as a reference, the aircraft's position can be indicated by various parameters that help a pilot return to the correct ground referenced path. A line joining the current aircraft position with the closest point on the desired path will be perpendicular to the desired track. The distance between these two points is termed the Cross Track Error (XTE). The distance between the closest point and the current waypoint is the along track Distance to Go. The angle between the current track and the desired track is the Track Angle Error (TAE).

A line joining the current aircraft position with the current waypoint can be used to define other distances and angles. The angle between this line and the desired track is the Cross Angle Error (XAE). The distance along this line segment is termed the straight line Distance To Waypoint (DTW). Also, given a relative wind the aircraft speed will translate into a Ground Speed (GS) that measures

[1] Considerations of sideslip angle and angle of attack are unimportant to this discussion as long as the aircraft is considered to be pointing along its wind referenced velocity vector.

how quickly the aircraft is moving over the Earth. Using the current Ground Speed and the Distance to the Current waypoint it is possible to estimate the time to the next waypoint (ETW). This is sometimes referred to as the Estimated Time to Arrival (ETA).

2.2 The Global Positioning System

It is really the advent of the Global Positioning System (GPS) that has made it possible for information such as TAE to be available to a pilot in real time. Devices which use GPS to provide course guidance have rapidly decreased in price and increased their abilities since their introduction. Appendix B describes in a little more detail the workings of GPS.

Using GPS it is potentially possible to solve the problem of creating approaches into airports that previously couldn't be accommodated using traditional navigation aids. GPS provides precise positioning at any location and allows the use of waypoints at arbitrary points. Instead of requiring fixes from at least two ground based beacons to locate a waypoint, waypoints are located by reference to the satellites. In the long term it is feasible that all the current VOR's, DME's and NDB's could be removed from service and completely replaced by a GPS system thus cutting the maintenance and running costs of these pieces of equipment. Finally, GPS provides a global system that is a common, redundant reference and thus protects against the problem of tuning the wrong frequency or two aircraft operating on different navigation aids. Of course GPS has its own problems and many of these have yet to be studied.

2.3. Instrument Approach Procedures

Although still of value and frequently used, none of the standard radio navigation instruments typically found in aircraft cockpits are formally required when it is possible to navigate by visual contact with the ground. This is considered flight under Visual Flight Rules (VFR) and it applies below specific altitudes and under Visual Meteorological Conditions (VMC). It is when this ability is lost that being able to determine an aircraft's position using instruments alone becomes paramount. Under these circumstances an aircraft is classified as flying under Instrument Flight Rules (IFR) which places special restrictions on the pilot, both to guarantee the safety of the aircraft and also to protect other aircraft flying in the same area. According to the regulations it is actually possible for two aircraft in the same area (and subject to the same local weather) to be flying under different sets of flight rules, although under this case it is the responsibility of the IFR aircraft to maintain visual separation from VFR aircraft. This is handled by the regulations and it is important only to realise that any aircraft legally classified as flying under IFR (even under VMC) must rely on instruments alone to navigate and maintain attitude.

Airports are not always going to have clear skies around them and so it is necessary to have procedures that allow aircraft under IFR to approach and land at an airport with less than perfect visibility. These are called Instrument Approach Procedures and they have associated with them

Instrument Approach Plates that document exactly what needs to be done at each stage of the approach. Each procedure is written so that sufficient safety is guaranteed for the aircraft to avoid natural and man made obstacles within the approach area. Exact definitions for the margins of safety are mandated in the Standard for Terminal Instrument Procedures (TERPS). The basic idea for all approaches is to allow aircraft flying between airports (the Enroute phase of flight) to arrive nearby the destination airport and then be channelled down to the currently active runway.

In the past all approaches have been defined by using references from local ground based navigation aids (VOR beacons, etc). Appendix B contains some brief descriptions of how these navigation aids have developed and how they operate. The inherent inaccuracies involved with a pilot using these aids to locate the aircraft mean that the procedures require large areas clear of obstacles in order to ensure the appropriate level of safety. This often makes it difficult for planners to write procedures for some airports. Without a published procedure or if the weather is worse than the minimums specified on the available procedures, an approach cannot be made to that airport or runway often causing the airport to close to IFR traffic. It is not always obstacles that cause this to happen but occasionally the lack of sufficiently well placed navigational aids prevents the establishment of a procedure for use in sufficiently poor weather conditions with the required safety margins. If a method can be implemented that opens these airports to IFR traffic then the volume of traffic at many airports could be reduced and also more secondary landing sites would be available during times when other airports are closed. Both of these outcomes will increase the safety of aviation in general. Furthermore, if safety boundaries around the current approaches can be reduced without decreasing the overall level of safety (ie. by helping aircraft to stay closer to the desired course) then the costs of creating approaches can be reduced.

2.4. TSO C-129

Technical Standard Order (TSO) C-129 [FAA95] was issued by the Federal Aviation Administration (FAA) of the United States as an amendment to RTCA/DO-208 [RTCA91] to enable the use of GPS based RNAV systems as airborne supplemental navigation devices for use with non-precision approaches under Instrument Flight Rules. It was updated to TSO-C129a in late 1995. This TSO enables GPS systems to be used as the first source of navigation information as long as a different navigation system is available to be used for navigating to an alternate airport should the GPS approach not be available. By allowing GPS to be used in this fashion the first step has been taken towards implementing a GPS dependent navigation system.

This TSO documents the regulations all such GPS receivers must satisfy. In particular, the unit must be able to autonomously monitor the integrity of the GPS network and report if there is insufficient information to provide safe guidance. This ability is called Receiver Autonomous Integrity Monitoring (RAIM) and is described in more detail in Appendix B. Also, the receiver is required to contain an updatable electronic database of the locations of all airports, VORs, NDBs and all named waypoints shown on en route charts, terminal area charts, Standard Instrument Departures (SID) and Standard Terminal Approach Routes (STAR) that are current within the area in which the device will be operated. It is this database which really contains the power of the system.

To actually provide guidance information, each device must present XTE information on a CDI located in the pilot's primary field of view. Other display requirements are also set forth in the document. Of particular relevance to this study is the requirement to calculate and display TAE.

Paragraph (a)(3)(vii) of the TSO relates:

> 2. Equipment certified to class A1, shall, in addition to the requirements for class A2:
>
> a. Provide a numeric (digital) display or electrical output of cross-track deviation to a resolution of 0.01 nm for deviations less than 1.0 nm.
>
> b. Compute and display track angle error (TAE) to the nearest one degree. Track Angle error is the difference between desired track and actual track (magnetic or true). In lieu of providing a numeric display of track angle error, non-numeric track angle error may be displayed in conjunction with the display required in paragraph (a)(3)(viii) of this TSO.

In the above paragraph from the TSO, class A equipment is defined as those devices which contain a GPS sensor integrated with a navigation capability. This is the typical unit that would be installed in aircraft that do not have data bus driven avionics systems, ie. regular GA aircraft. The class A2 devices provide only en route and terminal navigation capabilities and are not permitted to be used for executing non-precision approaches. Class A1 devices are further certified to provide navigation for non-precision approaches. The display alluded to by reference to paragraph (a)(3)(viii) is a non-numeric XTE display.

Several manufacturers are already supplying GPS receivers that have been demonstrated to comply with the class A1 TSO specifications. These TSOed receivers thus provide some means of displaying TAE information and it is important to study how this display affects pilots that are using these devices for the first time. Also, few manufacturers have followed the recommendation of the TSO which suggests that an analog display of TAE may provide the best overall tracking performance. They typically only use numeric TAE and one manufacturer has, in fact, been given permission to convert the TAE number into a related but substantially different indication. It was the purpose of this study to investigate the validity of the TSO recommendation for use of analog TAE displays.

2.5. Preliminary TAE Display Study

Reference will be made throughout the following pages to a brief study that was conducted as a preliminary look at the issues of presenting TAE information to pilots [Oman95]. This smaller study looked at five different display formats. In all, six pilots were run through the experiment protocol. The main finding of that study was that use of analog TAE displays resulted in a statistically significant improvement in performance during certain phases of the approach, without a demonstrable change in workload. Both analog and numeric TAE presentations were included but the study did not directly compare performance between numeric TAE information only and analog TAE. Inter-subject differences in performance were significant and on the same order as the display effects seen, hence one of the principal limitations of the study was that, because only six pilots were used, it was difficult to estimate the performance for a large population.

The experiment protocol for the study used two approach geometries and five display presentations. This required the subjects to attend the laboratory on two consecutive days, primarily to prevent confusion between two of the display formats, only one of which was used on each day. The two approach geometries both included a 90° turn onto the final approach heading, duplicating the procedure that is favoured for real GPS approaches and which is referred to as a "T geometry". In order to provide some manoeuvring while the displays were operating at the highest sensitivity and in a first attempt to simulate a navigational blunder, a "dog-leg" was incorporated into the descent to landing of one of the approach geometries. Pilots were required to execute two 45° turns separated by only 2 nm immediately after passing the Final Approach Fix (FAF) and starting their descent. This was referred to as the "Crooked-T geometry".

Five different display combinations were used, one of which did not include any TAE information. The other four included both numeric and analog displays of TAE. Most of the displays were similar to those used for the study which is the subject of this thesis and their definitions and descriptions are included later in Chapter Three. The actual formats used for the preliminary study included an early version of the Track Vector display, two different versions of the Triangle display and a combination which incorporated the use of the aircraft's HSI instrument to display XTE. The two different Triangle displays were identical except in the sign of the analog presentation of TAE. One was called the Triangle/Same display, the other the Triangle/Opposite display. Each of these displays was basically identical to the other except that the TAE needle on the Triangle/Opposite display moved in a sense that was the reverse of that used on the Triangle/Same display. These two displays were never used on the same experiment day to prevent the inevitable confusion that would result. To remove order effects, half of the pilots used one of the displays on the first day and the other on the second, while the other half of the pilots had this order reversed.

Primary results from the data analysis showed that pilot's preferences were in favour of the HSI combination display, which, in effect was what they were familiar with. They did, however, subjectively agree that the TAE displays were preferable to the display which only gave XTE information. Pilots reported that given the choice, they used the analog TAE data. It was found that performance suffered when the XTE display was located outside the pilot's primary field of view. A strong majority of pilots preferred the Triangle/Same display over the Triangle/Opposite display and their tracking data suggested that their performance was better using the Triangle/Same display. No significant effects of workload were found, suggesting that pilots maintained a constant workload and allowed their performance to vary when presented with the new TAE information. In terms of flight technical error performance, the Track Vector display showed the best tracking during final approach, although many pilots had difficulties with this display when executing the 90° turns. The overall conclusion was that TAE certainly helped with both the Track Vector and Triangle/Same displays showing performance improvements over a display with no TAE information in various portions of the approach. Some other things that were noticed were that TAE seemed to help pilots recover more quickly from blunders and to be able to track complex routes more easily.

Not discussed in the final report was the fact that that study was the first operational use of an upgrade to the turbulence model used within the Frasca simulator. The original turbulence model was subjectively felt to be non-representational of real turbulence and was too easy to fly. Using the NLR method [Jansen81, Moesdijk78] a new model was developed which gave a turbulence that was "patchy" in character. The preliminary study demonstrated that this turbulence better represented real world turbulence. Some of the pilots had difficulty flying the simulator and their tracks showed a more realistic tendency to wander around the desired course. Also developed for the preliminary study was a method for creating altitude dependent winds during a simulation run in the Frasca. Both of these models are discussed later in Chapter 4 and also in the report on the preliminary study [Oman95].

3. TAE DISPLAYS

For the pilot, the advent of GPS provides the opportunity to bring a very accurate measurement of aircraft position into the cockpit. This position is referenced to a standard Latitude and Longitude grid and can thus be used by an avionics device to automatically locate the aircraft relative to electronically stored geographical information - anything from simple waypoints to symbolic or even full topographic maps. GPS can also determine the aircraft's ground referenced velocity and it is because of this ability that the presentation of TAE information becomes possible.

3.1. Navigating an Aircraft

At the highest level, navigating an aircraft is identical to navigating a car or travelling of any form. The basic idea being to decide how to get from point A, the starting position, to point B, the final destination. Intermediate steps along the way can be thought of in a similar manner to that of the original problem. Much research is devoted to these processes and the cognitive tasks associated with modifying them along the route because of new hazards arising or changes in prior assumptions.

At some point the navigation task comes down to a tracking task. That is, a pilot has a desired course that is to be followed and wishes to keep as close to that course as possible. The literature on manual control mostly uses the results from simple computer generated tasks in which the subject must use a cursor to track a target. Most authors are in agreement with McRuer, et al. and the Cross-over model they developed that gives a good approximation to the transfer function humans adopt when controlling a process [McRuer65].

Tracking a desired course in a real aircraft, however, differs from laboratory tasks in at least one significant aspect - it is not possible to provide full attention to a single task of tracking one variable. A pilot is required to prioritorize all of the tasks necessary to keep the aircraft flying on course. There are tasks such as setting radios and communicating with aircraft controllers, juggling paper maps and approach charts and thinking about what tasks might be required in the near future. During the approach to land at an airport the workload is particularly high with the additional requirement that flaps need to be lowered, as does the landing gear and the ground is naturally down there and moving closer. This makes the pilot's job very much one of sampling the relevant information sources in order of their importance: firstly keeping flying and secondly arriving at the correct destination.

For many pilots this list of priorities tends to run something like: maintain attitude, maintain speed, maintain vertical speed, correct altitude, correct relative offset from desired course, deal with other problems. Obviously, these priorities are not always strictly adhered to nor is it always necessary to take action on each of them but it gives a general idea of some of the thought process. This order of priorities generally applies because of the tendency to separate the various control loops that are active and stabilise the inner ones first [McRuer67]. The aircraft yoke directly affects roll rate and pitch rate and so the aircraft attitude is only one integration removed from these inputs. Similarly, the throttle directly affects the rate of change of airspeed. Pilots are often taught to treat pitch angle as controlling airspeed and use the throttle for adjusting vertical speed, but no matter how these tasks are performed they form a set of inner loop control actions that stabilise the aircraft and keep it in the air.

Stabilising the aircraft sets a particular attitude, airspeed and vertical speed. At this point the pilot needs to know what is to be done with the aircraft. In general, the task is to maintain position relative to some desired trajectory. It then becomes the duty of the pilot to select an appropriate attitude and vertical speed to maintain the current position or fly back to the desired course. Altitude tends to come first in the list because it is only two integration steps away from the control inputs available to affect it. Position, however, requires three integrations from the pilots yoke control of roll rate to the control variable of cross track error (XTE). A triple integral plant is something that most people cannot control. The task is often made more difficult because it is hard to estimate the rate of change of XTE from a low sensitivity presentation. Fortunately, the flying task itself is made easier by providing pilots with displays of intermediate state variables that reduce the problem to one of controlling a number of nested single integration loops - still difficult but not impossible.

To be even able to control XTE, some measurement of it needs to be presented to the pilot. Classical navigation displays rely on angular information derived from VOR's. Doing so means that deviations from the desired course appear as angular measures on the HSI instrument. For this reason routes have often be defined as 'follow 294 radial from Delta VOR'. Navigating was a matter of picking up one VOR radial, following it for a set time based on an estimated ground speed and then turning to intercept a second VOR radial, etc. until the destination was reached. An RNAV system, on the other hand, can map out any arbitrary trajectory and present a display to the pilot showing the deviation from the current course. It can read the deviations from its navigation sources and convert them to a linear measure of XTE.

3.2. The Advantage of TAE Information

One common strategy taught to pilots to control XTE (or XAE as would appear on an ILS or HSI display) is to establish some wings level orientation relative to the desired course and then watch the XTE needle. By noting in which direction it is moving and how quickly it is doing so, suitable corrections to the aircraft attitude can be made so as to modify the current heading in a way which brings the aircraft closer to the desired position. Once this new configuration has been established the XTE needle can be checked again and further corrections made until the aircraft has been moved to the desired location relative to the course. It then becomes necessary to maintain that position. Thus the pilots use attitude to select appropriate headings which move the aircraft in such a way as to bring the XTE closer to the desired value (usually zero).

Initially, many pilots will point the aircraft in the direction of the desired track, even though they are still off-course, and then watch the XTE needle to see if it moves. In the presence of cross winds it will indeed move and then the pilot must correct the aircraft's heading until the heading which stops the needle movement has been found. Some pilots call this the "freeze heading". Each time a correction is made it is necessary to watch the XTE indication to determine whether or not it is moving. It is likely that this heading will not change dramatically over short periods during a flight, thus making it a little easier to find the correct heading the next time. During the process of finding this heading the aircraft will most likely drift away from the desired course requiring further corrections to the aircraft position. It is easy to see how a poor strategy could make this task very difficult and, in the presence of other distractions, particularly approaching waypoints and changes in the desired course, quite demanding.

An abstracted version of the control loops involved is shown in Figure 3.1. The inner most loop, controlling roll angle, is fairly simple for a trained pilot to control. Only one integration is involved and the attitude indicator provides a direct and rapid readout of the controlled variable. Pilots are also taught to concentrate on the attitude indicator, scanning it very regularly and often. This all means that the eigenvalues of the control loop can be fairly easily manipulated so that they provide an appropriately fast and stable response to desired roll angle inputs. Around this loop the pilot needs to further control the aircraft heading. With a good quality RMI the pilot has an effective readout of the current heading and can use his command of the roll angle loop to close the path around the heading loop, thus commanding roll angles to direct the aircraft heading to the desired value. All of this can be thought of as the "Inner Loop" control of the aircraft because it is concerned only with references located within the local wind reference frame. Stabilising these two loops allows the pilot to command the aircraft to any orientation relative to the air mass directly about the aircraft. Making a quick linearised analysis of these two loops, assuming that the pilot can be modelled as a gain and first order lag [Clement67] associated with reading the displays and making control inputs, indicates that the combination of these two loops will produce a system that has four poles with one in particular located fairly close to the origin. This pole is primarily associated with the heading loop.

The third loop in Figure 3.1 represents the control of XTE. This can be thought of as the "Outer Loop" - here external reference to the ground and the desired course is required to define the navigation parameters involved. In the linearised analysis this loop adds one more pole at the origin associated with the integration of TAE. The combination of this pole and the reasonably slow moving pole from the Inner Loop makes it difficult to shift the poles away from the origin and stabilise the closed loop. For this reason pilots can no longer simply act as a proportional controller, it becomes necessary to add lead to this Outer Loop, that is try to estimate the derivative of XTE, so that the poles can be drawn away from the origin and positioned so as to provide a reasonable response to commanded XTE inputs. It is really this lead generation that is difficult to achieve, particularly when using low sensitivity displays that require relatively long delays to read the rate of movement of the needle. These delays move the Inner Loop poles closer to the origin making it even more necessary to determine an accurate estimate of the XTE derivative.

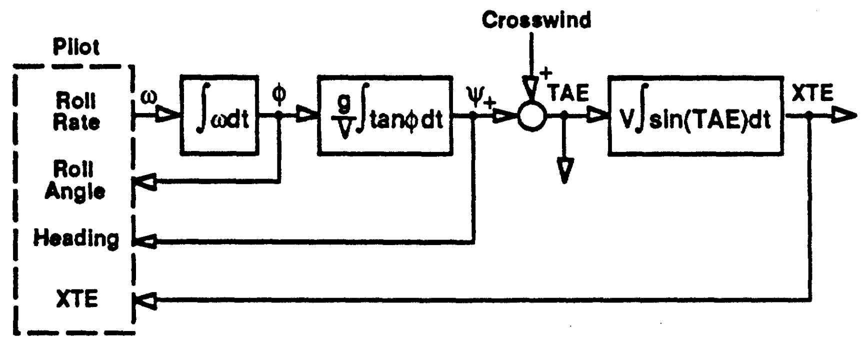
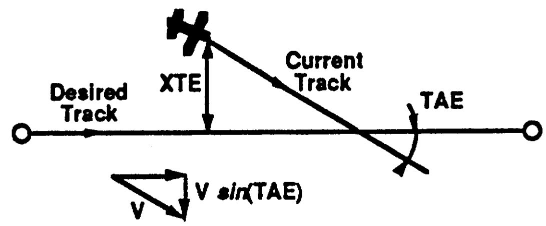


Figure 3.1: Control Loops involved with maintaining a desired XTE

RNAV systems already assist in the task of reducing XTE by providing a direct linear measure of that variable. They can, however, help further. Using their measurement of the aircraft's position and differencing this over time it is possible to estimate the current ground track. GPS systems can actually determine this estimate a little more directly. The current ground track can then be used to calculate TAE. As shown in Figure 3.2, TAE is approximately proportional to the derivative of XTE and is thus a direct measure of how quickly the XTE needle is moving and in what direction it will move. Assuming that more information makes the pilot's task easier, providing a feedback of TAE should help in the process of tracking the desired course.

This is the chief advantage of presenting TAE information to the pilot. By providing an indication which is directly related to the derivative of XTE the pilot is given a direct readout of the variable he needs to estimate in order to provide the necessary Outer Loop lead. This reduces his task back to one of simply providing a proportional controller.

Another advantage of a TAE measure is that, when combined with the current heading, an estimate of the crosswind disturbance can easily be determined. Applying this information to the tracking strategy described previously it becomes possible to immediately determine the desired heading that will stop the XTE needle moving. This should eliminate the cut and try process that is normally required. In fact, the TAE indication could, potentially, give a constant readout of a changing crosswind and the corrections required to update the desired heading to maintain the desired track.



$$\frac{d(XTE)}{dt} = V\sin(TAE)$$
$$\approx V \times TAE$$
$$where\ V = ground\ track\ velocity$$

Figure 3.2: Relationship between XTE and TAE.

3.3. Predictor Displays

Providing TAE information directly to the pilot accomplishes what the manual control literature refers to as providing lead. This is because of the predictive nature of such information, ie. it tells the pilot something about where the system is going before it actually goes there. Other methods have been devised to try to use this type of lead information to aid the flying task. In general they are called Predictor Displays. Formats that have been investigated range from simple first order quickened displays to full contact analog devices. Although not strictly a predictor display, a quickened display adds the first order derivative information onto the error signal thus making the needle move 'more quickly' in the direction it is changing thus providing some, albeit hidden, information about what is going to happen. A simple predictor display shows two needles, one which is the error signal itself and the other which is a future prediction of what the error signal will be. A contact analog display presents an artificial representation of the real world that can be overlaid with a real world view. Predictor information can be overlaid onto such a display to indicate what the view will be like in the future or to indicate where the aircraft will be in the future. Kelley presented some of the earliest information available on using predictor displays to aid manual control systems [Kelley60]. Kelley also developed the idea of using a tunnel, or series of predictions to achieve a type of road display to show either where to go or where the vehicle is going. Gallaher and Grunwald formalised the predictor idea by presenting mathematics for computing appropriate information in aircraft displays [Gallaher77, Grunwald85]. Predictor information has been shown to be basically helpful to a pilot and in general offers significant help in the control of high order plants (ie. with more than two integration steps) and systems with long delays. With any human controlled system, however, it is not usually possible to predict the control actions that are going to be made and thus any prediction information is limited in its range of validity. Most often some simple assumption about the future control inputs is used in the prediction calculation. Common assumptions include assuming the controls will remain in their current positions, assuming the pilot will return the controls to their neutral positions over the time of the prediction or assuming the controls will be modified to maintain the current rate of change of some or all state variables.

Typically, predictor information relies on the current (and past) states of the controlled system to project the state of that system into the future. When examining aircraft systems, variables are most frequently only referenced to a simple aircraft and wind model that ignores the fixed navigational reference. This simplifies the solutions and provides much useful information to a pilot performing air manoeuvres but ignores the fact that at some point reference needs to be made to the ground. While navigating an aircraft it is impossible to ignore this fact because the desire is to travel from one point on the ground to another. Contact analog displays address this issue by keeping track of where the aircraft is currently located and then superimposing the aircraft-axis related predictive information over the current view of the world. This has the effect of relating the predictor information to the ground without requiring direct calculations, it does, however, ignore the future effects of disturbances such as a persistent crosswind.

RNAV systems and particularly GPS RNAV bring to the cockpit the ability to cheaply and quickly provide ground track information to the pilot or an avionics system. This provides the potential to predict future ground track based on past collected information. The simplest form of this prediction is the value of TAE.

3.4. Analog TAE Formats

As already mentioned, the two advantages peculiar to TAE information are that it is closely related to the derivative of XTE and that the sum with heading provides a direct readout of crosswind. It is these pieces of information which should be highlighted in a display of TAE information.

For this study a number of constraints were also imposed to keep the displays within the realm of General Aviation (GA) avionics systems. All of these displays employed a simple linear CDI scale showing XTE. This was done because the TSO requires that a readout of XTE be always available. Three analog TAE display formats were used. They each highlighted a different aspect or method of presenting TAE information. Each was given a name which describes its primary aspect. These were 1) Triangle Same[2] , 2) Track Vector and 3) Predictor[3] . A fourth format, termed an Electronic HSI[4],

---

[2] The Triangle Same display was originally proposed by George Lydanne, FAA National Resource Specialist for flight management systems and who originally suggested this study.

[3] It is not really possible to attribute the Track Vector and Predictor displays to any particular source, although the actual displays used here were proposed and implemented by the investigators, namely Charles Oman and the author.

[4] The Electronic HSI display was originally proposed and developed by the author.

was also considered but dropped from the final experiment protocol. It is described in more detail in Appendix A. Many of these displays were originally used in the preliminary study. Small modifications were implemented for this experiment in light of the results of that study.

Obviously there are many other methods of displaying TAE information. One that has been implemented in all GARMIN GPS receivers is a value called Course to Steer which provides a track to follow that will bring the aircraft back to the desired track. This value automatically combines TAE and XTE information to provide a trajectory that will re-intercept and maintain the desired course. GARMIN refers to it as an "optimal" trajectory.

The basic precept for all of the formats is to present TAE information in a way that allows the pilot to quickly and most easily use that information to help guide the aircraft. This was necessarily to be achieved in as little space as possible. Obviously, it was important to keep all the formats as simple as possible to prevent overloading the pilot with too much extraneous information.

3.4.1. Specification Constraints

The overriding constraint for all of these displays was that they should be small enough to be easily retrofitted into a GA aircraft at relatively low cost. This criterion was based on the desire to make these displays useful and give them the potential to be implemented in real avionics. In typical GA aircraft the simplest method to add new equipment is to mount it in the main avionics rack. Since a GPS device would need to be added in this manner anyway, it was decided to restrict the displays to fit on the front of such a device. Frontal area available for this purpose is typically 6" wide and 3" high. Although the height can be arbitrarily designed, 3" was chosen because it also allows the possibility of locating the display within the primary field of view by replacing a standard instrument. The instruments within the primary set very commonly have a diameter of 3".

Of course, a GPS receiver needs to display numerous other pieces of information and must have some method of inputting commands and data. Thus not all of the space on the front of a 3" high GPS device will be available. Also, to cut costs, GPS manufacturers are going to use a display with the minimally required resolution. In fact, one manufacturer recently proposed a device with only a single line character display. These factors suggested that the displays needed to be very simple and able to be represented with a minimum of elements at a very low resolution.

Finally, if mounted in the avionics rack, the display would be out of the primary field of view. This location means that pilots would need to take their eyes from the primary instruments to look at their navigation information. Used in this way, as a secondary display, it is necessary that the information content be able to be quickly evaluated and easily interpreted so that concentration can return to the primary instruments.

3.4.2. Numeric Display of TAE

The simplest method of displaying TAE information is to present it numerically. It is not necessary to display this figure to a greater accuracy than one degree as changes of several degrees can occur within the one second period between GPS updates. (Also TSO C129 only requires a one degree accuracy.) The most difficult element of this type of display is showing the sign of the value, ie. in which direction is it pointing. A simple +/- indication does not readily correspond to any physical situation and has no mnemonic value. For this study the decision was made to follow the lead of the designer of the NorthStar M3 GPS receiver [Bennett95]. This implementation displays a caret (<,>) pointing in an appropriate direction. In this case, if the TAE is such that the aircraft is moving to the right of the desired course then a left caret (<) is presented. The reasoning behind this was that in such a situation the XTE needle indication will be moving to the left and thus the caret will remind the pilot of this. Also, in this same situation, in order to reduce the TAE to zero it is necessary to bank to the left. A useful rule-of-thumb then is that the caret points in the direction of the bank required to null the error.

Subjective comments from the pilots in the preliminary study suggested that any display should provide the ability assess the intercept condition at a glance to minimise the time spent scanning this display. They also found it worthwhile to be given useful rules-of-thumb by which to remember how to interpret each display and when it was telling them they were diverging or converging.

For the numeric TAE display, combining the caret indications with the position of the XTE needle it is possible to determine the intercept condition. Namely, if the needle is moving towards the centre of the scale then the aircraft is tracking towards the desired course. Also, once the needle reaches the centre, a bank in the direction of the arrow will reduce the TAE to zero and thus cause the aircraft to intercept the correct desired ground track.

3.4.3. Triangle Same

The "Triangle Same" or “T” display is perhaps the simplest of the analog TAE displays. The idea behind this display was to present TAE in a straight-forward manner. To this end, it shows TAE directly by superimposing a TAE CDI scale beneath the XTE scale. This second scale represents ±90° TAE and uses a triangle as the needle. The sign of the TAE is represented by placing the triangle on the same side as the XTE needle if the course is converging, or on the opposite side if the course is diverging. It is this correspondence which gives the display its name. That is a Triangle <u>Same</u> display is flown so that the triangle is on the <u>same</u> side as the XTE needle when intercepting the course. Also, it is necessary to bank in the <u>same</u> direction as you want the triangle to move. This means that while the XTE needle is a "fly-to" display, the TAE needle is a "fly-from" display.

Another possibility is to reverse the sign of TAE so that both indications become "fly-to" displays. Human Factors design principles would suggest this to be the most appropriate arrangement because of the compatibility of the two displays [Wickens92]. In the preliminary study, however, this configuration was shown to be poorly regarded by pilots and it demonstrated marginal decreases in performance over the Triangle Same format. Hence, such a format was not used in this study.

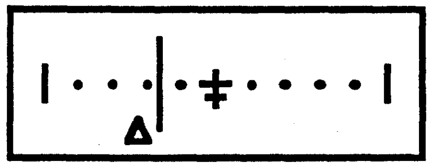


Figure 3.3: Triangle Same Format

The Triangle Same format highlights the TAE property of providing information on crosswinds by focusing attention on the zero TAE point. It is this point that gives the readout of current crosswinds. Pilot strategy when using this display is to turn the aircraft until the TAE needle reads zero. At this point the aircraft will track parallel to the Desired Track and by referencing the heading indicator the pilot can now determine the Desired Heading, the heading which corrects for crosswinds. Crosswinds are unlikely to change very drastically over a few minutes and thus the pilot can note this heading and use it as a reference on the primary instruments for maintaining a track parallel to the desired track. This strategy also demonstrates the ease with which the display can be used as a secondary instrument. A pilot need only refer to it occasionally to obtain updates on the Desired Heading and XTE.

Representing this display in a low resolution format is also extremely simple. The TAE needle could be any small symbol that distinguished it from the XTE needle. In a one line character display a half height triangle would work perfectly well. Another possibility is to represent it as a dot or group of four or five pixels moving along the bottom of the XTE scale.

3.4.4. Track Vector

The "Track Vector" or "V" display represents TAE by tilting the XTE needle by an angle equal to the TAE angle. Small vertical marks maintain the indication of XTE at large TAE angles. These marks also help to determine the zero TAE position. An arrow head on the tilted line indicates the direction of the desired track and allows the possibility of flying backwards along the course without any confusion. Also, when the XTE indication is off scale, the vector will continue to rotate with TAE so that it is still possible to tell how to make a correct intercept.

If the precise geometry of the situation is ignored, it is possible to see this display as a kind of mini-moving map with the vector representing the desired track as it would appear looking down through a narrow slot perpendicular to the current track direction of the aircraft. During the pre-briefing sessions of the experiment it was specifically suggested to the pilots that they might like to try to view the display in this way. The problem with the geometry arises because at large TAE angles the track should appear ahead of the aircraft but instead it is depicted off to the side at a distance that is measured perpendicularly to the desired track and not perpendicularly to the current track of the aircraft. Despite these problems, this display represents a very simple way of combining the two pieces of information in a way that pilots nonetheless found to be intuitive. Extracting the combination of TAE and XTE information from this display will hopefully be similar to examining a moving map. Some of the pilots from the preliminary study agreed with this assessment. However, not all did and it was partly due to their comments that the surrounding box was added and the vector extended to the edges of this box.

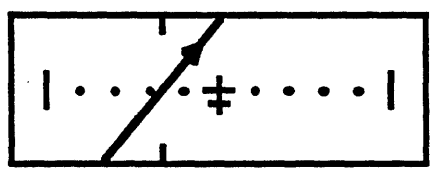


Figure 3.4 Track Vector Format

The Track Vector format highlights the TAE property of providing information on changes in XTE. The arrow head of the vector provides some measure of where the XTE needle is going and the combined needle will always move in the direction it is tilted. In this way the strategy for nulling the XTE is to fly towards the vector until it tips over to point at the centre aircraft symbol, wait for the needle to come in and then bank the aircraft so as to straighten the needle again. Of course, any time the vector is vertically aligned the aircraft will be tracking parallel to the desired course. When this is the case, the current heading can be noted and used as a reference for the desired heading just as with the Triangle Same display.

An earlier version of this display lacked the small vertical marks above and below the vector and the vector did not extend to the edges of the display box. In fact, this earlier version did not include the box nor the central aircraft symbol. These features were added after the preliminary study to combat the feeling that some pilots had of becoming confused at times with this display and to make it appear more map-like. This confusion was particularly apparent prior to the implementation of turn following, a computational procedure which causes the display to track a circular arc around the corner. (Turn following is discussed further in Chapter 4.) During a 90° turn the vector instantaneously flipped from tracking one leg to tracking the other leg and was difficult to follow.

Although implementing a lower resolution version of this display would be more difficult than for the Triangle Same, it should still be relatively easy to accomplish. The moving element can be represented by a simple line and the arrow head and XTE vertical marks may not be necessary. The biggest problem would be the resolution available for representing very small TAE values. This could potentially make it difficult to maintain a parallel course without reference to a digital readout.

3.4.5. Predictor

The "Predictor" or "P" display uses TAE to determine a prediction of what the XTE value would be in 15 seconds. This display shows future XTE on the same scale as the current XTE indication. Future XTE based on a first order prediction, simply extending the estimated current ground track ahead of the aircraft using a filtered version of the ground speed as calculated from finite differencing and averaging the previous five sample points. A dashed line represents the prediction. Different prediction times were tried and subjectively assessed to be either too jittery at the high resolution or not sensitive enough. Fifteen seconds represents 0.5 nm when travelling at a ground speed of 120 knots (the nominal approach speed used during the experiment). Fifteen seconds also represents a time constant on the order of the lateral modes of the aircraft model used. Subjective trials prior to the experiment suggested this would be a good value. Later questioning of the pilots after the experiment showed that almost all found the prediction needle to be neither too sensitive nor too sluggish.

The prediction needle was set up from and made shorter than the XTE needle to provide a sense of perspective in the display so that pilots could visualise it as being behind the XTE needle and thus remind them that the value being indicated was something in the future, ahead of the aircraft. Also, by joining the end points of the two lines in a mental picture it is possible to envisage the display as a kind of wall in 3D stretching along the desired track. This makes it appear like a horizontally viewed depiction of the route to be followed. Diagrams explaining this visualisation technique were shown to all of the pilots in the experiment and it was suggested that they might like to try using it while they were flying. These diagrams were part of the pilot brief which is included in Appendix C.

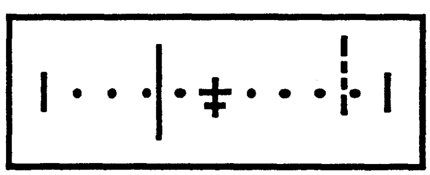


Figure 3.5: Predictor Format

The Predictor format uses the traditional notion of a prediction to assist in controlling a high order plant. Based on previous research into such predictor displays it was expected that this format, at least, should offer some assistance to pilots. Two strategies can be adopted when flying with this display. The simplest is to fly towards the prediction indication. With that needle centred, it is a sure thing that the main XTE needle will eventually return to centre. In fact the prediction will move away from centre a little before the main needle does, thus giving some prior warning that a deviation will soon develop. The second is to use the 3D wall idea to develop a mental image of the location of the desired course and then to fly up to and along the wall. This display requires the ability to distinguish the two needles, although the 3D aspects could be removed for display in low resolution.

4. EXPERIMENTAL APPARATUS

This chapter presents a top level description of all the hardware and software that made it possible to conduct each experiment run. In particular, included here are the names of manufacturers and model numbers of products used, as well as descriptions of their size and relative positioning. For the human factors engineer this information provides the details of visual angles, resolutions and arc separations that are necessary to give a complete description of the methods. Some details of the software coding are included to provide insight into the workings of the underlying processes that provided information and disturbances to the pilots while they were flying. This section is presented for the interested reader but is not necessarily crucial to the overall understanding.

4.1. Frasca Simulator

This experiment was conducted wholly within the Frasca Simulator located at the Volpe National Transportation Systems Centre (VNTSC). This simulator is a model 242 and was installed late in 1993. In provides a high fidelity simulation of a multi-engine aircraft cockpit, complete with real cockpit instruments where possible. It is a full instrument flight training device identical to those used by many flight schools for simulated instrument training. The lack of a motion base and visual system mean that it cannot be used for full visual flight training. As described later, a crude Out-The-Window (OTW) display was added to augment the decision processes of the test subjects.

4.1.1. Turbulence

The turbulence model included in this simulator was developed at the VNTSC and incorporated by the manufacturer into the simulator software. It was later added as a standard feature on all of their 242 simulators. This system allows the possibility of applying turbulence disturbances to four channels: roll, pitch, yaw and vertical speed. During the experiment the yaw and vertical speed turbulence was turned off with disturbances being added only to the roll and pitch channels. A full description of the model used to calculate the disturbance for each channel is included in the final report for the preliminary study [Oman95]. The basic principle behind the model is that each channel is the product of two filtered gaussian white noise processes which is then added to a third filtered gaussian process. This provides a 'patchy' disturbance that tends to increase for brief intervals that are randomly distributed over time. The time constants of the various processes are set so that the characteristic patch length was 2500 ft. At an airspeed of 120 knots this represents a 12.5 sec duration. This patchiness means that it is possible to have a period of relative quiet and then be hit by a strong gust. Such a disturbance profile requires a reasonably high sampling rate of the primary instruments, approximately 0.3 to 1 Hz, in order to maintain a desired attitude. If this scan rate is not achieved it is very easy to not notice the aircraft significantly deviate from the intended trajectory.

A "patchiness intensity factor" entered by the user controls the level of disturbance on all channels. For all of the experimental runs, this constant was set to 4.3. In the preliminary study a level of 5.0 was used, but this was perceived as unrealistically strong by most pilots. In comparison, in this study, most pilots found the turbulence challenging but few said it was too high. One pilot commented that he did not notice the turbulence. It was generally felt that the turbulence level was moderate to severe, perhaps like that associated with flying in the vicinity of a thunderstorm and a number of pilots commented that they would not normally continue to fly approaches under these conditions, although they were confident they could execute a safe landing if required.

4.2. Additional Computers

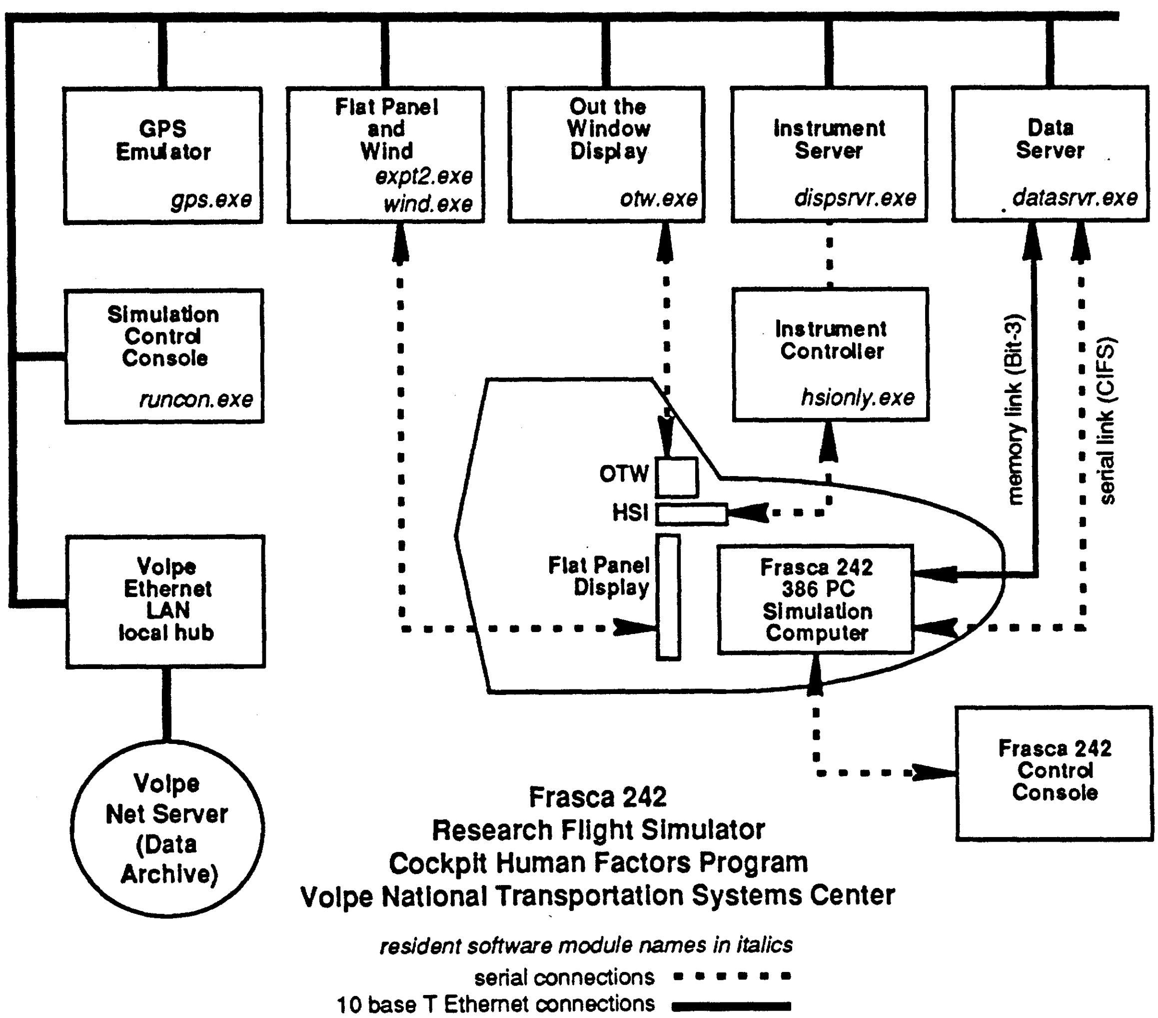


Figure 4.1: Arrangement of computers supporting the Frasca 242 Simulator.

In order to present additional displays to the pilot and to record the simulator variables, seven additional computers were interfaced to the Frasca and networked together. These computers were all Wintel style machines and generally each controlled a single aspect of the additional tasks required by the complete system. Chiefly, these tasks were: data collection from and command control to the Frasca, data storage and distribution, GPS calculations, simulated GPS receiver presentation, OTW presentation, low level servo control of certain instruments and calculation of altitude adjusted wind conditions. The wind calculations and presentation of the simulated GPS receiver were performed on a single Pentium class computer running Windows 95. The background calculations required for the GPS receiver were performed on a similar machine. Low level servo control was necessary only to remove a failure indication flag in the HSI. Doing so, however, required the use of two computers, one 286 class, the other a 386 class. The other computers were all 486 class computers running Windows for Workgroups 3.11.

These computers were all connected to each other with an Ethernet network using both TCP/IP and NetDDE connections. This network also connected them to the central VNTSC servers which were used to store a common database containing information on the approach profiles and to record the data after each experiment run. Figure 4.1 shows a schematic of the total system.

4.2.1. Simulation of Wind conditions

$$U = U_0 + \left(U_\delta - U_0\right)\left(b/\delta\right)^\alpha$$

$$\text{where } U = \text{wind velocity at h [ft / sec]}$$

$$\alpha = \begin{cases} 0.0072 \times R + 0.14 & 0 \le b \le \delta \\ 0.35 & b > \delta \end{cases}$$

$$R = \text{ground roughness height [ft]}$$

$$\delta = \text{boundary layer thickness [ft]}$$

$$U_0 = \text{wind velocity at ground level [ft/ sec]}$$

$$U_\delta = \text{wind velocity at height } \delta \text{ [ft / sec]}$$

$$b = \text{altitude above the ground [ft]}$$

Figure 4.2: Equation relating wind speed to altitude.

While the Frasca has a built-in turbulence model, the overall level of wind is set at a constant value in terms of direction and speed, neither of which vary with altitude. To add a little more realism to this and to enable programmed control of the wind direction, a module, that had been used in the previous study, was again employed to implement an altitude dependent wind velocity profile. For each run the wind direction was kept constant but was modified automatically between runs by information in the common database of approach profiles. The altitude dependence of the wind speed was characterised by four parameters: 1) ground level wind speed, 2) wind speed at the top of the boundary layer, 3) boundary layer thickness and 4) ground roughness. The wind speed at any altitude above the ground was then given by a power law with a variable exponent. Above the boundary layer this exponent was 0.35. Inside the boundary layer this exponent was a linear function of the ground roughness [Etkin80]. The exact equation used is given in Figure 4.2.

4.2.2. XTE and TAE Calculations

One of the supplementary computers was assigned to calculating all of the ground referenced variables. This was intended to simulate the operation of a GPS tracking engine and was thus called the GPS module. The variables calculated here were: GS, DTK, DTW, Ground Track, XAE, XTE, TAE and ETW. The module obtained information from the database of approach procedures to enable these calculations to be performed.

Current ground track was determined by calculating the difference between the current and previously sampled positions of the aircraft. TAE was then simply the difference between the current track and the desired track. The distance travelled in this time was also used to calculate the total distanced travelled. Ground Speed (GS) was calculated by recording the previous five time samples, totalling the time and distance travelled up to the current point and dividing them.

The difference between the current aircraft position and the current waypoint gave DTW and the direction to the next waypoint. The difference between this direction and the desired track (XAE) together with DTW was then used to calculate XTE and the along track distance to go. Finally, DTW and GS were used to calculate ETW.

Altitude related deviations were also calculated in this module in a very similar manner but they were not displayed to the pilot.

4.2.3. Automatic Turn Following

The most complex part of the GPS module was calculating the turn following. This procedure took two legs of the route and fitted a circle, that was tangential to both legs, inside the angle between them. The radius of the circle was chosen to provide a standard rate turn around the corner. Standard rate turns are defined to cover 360° in two minutes. The Ground Speed was used to determine the correct value to convert the distances to the time required. This dependence meant that the radius could change as the turn was being executed, particularly as the pilot turned into the wind while keeping a constant airspeed. Using this circle, TAE and XTE values were recalculated so that XTE was always measured perpendicularly to the circle. The Desired Track (DTK) was also calculated so that it updated as the aircraft flew around the turn. Along track distances did not account for the shorter distance travelled by following the circular spline.

The same procedure was also used to implement DME arc following. In this case the radius was set as a constant depending on the radius of the arc to be followed. Implementing the DME arcs in this way required the insertion of a dummy waypoint to define two tangential lines to position the arc. This waypoint was hidden from the pilot. Distances were adjusted so as to be always measured from the waypoint at the end of the arc. Along track distances were measured around the DME circle.

4.2.4. Data Recording

One of the primary tasks of the supplementary computers was to collect and recorded the state of the simulator at regular intervals during each experiment run. Data was collected at roughly 1 Hz by recording it directly to memory while each run was in progress and then storing it to disk at the completion of each run.

4.2.4.1. Sampling Rate

The sampling rate achieved was far less than ideal. On average one sample was recorded every 1.1 seconds with a standard deviation of 0.1 seconds, although the distribution of sample intervals was not gaussian. The actual samples were restricted to 55 msec (one tick) boundaries by the Windows clock making the median sample interval equal to 20 ticks. For any single run the minimum deviation below this median was about 7 ticks, whereas the maximum deviation above was generally about 12 ticks. Over all of the runs the maximum interval was 70 ticks (3.85 secs) and the minimum was 1 tick (0.055 secs). Because of this low and variable sampling rate time-course statistics were not considered when analysing the data. Instead all process statistics were referenced to the along track distance.

Inner loop control activity occurs at frequencies faster than a 1 Hz sampling rate can record, making it difficult to obtain direct relationships between pilot inputs to the aircraft and the aircraft output. Nevertheless, the gross features of XTE and aircraft position in general are fairly accurately recorded by this sampling. As for aircraft attitude, it is likely that the samples are sufficiently well spread to be independent and thus still maintain the statistical characteristics of the original random process.

4.2.4.2. Data Recorded

At each sample time a total of 35 variables were recorded. The full list is shown in Table 4.1. Most of these variables were directly retrieved from the Frasca and considered to be original data. All of the rest are calculations based on these original data. Unfortunately, no attempt was made during the recording to match the calculated variables with the appropriate data they depended upon. The delay required to make the calculations meant that this information was out of synchronisation with the original data by about one sample. Therefore, within the file, all variables retrieved directly from the Frasca are about one sample ahead of the calculated variables recorded at the same sample time.

Variable Name		Description	Variable Name		Description
RunCon Time		Time in seconds this sample was recorded. First sample is zero.	Altitude Error		Altitude Error in feet.
DataSrvr Time		Tick counts when this sample was retrieved from the Frasca (in msec)	Current WP		Five letter identifier of the waypoint that is the current target.
Latitude	•	Aircraft position in degrees latitude. Positive is West	CurWP Type		Symbol for the current waypoint. (eg, FAF, MAP, etc)
Longitude	•	Aircraft position in degrees longitude. Positive is North	Dist To Waypoint		Straight line distance to the current waypoint.
Altitude	•	Aircraft barometric altitude in feet.	Heading To Waypoint		Ground referenced direction to the current waypoint in degrees true.
Airspeed	•	Aircraft airspeed in knots.	Dist to End		Along track distance to reach the final waypoint.
True Heading	•	Aircraft heading in degrees true.	Reaction Time		Timing signal in msec between button presses. (NOT USED)
Pitch	•	Aircraft pitch attitude in degrees.	Right Engine	•	Right throttle position in percent of full
Roll	•	Aircraft roll attitude in degrees.	Left Engine	•	Left throttle position in percent of full
XAE		Cross Angle Error in degrees.	Elevator	•	Elevator control input position.
XTE		Cross Track Error in nm	Rudder	•	Rudder control input position.
Ground Track		Ground referenced aircraft track in degrees true.	Aileron	•	Aileron control input position.
TAE		Track Angle Error	Flap	•	Flap position in percent of full.
Ground Speed		Ground referenced speed in knots.	Gear	•	Gear position in percent of full.
ETW		Estimated time to waypoint.	Pitch Turb	•	Gust value added to Pitch attitude to simulate turbulence.
CDISensitivity		Sensitivity of the XTE CDI display in nautical miles.	Roll Turb	•	Gust value added to Roll attitude to simulate turbulence.
GAE		Glide Angle Error in degrees.	Yaw Turb	•	Gust value added to Yaw attitude to simulate turbulence. (NOT USED)
			VS Turb	•	Gust value added to Vertical Speed to simulate turbulence. (NOT USED)

• = Original Frasca Variable

Table 4.1: List of State Variables collected

4.3. Cockpit Display Hardware

Within the cockpit of the Frasca, a number of methods were available for presenting information to the pilot. Obviously, there were the standard cockpit instruments, but augmenting these were two LCD Panels. All of the cockpit instruments were located on a plane adjustably located between about 20" and 32" in front of the pilot's eye point and fitted within an area that was 45" wide and 40" high. The largest of the two LCD panels measured 8.25"x6" (10" diagonally) and was located in the centre of the main cockpit panel, between the primary instruments and the radio stack, approximately 7" below and 14.5" to the right of the centre of the pilot's view point. It had a 640x480 pixel resolution. This screen was used to present the simulated GPS receiver display to the pilots while they were flying approaches.

The second panel measured only 6.2 cm x 8.2 cm (4" diagonally) with a 117x320xRGB Trio pixel resolution. This panel was placed directly above the primary instruments, resting on top of the main cockpit panel. For most pilots this meant that it was at eye level when they looked directly ahead while seated in the left pilot's seat and was approximately located at the pilot's view point. Physically, it is actually a miniature TV set (Sharp, model no. 4M T30U) that was being used to mirror a computer display. The scan converter (AITech MultiPro CTV) used to accomplish this task converts a 640x400 pixel window and compresses it into a window small enough to fit inside the LCD resolution available. This screen was used to display the Out The Window (OTW) display. The resolution requirements of this display were minimal and thus the presentation did not suffer from the scan conversion process.

4.3.1. Simulated GPS Receiver

Perhaps the most important aspect of the experimental apparatus was the user interface to the simulated GPS receiver. It was this device which presented the new information about TAE to the pilot. All of the graphics were plotted on the main LCD panel located in the centre of the cockpit. No pilot inputs were required to operate any of the GPS functions. All of the flight plan programming and waypoint sequencing was handled automatically just as if the pilot had already selected the approach to be used and enabled the system. In a typical real world scenario this would most likely be the case. Regulations mandate that all instrument approaches be pre-programmed by the manufacturer and the pilot is simply required to select the appropriate one from the database. Once this has been performed further interactions are only required in special circumstances, such as when aborting before the MAP, executing a procedure turn or other manoeuvre requested by Air Traffic Control (ATC).

Output from the simulated GPS receiver consisted of three major elements: 1) a set of digitally displayed values, 2) three annunciator lights to inform the pilot of certain actions being automatically executed and 3) a CDI scale that presented analog versions of XTE and possibly TAE. The analog element was modified for different display formats. The four different formats used in the experiment were: 1) XTE Only, 2) Triangle Same, 3) Track Vector and 4) Predictor.

All of the data presented on these displays is delayed from the real state of the aircraft for numerous reasons. Firstly, the aircraft state needs to be retrieved from the Frasca and then communicated to the computer making the calculations. The delays required to perform these calculations are in themselves significant and then the results need to be communicated to the computer handling the presentation of the simulated GPS receiver. Although the actual delay was never measured accurately, rough measurements and subjective analysis showed that the information had about a 1 second lag. Most real GPS receivers present information with this order of latency thus making the simulation very similar to real time operation. The update rate was also maintained at 1 Hz to keep in line with the minimum requirements on actual devices. The code for the receiver display is included in Appendix F.

4.3.1.1. Digital Elements

All digital elements were common to all of the display formats. The data presented in this way were: 1) the identifier for the most recent waypoint passed (LAST), 2) the identifier for the waypoint currently scheduled as the next target (CURN), 3) cross track error in nautical miles (XTE), 4) track angle error in degrees (TAE), 5) straight line distance to the target waypoint in nautical miles (DTW), 6) desired track in degrees magnetic (DTK) and 7) ground speed in knots (GS). All of the angles were represented to an accuracy of one degree, distances to an accuracy of one tenth of one nautical mile and the speed to an accuracy of one knot. The accuracy of the digital measure of XTE was worse than that capable of being read on the analog CDI, particularly at the high sensitivity. The digital display of TAE included two arrows that indicated the sign of the angle as described previously for a numeric TAE display. The actual layout is shown in Figure 4.3. The smaller characters were approximately 5/32" high and the larger characters were 3/16". The total height of the display was 1.25" and located in the top left hand corner of the LCD panel.

LAST	**→ CURN**		
XTE	00.1	DTK	328
TAE	40 >	GS	120
DTW	00.1		

Figure 4.3: Digital Elements of the Simulated GPS Receiver

4.3.1.2. Annunciator Lights

Three simulated lights drawn just below the GPS receiver display were used to indicate to the pilot changes being automatically executed by the receiver. These lights were labelled: ARMD, ACTV and TURN. The ARMD and ACTV lights indicated changes in the relative phases of the flight. The ARMD light was programmed to illuminate when the receiver was armed for an approach, ie. within 30 nm of the destination airport. For this experiment, this meant that all experiment runs began with the ARMD light on. This light was programmed to start flashing at 3 nm prior to the FAF. This point represents the position at which a TSO'ed receiver is required to warn the pilot of an impending sensitivity change on the CDI scale. At 2 nm prior to the FAF the ARMD light began flashing also. This indicated the transition from Terminal Integrity Performance to Approach Integrity Performance and the beginning of the sensitivity change on the XTE scale from ±1 nm to ±0.3 nm. At the FAF, the ARMD light went out and the ACTV light came on steady to indicate that the receiver had activated the selected approach.

Figure 4.4: Annunciator Lights.

The TURN light indicated when the device was performing automatic turn following calculations. Anticipation of the turn was provided by flashing this light 0.2 nm before the computer calculated turn was due to begin. The light then came on solid and remained on for the entire time that the computer was using a splined turn circle to determine XTE and TAE values. At a nominal approach ground speed of 120 knots around a 90° turn the turn spline equated to a 0.64 nm (= $2/\pi$) radius circle tangential to the two legs of the course entering and leaving the turn waypoint.

4.3.1.3. Analog Element

The analog element was placed to the right of the digital elements. The enclosing rectangle measured 0.75"x2" making the combined display 5.25" wide. The baseline display included only a standard CDI for which the needle indicated XTE. This was referred to as the "XTE Only" or X display. The other formats added an analog TAE element to the standard CDI. These formats were discussed in Chapter 3. Notably, all formats included numeric TAE information - present in the digital readouts.

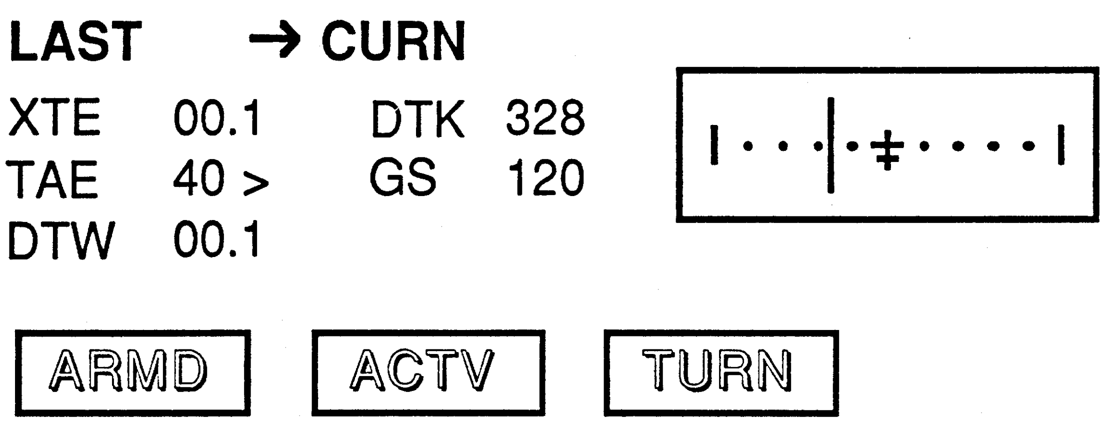


Figure 4.5: Simulated GPS Receiver Display. (Showing XTE Only Format)

5. THE EXPERIMENT

5.1. Motivation for the Design

The driving forces behind the design of this experiment made the task of sorting out all of the details complex at best. It was impractical to run a fully factorial experiment that examined all effects of display format, order of presentation, approach type and wind direction in the single day available to test each subject. The protocol written for the preliminary study was used as the framework for this experiment and modified in the areas that were found to be unworkable or needing revision[5]. Five displays were cut down to four. The missed approach portion of the approach was removed to reduce the length of each approach. The crooked T approaches of that study were replaced by a step disturbance and the final segment lengthened to 6 nm to accommodate this step. Finally, the straight T geometry was converted to an arc geometry.

5.1.1. Decision to Include Digital TAE in all displays

One decision that didn't affect the length of the experiment but still had profound implications was the decision to include numeric TAE information on all of the displays. This included the XTE Only format which was intended to be the baseline display and thus the control condition for the experiment. The preliminary study had shown that a display containing XTE but no TAE information did not provide as much assistance as one which included analog TAE information, so it was expected that a similar format in this study should provide an indication of the minimum level of performance that can be achieved with current instrumentation.

However, TSO C129 requires that TAE be available to be displayed in some form. It recommends that a combined display would likely provide the 'optimum of situation and control information for the best overall tracking' but the other formats were designed to address this point. Thus one display should use a purely numeric TAE display. Obviously, with time to study all possible displays, both a pure XTE display and also an XTE display with numeric TAE would have been used. This would have made it possible to determine the effect of simply adding numeric TAE information.

In an effort to reduce the overall number of displays the pure XTE display was dropped for two reasons: 1) this display had already been included in the preliminary study, 2) by adding an XTE with numeric TAE display it would be possible to directly contrast formats which contain analog TAE information with one that has purely numeric TAE. Basically, the TSO does require numeric TAE so it was thought prudent to include such a display in the current study. This made the experiment one which examined the question: what difference does it make if a manufacturer adds analog TAE information to the minimally required XTE and numeric TAE?

---

[5] See section 2.5. for a brief description of the preliminary study and the protocol used there.

5.1.3. Addition of the Step Disturbance on Final Approach

The preliminary study showed that TAE displays might affect how pilots performed both during manoeuvres and when re-intercepting the course after a blunder on the final approach sensitivity. The original design of having two 45° turns after the FAF was not realistic and proved difficult to analyse. This "crooked T" approach was originally implemented as a way of simulating a navigational blunder and re-intercept during the descent to the MAP. The step was devised as a better way to achieve this goal. By placing the aircraft at a known point it would be easier to analyse the results and be able to compare them across subjects and displays. The deviation was set to be almost full scale to represent a large mistake on the part of the pilot during this critical phase but to not require them to make a missed approach. Normally, if the XTE needle goes off scale a pilot would abort and go around to avoid any hazards on the edge of the approach path. Pilots were asked not to do this because of the detriment that would have had on the time available to complete the experiment. An XTE value just less than full scale should allow the pilots to begin a re-intercept without going off-scale, or, at least, doing so only briefly.

Of course, the other hazard with the display going off-scale is that the pilot then does not receive the same information as would be available while the needle is on scale. A value of 0.25 nm was chosen. This represented a 5/6 th scale deviation at the 0.3 nm sensitivity that was active during this segment. In the end it turned out that most pilots were able to begin a re-intercept without going off-scale even when the wind was blowing them away from the course. Those that didn't were generally not on scale before the step and the step actually moved them closer to the course.

A totally random step would have again made the analysis impossible so it was decided to place a single step at 3.7 nm prior to the MAP with the direction being equally distributed to the left and right of the desired course. One step in each direction was made with each wind condition so that equal numbers were available for comparison in every combination. Each step type was distributed within the entire experiment protocol to help prevent the pilots from being able to guess which way the step was going to occur. The distance before the MAP was chosen to allow sufficient time to re-intercept the course and also to make it difficult for the pilots to recall at what point the step occurred. The refresh rate and delay interacted sufficiently that the display of DTW never actually told them the correct figure, although an intuitive pilot should have been able to determine approximately when it would occur. Other distractions such as levelling off at the MDA together with the fact they were told it could occur anywhere between the FAF and MAP hopefully mislead most pilots. None of the pilots indicated that they had any knowledge of these patterns either during or even after the experiment was completed.

5.1.4. Reduction of Factorial Design

Just considering the formats presented in Chapter 3, a total of six display designs could potentially be studied. Combining this with the two step directions, the desire to included some manoeuvring in the low sensitivity portion of the approach and the necessity to add a number of wind conditions it is easy to see how a fully factorial design can become very long. As has been discussed above, two of the display formats - the pure XTE display and the Electronic HSI display - were dropped from consideration. This still left the wind conditions and approach types to be considered along with some means of presenting all of the information to the pilots without them determining the patterns involved and thus being able to guess the progress of the experiment. All up it was required to conduct the experiment within a single day to make it easier to recruit pilots.

Eventually, four display types, two approach types and two wind directions were chosen to represent the entire population of possibilities. This made for 4x2x2 = 16 experiment runs. At an estimated 15 minutes per run, including breaks, plus 4 practise runs and 2 hours for pre-experiment and post-experiment briefings that summed to 7 hours of experiment time per subject. In the end these estimates worked very well with most subjects taking between 7 hours and 8.5 hours to complete the experiment.

In order to complete each experiment run within 15 minutes it was necessary to limit the maximum length of the approach to something on the order of 20 nm. At a nominal approach ground speed of 120 knots this equates to 10 minutes of flying time and 5 minutes for the pilot to brief and debrief each approach. Not all of this time was required but it gave sufficient buffer space so that pilots could take a short break between each group of four runs.

5.1.5. Subject Selection

Pilots selected for this study were required to fill two criteria. Firstly, they needed to be multi-engine rated. This was because the Frasca is a multi-engine simulator and, although this feature was not of major significance to the study, it was felt that single engine pilots might become confused in the multi-engine environment. Secondly, since the experiment asks the pilots to complete a number of instrument approaches, they were required to have a current instrument rating. Instrument currency requires that a pilot has performed at least 6 landings and 6 takeoffs under Instrument Flight Regulations (IFR) in the last 6 months. Beyond this, pilots were selected on their ability to be contacted. An effort was made to increase the number of female pilots but this proved difficult, with only one woman being represented in the final sample population. A second female pilot was originally recruited but after her training and only 4 approaches it was obvious that she was unable to complete the instrument flying task. Subjectively, this appeared to be chiefly due to her age and the fact that she was used to having a second pilot operate the auxiliary systems.

Most pilots were recruited from local flight training schools where they worked as flight instructors. Ages ranged from 23 to 60 with a median of 31 and mean of 34. Total flight hours ranged from 446.7 to 10900 with a median of 1933 and mean of 2656. The number of instrument hours flown by each pilot ranged from 18.6 to 850 with a median of 175 and mean of 292. On average, pilots had spent 1/9 of their total hours flying instruments. Subjects were paid $10 per hour to participate in the experiment. The protocol was reviewed and approved by the Committee on the Use of Humans as Experimental Subjects affiliated with the Massachusetts Institute of Technology.

Subject Number	Age	Gender	Total Hours	Instrument Hours	Instructor Rating
3	60	M	3387	393	none
4	40	M	3100	700	exp CFII, MEI
5	27	M	1575	100	CFII
6	28	M	1220	100	CFII
7	23	M	700	80	CF/ME/AGII
8	27	M	1400	150	CFII, AGI
9	31	M	2290	260	CF/MEII
10	30	M	446.7	18.6	AGII
11	30	F	925	150	CF/AGII, MEI
12	38	M	2325	200	CF/MEII
13	31	M	3600	500	CF/MEII
14	40	M	10900	850	none
median	31		1933	175	
mean	34		2656	292	
range	23-60		446.7-10900	18.6-850	

Table 5.1: Subject Demographics

5.2. Experimental Procedure

The experiment consisted of training each pilot, running them through the 16 approaches and then de-briefing them to find out their opinions. The four displays were presented in four groups of four with one of each display in each group. An experiment matrix with four different subject types was created to keep track of the order of presentation. Aspects of each approach were varied so that the pilots would not get bored or realise some of the underlying patterns in the presentation. To enable later processing, though, each approach was based on a single geometry and modified in ways that were not likely to affect the pilot's performance. The generic approach geometry thus allowed the data to be converted back to a common reference after it was collected.

The main variations were included by providing four different approach plates that documented the required instrument procedures needed to be followed. Each plate could be used to run any of the four different geometry types of the generic approach.

5.2.1. Experiment Matrix

The main goal of the experiment matrix was to ensure that each display was run with each geometry type. So as to balance the presentation of displays across subjects four different sets of presentation order were created. Each subject was assigned to one of the orders of presentation. Table 5.2 shows the full experiment matrix. The codes in this matrix refer to the display format (upper case letters), geometry type (numerals) and approach plate (lower case letters). It was not possible to distribute all of these variables evenly so each subject type has one display and one approach type that never appear together. For each subject the display presentation is mirrored after the 8th run. Each approach plate, display and geometry type is used once in each group of four.

Subject Type	Run Number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	X1a	V3d	P4b	T2c	P3c	X2a	T4d	V1b	V4a	T1c	X3b	P2d	T3a	P1d	V2b	X4c
2	P2d	X3b	T1c	V4a	T2c	P4b	V3d	X1a	X4c	V2b	P1d	T3a	V1b	T4d	X2a	P3c
3	T3a	P1d	V2b	X4c	V4a	T1c	X3b	P2d	P3c	X2a	T4d	V1b	X1a	V3d	P4b	T2c
4	V1b	T4d	X2a	P3c	X4c	V2b	P1d	T3a	T2c	P4b	V3d	X1a	P2d	X3b	T1c	V4a

Code	Description
X	Display format: XTE Only
T	Display format: Triangle Same
V	Display format: Track Vector
P	Display format: Predictor
1	Geometry type: arc, left step, opposite wind
2	Geometry type: straight, left step, same wind
3	Geometry type: arc, right step, opposite wind
4	Geometry type: straight, right step, same wind
a	Approach plate: Marathon
b	Approach plate: Tavernier
c	Approach plate: Fedhaven
d	Approach plate: Ochopee

Table 5.2: Experiment matrix

5.2.2. Generic Approach Geometry

One basic geometry was used for all of the approaches. This was then modified by choosing two different approach types and two wind conditions that were combined to give the four geometry types. Finally, these were rotated to four different non-cardinal compass directions to provide the approach paths for the four different plates.

The two approach types chosen were a straight-in approach and a curved or arc approach. The vertical profile was the same for both. The only altitude change required was a descent to the MDA at the FAF. The descent was kept constant at 1900 ft with the MDA always being 400 ft above the field elevation. Over the 6 nm final approach segment, a 1900 ft descent approximately equates to a 3° glide slope. Travelling at the designated nominal approach speed of 120 knots required a descent rate of about 800 fpm in order to reach the MDA before the MAP.

The arc approach included a DME segment immediately after the IAF followed by a 90° turn onto the runway heading before continuing as the straight-in approach. The two wind conditions chosen were constant direction crosswinds that were set at 45° to the final approach course and originating either from the left or right side. These were combined with the step direction so that only four different combinations were needed. Of the two arc approaches, one was flown from each side of the final approach course thus requiring both a left and a right 90° turn at the end of the arc.

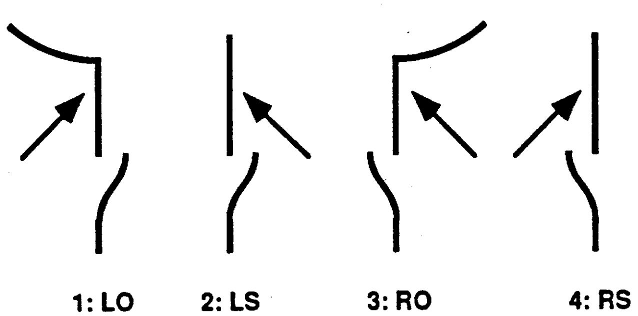


Figure 5.1: Approach Types used to combine wind, step and geometry features

The arcs were always flown when the step was away from the wind as it was felt that pilots would not be able to notice this fact while conducting the experiment. Partly, the reason for doing this was to always place the wind on the same side of the arc, allowing comparisons of the tracking data for this portion. (Assuming the aircraft is symmetric and the pilot has no bias towards a left or right crosswind.) The two straight-in approaches had steps in both directions but always into the wind. This meant that it would have been possible for an intuitive pilot to predict the wind and step direction on the arc approaches based on which direction the arc was to be flown. On the straight approaches, it was necessary for the pilot to determine the crosswind direction before the step direction could be determined. It was felt that sufficient variation was included in other aspects and the workload was sufficiently high to prevent most pilots from discovering these simple relationships. No pilots ever commented that they had been able to perceive these patterns.

The various combinations of step, wind and approach type are then all available by considering portions of each approach - except that the arcs were only executed with a single wind type, ie. one that blew into the centre of the arc. Naturally, this assumes that once the arc and the turn onto the final approach direction have been completed the straight portions of the experiment run can be compared to the straight portions of runs that didn't included an arc at the beginning.

The four different approaches types were labelled by number. They are shown diagrammatically in Figure 5.1. Type 1 was an arc approach with a step to the left side and a wind from the opposite side as the step. Type 2 was a straight approach with a step to the left side and a wind from the same side as the step. Type 3 was an arc approach with a right step and a wind from the opposite side as the step. Type 4 was a straight approach with a right step and a wind from the same side as the step.

5.2.3. Approach Plates

Four different fictitious approach plates were created that could each be used to run any of the four geometry types. Each plate had three IAF's, two for each arc approach and the third for the straight-in approach. Pilots were actually started a small distance from the assigned IAF. In the case of the arc approaches this offset was equal to 0.5 nm backwards along the tangential track at the start of the arc. For the straight approaches the aircraft was initially placed at 1.0 nm backwards along the final runway track.

The plates were based on the generic approach geometry and then rotated to non-cardinal compass directions. The four runway heading directions chosen were 052°, 138°, 213° and 296°. As a result, two of the final approach headings were to the South, requiring the subject to mentally rotate the map in order to visualise it track-up. Non-cardinal directions were used to make it difficult for the pilots to recall the exact numbers associated with each plate and therefore need to refer to the plate on each run. A fifth plate, rotated to 180°, was used for all of the training approaches. Each plate had a different field elevation so that the altitude numbers for the vertical profile were all different even though the profile remained constant relative to the ground. Plates were drawn as accurately as possible to reflect a real instrument approach plate. All plates are reprinted in Appendix D.

5.2.4. Subject Briefing and Training

After being contacted, each pilot was sent a briefing package that described the experiment and each of the different display formats. The full text of this briefing is reprinted in Appendix C. It was requested that they read this information before coming to the lab on the day of their experiment. Information in this package was, however, repeated during the pre-briefing and training process. Immediately upon arrival on the morning of each experiment up to date demographic data was collected from each subject, including information about their flying experience. Further insights into the subject's background and experience were often volunteered later in the day.

Following this, the subject was taken to the simulation lab and a pre-brief training session begun. Pre-briefing consisted of familiarising the pilot with the Frasca cockpit, a basic run down of the experiment and a teaching session to relate the concepts of the displays. Most pilots needed at least a simple demonstration of all the relevant devices and controls and their location in the cockpit. They were also allowed to fly the simulator in a free running mode to become accustomed to the dynamics of the modelled aircraft. The training session began with a quick in-cockpit demonstration of the four different display types and then adjourned to a meeting room were the concepts could be discussed in detail. Time allotted for this briefing was two hours. After the pre-briefing pilots were asked to sign an informed consent statement which acknowledged their voluntary participation in and comprehension of the experiment procedures.

Once the pilot was familiar with all of the procedures to be followed in the experiment and the workings of each display format they were asked to fly a number of practise approaches. At least four approaches were flown, one using each display type and geometry type. These approaches all used the same "Nassau" approach plate so that pilots would not memorise the four charts to be used during the subsequent experiment. By using only one chart during practise the workload was also reduced slightly allowing the pilots to concentrate on using the displays. During the practise approaches the overall turbulence level was progressively increased from 0 to 4.3. Pilots were occasionally reminded that these were practise runs and that they should try to learn about the displays. Even so, they were asked to complete the entire task and become accustomed to the Frasca, tuning the radios, reconfiguring the aircraft and checking for a horizon when at the MDA. Thus practising of all details were lumped into these four approaches. If desired, the pilots were permitted to run more practise approaches if they felt insufficiently prepared for the experiment. Four pilots asked for extra practise with a range of one to three extra approaches being requested for a maximum total of seven practise runs. After the training a one hour lunch break was taken.

5.2.5. Test Procedure

The basic test procedure was to present the sixteen experiment conditions to each pilot in the order prescribed by the experiment matrix. To assist the experimenter in this task a series of checklists were written that set out the order of presentation for each subject and permitted comments to be written during each approach. The checklists marked off the major milestones through the approach so that failures in the software could be detected as early as possible. They also helped the experimenter to remember all the tasks involved with each separate run. These lists are reprinted in Appendix D.

Each run was begun by setting the correct geometry type and approach plate into the control computer. This process froze the simulator and reset its position and attitude to the correct initial conditions. By placing the aircraft always at a known position and attitude at the start of each run it was hoped that data could be collected on how quickly pilots could use each display to determine the correct desired heading for the given wind condition. During this time the pilot was told which approach plate would be used and which IAF would be the starting point. The display type was also entered at this stage and the pilot informed of the format to be used for this run. The pilot was then asked to examine the plate and inform the experimenter when he was ready to fly. Once the simulator was ready and the pilot had indicated he was ready, the run was started. On two of every four approaches a question concerning the details of the approach plate was asked before the run began. Pilots were asked not to refer to the plate while answering these questions, although they were also not required to provide the correct answer. These questions provided a simple incentive to keep the subject carefully examining the approach plates.

During the run, the checklists were used to record the points at which the pilot made various changes to the aircraft configuration and also when radio calls were made. The pilot was asked not to deploy the landing gear or flaps before 3 nm before the FAF and to have completed doing so before descending. Before beginning a descent to the MDA the pilot was required to make a radio call on the published CTAF to inform local traffic of his intentions. This required tuning the radio to a new frequency as each plate had a different CTAF. Once at the MDA, the pilot was requested to monitor the OTW display and inform the experimenter when the horizon was in sight but to not descend to the runway until after the MAP. At the MAP a second radio call was required that informed the local traffic of the pilot's decision as to whether or not to continue with the landing or to go around. This call ended the run.

Finally, the aircraft was flown to a reasonable altitude and reconfigured for straight and level flight with the gear and flaps retracted. This placed the simulator controls close to the required positions for the next run. Upon completion the pilot was asked to provide a number of subjective workload measures. After every four runs the pilot was asked to take a break to help reduce the effects of fatigue.

5.2.6. Workload Reporting

The workload measures asked of the pilot at the end of each run were of two types. The first used a descriptive scale to assign a value between 1 and 10 to the level of workload. The scale uses a modified Bedford style that rates low scores as being associated with low workloads. A chart of this scale is shown in Figure 5.2. The chart asks the pilots to remember how much spare attention they had and to associate low workloads with periods when they felt they could have performed other tasks. Pilots were briefed on the use of this chart during the training period and a copy was posted in the cockpit beside them for direct reference. They were asked to report two values and to try to maintain the absolute reference of the scale by referring to the chart on each occasion. The two values represented the workload in two different phases of the approach - initial approach (prior to deploying the flaps and gear) and final approach. If the pilot felt a mistake had been made in a previous workload score they were permitted to correct that score after the fact, although no pilots took advantage of this.

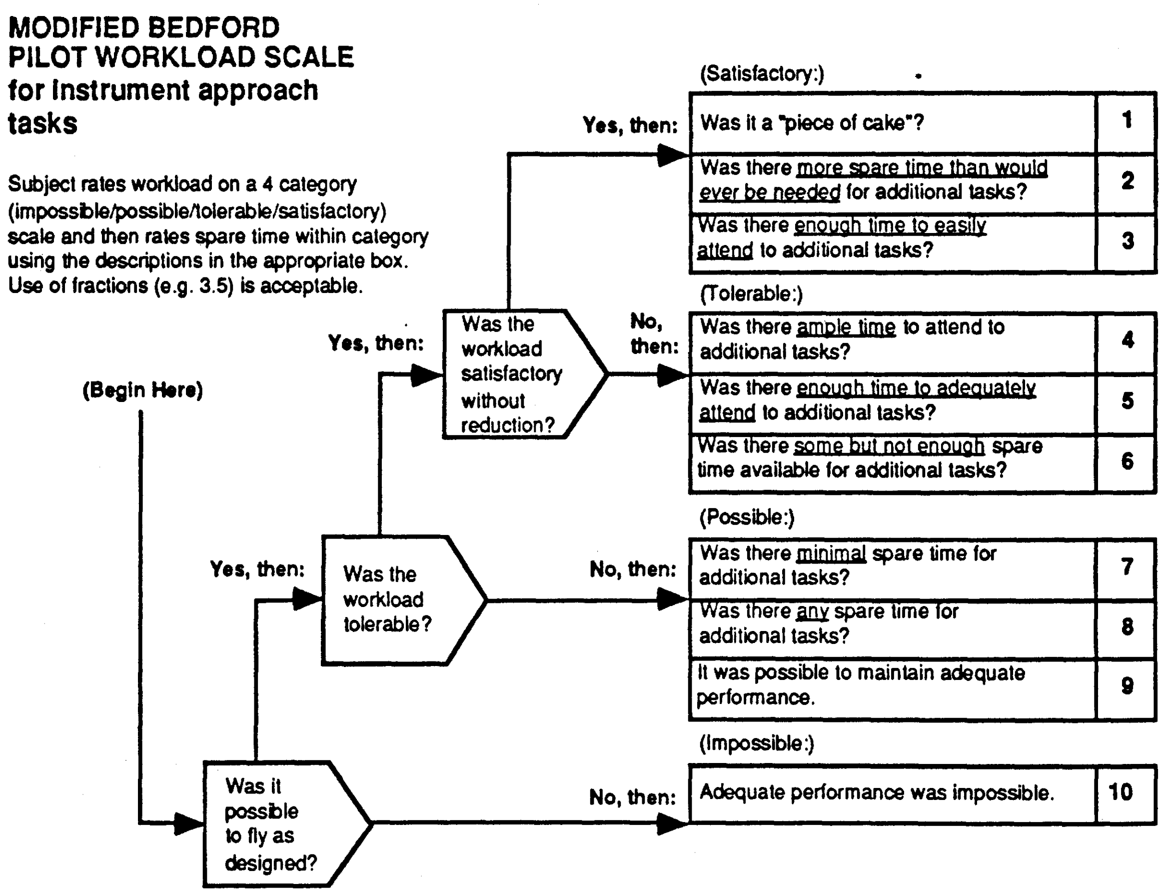


Figure 5.2: Modified Bedford Pilot Workload Scale.

The second measure was a relative ranking. After each approach the pilots were asked to compare the workload over the entire approach with the workload on previous approaches and to rate them accordingly. Rating was divided into the four groups of four so that pilots did not need to recall approaches made more than three runs previously. Also a running rating was maintained to help the pilots recall the difficulty of previous runs.

5.2.7. Subject De-brief

On completion of the experiment runs, the pilot was taken back to the briefing room and asked to complete a questionnaire that requested information on his subjective opinions. While completing the questionnaire and during later discussions of the experiment, a tape recording was made of all of the pilot's verbal comments. These were summarised to document each pilot's opinion.

The questionnaire is reprinted in Appendix C. It consisted of three portions. The first asked the pilot to compare each of the displays on a "head-to-head", pairwise comparison (tournament) basis and to record his preference for each by placing a mark along a scale between each display relative to that preference. The second section rated the displays by various aspects of their ability to help the pilot. In this section a relative ranking was recorded that represented the pilot's opinions when comparing all displays . No ties were allowed in the ordering of displays. Finally, the third section asked questions about how the pilot felt about the experiment and what previous experience he had.

5.3. Instrument Approach

Each run required a very similar instrument approach procedure. Once the pilot was told the IAF he could determine where he would begin in that procedure. An arc approach required some extra thinking and action before the main task of reconfiguring the aircraft and descending to the airport could be completed. During the approach the sensitivity of the XTE display was scheduled to change as required by TSO-C129. Once at the MDA a decision then needed to be made as to whether or not the landing could be completed.

5.3.1. Simulated DME Arc

IAF's located at the ends of the arc approaches required the pilot to follow the 7 nm DME arc from a remote VORTAC beacon. Of course, this was designated as a GPS overlay so the GPS receiver automatically calculated the desired arc and reported XTE and TAE values relative to this arc. While flying this procedure the DTK value also updated to reflect the present desired track. The arc itself was about 5 nm in length. The aircraft was initially located 0.5 nm before the IAF. At the end of the arc a 90° turn brought the aircraft onto the final runway direction. All through the arc altitude was to be maintained at 2300 ft above the field elevation. Pilots could use the time spent following the arc to complete some simple additional tasks. These included retuning the radio to the correct CTAF and turning on the fuel pumps ready for the descent.

5.3.2. Reconfiguration and Descent

For a straight approach, or after the turn had been completed on the arc approaches, the first task was to reconfigure the aircraft for the descent. This needed to be completed prior to the FAF but not before 3 nm. Most pilots completed this task with some distance to spare and thus needed to maintain altitude until the FAF.

Reconfiguring required the gear to be lowered, the flaps to be extended to 50%, the pitch on the propellers to be increased to the maximum and the carburettor heat to be turned on. Also a landing check should be completed to ensure that all of these things were done. Just before the FAF the radio call could be made and then once at the FAF the throttles were reduced to begin the descent.

A descent rate of 800 fpm generally kept the aircraft on the correct descent profile. If the descent had been started late, however, a greater rate was required. The throttle position to achieve this was not very much reduced from the level flight position and so pilots tended to descend too rapidly. Given this was a non-precision approach doing so was not penalised. In fact, many pilots try to intentionally "duck under" during this type of approach.

5.3.3. CDI Sensitivity Scheduling

The relevant portions of TSO C-129 section (a)(3)(xii) state:

> 1.b. Upon activation of the approach mode, the equipment shall provide a smooth transition from 5 nm non-numeric display sensitivity to 1 nm sensitivity.
> 1.c. At a distance of 3 nm inbound to the final approach fix, the equipment shall provide an annunciation indicating an automatic non-numeric display sensitivity change will occur.
> 1.d. At a distance of 2 nm inbound to the final approach fix, the equipment shall:
> ii. Provide a linear transition from 1 nm non-numeric display sensitivity to 0.3 nm sensitivity at the final approach fix.

The approach protocol used for this experiment began with the aircraft located within a 30 nm radius of the airport. The simulated GPS receiver was thus programmed to begin in the approach mode and already set at ±1 nm full scale sensitivity. As required by the above paragraphs of the TSO, the GPS receiver was programmed to provide a linear transition from 1 nm sensitivity to 0.3 nm sensitivity. This transition began at 2 nm inbound of the FAF and ended at the FAF. Comparing this with the description given earlier of the annunciator lights it can be seen that the pilot receives warning of both the initiation of this sensitivity change and of its completion.

5.3.4. Step Disturbance

After passing the Final Approach Fix (FAF) and during the initial descent the aircraft was shifted perpendicularly to the desired track. During this process the altitude was maintained but the aircraft heading and attitude were reset. The heading was set to the desired track (ie. the desired heading assuming no crosswind) and the attitude was set to both zero Roll and Pitch angles. Setting the Pitch angle to zero gave the aircraft a pitch disturbance. Preferably, this disturbance should have been removed but doing so was not possible using the available software. The Roll angle and heading were set so that all pilots would start at the same initial point before executing their re-intercept. This allowed for fitting of the responses, knowing the initial conditions. The shift represented an almost full scale deflection of the XTE needle. The pilots were then asked to re-intercept the original course as quickly as possible.

5.3.5. Missed Approach Procedures

The Missed Approach was never actually flown. A number of steps were taken, however, to make the pilot evaluate visibility conditions and arrive at a missed approach decision in as realistic a way as practical. The normal task for a pilot during a non-precision approach requires that a descent to the airport is arrested at the Minimum Descent Altitude (MDA) unless the airport is in sight. If the pilot can not see the runway threshold once the MDA is reached then this minimum altitude should be maintained until the Missed Approach Point (MAP) is passed. If the airport has still not become visible then the pilot is required to execute a missed approach - following the published missed approach procedure. The Frasca simulator used for this experiment lacks a visual system capable of fully simulating the view a pilot has from the cockpit. This makes it difficult to simulate the appropriate conditions that a pilot relies upon in normal flying to make the decision to land or to go around.

In an attempt to compensate for the lack of a full visual system, a small TV monitor was placed above the main cockpit instruments and used to display a simple Out The Window (OTW) view. This display showed two states. The first, a completely white screen, simulated the view while still under Instrument Meteorological Conditions (IMC). That is, as if the aircraft was flying within cloud. The second simulated the Visual Meteorological Conditions a pilot would expect to find when beneath the clouds close to the airport. It consisted simply of a horizon separating areas of solid colour. Light Blue was used to represent the sky and Light Grey the ground. These colours were chosen to appear realistic and also to reduce the contrast between the two states. These close contrast levels required the pilots to look directly at the display in order to notice when the state had changed. Although the actual runway was not depicted on this display, the pilots were asked to consider the horizon display as indicating that the runway was indeed visible.

The procedure required the pilots to monitor this OTW display while on final approach and inform the experimenter when they had the horizon in sight. The experimenter was then able to note the distance at which the horizon was able to become visible and compare it with the distance at which the pilot noticed that it had become visible. Once the MAP had been reached, the pilot was then required to make a decision as to whether or not to go around. This decision would be executed and the pilot was required to make a radio call announcing his intention. This differed from normal practise in that a descent to the airport would normally be started as soon as the airport was in sight. The protocol for this experiment asked the pilots to maintain altitude above the MDA until the MAP was reached before executing any decision. Data recorded beyond the missed approach point was ignored in all subsequent analysis. Pilots were asked to maintain altitude so that flight conditions and aircraft configuration for all pilots on all runs would be kept constant during the last miles before reaching the MAP.

Roughly one quarter of the runs were programmed to keep the pilot always within the clouds and he was thus expected to execute the missed approach procedure. Also, if the pilots deviated more than 70 ft above the MDA they would re-enter the clouds. On the runs that were programmed to show a horizon, the distance before the MAP at which the horizon could become visible was chosen randomly and uniformly distributed between 2.0 nm and 0.0 nm.

In the preliminary study all pilots flew the missed approach. The data from this segment of the approach showed little useful information and extended each run by a number of minutes. In the protocol for the present experiment it was felt that this segment could be eliminated, thus reducing the time required to execute each approach. The above procedures were then added to bring back the requirement on the pilot to execute the missed approach. In fact, it was felt that this added the cognitive task of making this decision to the workload of the final few miles - something that was missing when the pilot knew that the missed approach would always be flown.

5.4. Data Analysis

All of the data collected by following the above protocol for 12 subjects needed to be analysed. Before any other calculations were made, the first procedure was to import the data into a format readable by MATLAB and to transform the Latitude and Longitude positions into an XTE/Along Track Distance referenced axis system. Using these transformed sample points slices of any variable at any Along Track Distance could be easily obtained and used to calculate statistics. A second order model was also used to fit the XTE data for the last 3.7 nm of each run. Using slice information some statistics were calculated by aggregating all slices between two along track distances. In all much data was processed and the code used to do so is presented in Appendix G.

5.4.1. Normalisation Transformation

Transformation of the data into a form that could be used to compare the various experiment runs was achieved in a number of steps. Initially all Latitude and Longitude values were offset and scaled to represent nautical miles from the MAP. This was then rotated so that the final runway heading was due South, thus creating a coordinate system that roughly used the X-axis for XTE and the Y-axis for Along Track Distance. For the straight approaches no further transformation was required. The type 3 arc approaches were further flipped about the Y-axis to bring all of the arc segments to the second quadrant (the right side of the approach). Figure 5.3 shows the grid that was then used to transform the rotated arc approaches. This grid was "straightened" to convert the coordinates to a true XTE/Along Track Distance axis frame. Straightening was achieved by cutting the plane into four pieces and transforming each piece appropriately. The details of the transformation are coded into the routine SAVEROT.M included in the Appendices.

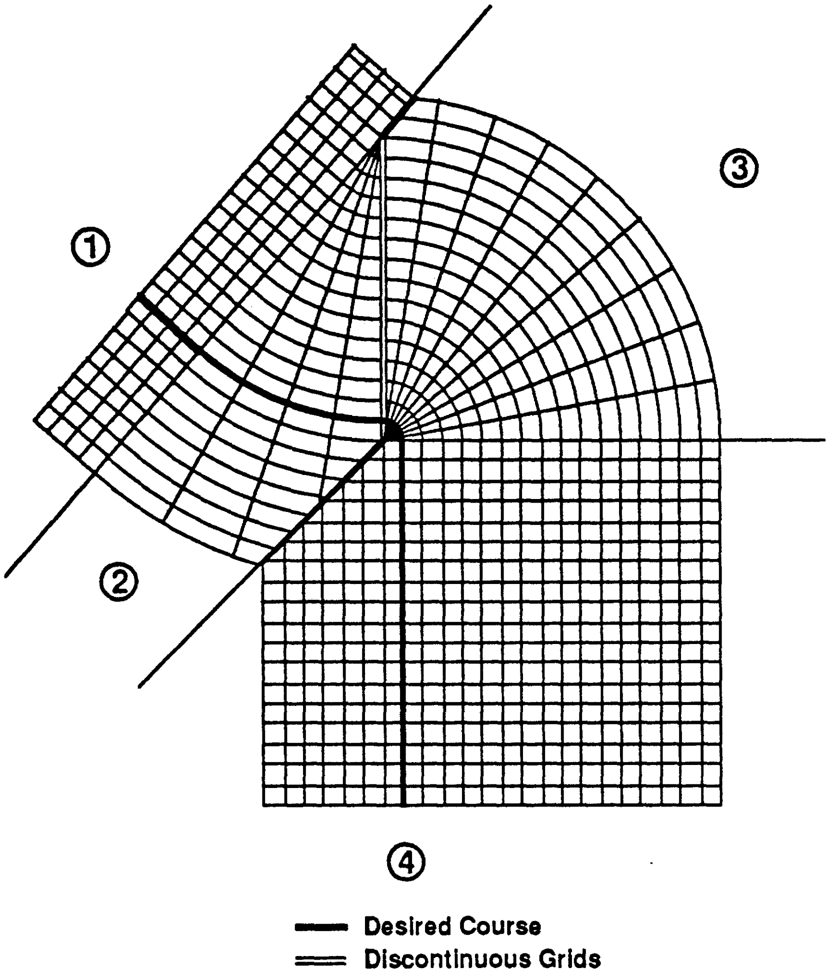


Figure 5.3: Transformation Grid used to straighten the arc approaches.

5.4.2. Along Track Slicing

All statistics were based on values extracted from the recorded data by taking slices at a specific along track distance. This is equivalent to hypothetically putting a very large screen at the slicing distance and marking the aircraft position, attitude and configuration at the moment it passes through the plane of the screen. Basing the statistics in this way achieves two things. Firstly, it removes the dependence on ground speed and therefore time of the various random processes. Instead the independent variable becomes along track distance to go. Secondly, it makes it possible to evenly sample the process. The poor time sampling of the computer systems used to record the data could be ignored and any new sample could be created by linearly interpolating between appropriate sample points.

Variable pilot performance meant that it was possible to have more than one aircraft trail at each along track distance. This occurs if the pilot ever flies back along the course, either by making a large blunder, turning too sharply and actually flying, perhaps only briefly, in a direction which increases the along track distance to go, or at the step disturbance because of the delays in the system which often caused the pilot to be stepped backwards a very short distance. This was solved depending on the application. For the majority of slices there was only one value, but when multiple values occurred the most recently recorded point closest to the end of the run was used. Very occasionally, the oldest, first recorded point was used. The method was chosen to reflect the geometry of the situation and the desired intent of the calculation being made. In the case of the step the most recent point was obviously the more appropriate because it had been recorded after the step had occurred.

5.4.3. Model Fit to Re-Intercept after the Step

The final 3.7 nm before the MAP represented the pilot/aircraft response to a step disturbance of known initial conditions. Using a second order linear model, a least squares fit was applied to the XTE vs Along Track data. The residuals were weighted with an exponential function of distance so that the later points were not as important to the fit as the initial response. This gave very good approximations for the initial response to the step in all cases. The exponential used decayed to half of its initial value after 0.69 nm.

$$A = \begin{bmatrix} 0 & 1 \\ |\lambda_2|\lambda_2 - \lambda_1^2 & 2\lambda_1 \end{bmatrix}$$

$$\lambda_1 = \text{first parameter}$$

$$\lambda_2 = \text{second parameter}$$

Figure 5.4: System Matrix for the model fit of the step response.

The parameters of the model included the initial rate of change of XTE as well as two values used to calculate the system matrix. Figure 5.4 gives the exact equations used to relate the parameters to the system matrix of the model. The net effect of these equations is that if the second parameter is negative then it will become the imaginary part of the eigenvalues of the matrix and the first parameter will become the real part. If, however, the second parameter is positive, then the matrix will have two real roots centred about the value of the first parameter and separated by twice the value of the second parameter. Another way to imagine this is to think of the first parameter as an anchor on the real axis in the s-plane. The second parameter then defines the roots relative to that anchor. If the second parameter is negative the roots will be perpendicular to the real axis, ie. a complex pair. If it is positive they will be parallel to the real axis, ie. a real pair either side of the first parameter.

Using this model it is doubtful how important the eigenvalues of the system matrix are towards characterising the response because of the low sensitivity of the solution to the values of the system roots. The smoothing that this model provides to the data track, however, gives an excellent approximation of various features of the response. Namely, the first intercept after the step, the maximum intercept angle and the maximum overshoot were all well captured.

The actual statistics gathered were 1) the along track distance to the first zero XTE crossing (ZERODIST), 2) the intercept angle achieved at an XTE distance of 0.15 nm (INTANG), 3) the maximum value of XTE (MAXVAL), 4) the maximum value of XTE after the first zero XTE crossing (MAXREMN) 5) the eigenvalues, natural frequencies and damping ratios of the model roots, 6) the maximum intercept angle (MAXANG) and 7) the location of the maximum intercept angle (ANGLOCN).

5.4.4. 95% Limit Plots

One method used extensively to aggregate the information from all the different subjects and to compare their performance across displays was the calculation of 95% confidence limits on the mean of XTE based on the sampled variance of one Along Track slice. These limits show where the aircraft is expected to be found 95% of the time and show how wide a path is required to keep most aircraft safe as they follow the approach. As in [Oman95] the procedure was to calculate the sample mean and standard deviation and then determine the 95% bounds by adding 2 standard deviations to the mean. In order to account for the sampling error in the standard deviation and provide an upper limit on these bounds a correction was made based on the 95% confidence interval for the value of the sample standard deviation. The formulae for the bounds are then $M \pm 1.96 * L^{0.5} * S$, where M is the sample mean, S is the sample standard deviation and L is a correction based on the chi-squared distribution for the error of the sample variance. This correction is equal to the tabulated $p=0.025$ value for $\chi^2/(n-1)$.

Calculations of this value were made for slices at regular 0.2 nm intervals and plotted against Distance To Go, producing an envelope of XTE points. In plotting the results, a T-test on the significance of the mean being different from zero was marked by a cross (x) mark on those points that did not pass at a two tail probability level of 0.05.

Pairwise comparison plots were also made to examine the various combinations of displays. A second T-test was used to find slice positions at which the means were significantly different. These were marked by a cross (x) symbol on the plots. An F-test, assuming a common population, was calculated to compare the variances at each slice position. Those that did not pass at the $p=0.05$ level were marked with circles (o). If a slice did not pass the T-test, then the F-test was not calculated due to the high likelihood that the assumption of a common population was invalid.

It was first necessary to introduce a means of removing outliers from the data due to the large effect some of these points had on final results. The method used was to calculate means and variance for any data set from a single slice position and determine which points (if any) fell outside the two tailed 99.99% range based on a normal distribution. This limit is equivalent to ±4 standard deviations as recommended by Sachs [Sachs84]. To find these points the mean and standard deviation of the currently valid population was used to identify the point most likely to be an outlier. This was then removed, new statistics calculated and the point rechecked to determine if it really was an outlier. This process was iterated until the list of outliers stabilised. The outlier points are marked on the plots showing single displays by a plus (+) character.

These plots, thus show the results of a large number of statistical tests. If all of these test were independent 1 in 20 of them would come out true simply by chance. They are not all independent, however, with nearby tests being correlated due to the nature of the flying task. Points spread sufficiently far apart should be independent for the same reason. This is not always true because of the design of the experiment. For example, the step response on the arc approaches will always be in the same direction as the initial response to the wind at the start of the run because the wind was always blowing in that direction.

Interpreting the pairwise comparison plots, then comes down to a matter of looking for large groups of slice positions that pass the significance test. One or two isolated points are likely to be there just by chance. A series of four or five points, on the other hand, is likely to indicate a significant effect.

5.4.5. Along Track Statistics

A set of statistics was collected which aggregated a series of along track slice values into a single number. In retrospect, perhaps the most important of these was the Roll RMS value. This statistic took the value of Roll Attitude sampled at 0.05 nm intervals for the entire run and calculated the RMS value. The 0.05 nm sampling rate ensured that at least one sample occurred for every real sample in the data while only occasionally sampling more than once between each sample.

Other similar statistics calculated along track values of RMS for XTE. These were separated into five, two mile long portions. The first section collected XTE values from 16 nm to 14 nm, during the DME arc portion of the approach. It was labelled Arc XTE RMS. The second section collected values from 10 nm to 8 nm, just after completing the 90° turn in arc approaches and soon after the start of the run in the straight approaches. It was labelled Turn XTE RMS. The third section collected statistics for the two miles immediately before the FAF, 8 nm to 6 nm. It was labelled Sens XTE RMS. The fourth section extended from the FAF two miles into the descent. It was labelled FAF XTE RMS. The final section aggregated the XTE data just before the MAP. It extended from 2 nm to 0 nm and was labelled MAP XTE RMS. These sections skipped the portions of the approach during which manoeuvres were being performed. At these times the mean was most often not zero and thus didn't provide information on straight tracking.

All of the RMS statistics were log transformed before performing any statistical tests which assumed a normal distribution. Obviously, RMS values can't be less than zero and thus tend to follow a Poisson like distribution, rather than a normal distribution. The log transform suitably reshaped the distribution to be more gaussian.

6. RESULTS

The main hypothesis driving this experiment was that the addition of analog TAE information to a general aviation cockpit will help a pilot to fly instrument approaches better. What exactly is meant by "better" needs to be defined but in a general sense it refers to the pilot's ability to follow the desired course more closely, possible with less effort and perhaps with a better idea of the current situation. Firstly, results from the experiment show that analog TAE does indeed have an effect on how pilots fly, both on their Inner Loop control strategy and their level of workload. From there it is a matter of "well, it all depends". For some pilots the analog TAE displays show some obvious improvements in the pilot's ability to track the desired course during certain critical portions of the approach, in particular the last few miles before the MAP - the short final segment. Other pilots, however, had a hard time performing the experiment and were perhaps hindered by the extra information contained in the new displays.

6.1. Dividing the pilots: Roll and Pitch RMS statistics.

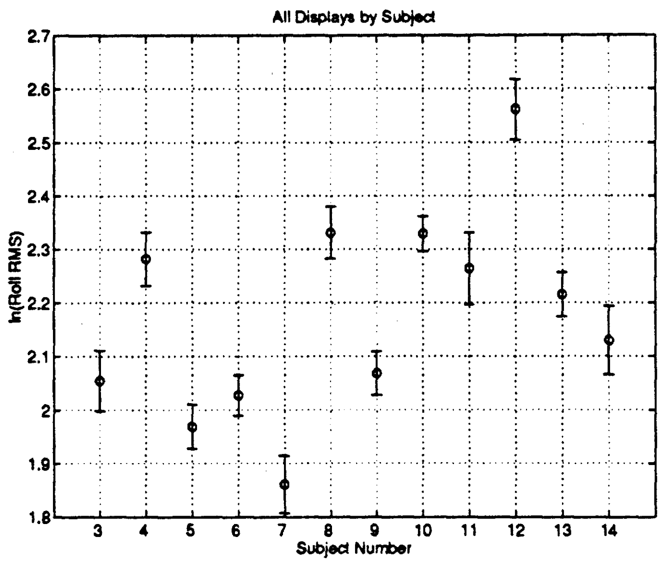


Figure 6.1: Mean value of ln(Roll RMS) plotted by subject for all displays

During the search to find evidence to prove or disprove the hypothesis it became apparent that the pilots could be categorised by the mean of their Roll RMS statistic. This number, in a very general way, represents the Inner Loop control being applied by the pilot averaged over one entire run. A high Roll RMS means that the aircraft was experiencing large bank excursions. One of two interpretations can be given to this: either the pilot was having trouble reacting to the turbulence, which was being input as a roll angle disturbance, and let it pass unattenuated or possibly amplified, or the pilot was easily compensating for the disturbance and using frequent bank angle changes to aggressively track the course. A low Roll RMS means that small bank angles were being experienced for short periods. This could imply that either the pilot was compensating for the roll angle disturbance extremely well and causing it to be well attenuated before it could manifest itself as large bank excursions, or that there was only a small disturbance which the pilot didn't feel he needed to modify significantly.

Figure 6.1 shows the mean values of ln(Roll RMS) for all of the subjects. Using the median value of 2.16 it was possible to split the pilots into two groups. Most subjects fitted easily into one of these two groups. Subject 13 fell right on the boundary line but was placed into the upper group. Both of subjects 12 and 14 could also, potentially, have been classified either way but were placed into the obvious group. Thus pilots 4, 8, 10, 11, 12 and 13 form a group - the High Roll RMS group - and pilots 3, 5, 6, 7, 9 and 14 form another group - the Low Roll RMS group. This formed a median split of the pilots. Figure 6.3(b) shows the mean of ln(Roll RMS) when the pilots were divided in this way.

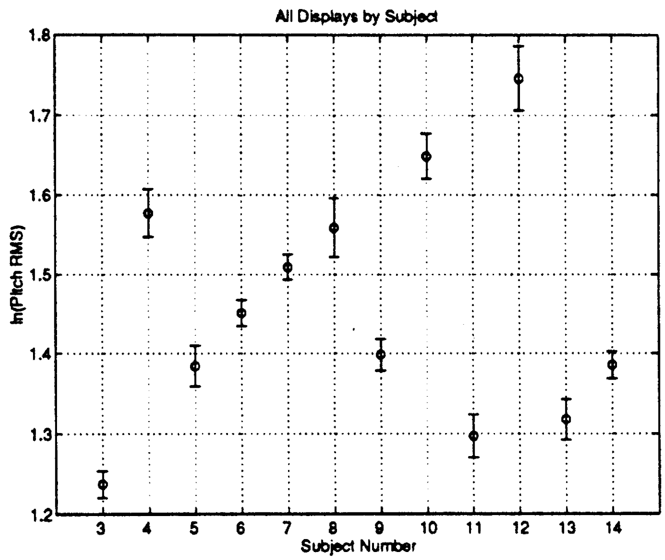


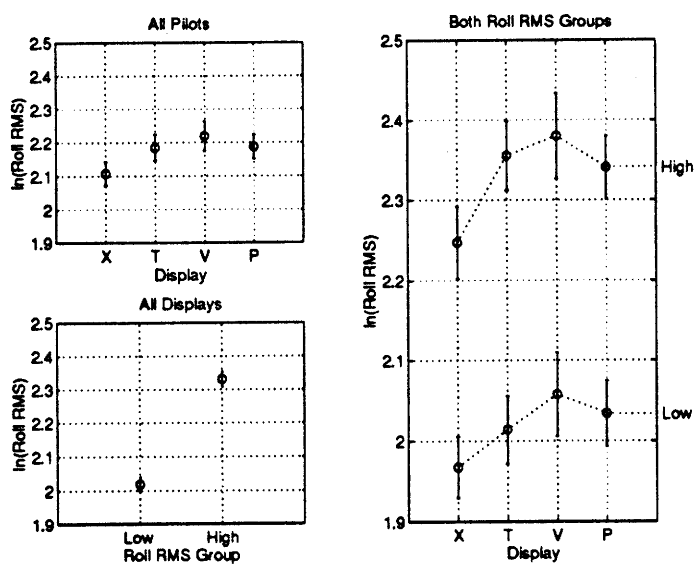
Figure 6.2: Mean value of ln(Pitch RMS) plotted by subject for all displays

The same procedure was also be applied to the Pitch RMS statistic. Figure 6.2 shows the mean values of ln(Pitch RMS) for each of the pilots. The median in this case was 1.44 and the two groups formed were very similar to those using the Roll RMS statistic. The only differences occurred with subjects 6, 7, 11 and 13. In these cases subjects 6 and 7 previously fell into the low Roll RMS group, but using this method fell into the high Pitch RMS group and subjects 11 and 13, previously in the high Roll RMS group were classified into the low Pitch RMS group. A regression on Roll RMS vs Pitch RMS shows a reasonable correlation between them ($R^2 = 0.18$, $F(1,222) = 48.7$, $p < 1e\text{-}10$).

6.2. Analog TAE does affect how pilots fly.

The simplest result was that analog TAE displayed on a secondary device outside of the pilot's primary field of view during challenging instrument approach conditions did indeed affect how a pilot flew the simulator. An ANOVA analysing ln(Roll RMS) shows a significant effect of display type ($F(1,190) = 4.06$, $p < 0.046$). In this case the displays were grouped into two categories: 1) no analog TAE information (XTE Only [X] display) or 2) analog TAE information (Triangle Same [T], Track Vector [V] and Predictor [P] displays).

Figures 6.3 plot the mean value of ln(Roll RMS) by display and RMS group. Figure 6.3(c) clearly shows the split into High and Low Roll RMS groups as well as the categorisation into analog TAE and no analog TAE displays. What can be seen in this plot was that the display without analog TAE information was associated with a lower mean value of ln(Roll RMS). This was true for both groups of pilots, although slightly less so for the low Roll RMS group. This is worth noting as it will later be shown that low Roll RMS, when considered by subject, is associated with low XTE RMS (and thus better performance) but, in many circumstances, the analog TAE displays show lower XTE RMS than the non-analog display. Also worth noting, on a similar vein, is that the Track Vector [V] display tended to have the highest mean value of ln(Roll RMS).



<u>Figure 6.3</u>: **a) [Top Left] Mean value of ln(Roll RMS) plotted by display for all subjects, b) [Bottom Left] Mean value of ln(Roll RMS) plotted by subject group for all displays, c) [Right] Mean value of ln(Roll RMS) plotted by display and subject group.**

6.3. XTE Comparisons

XTE provides an easily examinable measure of a pilot's ability to track the desired course and was used as the basic performance measure to determine if any displays could improve performance. Two methods were used to compare XTE. Firstly, the 95% limit plots were used to gauge the magnitude of any effects. These plots show numerous statistical test results and so need to be examined with some care. Each test has a 1 in 20 chance of succeeding and thus isolated positive results may not be meaningful. Also, adjacent slices are not independent and so small groups might also be questioned. A long sequence of positive results, however, does suggest a significant result. Secondly, conglomerate, along track statistics were gathered on certain 2 mile segments of each experiment run and analysed in support of the results made apparent by the 95% limit plots.

These comparisons point to a number of significant findings and several important trends. When examining all pilots as a single group it was apparent that the Track Vector [V] display was the only one which had a demonstrable effect of improving XTE performance by reducing XTE RMS. This effect showed up only in the arc portion of the approach. Using the Roll RMS value to split pilots into the two groups, however, showed that the low Roll RMS pilots had significantly better performance over the entire approach when compared with the high Roll RMS group and demonstrated significant effects of the various displays during the descent portion of the approach. But the high Roll RMS group showed little benefit and possibly some detriment from using analog TAE displays during this type of high intensity instrument approach.

6.3.1. All pilots

Before dividing the subject population into sub-groups it is important to look at all pilots and determine if there are any effects which can be shown to influence everyone. Figures 6.4, 6.5 and 6.6 show the 95% limit envelopes for the various displays and the comparisons between all pairs of displays. Using these as a starting point it is easier to find the significant results of this study.

In Figure 6.4 it is possible to compare the individual aspects of each display. Firstly, it is important to note that very few points have been excluded due to being outside the arbitrarily assigned 99.99% cut-off for outliers. There are five outlier points on the Track Vector plot that have been hidden because of the XTE axis scaling. They occur at 13.4, 13.2, 10.0, 9.8 and 9.6 miles. Generally the sharp changes in the envelope width are caused by changes in the number of tracks included in each analysis. These changes can occur either because the normalisation transformation causes the track to jump across a large distance (primarily across the 90° turn) or because points have been eliminated by the outlier rejection routine.

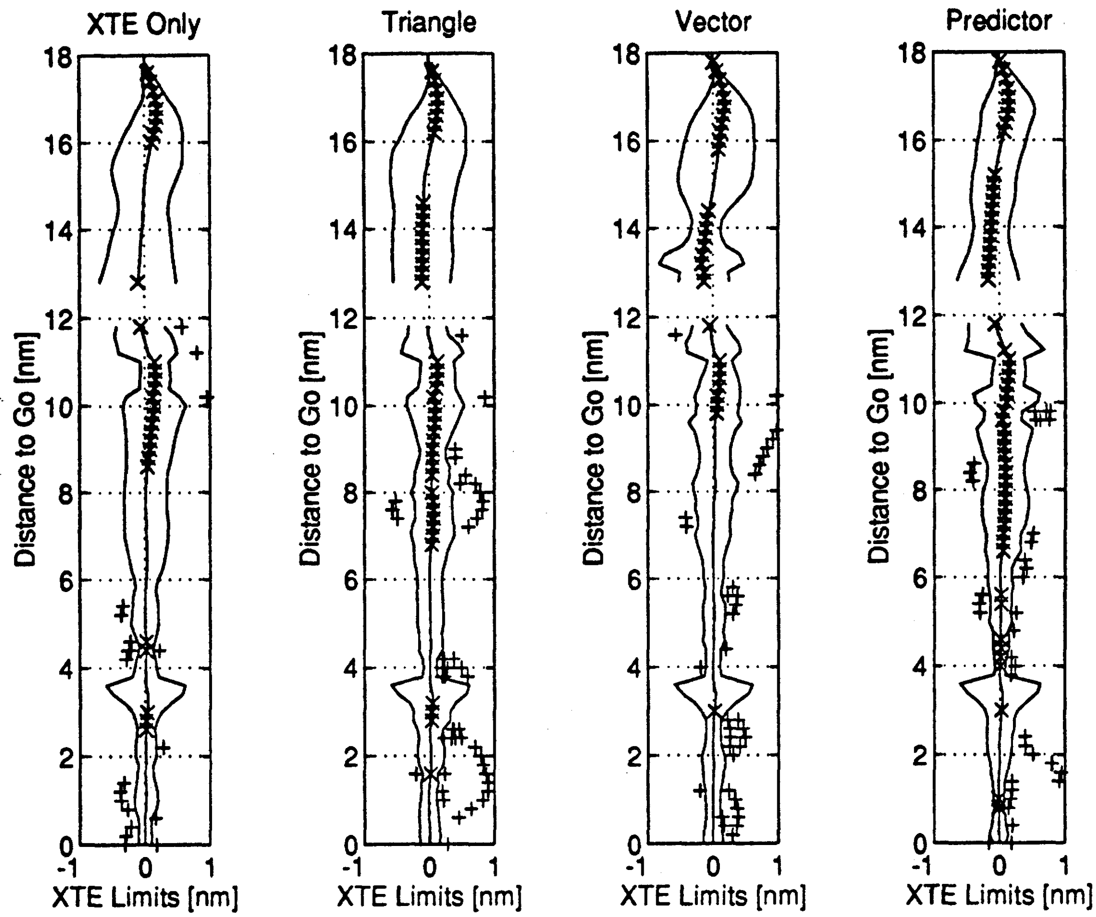


Figure 6.4: Plots of the 95% limit envelopes of all displays when all subjects are included in the analysis.
a) [far left] XTE Only display, b) [middle left] Triangle Same display,
c) [middle right] Track Vector display, d) [far right] Predictor display.
Tracks have been normalised for a crosswind from the right of the aircraft (from the left in the figure).
Both Arc and Straight approaches have been included.
Plus (+) marks indicate points excluded from the computation because they are outside the 99.99% limit.
Cross (x) marks indicate slices whose mean is significantly different from zero based on a 5% T-test.

The line showing average XTE deviation (the one with the cross [x] marks) clearly shows that pilots were generally blown down wind when they were unable to track zero XTE. The only exception occurred in the arcs and it indicates that the pilots were drifting outside of the second half of the arc, perhaps purposefully erring towards the side closest to the MAP. During the arc segment (18 to 13 miles) it is apparent that the average XTE deviation was initially deflected away from the prevailing wind and toward the centre of the arc. All displays crossed zero XTE at roughly the same distance and then it was only the XTE Only display which continued to track zero XTE, although the variance simultaneously increased. These plots include both straight and arc approaches so it is difficult to see what is happening between 12 and 10 miles. After the turn (from 10 miles) it was the Track Vector display which first returned to tracking zero XTE after a small deviation downwind. From the FAF (at 6 miles) until the MAP is reached (at 0 miles) all displays basically followed zero XTE with a small downwind disturbance being caused by the step. This disturbance is primarily due to the step itself and the reinitialization it performs on the aircraft state. Once the aircraft has been placed in its new position it takes the pilot a short period of time to initiate a correct intercept heading. During this time the aircraft was inevitably blown downwind because the initial heading was set up to be parallel to the runway heading and thus did not include the appropriate correction for the cross-wind.

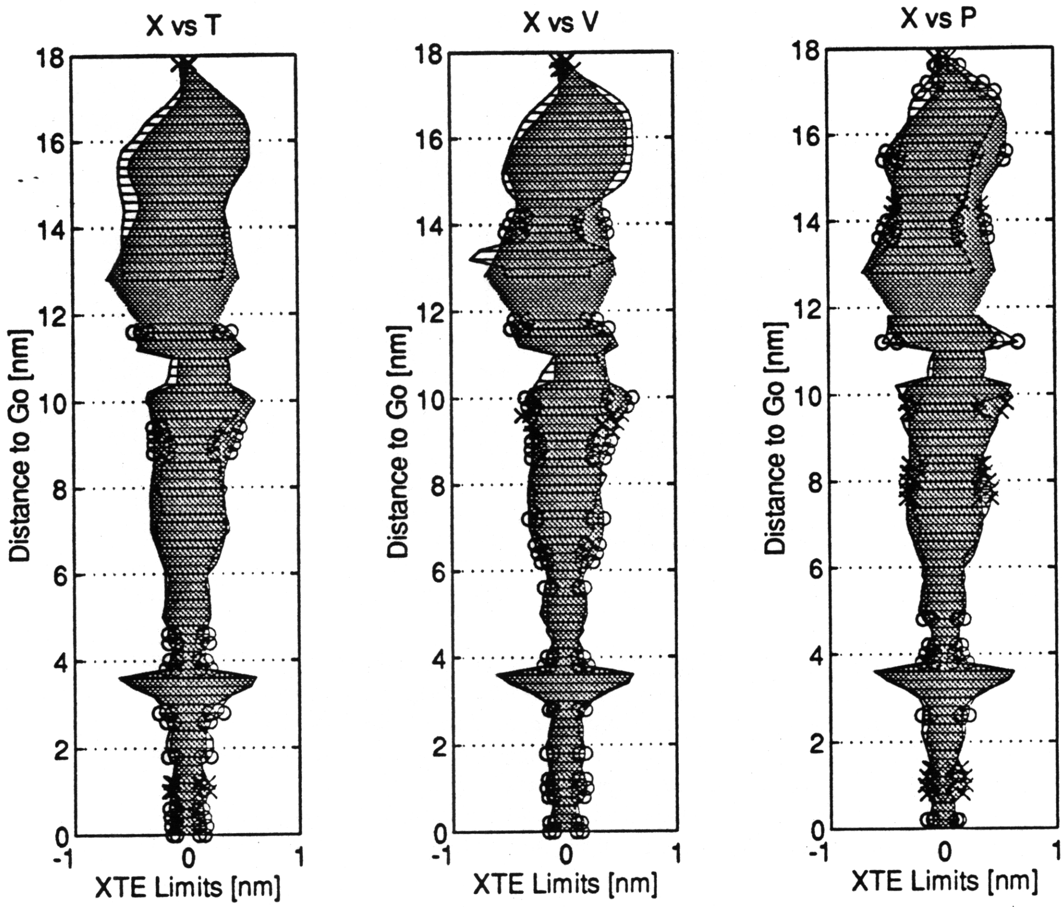


Figure 6.5: Plots comparing the 95% limit envelope of the XTE Only [X] display with those of the analog displays when all subjects are included in the analysis.
a) [left] XTE Only [X] display (grey shading) versus Triangle Same [T] display (horizontal lines)
b) [middle] XTE Only [X] display (grey shading) versus Track Vector [V] display (horizontal lines)
c) [right] XTE Only [X] display (grey shading) versus Predictor [P] display (horizontal lines)
Cross (x) marks indicate slices whose means are significantly different from each other based on a 5% T-test
Circle (o) marks indicate slices whose variances are significantly different based on a 5% F-test.

Comparisons between the envelopes are better shown in Figures 6.5 and 6.6. Figures 6.5 compare the XTE Only display with all of the analog TAE displays. These plots show the main advantages and disadvantages of the analog formats. Firstly, it can be seen that during the arcs the Triangle Same display can not be distinguished from the XTE Only display. The Track Vector and Predictor displays were marginally better than the XTE Only display during the arc, particularly after the initial disturbance had settled out. Soon after the turn the Track Vector and Triangle Same displays showed a small decrease in the size of the XTE envelope over the XTE Only display. This continues for the Track Vector display, in some small part at least, right down to the step (at 3.7 miles). The other displays showed little improvement over the XTE Only display. The Triangle Same display actually showed a decrease in XTE performance after the step and continuing down to the MAP. An ANOVA on the logarithm of the MAP XTE RMS statistic (RMS XTE for the two miles immediately before the MAP - 2 nm to 0 nm) showed, however, that the Triangle Same display was not significantly different from the other three displays, or even the XTE Only display by itself (Figures 6.14), although, the MAP XTE RMS statistic ignores the data immediately after the step, thus excluding the portion where the Triangle display appears to differ most from the other displays. The details of the effects of the step and the reintercept after this disturbance are covered later in the results from the model fit.

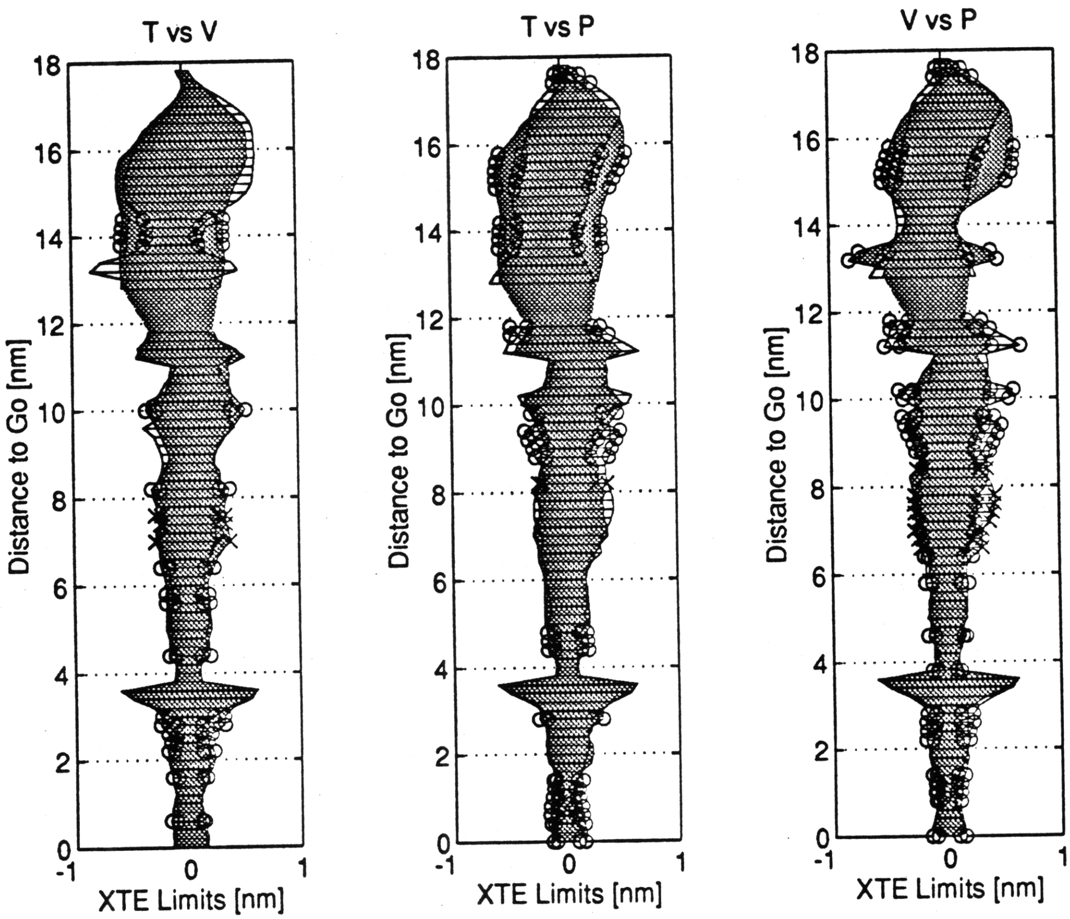


Figure 6.6: Plots comparing the 95% limit envelopes of all of the analog displays when all subjects are included in the analysis.
a) [left] Triangle Same [T] display (grey shading) versus Track Vector [V] display (horizontal lines)
b) [middle] Triangle Same [T] display (grey shading) versus Predictor [P] display (horizontal lines)
c) [right] XTE Only [V] display (grey shading) versus Predictor [P] display (horizontal lines)
Cross (x) marks indicate slices whose means are significantly different from each other based on a 5% T-test
Circle (o) marks indicate slices whose variances are significantly different based on a 5% F-test.

Figures 6.6 compare the envelopes of all of the analog TAE displays. These plots show that, during the arcs, the Track Vector and Predictor displays were similar to each other and both were frequently better than the Triangle display. After the turn and down to the FAF, the Predictor envelope is frequently wider than the Track Vector envelope. In many instances this is due to the offset of the Predictor envelope from a zero mean. Overall, the Track Vector envelope is narrower than those for both of the other displays. The Predictor displayed tends to be an improvement over the Triangle Same display along most of the run. The exception was soon after the turn, perhaps indicating a difficulty for most pilots when recovering from a fast manoeuvre using this display.

Examining the most compelling points from the above subjective description of the 95% limit plots brings out only one strong effect. This effect is that, for all pilots, it was the Track Vector display which gave a significant improvement in performance, but it did so only during the arc segment of this approach - at least when only examining the along track aggregate statistics by use of ANOVA tests. The Arc XTE RMS statistic (XTE RMS computed by aggregating the samples taken between 16 and 14 miles) showed a significant effect due to the Track Vector display when compared with the other three displays ($F(1,108) = 4.37$, $p < 0.039$) and also due to the Roll RMS group of the subject ($F(1,108) = 26.4$, $p <$ 1e-5) but there was no significant interaction. This is shown clearly in Figures 6.7 which plot the mean value of ln(Arc XTE RMS) when split into these various categorisations.

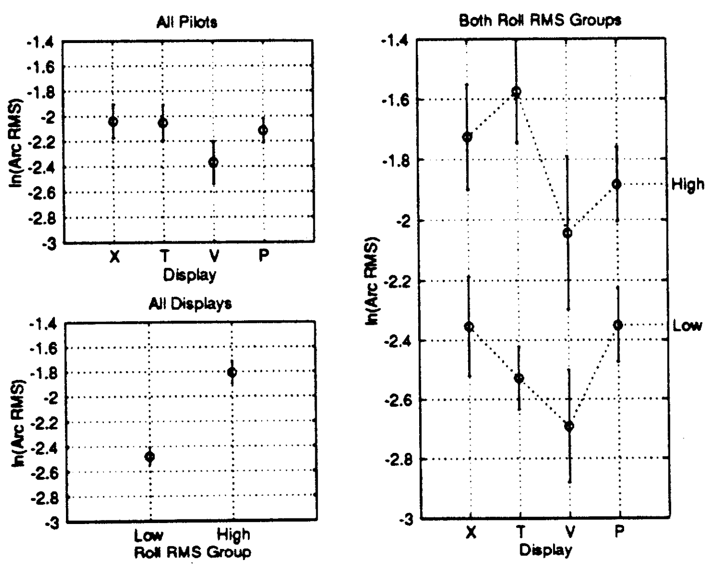


Figure 6.7: a) [Top Left] Mean value of ln(Arc XTE RMS) plotted by display for all subjects, b) [Bottom Left] Mean value of ln(Arc XTE RMS) plotted by subject group for all displays, c) [Right] Mean value of ln(Arc XTE RMS) plotted by display and subject group.

It is interesting to note that this plot also shows that pilots in the high Roll RMS group tended to perform less well with the Triangle Same display, although the effect was not significant. Also, the effect of improved performance using the Track Vector was not significant in either of the individual groups but it is apparent from this plot that both groups of pilots did benefit from this display.

Figures 6.7 also show an important comparison between the two pilot groups. Namely, that the low Roll RMS group generally also had the lowest XTE RMS - in this case demonstrated during the Arc segment of the approach. This trend was consistent through all of the XTE RMS statistics. Comparing this plot with the similar plot of Roll RMS it can be seen that (at least for the low Roll RMS group and partially for the high Roll RMS group) the mean value of Arc XTE RMS was inversely related to the mean value of Roll RMS when compared by display but directly related when compared by subject type. Thus, it was the analog TAE displays which showed the lowest XTE RMS even though they were associated with a higher Roll RMS. By virtue of the fact that the subjects in the low Roll RMS group had the lowest XTE RMS there would seem to be a contradiction. Pilots in the low Roll RMS group were able to fly closer to the desired track using fewer roll excursions than those in the high Roll RMS group. When these pilots used the analog TAE displays they, however, increased their roll excursions and by doing so reduce their XTE RMS. It is highly likely that the extra information in the analog TAE displays was used by the pilots to make sharper and more accurate corrections to their flight path, thus trading some of their inner loop (the stabilisation of aircraft attitude) performance in order to obtain a higher outer loop (the control of XTE) performance. The high Roll RMS pilots appeared similarly able to use this skill but not as consistently.

6.3.2. Low Roll RMS pilots

The most striking features of the 95% limit plots for the low Roll RMS group of pilots (Figures 6.8) are the consistency and narrowness of the envelopes, particularly in comparison to the same plots for the high Roll RMS group (Figures 6.11). These pilots made far fewer large excursions away from the desired course and also consistently maintained a low XTE centred on the course. This is particularly apparent during the Arc segment of the approach. Although it is difficult to infer any results about the turn from these plots because of the combination of the two approach types at this point, it is very interesting to note that the Track Vector display shows no widening of the envelope due to the execution of the turn. This suggests that using this display these pilots were able to follow the computer directed spline around the corner very well.

The two sets of comparison plots shown in Figures 6.9 and 6.10 illustrate a number of other interesting trends for this pilot group. During the arc segment, it is difficult to note any effects, except, perhaps, for a slight improvement of the Track Vector display over the XTE Only display. This is reflected in Figure 6.7(c), although, that plot suggests a larger difference between the Track Vector and Predictor displays that is not apparent in the 95% limit plot (Figure 6.10(c)). Either side of and during the turn, the Track Vector display showed great improvement over all of the other displays, but an ANOVA did not show any significant effect when examining the Turn RMS statistic. During the straight tracking portion after recovering from the turn and prior to the FAF, there appeared to be no differences between any of the displays.

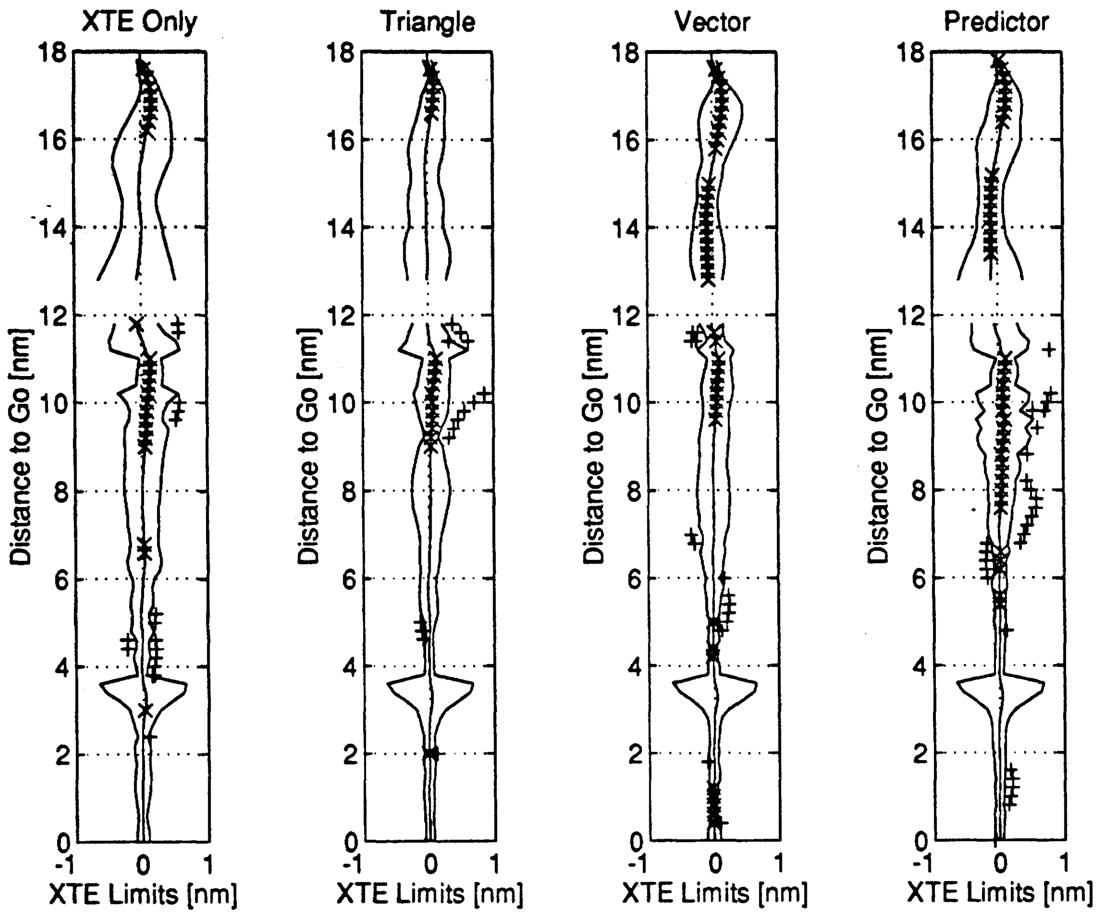


Figure 6.8: Plots of the 95% limit envelopes of all displays when only the Low Roll RMS subject group is included in the analysis. a) [far left] XTE Only display, b) [middle left] Triangle Same display, c) [middle right] Track Vector display, d) [far right] Predictor display. Tracks have been normalised for a crosswind from the right of the aircraft (from the left in the figure). Both Arc and Straight approaches have been included. Plus (+) marks indicate points excluded from the computation because they are outside the 99.99% limit. Cross (x) marks indicate slices whose mean is significantly different from zero based on a 5% T-test.

The greatest effect occurred just prior to and during the initial portion of the descent. This segment (from 7 nm to 4 nm), referred to as 'Long Final', showed significant improvement over the XTE Only display for all of the analog TAE displays. Amongst the analog TAE displays, the Track Vector and Triangle displays were similar to each other with the Predictor display showing some minor improvements over both of them.

An ANOVA on ln(FAF XTE RMS), however, showed no significant effects chiefly because of the low number of points available for this analysis. There was a similar trend during the Short Final segment (ie. after the step, just prior to the MAP), but it was not significant under either analysis. The comparisons between the various analog TAE displays during the Short Final segment showed these all to be similar.

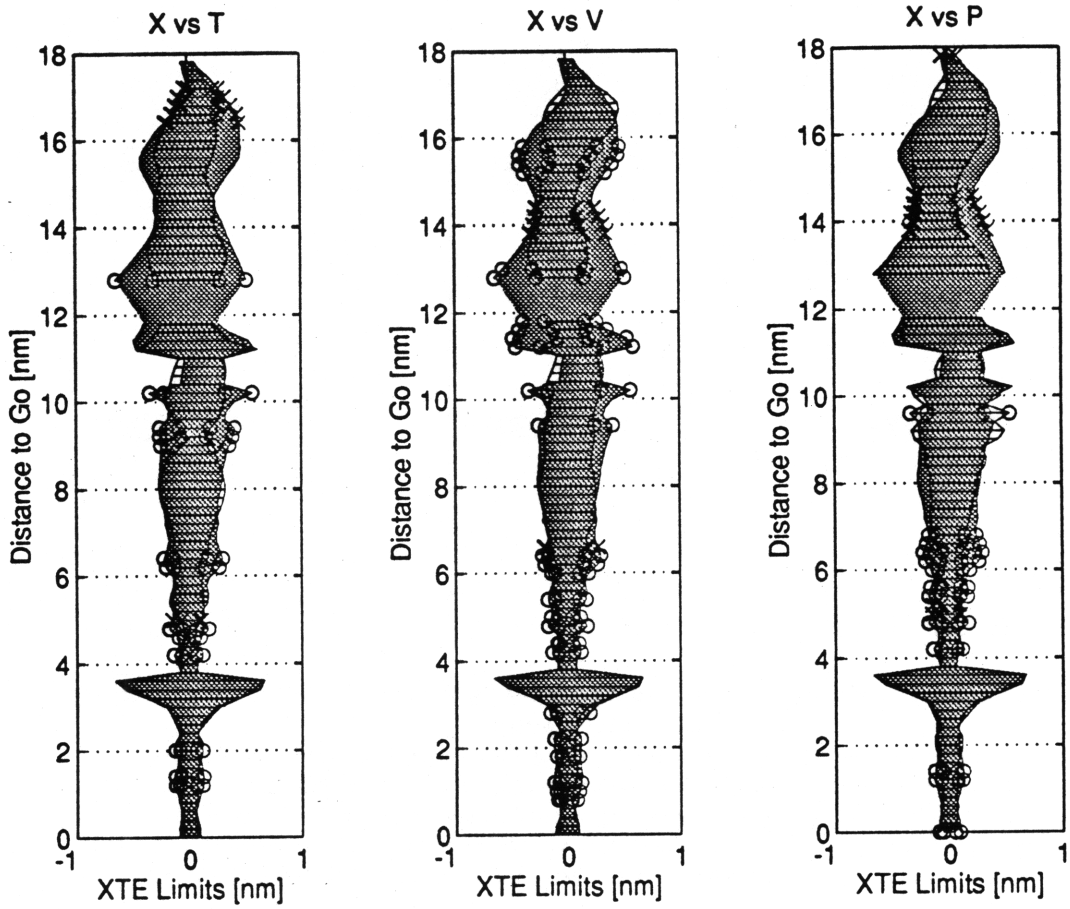


Figure 6.9: Plots comparing the 95% limit envelope of the XTE Only [X] display with those of the analog displays when only the Low Roll RMS subject group is included in the analysis.
a) [left] XTE Only [X] display (grey shading) versus Triangle Same [T] display (horizontal lines)
b) [middle] XTE Only [X] display (grey shading) versus Track Vector [V] display (horizontal lines)
c) [right] XTE Only [X] display (grey shading) versus Predictor [P] display (horizontal lines)
Cross (x) marks indicate slices whose means are significantly different from each other based on a 5% T-test
Circle (o) marks indicate slices whose variances are significantly different based on a 5% F-test.

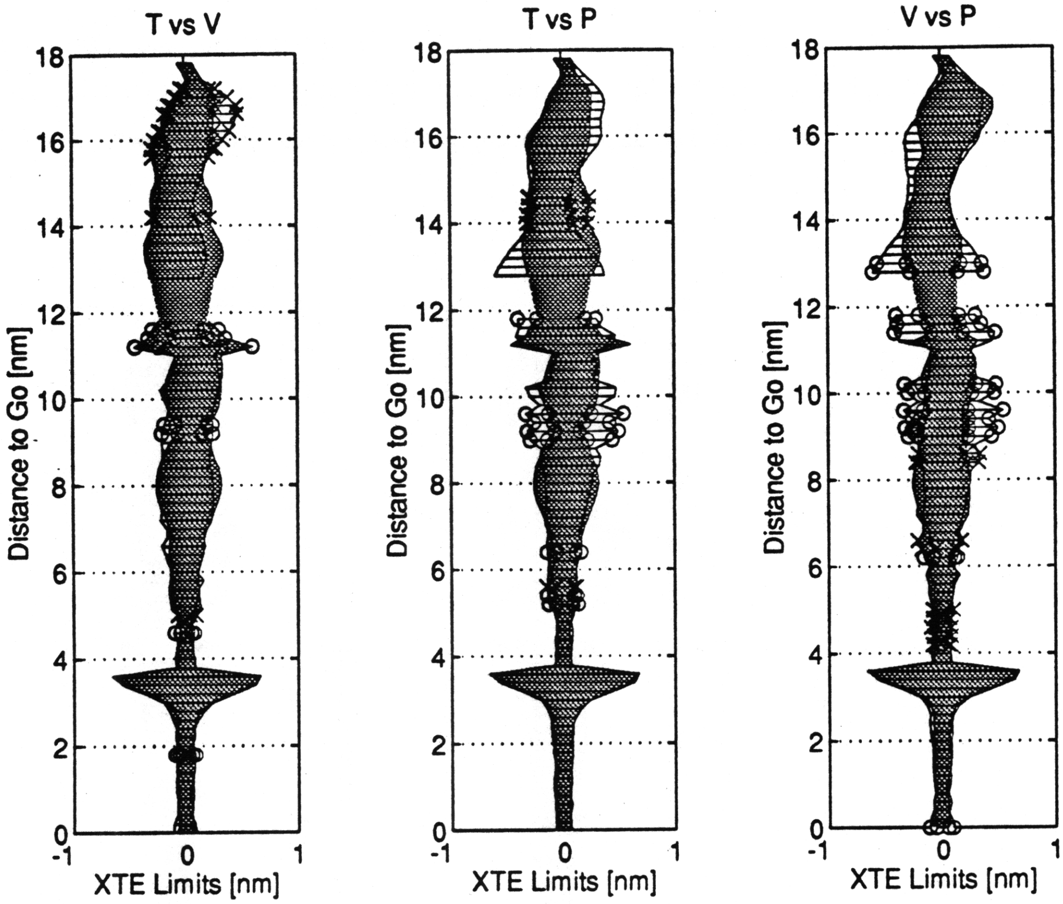


Figure 6.10: Plots comparing the 95% limit envelopes of all of the analog displays when only the Low Roll RMS subject group is included in the analysis.
a) [left] Triangle Same [T] display (grey shading) versus Track Vector [V] display (horizontal lines)
b) [middle] Triangle Same [T] display (grey shading) versus Predictor [P] display (horizontal lines)
c) [right] XTE Only [V] display (grey shading) versus Predictor [P] display (horizontal lines)
Cross (x) marks indicate slices whose means are significantly different from each other based on a 5% T-test
Circle (o) marks indicate slices whose variances are significantly different based on a 5% F-test.

6.3.3. High Roll RMS pilots

The main impression from the 95% limit plots for the high Roll RMS group of pilots (Figures 6.11) is that the envelopes are more jagged and much wider than the low RMS group counterparts. It is also easier on these plots to notice the strong influence of outliers. Between 10 and 8 miles in the Track Vector envelope it is apparent that the large spike just after 10 miles was due to the single outlier that is later removed by the 99.99% outlier exclusion but which re-enters the analysis just prior to 8 miles. An examination of the individual ground tracks showed that this single track was also mostly responsible for the peak just before the turn (at 13.2 miles) and was, in fact, a case in which the pilot turned too early and completely cut the corner. This track registered no slices between 13.2 miles and 9.6 miles.

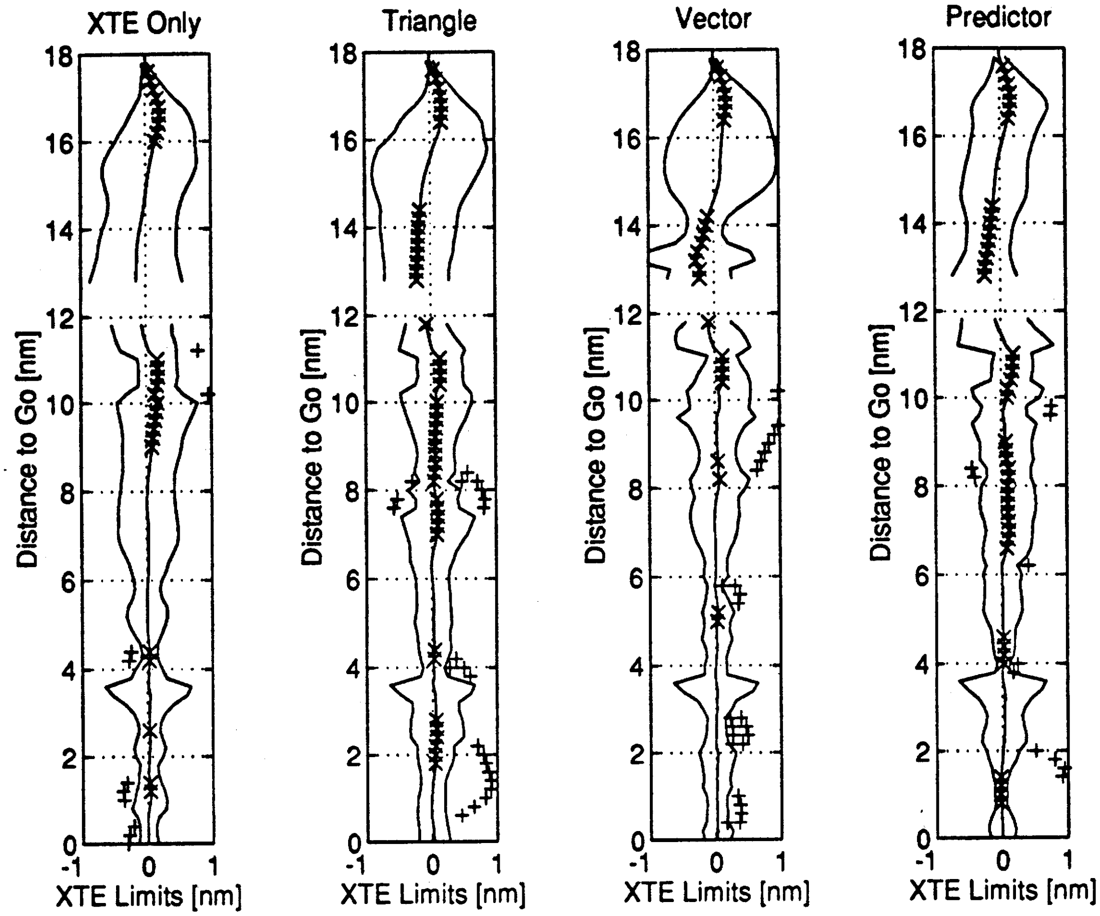


Figure 6.11 Plots of the 95% limit envelopes of all displays when only the High Roll RMS subject group is included in the analysis. a) [far left] XTE Only display, b) [middle left] Triangle Same display, c) [middle right] Track Vector display, d) [far right] Predictor display.
Tracks have been normalised for a crosswind from the right of the aircraft (from the left in the figure).
Both Arc and Straight approaches have been included.
Plus (+) marks indicate points excluded from the computation because they are outside the 99.99% limit.
Cross (x) marks indicate slices whose mean is significantly different from zero based on a 5% T-test.

For these pilots, the comparisons between the XTE Only display and the analog TAE displays (Figures 6.12) showed little advantage of the analog displays. During the arc portion, the Triangle Same [T] and Track Vector [V] envelopes are actually wider, although this was not a significant effect. Given the evidence of the Arc XTE RMS statistic, it is possible that the widest section of the Track Vector envelope during this segment was caused by only a very few tracks. If these had been excluded this display may have shown a closer resemblance to the Predictor display, as indicated by Figures 6.7. All through the approach the analog displays often show a wider envelope than the XTE Only display. During the Short Final segment many of these came out to be significant in the 95% limit plots, although an analysis of the MAP XTE RMS statistic showed no significant effects.

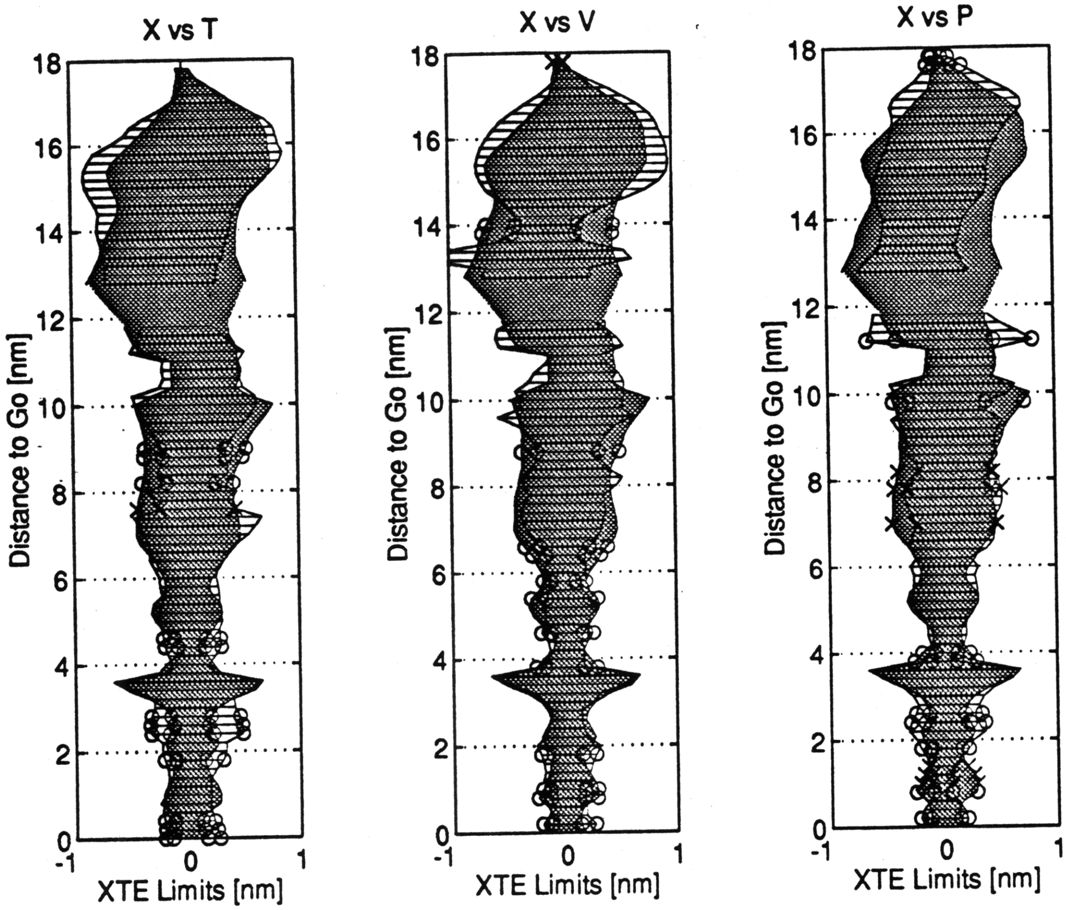


Figure 6.12: Plots comparing the 95% limit envelope of the XTE Only [X] display with those of the analog displays when only the High Roll RMS subject group is included in the analysis.
a) [left] XTE Only [X] display (grey shading) versus Triangle Same [T] display (horizontal lines)
b) [middle] XTE Only [X] display (grey shading) versus Track Vector [V] display (horizontal lines)
c) [right] XTE Only [X] display (grey shading) versus Predictor [P] display (horizontal lines)
Cross (x) marks indicate slices whose means are significantly different from each other based on a 5% T-test
Circle (o) marks indicate slices whose variances are significantly different based on a 5% F-test.

Comparisons between the various analog TAE displays (Figures 6.13) again showed only minor differences and insignificant trends. If anything can be read from these plots, perhaps it is that during the arc segment the Predictor [P] display showed an improvement over both of the other displays. The reverse, however, is true in the Short Final segment. In between, it is perhaps the Track Vector [V] display which showed the narrowest envelope. Neither did any of the along track, aggregate statistics support any significant findings.

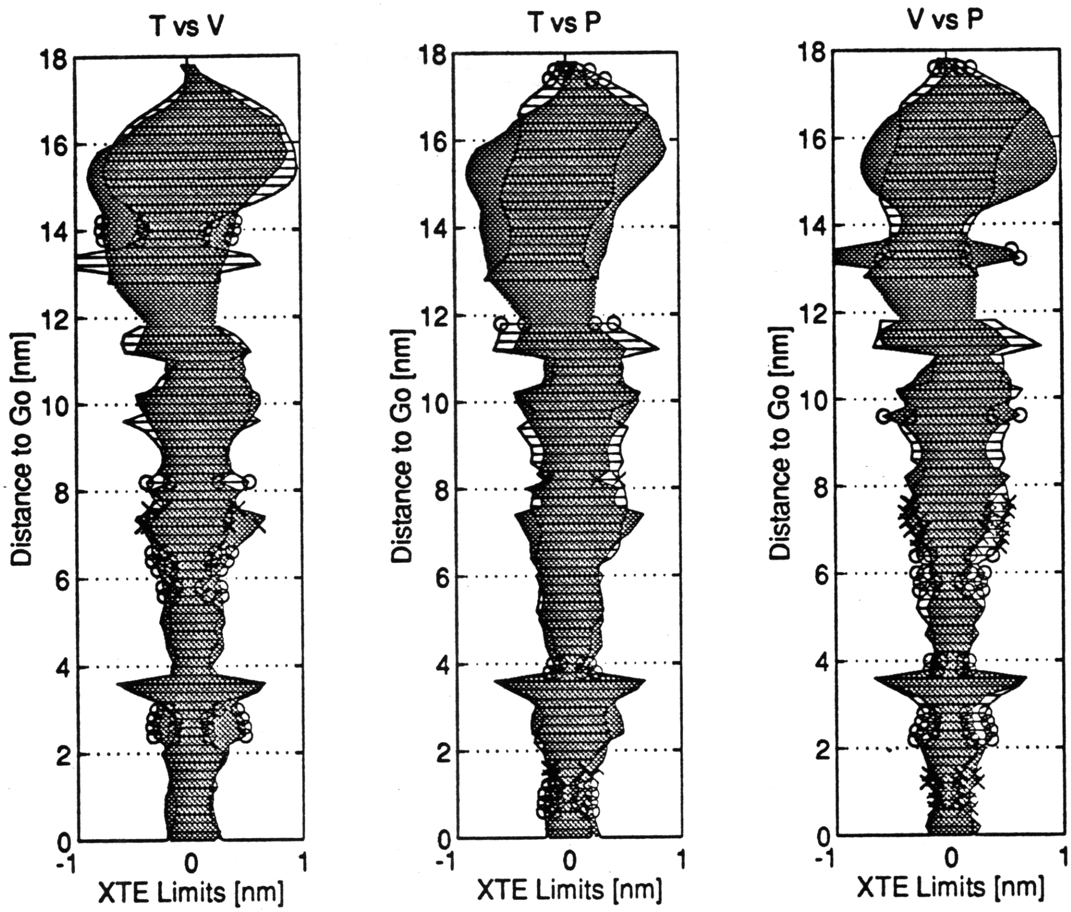


Figure 6.13: Plots comparing the 95% limit envelopes of all of the analog displays when only the High Roll RMS subject group is included in the analysis.
a) [left] Triangle Same [T] display (grey shading) versus Track Vector [V] display (horizontal lines)
b) [middle] Triangle Same [T] display (grey shading) versus Predictor [P] display (horizontal lines)
c) [right] XTE Only [V] display (grey shading) versus Predictor [P] display (horizontal lines)
Cross (x) marks indicate slices whose means are significantly different from each other based on a 5% T-test
Circle (o) marks indicate slices whose variances are significantly different based on a 5% F-test.

Figures 6.14 show plots of the mean value of ln(MAP XTE RMS). Although there were no significant effects between displays, some of the trends also apparent in the 95% limit plots are more easily summarised this way. Particularly, this plot highlights the trend that the Triangle Same [T] display had a larger XTE RMS especially in the high Roll RMS group of pilots. The only significant effect was that of Roll RMS group ($F(1,190) = 51.2$, $p < 1e\text{-}10$).

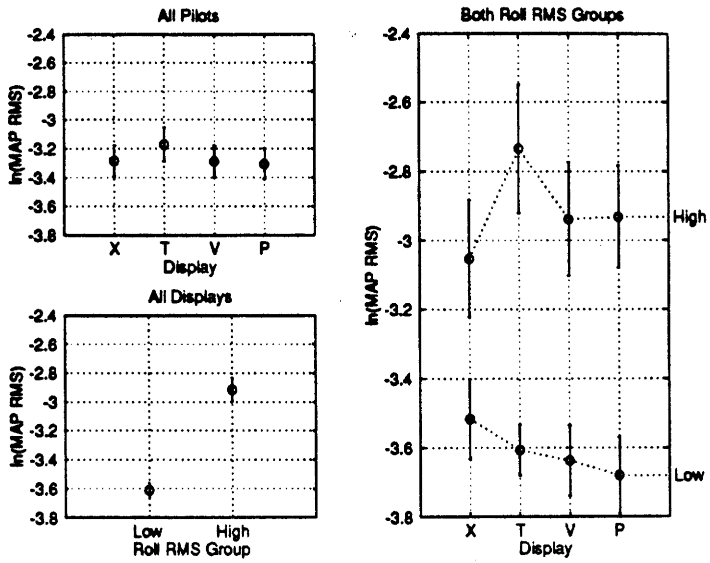


Figure 6.14: a) [Top Left] Mean value of ln(MAP XTE RMS) plotted by display for all subjects, b) [Bottom Left] Mean value of ln(MAP XTE RMS) plotted by subject group for all displays, c) [Right] Mean value of ln(MAP XTE RMS) plotted by display and subject group.

6.4. Intercept Statistics

Second Order model fits to the step disturbance showed excellent smoothing of the tracks. Figures 6.15 show the results for one of the subjects as an example of how these fits turned out. Each of the plots in this figure shows XTE plotted against distance from the point of the step disturbance. The dashed line represents the original data and the solid line the model fit, the parameters for which are given under the plot.

Of all of the statistics gathered from the model fitting process the most interesting appeared to be MAXANG. This value represented the maximum intercept angle used by the pilot on that run as recorded in the smoothed track of the model fit. In particular, it appeared to be characteristic of the general intercept strategy intended by the pilot. Figures 6.16 show plots of the mean value of this statistic both by display and by Roll RMS group. An ANOVA using MAXANG as the dependent variable demonstrated strong effects of display type ($F(3,144) = 4.33$, $p < 0.006$) and individual subject ($F(11,144) = 2.24$, $p < 0.016$) but no interaction effect. Using Roll RMS group as a factor instead of the individual subjects showed only a marginal effect ($F(1,184) = 3.15$, $p < 0.08$).

This marginal effect of Roll RMS group is most likely due to the fact that most of the variation in the intercept angle appeared to be associated with the high Roll RMS group. In particular, the low Roll RMS pilots did not appear to change their intercept response when using the various different displays. This confirms what was evident in the limit envelope plots - that it was not possible to distinguish the envelopes for these pilots during the final stages of the approach. The high Roll RMS pilots, on the other hand, showed significant differences between the displays, mostly because of the shallow intercept angles used when flying with the Predictor ("P") display. Also, the Low Roll RMS pilots showed a tendency towards using an intermediate angle that was similar to that associated with the XTE Only display in the High Roll RMS group. This angle, about 25°, appears to be what might be considered the typical intercept angle when not using TAE information. (That is assuming that the Low Roll RMS pilots started to ignore the analog TAE elements during the final approach as is suggested by this analysis and the previous XTE envelope and XTE RMS analyses.)

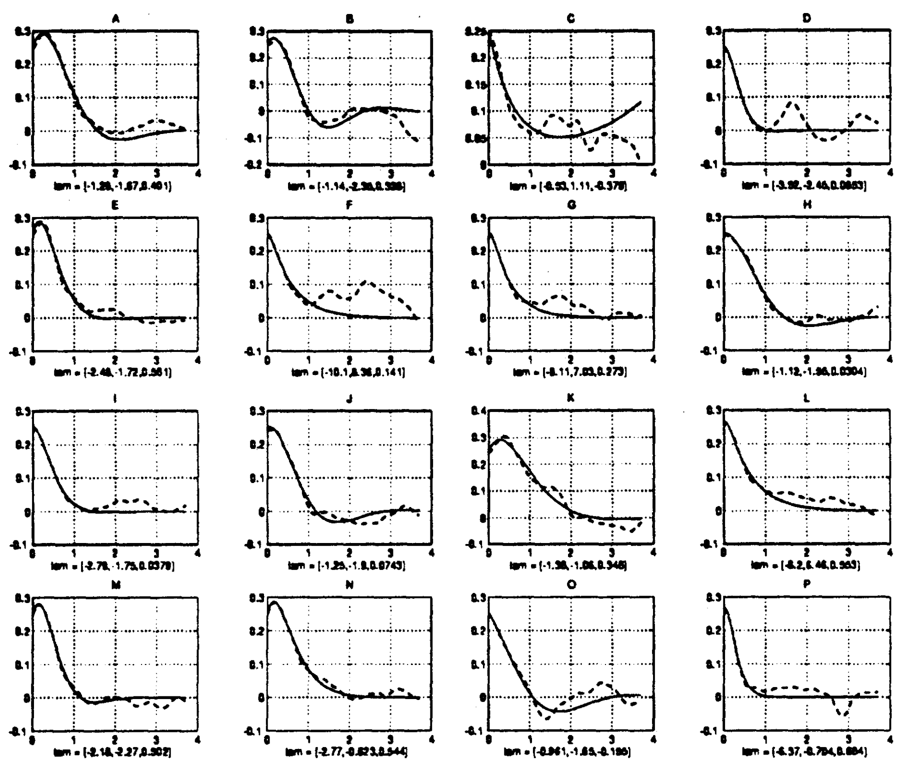


Figure 6.15: Individual model fits for subject 7. Plots of XTE vs distance from the step.
Dashed line is original data, Solid line is step fit.
The values under each plot recorded as lam = [l,m,n] show the model parameters.
l and m are the two values used to calculate the system matrix, n is the initial rate of change of XTE.

Examining the mean values of MAXANG for each display type within the High Roll RMS group it is possible to see differences between the analog TAE displays. Taking the mean angle achieved with the XTE Only ("X") display as a reference, there appeared to be two types of analog TAE displays. The trend was for the Predictor display to be associated with a shallow intercept angle and the Track Vector and Triangle Displays to be associated with a steep intercept angle. The shallow intercept angle associated with the Predictor display is what would be expected when the pilots used this display by following the prediction indication. Doing so would bring them back to the desired course on a gentle exponentially decreasing intercept with the maximum angle achieved being a function of the prediction time. The results suggest that the pilots were indeed following this strategy. Steep intercept angles for the Track Vector and Triangle displays, on the other hand, suggest that these were being used to intercept the course more aggressively than would be the case with the standard XTE display and that perhaps the TAE information was being used to attempt a faster intercept.

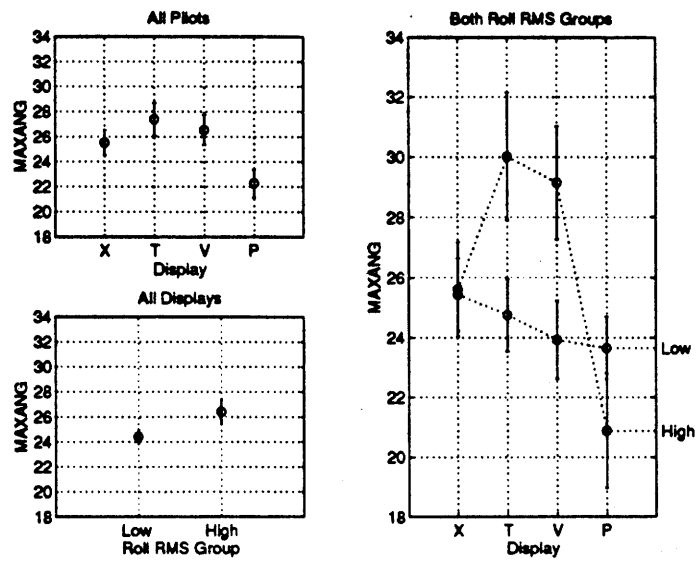


Figure 6.16: a) [Top Left] Mean value of MAXANG plotted by display for all subjects, b) [Bottom Left] Mean value of MAXANG plotted by subject group for all displays, c) [Right] Mean value of MAXANG plotted by display and subject group.

A plot of the ZERODIST statistic as shown in Figure 6.17 also shows the differences associated with the Predictor display. This statistic records the distance from the step at which the smoothed track first crossed the desired track - a measure of how quickly the pilot initially returned to zero XTE. In comparison to the MAXANG statistic this value looks at how the pilots completed the intercept. An ANOVA on the ZERODIST statistic showed significant effects of both individual subject ($F(11,144) = 1.91$, $p < 0.042$) and display type ($F(3,144) = 3.72$), $p < 0.0131$) but no interaction effect nor an effect due to Roll RMS group. In this case both sets of pilots demonstrated an equal propensity to have longer distances to intercept when using the Predictor display. Examining the individual data it was apparent that the larger mean zero crossing distance was mostly due to the high number of model fits using this display which never actually cross zero, in which case the zero crossing was recorded as the maximum distance possible (3.7 nm). (Generally, these never cross zero because the resultant system matrix had real roots and exhibited a strict exponential decay.) Even so, it still shows that the tendency was for pilots to use this display to intercept more gradually, although, as seen previously with the MAXANG statistic, the Low Roll RMS pilots appeared to always begin the intercept using the same strategy.

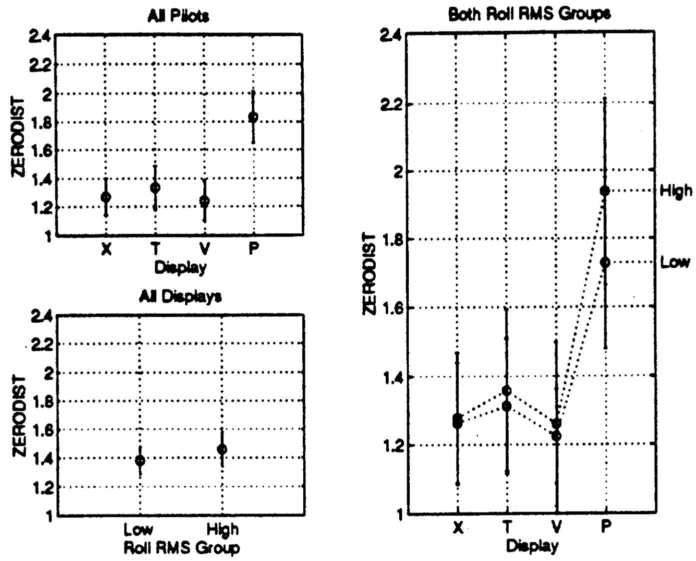


Figure 6.17: a) [Top Left] Mean value of ZERODIST plotted by display for all subjects, b) [Bottom Left] Mean value of ZERODIST plotted by subject group for all displays, c) [Right] Mean value of ZERODIST plotted by display and subject group.

6.5. Pilots' Opinions

6.5.1. Questionnaire Answers

Analysis of the post-session questionnaire data showed that pilots all had their own opinions about which display was best. Examining the five different methods of ranking the displays (tournament ranks derived from the head-to-head comparisons [HTH], ease of interpretation [EOI], effect on flight path accuracy [FPA], effect on overall performance [OP] and effect on overall workload [OW]) ease of interpretation was the only one which demonstrated any effect of display (Friedman rank ANOVA, $p < 0.02$). In general, as judges of the various displays, the pilots were demonstrably not of the same mind. Based on the rank sums for each display this would suggest that pilots were of the opinion that the XTE Only display was the easiest to interpret, the Track Vector was the hardest to interpret and the Triangle Same and Predictor displays were tied in the middle.

There was similarly no general agreement between pilots in each of the two Roll RMS groups, except, again, on the issue of ease of interpretation. In this case all but one of the Low Roll RMS group pilots listed the XTE Only display as the easiest to interpret, however, votes within that group for the other rankings were distributed equally among the other three displays.

Taking the rank sums at face value, ignoring the fact that the analysis shows no significant differences, and using them to assign overall ranks to all of the displays based on each of the opinion measures showed that the XTE Only display was ranked lowest for all of them. That is, forming a composite ranking based on all the display scale measures chosen for this study, the XTE Only is ranked as the most preferred because it consistently had the lowest score (Table 6.1).

Display Format	HTH	Display Preference Scales EOI	FPA	OP	OW
Predictor	33	33	33	30	32.5
Track Vector	32	36	33	35	32.5
Triangle Same	30.5	33	29	31	33
XTE Only	24.5	18	25	24	22
Friedman ANOVA	$p < 0.537$	$p < 0.02$	$p < 0.532$	$p < 0.377$	$p < 0.233$

Table 6.1: Rank Sums for the various preference scales summed by display for all pilots.

6.5.2. Workload Ratings

ANOVA analysis of the workload ratings collected after each approach showed significant effects of both subject and display. Ratings for the initial portion of the approach had display ($F(3,144) = 3.63$, $p < 0.015$) and subject ($F(11,144) = 12.2$, $p < 1e\text{-}15$) as significant factors but the cross factor was not significant. Figures 6.18 plot these mean values by display and Roll RMS group.

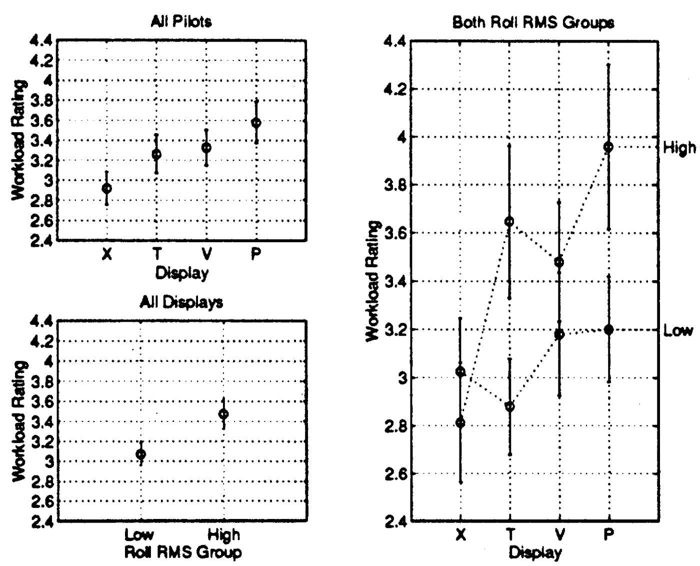


Figure 6.18: a) [Top Left] Mean value of initial workload rating plotted by display for all subjects, b) [Bottom Left] Mean value of initial workload rating plotted by subject group for all displays, c) [Right] Mean value of initial workload rating plotted by display and subject group.

The workload rating for the final portion of the approach had display ($F(3,144) = 4.33$, $p < 0.006$) and subject ($F(11,144) = 12.8$, $p < 1e\text{-}15$) as significant factors, however there was also a strong interaction effect ($F(33,144) = 2.3$, $p < 0.02$). Figures 6.19 show plots of the mean workload ratings of the final portion of the approach when grouped by display and Roll RMS group.

The effects due to Roll RMS Group on the workload ratings were also significant for both the final portion ($F(1,190) = 11.9$, $p < 0.0008$) and the initial portion ($F(1,190) = 4.66$, $p < 0.033$) of the approach. In general, the low Roll RMS pilots tended to give lower workload ratings for both segments. Notably, however, both groups gave comparable ratings for the XTE Only display in both segments of the approach. Their ratings differ mainly in the analog TAE formats with the High Roll RMS group giving ratings higher than those given by the Low Roll RMS group on the same display.

During the initial portion of the approach, pilots rated the Predictor as associated with the highest workload. The XTE Only display was generally associated with the lowest workload ratings and the Triangle Same and Track Vector displays were somewhere in between. On final approach the Track Vector display had the highest mean workload rating with the XTE Only display again having the lowest mean ranking. In this case the Triangle Same and Predictor displays were closely matched in the middle position. Generally, the initial portion of the approach was associated with lower mean workload ratings than the final portion of the approach. This can be seen by comparing Figure 6.18(a) and Figure 6.19(a).

Workload ratings ranged from 1.5 up to 8 for the initial workload score and up to 9 for the final workload score. The difference between the highest and lowest mean workload ratings by display is about one rating point which represents a 15% to 20% increase based on the 95% range of all workload ratings.

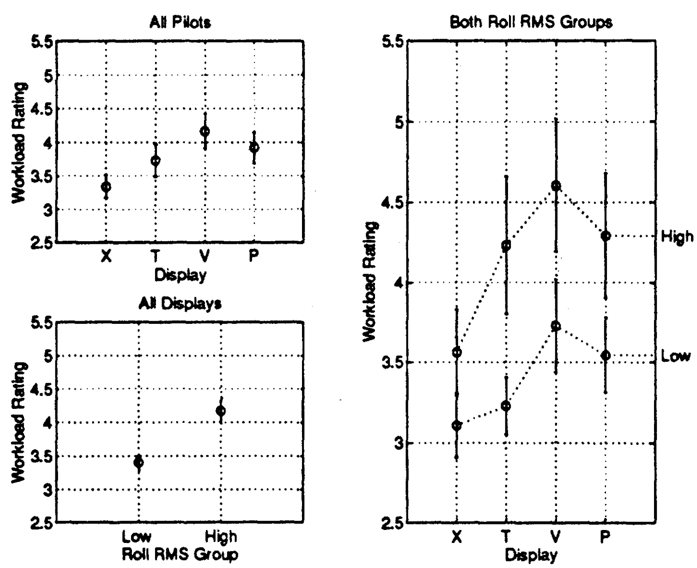


Figure 6.19: a) [Top Left] Mean value of final workload rating plotted by display for all subjects,
b) [Bottom Left] Mean value of final workload rating plotted by subject group for all displays,
c) [Right] Mean value of final workload rating plotted by display and subject group.

An ANOVA using subject and geometry (arc or straight) as factors showed no significant effect of geometry for the workload rating of the final portion of the approach, however, geometry ($F(1,168) = 6.33$, $p < 0.013$) and subject ($F(11,168) = 11.7$, $p < 1e{-}15$) were both significant for the workload rating of the initial portion of the approach. There was no significant interaction effect in this analysis. Not surprisingly, it was the Arc approaches which were associated with the highest mean workload ratings during the initial portion of the approach.

Examining the workload ranking data collected during the experiment it was apparent that the Predictor and Track Vector displays were ranked highest in terms of workload against the other two displays. The order of the rank sums put the Predictor display highest followed by the Track Vector, the Triangle Same and the XTE Only display as the lowest. There was a significant effect of display (Friedman Rank ANOVA, $p < 0.0009$).

In comparison to the preliminary study, the workload results recorded here contained much larger effects from the main experiment variables. The preliminary study reported that no variations were found in the workload scores suggesting that pilots maintained a constant workload level and varied their performance. In this study workload ratings may not be able to be compared between pilots due to the high variability between them but, despite this, the displays showed consistent effects associated with workload. Typically, pilots rated the Track Vector and Predictor displays as having the highest workload. The Triangle Same display rated next and the XTE Only display was fairly consistently rated as having the lowest workload. Differences between the two studies might be attributable to the reduction in the turbulence level. Perhaps, at the lower turbulence level of this experiment, the pilots had a small surplus of attention and were thus able to concentrate more on the tracking task - improving their performance at the cost of an increased workload. It is also possible that these effects existed in the previous study and that the increase in the number of subjects helped to make them apparent and, finally, this time around the pilots were given better instructions on using the Workload Scale helping them to be more consistent in their evaluations.

6.5.3. Training Effects

When the workload data was examined relative to the run number, there was a definite tendency for the pilots to report lower workload ratings later in the experiment, although, ANOVA tests showed that run number was not a significant effect on either workload rating or was at best a marginal factor. The trend towards lower ratings was primarily due to the low Roll RMS pilots. Considering the initial ratings of only the first four runs for the low Roll RMS pilots, run number was a significant factor ($F(3,20) = 3.10$, $p < 0.05$). Figures 6.20 plot both sets of workload ratings for all pilots and both subgroups. As can be seen in this figure the low Roll RMS pilots showed a sharp decrease in their workload ratings over the first four runs, particularly for the initial portion of the approach. The high Roll RMS group showed no such decrease.

In contrast, there was no apparent effect of run number on the various XTE RMS statistics, suggesting that the low Roll RMS pilots chose to maintain a consistent performance level even though their workload was decreasing due to increasing familiarity with the experiment and the displays.

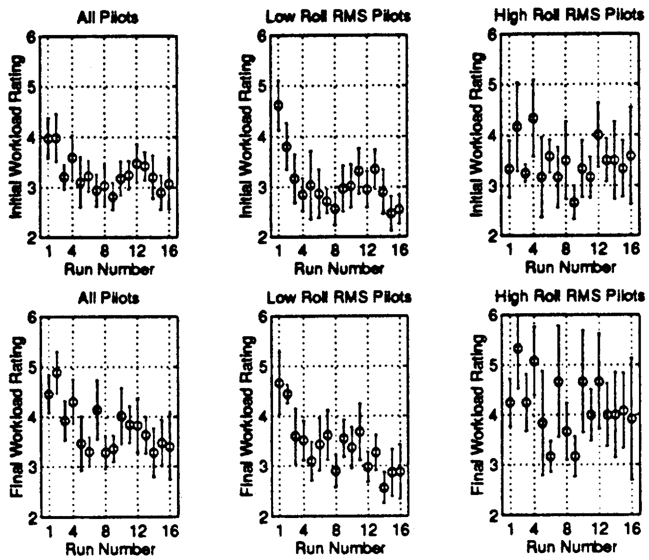


Figure 6.20: Mean Workload Ratings plotted by run number.
a) [Top Left] Initial Workload for all pilots, b) [Bottom Left] Final Workload for all pilots,
c) [Top Middle] Initial Workload for low Roll RMS pilots, d) [Bottom Middle] Final Workload for low Roll RMS pilots
e) [Top Right] initial Workload for high Roll RMS pilots, f) [Bottom Right] Final Workload for high Roll RMS Pilots

7. CONCLUSIONS

This study set out to determine whether or not analog Track Angle Error (TAE) displays were in any way better then displays that only presented numeric information. The means of investigating this issue was to present a number of TAE displays to pilots during a very demanding instrument flying task. Each of the displays was placed out side of the primary field of view and set up to resemble what might be available on a retrofit type device available to General Aviation operators. The basic result was that analog TAE displays did affect the way pilots fly under these circumstances.

The main effect was on the inner loop control task as measured by the Root Mean Square (RMS) value of the Roll Attitude (Roll RMS), the general trend being that the analog TAE displays were associated with higher Roll RMS values than the display with only numeric TAE information - an increase of about 10% in the mean RMS value. This percolated into the outer loop showing some trends associating the analog displays with differences in the widths of 95% probable location envelopes and the RMS values of certain Cross-Track Error (XTE) statistics. In particular, all pilots showed an association between the analog displays and low values of RMS XTE while tracking a circular arc - a difference of about 20% between the XTE RMS values for the XTE Only and Track Vector displays.

Certainly, however, the greatest differences showed up between pilots themselves, with all pilots demonstrating varying degrees of skill at maintaining close proximity to the desired track. A median split of the pilots based on their inner loop performance (ln(Roll RMS)) helped to remove some of these inter-subject effects and tease out some more information about the relationships between the displays. Those pilots in the group with mean RMS Roll values (inner loop performance) greater than the median were associated with very wide 95% probable XTE envelopes and high mean values of all of the RMS XTE (outer loop performance) variables. The other half of the pilots were associated with narrow envelopes and low mean RMS XTE statistics. Even so, within both groups, the association of the analog displays with high RMS Roll values and low RMS XTE values remained.

Comparing the workload results with the XTE tracking results it is apparent that the displays linked to high workload ratings, namely the Track Vector and Predictor, were also associated with the best XTE tracking performance. This suggests a trade-off between XTE performance and workload where the extra performance comes at the cost of extra workload, although the magnitude of the extra workload decreased with time.

The primary conclusion is that, for all pilots, analog TAE displays provide general help in reducing XTE most effectively during manoeuvres at low sensitivities of the XTE CDI scale. Such a conclusion is supported by the evidence of lower RMS XTE statistics during the arc portion of the approach for the Track Vector display and also by the fact that there was little evidence of differences between the analog and numeric TAE displays for the high XTE CDI sensitivity portions of the approach. A simple analysis of the value of TAE information would suggest that this would be the case for any presentation of TAE, whether numeric or analog, as, during the low sensitivity phases, TAE should provide fast information about the slow motions of the XTE needle. Why it was that the numeric display did not evidence this trend is perhaps worth further study, although the subjective analysis by the pilots themselves would suggest that the reason is simply that they did not find it sufficiently easy to incorporate the numeric TAE reading into their scan and therefore did not use it.

A supplementary conclusion is that some pilots were able to use the new analog TAE presentations better than others. This is evidenced by the strong differences between the median split pilot groups. The low Roll RMS pilots had narrower XTE envelopes when using analog TAE displays, not only when tracking the arc on the low CDI sensitivity, but also after the change to the high sensitivity when initiating the descent. During the intercept, and on short final, no performance advantage of analog TAE displays was apparent, except that intercept angles were consistently shallower with the predictor display. In contrast, the high Roll RMS group had poor XTE approach performance overall.

Of the three analog formats experimented with in this study it was perhaps the Track Vector display which showed the best qualitative effects. Indeed, it was only the Track Vector display which showed any real differences in XTE RMS statistics for all of the pilots. This display was also associated with a faster return to zero mean XTE after the 90° turn, than the other two analog displays. Within the low Roll RMS subject group, this display was associated with narrow envelopes during and immediately after the turn suggesting it was also helpful for this high rate manoeuvre. Along with the other two analog displays it was also associated with narrower envelopes during the initial descent. On the other hand, however, higher workloads were associated with this display than with the Triangle Same and XTE Only displays.

In general the analog displays were associated with higher performance but at the cost of a little extra workload. In fact, the Track Vector display has demonstrated up to a factor of two improvement in XTE performance over the XTE Only (numeric TAE) display when the CDI scale was at the low sensitivity. However, the Track Vector display was also associated with close to an 18% increase in workload.

To assess the relative advantages of each display, overall rankings by both flight performance and workload were summed to arrive at a single rating figure for each format. The results are shown in Table 7.1. A bias towards the flight performance ranks was applied, in order to break the tie, as these were considered more important than workload considerations. (A value of two is used in the table but any bias greater than one and less than three will give the same order.) The resulting ranking, from best to worst, was: Track Vector, numeric TAE (XTE Only), Predictor and Triangle Same. With the exception of the Triangle Same display, the analog TAE displays produced consistently better XTE tracking then the numeric TAE display. However, these advantages were offset to some degree by higher workload ratings. The XTE Only display, while not showing the best XTE performance was certainly associated with the lowest workload measures.

Display Format	Rankings Flight Performance (x2)	Workload (x1)	Weighted Rank Sum
Track Vector	1 (best)	3	5
Predictor	2	4 (highest)	8
Triangle Same	4 (worst)	2	10
XTE Only	3	1 (lowest)	7

Table 7.1: Final display rankings based on all of the results.

Of significant interest is the association between high RMS Roll and low RMS XTE when examining the data by display, but the reverse association when examining the data by subject. In particular it was the low Roll RMS group that had the lowest RMS XTE statistics, particularly when using the Track Vector display, but it was the Track Vector display which was associated with the highest RMS Roll values. Multi-loop manual control considerations provide a straightforward interpretation. Roll angle is simultaneously a measure of inner loop output and input to outer loop control of XTE. Those pilots with a poorer mastery of the inner control loop, possibly because of deficiencies in instrument scan resulting in greater inner loop delay, were not generally able to gain any advantage from the analog TAE displays. These pilots reported consistently higher workload ratings, particularly when using distracting analog formats. On the other hand pilots with better inner loop control performance appeared consistently able to take advantage of the analog TAE information by using higher roll angles to better control the outer loop and reduce their XTE when CDI sensitivity was low, as when tracking the arcs, and during the initial portions of the descent with high CDI sensitivity.

7.1. Recommendations for Further Study

Neither this study nor the preliminary study directly examined the differences between a numeric TAE display and a format which included no TAE information. In this study all formats had TAE information in some form and in the preliminary study all of the TAE displays included an analog element. In the end neither study provided any insight into how much pilots used the numeric only information. The first study showed that TAE information was indeed helpful and this study showed that analog was better than numeric TAE. Given the high final ranking of the XTE Only display in this study and pilots' comments that they variously did and did not use the numeric TAE information it would be sensible to include a comparison of no TAE with numeric TAE in any follow up work. This should be designed so as to tease out the details of when and how pilots use numeric TAE information.

Of the displays, it was the Track Vector which might be considered closest to a Moving Map type display. It was also this display which appeared to offer the most advantages. Most manufacturers are currently investing in the production of GPS devices that incorporate the ability to display a Moving Map type presentation. Further study could profitably focus on the use of such displays to either completely replace the analog TAE designs examined in the current study or which use these TAE displays to supplement a moving map display. Questions which should be answered are whether moving map displays actually provide as much TAE information as is contained in a dedicated analog TAE display or whether and how analog TAE and XTE CDI displays can be used to supplement moving map displays. It is generally felt that moving map displays should not themselves be used for navigation chiefly because of the low XTE sensitivity they provide. The Track Vector display could potentially be used in conjunction with a moving map to provide an increased XTE sensitivity while still retaining the character of the moving map, either by placing them both on one display or by putting the TAE display inside the pilot's primary field of view. An important regulatory question is whether the moving map and analog TAE displays should be permitted to be displayed together on a single display, or whether, to avoid the map being used as a CDI display, the map should be required to be hidden while engaged in primary flying tasks.

The preliminary study also showed that the magnitude of the detriment caused by moving the XTE CDI and TAE displays out of the primary field of view was significant. What still needs to be examined is the effect of moving a display of TAE into the primary field of view. Perhaps the Electronic HSI display discussed in Appendix A would be a suitable and cheaply implementable way of performing this modification and would provide superior tracking performance over the non-primary field displays already investigated. Given the evidence of the standard HSI results from the preliminary study and the tracks from the two pilots which used the Electronic HSI as a non-primary field instrument it is likely that this will indeed show great benefits even at the low update rate available from a standard GPS receiver. Such a study would need to be associated with research into the effects of placing what is effectively an automated OBS needle into an environment in which pilots will still need to manually operate the OBS needle during non GPS approaches. In fact, the general effects of the dual functionality that the HSI instrument would acquire would also be of interest. Is it important and are pilots able to easily switch between the angular XTE CDI scale on an HSI with a manual OBS indicator and a linear scale XTE CDI on an HSI with an automated OBS indicator?

Finally, it would be interesting to attempt to investigate the manual control issues associated with the relationship of inner loop control to outer loop performance when pilots are involved in the type of task that was associated with this experiment. It was not possible to study this in great detail in this work because of the low fidelity of the control input recordings. A much higher and more even sampling would be necessary to accomplish this.

APPENDIX A: ELECTRONIC HSI DISPLAY

Originally, the fourth display in the protocol was the Electronic HSI format showing Current Track on the RMI. The first two subjects were run using a protocol that used this display in place of the XTE Only display. It was dropped from the experiment for two main reasons: 1) it took up more space than was allowed in the design parameters for the displays, 2) it was decided that the XTE Only display needed to be included to allow for a baseline display against which to compare the other TAE formats. Table A.1. shows the demographic data for the two subjects that used this display and which were excluded from the analysis.

Subject Number	Age	Gender	Total Hours	Instrument Hours	Instructor Rating
1	43	M	2910	390	CFII
2	32	M	1110	123	CFII

Table A.1: Subject Demographics for the two excluded subjects

Although it is only a subjective analysis, the first two pilots showed very good performance with this display compared with the other TAE formats. No doubt this was due to the familiarity of this format and the fact that it did not appear to really provide any new information even though it actually did. A version of this display replacing a standard HSI (and thus in the primary field of view) would perhaps provide the best performance of any device yet considered. This is partly supported by the result in the preliminary study that the HSI format used then was amongst the better displays. By providing aircraft heading on the RMI scale the original functionality of the HSI could be kept and the TAE updates to the OBS needle every second or so from the GPS would probably not provide any distraction.

A.1. Description of the Electronic HSI

The "Electronic HSI" or "E" display presents TAE and XTE on an electronic version of an HSI like instrument. XTE is presented on the CDI scale in the centre of a compass rose. This XTE scale rotates with an OBS like needle that is slaved to TAE. The deviation of the OBS needle from vertical gives a direct reading of TAE. This display corrects the geometry problems with the Track Vector by correctly rotating the desired track relative to the aircraft position. It is also a display that many pilots already find familiar.

The compass card (RMI) can be used to indicate either current track or current heading. By indicating current track, all elements on the display are represented in a common axis system. In this configuration the OBS will always point to the compass direction which is the desired track. While following a curved segment or during a turn this should follow the desired track as it changes around the curve. Indicating current heading on this card confuses the axes systems but also provides a direct readout of the desired heading to follow any angular deviation from the course. For example, to track at 45° from the course it is a simple matter of flying to put the OBS at 45° and then the value at the top of the display will be the desired heading to maintain that track. In this case, however, the value pointed to by the OBS needle is a complicated function of the desired track, the current heading and the crosswind and is probably not a useful indication.

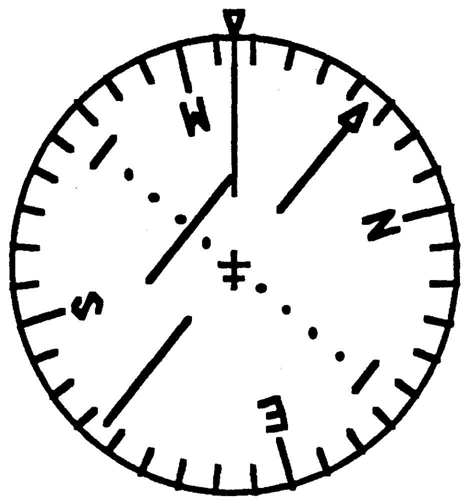


Figure A.1: Electronic HSI Format

The Electronic HSI format combines the properties of both advantages of a TAE measurement, particularly when used with the current heading indication on the compass card. As mentioned this provides a direct reading of desired heading. Also, the geometry of the system automatically indicates how quickly the aircraft is approaching or diverging from the course by showing the correct relationship between the aircraft and the current course. Unfortunately, a low resolution version of this display is not really feasible. This display would work best as a replacement for an existing HSI or to add an HSI to an old aircraft. One possibility for implementing this display is to augment an already installed HSI with a motor that could move the OBS needle in response to changes in TAE.

APPENDIX B: A HISTORY OF AIRCRAFT NAVIGATION

This Appendix presents, at least in part, a very brief summary of the basics of aircraft navigation. It is not intended to be an instructional manual and any reader interested in greater detail is referred to the many basic training books on the subject or to references describing the details of cockpit instruments and aircraft navigation.

B.1. Past and Present Navigation Instruments

The development of aircraft navigation systems is an interesting topic in its own right. This section presents a limited historical overview of that development, highlighting the major advancements and providing a framework that describes the environment in which this project was created.

Perhaps the greatest ever advancement in aircraft navigation came with the introduction of radio navigation aids. Prior to that point it was all up to the pilot to recognise known landmarks and navigate much the way any person located in a landscape devoid of man made features might do. Radio gave the ability to create landmarks that could be "seen" from large distances and which could also be very simply identified. Such "landmarks" were created as beacons or navigational aids that transmitted radio frequency radiation to any willing and able to receive it. In order of development, these aids were variously termed AN ranging, Non-Directional Beacons (NDB), VHF Omnidirectional Ranging (VOR), Distance Measuring Equipment (DME), Long Range Navigation (LORAN) and eventually Global Positioning System (GPS). Many of these aids are very costly to install and maintain and because an aircraft requires a clear line of sight to be able to use them a large number are required to cover any reasonable area. The current navigation systems in use today require constant calibration and maintenance of hundreds of NDB, VOR and DME beacons around the continental United States alone.

To use these beacons a pilot must tune the correct frequency on a receiving radio which is connected to an appropriate cockpit display. Typical such displays in traditional cockpits are the Automatic Direction Finder (ADF), Direction Measuring Equipment (DME), ILS Course Deviation Indicator (CDI) and the Horizontal Situation Indicator (HSI) . These are commonly called 'round dial' instruments because of the heritage of displaying them on mechanical displays with a round clock type face.

The advent of miniaturised electronics, micro-processors and multi-functional electronic displays is dramatically altering the cockpits of all aircraft. Unfortunately, the majority of General Aviation (GA) aircraft still use the old systems and cannot afford the upgrade costs to completely overhaul their cockpits. In this environment it is necessary to develop ways for GA pilots to benefit from the advances in navigational technologies without severely impacting their costs or their cockpits.

B.1.1. AN Ranging

AN Ranging was one of the earliest radio navigational aids used by pilots. Using this system a pilot would tune his communication radio to a transmitter located at the destination airport and listen to the message being broadcast. This message would either be morse code for "A" (dot dash) or "N" (dash dot). The signal was divided into four quadrants with the "A" message being broadcast in the first and third quadrants, and the "N" message in the other two quadrants. In this way, when the aircraft was between two quadrants the signals would interfere so that the pilot would hear a continuous tone and could follow this boundary by maintaining the aural indication. Generally, the quadrants would be established so that this would bring the aircraft down along a runway centreline. The system is no longer in use.

B.1.2. Automatic Direction Finder (ADF)

Another simple, early system was the Direction Finder. This started as literally a loop antenna which the pilot could manually rotate to find the strongest signal from a tuned radio beacon and thus determine its approximate direction relative to the aircraft. This was later automated requiring the pilot only to tune the correct frequency. The result was displayed as a needle on a fixed card indicating the direction relative to the aircraft.

The beacons used for tuning an ADF are referred to by the general term Non-Directional Beacons (NDB). These can be any simple radio transmitter and initially they were the local broadcast radio stations. Eventually, the FAA established a number of beacons designated for air navigation. Many NDB's were located along the extended centreline of a runway. Using these beacons it was possible to fly to the destination airport and then follow a particular magnetic direction away from the beacon that would bring the aircraft down along the runway direction. Co-located with these beacons is a special type of beacon called a marker beacon. When directly overhead of these marker beacons, visible and aural indicators alert the pilot to this fact. These beacons were later used to augment the ILS system. Many GA aircraft flying today still have an ADF display, although it is likely to be integrated with an RMI.

B.1.3. Remote Magnetic Indicator (RMI)

The advent of the Directional Gyro (DG) allowed avionics manufacturers to provide the pilot with a display of aircraft heading relative to magnetic North by means of a Remote Magnetic Indicator (RMI) display. The drift in the gyro requires that this display be corrected by reference to a compass but the ability to show this information has some significant advantages.

Chiefly, by adding an ADF needle on top of the RMI it suddenly becomes much easier for a pilot to follow published magnetic headings. The RMI, effectively, automatically performs the addition of aircraft heading to the ADF indication thus presenting it relative to magnetic North rather than as a relative bearing from the aircraft.

B.1.4. Very High Frequency (VHF) Omnidirectional Range (VOR)

An improvement on the simple NDB/ADF system was to add the ability to provide arbitrary directional tracks to and from a radio beacon. The VHF Omnidirectional Range (VOR) beacons provided this information. Using a VOR receiver a pilot is able to follow any desired magnetic track, or radial, to or from the beacon and to know how far off that track he is currently located. This is achieved in a way that it vaguely similar to the AN ranging system in that interference between signals indicates the correct course. Only the pilot doesn't need to listen to the signal itself. Also an intelligent use of signals provides for the ability to track any desired radial and also the angular deviation from that radial.

The angular deviation is typically displayed on a Course Deviation Indicator, which was originally a simple horizontal needle and a number to tell the pilot what magnetic direction was tuned. This control used to change this number is called the Omni-Bearing Selector (OBS) and can be used by the pilot to select the desired radial to follow. A supplementary flag indicates whether the aircraft is flying towards or away from the beacon. The scale on the CDI typically ranges between $\pm 10^{\circ}$. Due to the angular nature of the XAE measurement, though, as the aircraft flies closer to the beacon source, the linear sensitivity of this display increases. Also, in the presence of a crosswind, flying the heading indicated by the OBS will cause the CDI needle to slowly drift to one side and thus a small angular correction needs to be found that will stop the needle moving. This is a type of display called a "fly-to" display because, in order to centre the needle, it is necessary to fly the aircraft towards the needle. That is when the needle is to the left of centre, it is necessary to fly to the left and if it is to the right then it is necessary to fly to the right. When flying backwards along the desired radial, ie. away from the VOR, it is necessary to reverse these rules when using a simple CDI scale.

Today there exist numerous VOR beacons located around the United States, including at important regional airports. Combined with the network of NDBs they provide the primary means by which aircraft navigate. Each has its own waypoint designator and they are also used to define numerous other waypoints, associated air routes and intersections of air routes.

B.1.5. Instrument Landing System (ILS)

The Instrument Landing System (ILS) uses different technology to that of the VOR but it can be considered to be like two VOR beacons, one in the horizontal plane, called a Localiser, and the other in the vertical, referred to as the Glide Slope beacon. When placed appropriately relative to a particular runway it can be used to not only guide an aircraft along the correct ground track that is the runway centreline, but also along a glide slope that will bring the aircraft down exactly to the runway threshold.

The set of beacons that comprise an ILS usually begin with an NDB located at the start of the runway approach. This is co-located with the Outer Marker, a simple marker beacon. An ILS equipped aircraft will display aural and visual indications that the aircraft is directly above this beacon. Closer in to the runway threshold is a similar radio signal called the Middle Marker. These markers simply alert the pilot to the their distance from the threshold which is published on an instrument approach plate. Finally, there are the localiser and glide slope beacons.

The actual cockpit display combines two CDI needles, one for horizontal deviation and one for vertical. Both needles display angular deviations with maximum deviations of $\pm 2.5°$. Around these needles is a compass card to represent the OBS. As a memory aid, this card can be rotated to choose a particular radial to follow, but it has no effect on the CDI indications when used for tracking an ILS. The instrument is used by tuning a particular localiser frequency and then flying to centre the two needles. As with the VOR CDI, this is also a "fly-to" display. Occasionally, approaches are published which require flying away from the localiser and in this case the sense of the needle movement is reversed. With both needles always centred the aircraft will be guided to the runway threshold.

The instrument can also be tuned to and used to follow a VOR and, in this case, the OBS needle is used to set the desired radial to follow.

B.1.6. Horizontal Situation Indicator (HSI)

Initial VOR and ILS cockpit displays were only very basic. The next major advancement was the integration of two or three previously separate devices into a single instrument. One of the more important of these was the Horizontal Situation Indicator (HSI). An HSI device combines an RMI with a CDI driven by a VOR or an ILS. The OBS indicator for the VOR is implemented with a needle that rotates around the inside of the compass card of the RMI. Attached to the OBS, and perpendicular to it, is the CDI scale. The CDI then indicates angular deviation (XAE) from the radial being indicated by the OBS.

This integration makes it much easier for a pilot to navigate VOR radials because it automatically rotates the navigational coordinates into the aircraft referenced coordinate frame. Flying away from the beacon then becomes no different from flying towards it - previously it had been necessary to mentally reverse the sense of the CDI needle. The basic indication of this display is that when the CDI is centred the aircraft must be located somewhere along the radial line drawn from the VOR in the direction indicated by the OBS.

B.1.7. Distance Measuring Equipment (DME)

The advent of low cost Distance Measuring Equipment (DME) receivers allowed them to be used extensively by the GA fleet. These typically present the distance measured between the aircraft's current position and the beacon source on a digital display. The avionics equipment also uses this information to provide an estimate of the aircraft's ground speed. DME displays can be used to follow what is termed a DME arc. This is a curved approach procedure during which the pilot maintains a constant distance from the DME beacon, following a segment of a circle of that radius.

B.1.8. Area Navigation (RNAV) Equipment

Integrated Circuits and electronic miniaturisation brought the cost and weight of avionics equipment into a realm in which it was possible for most aircraft to have at least two VOR receivers. Using these and some extra microprocessors it became possible to compute phantom VOR locations and navigate using them. This type of device is termed Area Navigation (RNAV) equipment. Typically they are used by tuning at least two known VOR beacons and then programmed to present a phantom VOR at a specified location at which no VOR is physically present. This ability enables much simpler navigation to airports that do not have their own VOR. In such a situation the phantom VOR is simply placed at the end of the runway and the radial along the runway centreline is set on the OBS. It is also possible to set this up for flying parallel routes, for example, "3 nm to the left of the 340 radial from FGX VOR". The FAA originally thought that being able to fly parallel routes would provide new options for defining flight procedures but they turned out not to be practical.

Unfortunately, this system proved to not be as useful as first imagined because the accuracy was not consistent and depended on the geometry of the physical VORs that were used. Typically, the VORs were selected by the pilot and thus errors could be easily made, poor geometries chosen or new settings neglected when the previous VORs became inappropriate. This prevented its use for instrument approaches because the accuracy could not be guaranteed. Most GA operators didn't install them and so they were not cost effective.

B.1.9. LOng RAnge Navigation (LORAN)

Long Range Navigation (LORAN) systems were initially (LORAN A) very cumbersome and only used on ships or in large aircraft because of the high weight and power requirements of the Cathode Ray Tube (CRT) displays they used. LORAN C receivers fixed this problem by using miniaturised electronics and lighter displays which made them popular in GA aircraft. This system relies on the locations of master beacons on the ground that are coupled with a number of slave beacons to present information to the aircraft enabling the avionics equipment to place the aircraft at a particular point on a map. The basic concept is that a signal from the master beacon and its synchronised counterpart from one of the slaves can be used to place the aircraft on the locus of points that are a constant difference in distance from both stations. By pairing more than one slave beacon with the master it is possible to use the intersections of the resulting hyperbolas to place the aircraft. Originally, this was done by creating maps which depicted these hyperbolic grids but later advanced by storing the maps electronically and enabling the avionics to retrieve latitude and longitude coordinates for the aircraft. The system was still not sufficiently accurate, though, to be used for Instrument Approaches and performance was often erratic during rain storms making it only useful to VFR pilots, high altitude operations and marine use.

Micro-processors were required in these devices and it was a simple matter to use them to add other functionality. This included the ability for the user to define his own waypoints and even sequence them into routes. These routes could then be used to calculate navigational parameters such as XTE, Current Ground Track and Ground Speed. Some manufacturers also began to implement databases of aviation waypoints including VOR locations, airports and the boundaries of restricted zones, providing the pilot with a great deal of automatically available situational information.

B.1.10. The Global Positioning System (GPS)

The most significant recent advance in aircraft navigation has been the widespread introduction of the Global Positioning System (GPS). GPS was developed in the late 70's and 80's for use by the United States military as a means to target and guide long range missiles [Ref]. The system was first operative in 1985 and the full satellite constellation was brought on line in 1995. The system uses 24 (3 backup) satellites in low orbits (10,900 miles) at an inclination of 55° to broadcast spread spectrum signals that can be used to determine the position of a receiver at almost any location around the Earth. The basic principle is that a receiver picks up the signals from a number of satellites and by analysing the waveform can determine how long the signal has taken to arrive at the antenna. Using this information and the knowledge of the exact positions of the satellites (which is encoded in the message sent from each satellite) it is then further possible to calculate the location of the antenna. Signals from at least four satellites are required to determine a full three dimensional position because it is also necessary to solve for the error in initial transmission time of the coded messages from each satellite [Wells89]. Typically, a GPS receiver, when used for IFR non-precision approaches, will calculate a new position once every second.

The GPS receiver itself returns the Latitude and Longitude coordinates, the altitude and even the velocity of the antenna to the navigation processor which can use this information to locate the aircraft within its database of stored waypoints and related aviation information. These abilities place many GPS systems into the general category of RNAV devices. The FAA has mandated that GPS receivers to be used in aircraft as an aid for the execution of non-precision instrument approaches must include such a database and incorporate into it a complete list of all published waypoints in the area in which the aircraft will operate. These published waypoints can then be used to navigate the aircraft. The accuracy of this system is sufficiently great that it has been approved for use in executing Non-Precision Instrument Approaches. It is TSO C-129 which standardises the requirements of a system that is certified for IFR non-precision approaches.

The accuracy of the GPS system is, however, affected by a number of factors. Firstly, the simple geometry of the satellites can potentially cause large errors if the angles between them are very small. This is particularly true in the measurement of altitude, although the receiver tries to minimise this error by choosing the best available geometry at any moment. Satellites close to the horizon are excluded from the navigation solution because there is too much interference from the ionosphere. Secondly, the satellites are constantly orbiting the Earth and shifting their positions in the sky. At any time it is possible for a satellite to drop out of view or for a new satellite to come into view and cause a sudden jump in the navigation solution. Finally, the true speed of the transmission or the path it has taken cannot be known precisely due to the variations in the atmosphere and problems with reflection of the signal. Intelligent receivers use sophisticated radio techniques to eliminate multi-path errors, while the effects of the atmosphere can be determined by listening to more than one frequency.

Two versions of GPS are available. The C/A(Coarse Acquisition)-code or civilian version transmits publicly available coded messages from each satellite. It is also known as the Standard Positioning Service (SPS). The accuracy of this system is intentionally degraded by the US Department of Defence by purposefully introduced clock errors and satellite position reports referred to as Selective Availability (SA). The P(Precision)-code or military version transmits restricted access coded messages on a separate frequency. This is also known as the Precision Positioning Service (PPS). Using both signals it is possible to correct for atmospheric disturbances as well as eliminate the SA dithering allowing for much higher accuracies to be attained. GA users can generally only access the civilian version.

Receiver Autonomous Integrity Monitoring (RAIM) checks consistency among the ranging signals from different satellites, the technical health of the GPS system and confirms the accuracy of the navigation solution. In order to pass the RAIM checks and allow the GPS receiver to be used for supplemental navigation redundant satellites are required. Typically, about six satellites need to be available to determine the navigation solution in order to provide the necessary redundancy in the measurement and guard against the case that one of the satellites is sending a false signal or that the geometry is not suitable.

APPENDIX C: BRIEFING FORMS

C.1. Pilot Briefing Package

INFORMATION FOR PILOTS PARTICIPATING IN VOLPE-MIT GPS PREDICTOR DISPLAY STUDY

Charles M. Oman and Scott A. Rasmussen

Volpe National Transportation Systems Center,
Cockpit Human Factors Program, DTS-45
and
MIT Department of Aeronautics and Astronautics, Man Vehicle Laboratory,
Contract DTRS-57-92-00054-TTD27

February, 1996

TABLE OF CONTENTS:

1. INTRODUCTION

The goal of this project is to study pilot performance and workload during simulated non-precision GPS approaches, using several different ways of depicting flight path deviation information. The study is sponsored by the FAA, and will help establish design guidelines for the displays used on the next generation of panel mounted general aviation GPS receivers.

In this study, we are asking a dozen current, instrument rated, multi-engine pilots to come to the Volpe Center and fly a series of 16 test approaches in our Frasca 242 simulator. We'll train you up on the simulator and the four different types of displays being used, and give you the chance to practice with them beforehand. You will be paid $10 per hour for each day's session, which should last about 6 hours. If you are coming from outside of greater Boston, keep track of your mileage, so we can arrange reimbursement (25 cents/mile). A map of Cambridge and the Volpe Center is attached. Park in the Visitor Parking Lot on Broadway. Meet us at the scheduled time in the lobby of Building 1. We will get you a badge and a parking pass.

Please bring your logbook with you, so we can update our database on your pilot experience. Your logbook information, as well as all the other data we collect in the simulator, will be kept completely confidential. We refer to subjects only by a letter code and never by name in all publications. We will ask you to sign an "informed consent" statement prior to testing. A copy of the form is attached.

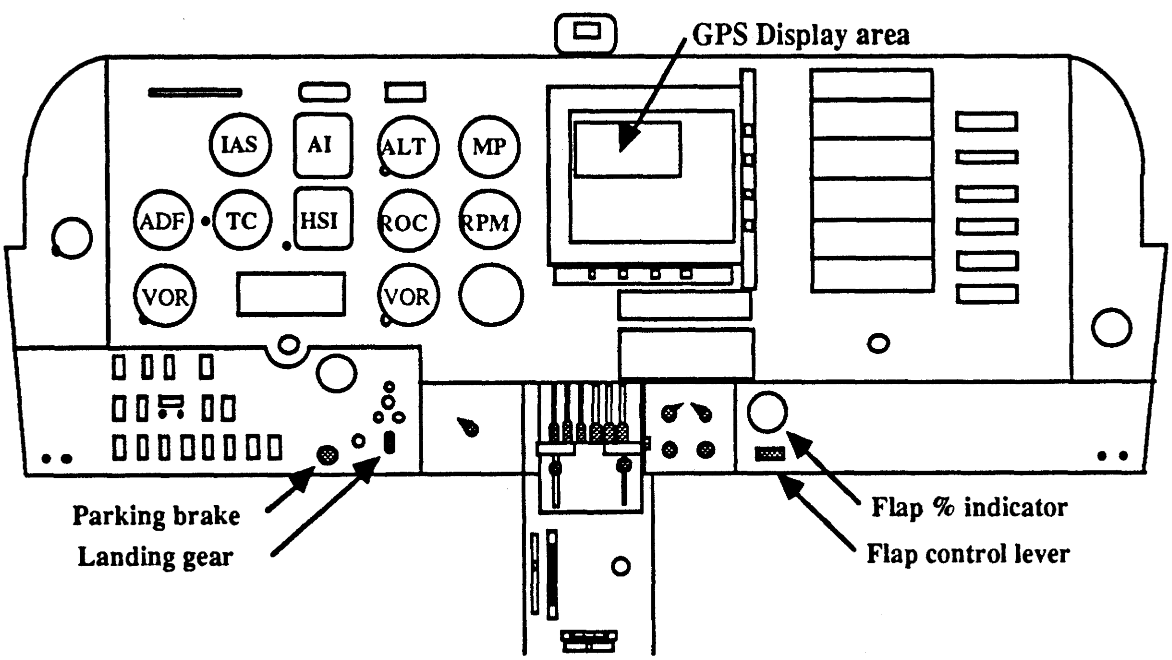
The purpose of this briefing package is to give you some advance information on the simulator, the displays, the approaches you will fly, and the way we will ask you about workload and display preferences. If you read this information beforehand, you will help us use your time efficiently, and you'll have more time to fly. We suggest you skim the whole package first, to get the main ideas, and then re-read the sections on displays and approach hints more carefully. If you have questions, either make some notes, or feel free to call us ahead of time, and we'll be happy to answer them.

2. SIMULATOR AND DISPLAY INFORMATION

If you are already familiar with Volpe's Frasca simulator, you can skip to section 2.2

2.1 Frasca 242 Simulator

Volpe's Frasca flight simulator simulates a generic GA light twin cockpit. There is no outside visual display, but the panel instruments are realistic. The flight equations in the simulator computers approximate the behavior of a Piper Aztec in crosswinds and patchy turbulence. You will have to maintain a good instrument scan to fly the approaches properly.



The Frasca instrument panel layout is a conventional "T". Beneath the attitude indicator is a Horizontal Situation Indicator (HSI). The XTE needle in the center of the HSI will not be used (and is masked from view), but you can use the HSI as a head indicator, and use the two heading bugs as memory aids. The primary panel has VOR and ADF heads, but you won't be using them, since we'll be flying GPS approaches. In the row beneath the primary panel are the electrical switches, parking brake knob, and landing gear switch. The engine instruments and the throttle, prop, mixture controls, carb heat and elevator and rudder trim wheels on the center console are conventional. On the left side of the control yoke (your "flying hand"), you'll find an electric elevator trim switch, a push to talk mike switch and an autopilot disconnect button. In the center of the yoke is an approach plate clip. On the right side of the panel - in front of the right seat - are the fuel selector levers, the flap control switch and flap % indicator. It is important to remember that the flap switch isn't a "set-and-forget" control, as on some aircraft you may have flown. The flap switch has three positions: up, stop, and down. To lower the flaps to the 50% position normally used for approaches, you must put the switch in the down position, monitor the indicator till in reaches 50%, and then put the switch in the stop position. The flaps drop about 10% per second, so you can start the flaps down, count to five, and then look back to verify their final position.

The Frasca is equipped with a King autopilot, flight director and avionics.
Since you will be manually flying GPS approaches, and the transponder code will be preset for you, you will only be using the COMM radios. The radios only display frequencies in 10 kHz increments, but tune to 25 kHz, so if you're asked to tune 118.025 MHz, for example, set the radio to 118.02 MHz. You will wear a headset with voice activated boom mike, and use it to talk with the experimenter and make simulated CTAF calls during each approach, as detailed in section 2.4.

You'll find dual NAVs, ADF, DME, and a NorthStar M3 GPS receiver. But you won't be navigating with these. Instead, you'll be using a simulated receiver, presented on the large display above the center console.

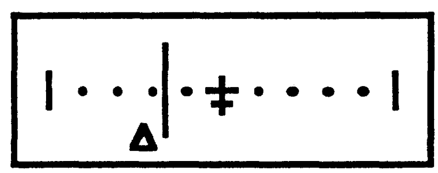
2.2 Simulated GPS Receiver:

The simulated GPS RNAV (ARea NAVigation receiver) appears on a color LCD display located above the center console. We will pre-program the simulated receiver for you prior to each approach, so you don't have to program anything, and waypoints will sequence automatically. As you fly past each waypoint, the current waypoint is automatically switched to become the last waypoint, and a new current waypoint appears. There will be no control knobs to turn, buttons to push, or modes to change.

As shown below, numeric information will be displayed on the left side, always in the same arrangement: The receiver shows the 4 letter name of the last waypoint, followed by an arrow pointing to the name of the current (next) waypoint. On the three lines below, numeric values appear for the cross track error (XTE), the desired track (DTK) in degrees magnetic, the ground speed (GS) in knots, and distance to the current waypoint (DTW) in nautical miles. TAE is Track Angle Error. More about that in a minute.

LAST → CURN

XTE	00.1	DTK	324
TAE	40 >	GS	120
DTW	00.1		



ARMD ACTV TURN

Information on how far off course you are is also shown on the right side of the display in graphical form. In these experiments, we will be trying four different formats. We'll explain each of these formats later, in section 2.3.

When you arrive at the end of a straight leg, and are about to turn onto another straight leg, a "TURN" annunciator light beneath the display will begin to flash when you are within 0.2 nm of the fly-by waypoint. The receiver calculates an appropriate standard rate turning centerline for you that will bring you smoothly onto the next leg after you pass the waypoint, and the cross track error displays show your distance from this turning centerline. The waypoint names will sequence in the middle of the turn. The receiver will continuously update the DTK angle so you can have an idea how to adjust your heading.

Note that the TURN annunciator isn't on continuously on curved legs - a DME arc segment for example- that require other than a standard rate turn. On curved legs, the receiver continuously calculates deviations from the curved leg centerline, and updates DTK as usual, but the TURN light won't come on unless the leg ends at a waypoint where an additional standard rate turn is needed. In this case, the TURN light comes on 0.2 miles from the end of the curved leg, and the computer calculates the turn centerline.

The ARMD and ACTV annunciator lights indicate what phase of the approach you are on, and the sensitivity of the graphical display, as explained below.

Not shown in the figures, but beneath the ARMD, ACTV and TURN lights is a GO AROUND annunciator light. This light will change color when you reach the MAP. It will start as a dark grey and change to bright green once it becomes active at the MAP. On a real system, you'd push the annunciator button if you decided to initiate a GO AROUND prior to the MAP. In these experiments, we'd like you to do your best to stay on the approach, and not fly a GO AROUND until you have passed the MAP. Then, if required, you can press the GO AROUND button by pressing the closest large black button in the row under the screen. This will initiate the missed approach sequence which adjusts the CDI sensitivity back to ±1 mile full scale deflection and sets the approach mode back to ARMD.

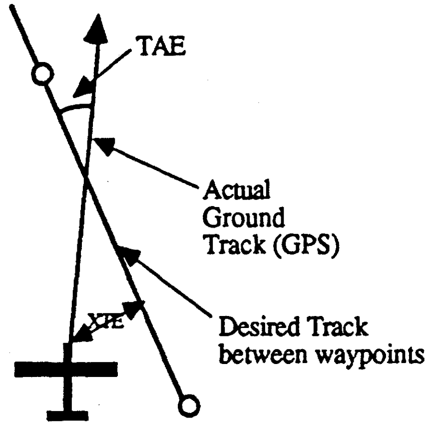
2.3 Graphical display formats and how to use them

As you know, this experiment compares four different ways of showing course deviation information on GPS receivers in graphical format. One aspect that is identical across all four formats you'll use is the way that cross track error is shown. With all four formats, XTE is presented on a "ten dot" scale, reminiscent of VOR or ILS. As usual, you will always "fly-toward" the needle. However, the needle will behave differently than with VOR or ILS as you approach each waypoint, because cross track error ("XTE") information is displayed as distance off the desired course, rather than as an angle. As you well know, with VOR and ILS, the needle becomes more sensitive as you approach the station. However, with RNAV receivers, sensitivity on each leg is constant. With TSOed GPS RNAVs, the enroute is +/- 5 miles full scale. As you fly the transition to the approach, the ARMD annunciator comes on, and needle sensitivity is automatically increased to ± 1 mile full scale. In our experiments, you'll start each approach at this sensitivity. As you fly along the intermediate leg, the needle sensitivity remains constant. Three miles from the FAF, the receiver will start doing its final GPS satellite visibility checks for final approach, and the ARMD light will start to flash, to warn you that the needle sensitivity is about to change. When you are two miles from the FAF, the needle sensitivity will begin to gradually increase, and the ACTV annunciator light will start flashing as well. This is done so you can fly the final leg more accurately, though certainly you have to look at the needle more often. As you pass the FAF, the needle sensitivity has reached ± 0.3 miles full scale, and is then frozen at this value for the entire final approach leg. If the GPS satellites are predicted to provide reliable information for the whole final approach, the ARMD light goes out, and the ACTV light comes on steadily, indicating that you may proceed on final approach. When you reach the MAP, should you decide to go around, needle sensitivity will return to ±1 mile. The various annunciator light combinations thus provide a useful reminder of FAF and MAP waypoint passage, and the status of the needle sensitivity.

Incidentally, if at any time during the entire approach, the XTE needle ever goes off scale at the left end, a < symbol will appear to the left of the [symbol. Offscale at the right end, a > symbol appears.

Everything we've said so far is typical for existing TSOed GPS receivers. So what is different about the displays we're studying in this experiment? The displays we're studying provide more than simple course deviation information. They provide additional information that will let you anticipate the motion of the needle, once you have learned to use them properly. Many pilots say they could fly GPS approaches better if they could somehow anticipate XTE needle movement, other than just increasing needle sensitivity. Increasing needle sensitivity helps, but if the sensitivity is set too high, the needle can become very difficult to follow. (You know first hand how difficult it is to follow a VOR or ILS needle very close to the station.)

Another way to allow the pilot to anticipate needle movement is to show information on "track angle error" (TAE). Track Angle Error is the angular difference between the direction of your actual track over the ground and that of your desired track:

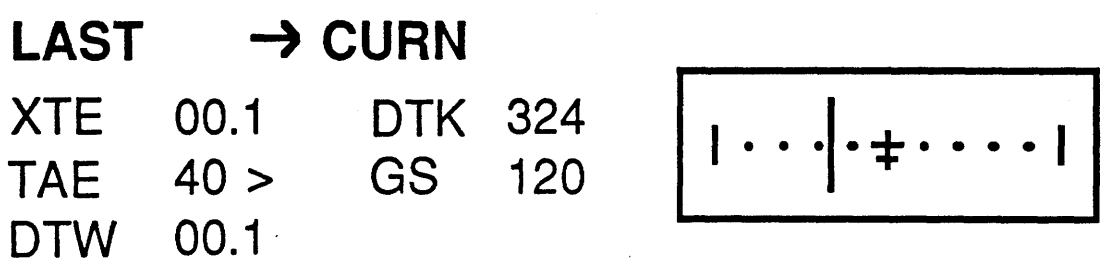


If your TAE is 20 degrees to the right of the desired course, you need to turn 20 degrees left to parallel the desired course. If you see that your TAE is zero, you know your aircraft is paralleling the desired course. When both TAE and XTE are zero, you are on course <u>and</u> on centerline. TAE as computed in your GPS receiver lags behind your true track angle by a few seconds. But if you know your TAE, you can use it to help anticipate XTE needle movement. TAE determines your cross track velocity, which is just the rate of change of the XTE needle.

But what is the best to display TAE information?

<u>XTE Only Display</u>

The simplest display that we are going to ask you to use in this experiment presents Cross-Track Error information in the form of a traditional CDI. It includes a number of digital readouts that will be common to all of the displays we will present to you. [Most of these were explained back on page 4 of this document.] One of them is a numeric representation of your Track Angle Error. You can use this number to give you the information described above about your current ground track and its relation to your desired course. The display includes arrows (< and > symbols) which, technically, show the sign of your TAE. They indicate which way you should turn so that you will return to paralleling your course. Thus if the display shows an arrow pointing right (>), then the desired compass direction is greater than the compass heading you are currently tracking. In this case, if you wanted to return to paralleling the desired course line, then you would need to bank to the right. The arrows also indicate how the XTE needle will move. If an arrow is shown pointing right, then the XTE needle will move to the right. (Whether you are intercepting the course or diverging from it will depend on which side of center is the XTE needle.) The magnitude of the TAE number shows how quickly the needle will move.

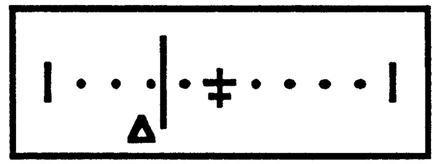


Triangle TAE Display

It may be simpler, and more intuitive to pilots, to show TAE information graphically. One way to do this is as a moving symbol on a scale. In one proposed format, TAE is shown as a Δ symbol, moving left and right just beneath the same ten dot CDI scale used to display the XTE needle.

Full scale deflection of the TAE Δ pointer is chosen to correspond to the largest useful course intercept angle expected, which is 90 degrees. In the situation shown above, the aircraft is off course to the right, but intercepting the desired course at a 40 degree angle.

LAST		**→ CURN**	
XTE	00.1	DTK	324
TAE	40 >	GS	120
DTW	00.1		



When you fly with this display, you'll discover that:

- When you bank right, the TAE triangle moves right. When you bank left, the triangle moves to the left. This is easy to remember.

- When you see the triangle and the needle on the same side (as in the figure above), you are intercepting the desired course. If the triangle and the needle are on opposite sides of center, you are diverging from the desired track. (Mnemonic: Same Side: safe) Take immediate action to move the triangle over to the same side as the needle. To do this:

- Use bank angle to "fly" the triangle over outside the needle, and then wait for the needle to return to center. You can choose a large intercept angle initially, and then reduce it as you approach course centerline, so you won't overshoot. As the needle approaches center, center the triangle along with it.

- Whenever you notice the triangle is centered, make a mental note of your heading on the HSI. Flying this "magic number" will always make your ground track parallel to the desired track. You can set the heading bug on the HSI to this. Once you have the bug set, you can make small adjustments to stay on course centerline by flying the left or the right side of the heading bug.

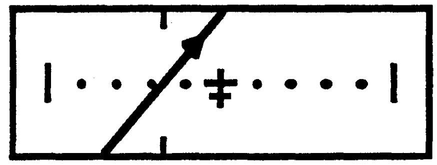
Being able to determine the heading that makes your ground and desired tracks parallel is particularly useful under changing wind conditions. There will often be crosswinds in the approaches you will be flying in the simulator. If you had only XTE information, you must use a cut-and-try technique to determine the proper wind correction angle. You are familiar with this problem from your experience tracking VOR and ILS CDI needles. Typically you try a succession of wind correction angles and watch which way the needle drifts. Eventually you find a heading which keeps the needle centered. If your GPS receiver displays both XTE and TAE, you can skip this cut-and-try business - just fly to keep the needles centered.

Track Vector Display

We are also studying a second format called a "Track Vector" display, in which TAE and XTE information is combined. The CDI "needle" is now a vector arrow which moves left and right in proportion to XTE, and simultaneously rotates by an amount equal and opposite to TAE.

LAST → CURN

XTE	00.1	DTK	326
TAE	40 >	GS	120
DTW	00.1		



When you fly with this display, you'll learn that:

- Most people find this display easier to interpret if they imagine it to be a downward looking track up map view of the aircraft (note aircraft symbol in the center), where the vector represents the desired track. In the situation displayed above, the aircraft is off course to the right, but intercepting at a 40 degree angle.

- If the vector is "leaning" towards the center of the display, you are intercepting the desired course. The vector always tends to slide sideways in the direction that it is tilted. When the vector is leaning away from center, you are diverging from course. Learn to distinguish the "safe" from the diverging situations at a glance. If you are diverging, you should take immediate action to make the vector lean the other way, towards the center. To do this:

- To bring the vector toward upright, bank the airplane in the direction the vector is tilted. In the example in the figure above, roll gently to the right.

- When intercepting, you can choose a large vector tilt angle initially, and reduce it as you approach course centerline.

- Once the vector is upright, the arrow will line up with the little fiducial marks at the top and the bottom of the window, and you know your ground track is parallel to the desired course. Make a mental note of the current heading, or set your HSI heading bug to it, and use it as a reference point when making small corrections.

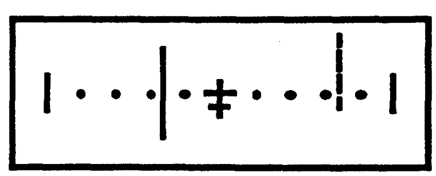
- When XTE goes off scale, TAE information is still available, so you know how much of a "cut" you're taking when you reintercept.

<u>XTE Predictor Display</u>

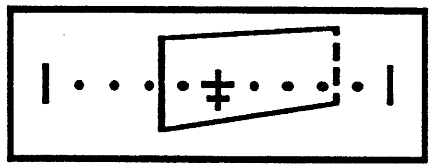
An alternative to these two formats is a Cross Track Error predictor display. This display shows cross track error in conventional fashion using a thin needle. However, in addition, the computer makes a continuous prediction of what the cross track error will be fifteen seconds into the future, based on current ground speed and heading, and displays this to you using a second needle, which is dotted, somewhat shorter, and shifted up so you can see the top of the dotted needle when both needles are aligned. The sensitivity used for both needles is the same (±1 mile, changing to ±0.3 miles as you approach the FAF.)

LAST → CURN

XTE	00.1	DTK	328
TAE	40 >	GS	120
DTW	00.1		



Some pilots find this type of dual needle format simpler to interpret than the Triangle TAE display, since both pointers show the same quantity (distances, rather than distance and angle). Pilots who prefer to visualize their leg intercept angle can still do so, using a simple mental visualization: They imagine solid and dotted needles as connected at the tops and bottoms, so they "see" a three dimensional rectangle, nearly end on, looking "in" from one side or the other. It is a segment of an earth vertical plane containing the desired track about one half mile ahead of the aircraft, with the dotted end farther away.



This simple 3-D visualization trick allows you to "see" your track angle error, even though it isn't explicitly shown. In the situation above, the aircraft is off to the right of track, but it's present track will intercept the desired course about 1/6 of a mile ahead (1/3 of the way from the near end of the half mile long rectangle).

When you fly with this display, you'll discover that:

- When you bank left, both needles move right (a "fly to" display).

- When solid and dotted needles are on the opposite sides of center or the dotted needle is closer to the center than the solid needle, you are intercepting the desired course ("safe" configuration). If the dotted needle is on the same side as the solid needle, but further from the center, you are diverging. Take immediate action to move the dotted needle inside the solid one. To do this: bank toward the dotted needle.

- Use bank angle to "fly" the dotted needle to the center, or somewhat beyond, and then wait for the solid needle to return to center. If you visualize the needles in three dimensions, you can easily adopt a large intercept angle initially, and then reduce it as you approach course centerline, so you won't overshoot. It won't bother you that the dotted needle is on the opposite side of center.

- Whenever you see the dotted and solid needles superimposed, make a mental note of your heading on the HSI, and as with the other displays, use it as your "magic number", set a heading bug, and after intercept use it to make small adjustments to the left or right of course.

- One strategy for following this display is to keep the dotted line centered by always flying towards it whenever it deviates and pretty much ignoring the solid line showing your current XTE. You know that if the dotted line is centered then the solid line will eventually return to center also. This works because the dotted line is showing where the solid line will be in 15 seconds. This simplifies the use of this display but it gives you a reasonably slow intercept and only moderate performance. You can improve your performance by visualizing the flight path as described above.

So that's basically it. When you come in to fly the simulator, we'll brief you again on these different GPS display formats, and answer any questions you have. You'll have a chance to experiment with each of them first hand, develop your own rules of thumb, and practice until you feel comfortable.

2.4 Approach Procedures

After your practice approaches, we'll record data while you fly 16 more GPS approaches using one of the display configurations. You'll discover the approaches are to a variety of different airports, and have different approach headings and altitudes. Before each approach, you'll be given the corresponding paper approach plate, and enough time to review it thoroughly. You will be free to refer to the chart as you fly. There is a chart clip on the yoke if you need it.

A typical approach plate (one you'll use for your practice approaches) is shown on the next page.

Assume that you are flying practice instrument approaches to unattended fields, and that prior to the start of the simulation, you have been vectored close to an IAF. The experimenter will re-initialize the simulator on a heading nearly parallel to the initial approach leg. The GPS approaches used in the experiment will typically involve a curved or straight leg to an intermediate waypoint (IF), followed by a five mile intermediate leg, and then a six mile final approach leg. The curved initial legs are charted as DME arcs. The GPS knows this, and will guide you on the appropriate arc from the charted IAF to the IF, and alert you when it is time to TURN at the IF onto the intermediate leg.

Prior to each approach, we will tell you the name of the waypoint you will start near. Be sure to prebrief the missed approach instructions, since you may be needing them! When you have finished reviewing the chart:

- Check that the Frasca is configured for level flight at 120 knots, with flaps and gear UP. The numbers are:
    Prop - 2400 RPM
    Throttles - 15 inches MP
    Boost pumps - off

- As soon as the simulation begins, intercept the initial approach leg. Remember that you may have a crosswind. Once on course, keep the XTE centered to within a dot or two.

- Fly the entire approach at 120 knots, maintaining this airspeed ± 10 knots, and the approach altitudes shown on the chart ± 100 feet. This may be challenging in the turbulence, but do your best, and try to be smooth. These standards are similar to those on the practical test you took for your instrument rating. Our computers will be continuously recording the Frasca's flight path and airspeed, and measuring your deviation from the published approach, your "flight technical error".

- As you fly along the initial approach leg, look up the CTAF frequency on the chart, and pretune one of the COM radios so you will be able to make a CTAF call just outside the FAF. You can start your cockpit preparations for landing.
    Boost pumps - on
    Fuel - fullest tanks
    Carb Heat - on

So we can compare initial approach data across runs with the aircraft in the identical configuration, we ask everyone not to reconfigure flaps, gear, and props for landing until you get within three files of the FAF.

- If you discover that you've made some procedural or navigational mistake while flying, <u>immediately tell us</u>. It will be very helpful to us when analyzing your data to know exactly when it happened. Be brief, keep flying, and we'll ask you for the details after the approach is over.

- When the ARMD light starts flashing 3 miles prior to the FAF, but not before, reconfigure for gear/flaps down level flight:
    Flaps 50%
    Gear down
    Props - 2750 (highest RPM)
    Throttles - 21 inches MP

- As you come up on the FAF, but prior to it, self-announce your approach on the CTAF frequency. For example: (AIRPORT) TRAFFIC, FRASCA 123SH, TWO MILES FROM (FAF WAYPOINT NAME) INBOUND, AT (ALTITUDE), PRACTICE GPS APPROACH TO (RUNWAY NUMBER) AT (AIRPORT).

- As you go over the FAF, adjust the power back to 18 inches, and initiate a descent. The approaches call for a relatively high rate of descent, so you will need to start your descent promptly in order to get down to the MDA prior to the MAP. Be aware that there may be a crosswind present, and that it may change with altitude.

- Do your final approach check as usual, e.g. GUMP.

- At some point on final approach, the computer will sidestep the aircraft a quarter mile to one side of the final approach course. We do this to see whether the displays help you reintercept a final approach course. When this happens, use your displays to reintercept the final approach course <u>as smoothly and quickly as possible</u>.

• During your descent, occasionally look up and check the small "breakout" video monitor mounted above the instrument panel. When you're in the clouds, it will just show grey murk. When you breakout beneath the deck, you'll see a grey earth/white sky horizon line. Report to us on the intercom when you notice you have broken out. (No ground features - such as the airport - will be shown, incidentally).

• Stop your descent just above the published MDA. Be careful not to descend below the MDA. If you do, you'll hear about it from the controller. Throttles forward to 21 inches MP again should do it.

• If you have broken out, when you reach the MAP, but not before, initiate a descent, and make a CTAF call that you are landing. [For example:(AIRPORT) TRAFFIC, FRASCA 123SH, ON SHORT FINAL RUNWAY (NUMBER) AT (AIRPORT)].

• If you have not broken out when you reach the MAP, initiate a go-around, and make a CTAF announcement that you are going around [for example: (AIRPORT) TRAFFIC, FRASCA 123SH, AT (MAP WAYPOINT), ON THE MISS AT (AIRPORT)]. Retract gear and flaps, and set up a cruise climb at 2400 RPM and 24 in. MP.

• At this point the experimenter will tell you on the intercom that the run has ended. Reconfigure the Frasca for level cruise at 120 knots, gear and flaps up, in preparation for the next approach. (15 inches, 2400 RPM) Notify the experimenter when you are ready.

• We'll freeze the simulation, and then ask you for details on any mistakes you remember making during the run, or problems you had interpreting the displays. Then we will ask you some questions about your workload on the most recent approach, and after every four approaches, questions about your display preferences. At the end of the session, we'll do a debrief, and ask you to complete a final written questionnaire on your display preferences.

2.5 Reporting Workload

We want you to tell us about your workload using the "Modified Bedford" method, illustrated in the figure below. It seems complicated at first, but after a little practice it is easy to do. The method asks you about how much "spare time" you had to handle additional tasks or likely contingencies.

First, judge whether the workload was "impossible", "possible", "tolerable", or "satisfactory", and then judge your spare time within the category you chose by using the descriptions in the appropriate box on the right.

We'll ask you to make two judgments: workload for the initial and intermediate portions of the approach (prior to ACTV light flashing), and for the final approach portion, including the reintercept during the descent.

If the workload was so high that there was no way you could ever fly the approach, then the workload is obviously "impossible", and you rate it "10".

If you found you could do the approach, but you were really busy, had minimal or no spare time to do anything else, and the workload is so intolerable you sure wouldn't want to have to fly approaches that way regularly, then judge the workload as "possible". Pick a number between 7 and 9 to describe it, using the descriptions in the appropriate box. By the way, you can use fractions (e.g. 7.5) anytime you need to when reporting on the Bedford scale.

If the workload was acceptable, but you didn't have enough of a spare time reserve so you could easily attend to additional tasks, then judge the workload as "tolerable". Pick a number between 4 and 6 to describe it, using the descriptions.

If you had enough time to easily attend to all likely additional or contingency tasks, or more, then judge the workload as "satisfactory", and assign it a number between 1 and 3, using the descriptions.

Judge workload as accurately as you can. Use fractions, if it seems appropriate. For example, rather than saying "workload was 4 to 5", say "workload was 4.5".

After every fourth approach, we'll ask you how you would rank the four most recent approaches in terms of overall workload. We'll also give you a set of four cards showing the four different types of displays, and ask you to arrange them in order of your overall preference, from best to worst.

During the debrief after the session, we'll ask you some more questions on your preferences, and reasons for them, and ask you to rank the displays, from best to worst, in terms of:

- ease of interpretation
- effectiveness in help you to fly accurately
- your overall preference.

C.2. Informed Consent Statement

Informed Consent Form (Rev 5 12/22/95)

PILOT PERFORMANCE DURING GPS INSTRUMENT APPROACHES

I have been asked to participate in a study investigating GPS guided approach procedures. The study is being conducted by scientists from MIT and the U.S. Department of Transportation.

The study's purpose is to examine the influence of different approach characteristics (such as approach angle and cross wind) and instrument panel displays on the pilot's ability to accurately follow a flight plan using a GPS navigation system. If I agree to participate, I will be asked to fly a series of non-precision approaches in a fixed base Frasca flight simulator, configured to represent a Piper Aztec.

I am at least 18 years of age. I agree to bring my pilot logbook, and allow the investigators to copy flying background related data from it.

The study will take approximately six hours to complete during a single testing day. There are no unusual physical risks involved in participating in this ground based simulator study. I have been shown the building fire exits and extinguisher locations. During the experiment, I will be asked to answer questions in order to measure my mental workload during different phases of the approach. My name will be kept strictly confidential and not be associated with my data in any publications. I understand I may decline to answer questions, and I may ask questions about the study at any time, and that my participation is strictly voluntary. I am free to withdraw at anytime without penalty.

I understand that I will be paid $10 per hour for participation in the study. Payment will be prorated if I withdraw before completing the experiment. If I am driving from the outside greater Boston area, transportation will be compensated at a rate of $0.25/mile.

I understand that by my participation in this study I am not waiving any of my legal rights. (Further information may be obtained by calling MIT's Insurance and Legal Affairs Office at 617 253-2822.)

I understand that I may also contact the Chairman of the Committee on the Use of
Humans as Experimental Subjects, MIT, telephone 253-6787, if I feel I have been treated unfairly as a subject.

I have been informed as to the procedures and purpose of this experiment and agree to participate.

______________________________ __________

Participant Date

______________________________ __________

Witness Date

C.3. Training Protocol

Before the Experiment

Make initial call
Tell them how much to be paid and reimbursement for mileage
- Boston Metro gets no reimbursement (inside 495)
- everyone else does

Send pilot brief with note about when to turn up and where
Ask to bring log books

Day of Experiment

Copy name, address and payment form
Print copy pilot brief to give pilot on day of experiment
Print copy of briefing sheets showing displays
Open the subject database
Pilot shows up
Fill out name and address sheet
get informed consent statement signed
Take log book and update database
Enter that we used them this day
Give pilot a brief overview of the experiment
- try to get a sense of whether the pilot has read the brief
- ask if they have any questions

Walk pilot through the Frasca to show them physically everything
- here's a typical approach plate
- fly for a while
- here are the numbers
- here is the display where things will appear

Talk the pilot through an approach
- talk through everything but the details of the displays
- when to set radio frequency and do radio call
- when to reconfigure - why it is important to do it where we want
- decision on go around announcement

Talk to pilot about the display
- Top level Stuff
  - Annunciator lights
  - Displays will work automatically
  - Turn Anticipation
- XTE display
  - Digital displays
    - Abbreviations
  - Shows Cross Track Error
  - Fly-To indicator
  - Off-Scale indications
  - Q What does converging look like?
  - Q What does diverging look like?
  - Q What happens to the display when you bank?
  - Q What does off scale look like?
- TAE Explanation
  - TAE is a number that tells you your magic number
  - When it is zero you should make a note of your heading
  - This number will tell you what heading will null the display

Explain how the arrows work on the digital display
Break control into two parts: Intercept and Tracking
Intercept requires picking an intercept angle and flying to null the XTE
Tracking then means following the magic number to keep the XTE zero
Teach technique of flying the bug to null errors
Don't chase the TAE, treat it as a secondary instrument

Triangle Display

Numeric value of TAE is simply shown as position of the triangle
Easy way to remember the scale is that intercepts greater than 90 are useless
Thus 90 is full scale
To fly, pick a suitable intercept angle with the triangle on the SAME side as the XTE
Fly the triangle in with the XTE needle
The triangle will move in the SAME direction as your bank.
If you want to move it right, bank right.
You should know the intercept appearance of the display
Q What does converging look like?
Q What does diverging look like?
Q What happens to the display when you bank?
Q What does off scale look like?

Track Vector

This is the same as the XTE scale which is just tilted to represent the TAE
Because of the tilt this has a geometrical interpretation
Think of this as a mail slot or god's eye view of the track
To choose the intercept angle you can just judge the angle of the vector
Anytime you see the vector upright that is a TAE of zero. Even when off centre
This gives you your magic number. Note it.
Point out the little verniers which tell when the vector is upright
When off scale you can still see the TAE.
Q What does converging look like?
Q What does diverging look like?
Q What happens to the display when you bank?
Q What does off scale look like?

Predictor

TAE is not literally shown
Instead the predicted displacement is shown 15 seconds into the future
TAE is still there on the display by visualising it in 3D
This display is very much like the track vector but with the vector horizontal
Q What does converging look like?
Q What does diverging look like?
Q What happens to the display when you bank?
Q What does off scale look like?

Workload Evaluation

After each run we will ask for two workload scores
Workload ranks are a scale within a scale
First decide whether workload was satisfactory, tolerable, possible, impossible
Tolerable: OK but I would like it reduced
Possible: Could be done but wasn't safe
Then decide what level within each adjective
Can choose fractions
After a few runs you will likely be rating the runs relatively to each other
Try to keep the absoluteness of the scale

C.5. Pilot De-brief Questionnaire

Pilot Initials______________________ Date__________

VOLPE-MIT GPS RECEIVER DISPLAY STUDY
Post Session Questionnaire

We'd like you to make some "head to head" subjective comparisons of the four displays you flew with. To do this, we'll ask you to indicate your preference by using a pencil indication on a horizontal "preference scale". If you had to make an approach in marginal weather, and you were given the choice between the two displays, which would you choose?

I--------------------------------0--------------------------------I

Strongly prefer **Track Vector**	**Strongly Prefer** **Triangle TAE**

I--------------------------------0--------------------------------I

Strongly prefer **XTE Predictor**	**Strongly Prefer** **Triangle TAE**

I--------------------------------0--------------------------------I

Strongly prefer **XTE Only**	**Strongly Prefer** **Triangle TAE**

I--------------------------------0--------------------------------I

Strongly prefer **XTE Predictor**	**Strongly Prefer** **Track Vector**

I--------------------------------0--------------------------------I

Strongly prefer **XTE Only**	**Strongly Prefer** **Track Vector**

I--------------------------------0--------------------------------I

Strongly prefer **XTE Only**	**Strongly Prefer** **XTE Predictor**

Overall, how do you rank the 4 displays in terms of:

A. Ease of interpretation
where 1 = easiest, and 4 = hardest

Track Vector___________
Triangle TAE___________
XTE Predictor__________
XTE Only_____________

Comments:____
____

B. Effect on flight path control accuracy
1 = most accurate 4 = least accurate

Track Vector___________
Triangle TAE___________
XTE Predictor__________
XTE Only_____________

Comments:____
____

C Your overall preference, where
1 = most preferred; 4 = least preferred

Track Vector___________
Triangle TAE___________
XTE Predictor__________
XTE Only_____________

Comments:____
____

D. How would you rank the displays, in terms of their effect on overall workload?
1 = least workload ; 4 = most workload

Track Vector___________
Triangle TAE___________
XTE Predictor__________
XTE Only_____________

Comments:____
____

General Questions:

Were there enough practice approaches?

How many practice approaches would you recommend we use for each type of display?

Track Vector___________
Triangle TAE___________
XTE Predictor__________
XTE Only_____________

When did you use the numeric TAE information presented on the GPS receiver? Y N (circle one)
Were the < and > symbols beside numeric TAE useful? Y N

With the TAE triangle display, did you have any trouble remembering which way to turn to center the TAE triangle?
Y N (circle one)

With the XTE predictor display, did you have any trouble remembering which way to turn to center the dotted predictor line? Y N (circle one)

Did you try to visualise the XTE predictor in 3 dimensions? Y N (circle one)
Did you find this useful Y N (circle one)

With the XTE predictor display, how did the dotted predictor bar movement seem during final approach?

OK Too sensitive Not Sensitive Enough (circle one)

When did you use the DTK information presented on the display?

When did you use the Ground Speed (GS) information presented on the GPS receiver?

On which leg did the turbulence seem strongest? (circle which)

DME arc
Intermediate Leg
Final approach
Same on all legs

What manufacturer/model aircraft have you flown the most in the past 6 months?
Does this airplane have an HSI? Y N (circle one)

Approximately how many actual flight hours experience have you had (if any) using a GPS receiver?
Which manufacturer/model(s)?
Was a moving map display available? Do you use it?

Approximately how many flight hours experience have you had (if any) using a LORAN receiver or RNAV system?
Which manufacturer/model(s)?
Was a moving map display available? Do you use it?

APPENDIX D: EXPERIMENT FORMS

D.1. Experiment Checklists

D.1.1. Experiment Checklist

SAMR/96/8/24/0307

Subject #	Pilot's Name	Subject Initials	Subject Type	Date Tested	Comments
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

D.1.2. Subject Checklist

SAMIR/96/8/24/0307

Pilot Name: ....

Date: ....

1

Test #	Flight Plan	Display Code	Rank W	Rank P	File Name	Time Start	Time Stop	Comments
P1	2p	T						
P2	4p	V						
P3	3p	P						
P4	1p	X						
1	1a	X						
2	3d	V						
3	4b	P						
4	2c	T						
5	3c	P						
6	2a	X						
7	4d	T						
8	1b	V						
9	4a	V						
10	1c	T						
11	3b	X						
12	2d	P						
13	3a	T						
14	1d	P						
15	2b	V						
16	4c	X						

D.1.3. Flight Checklist

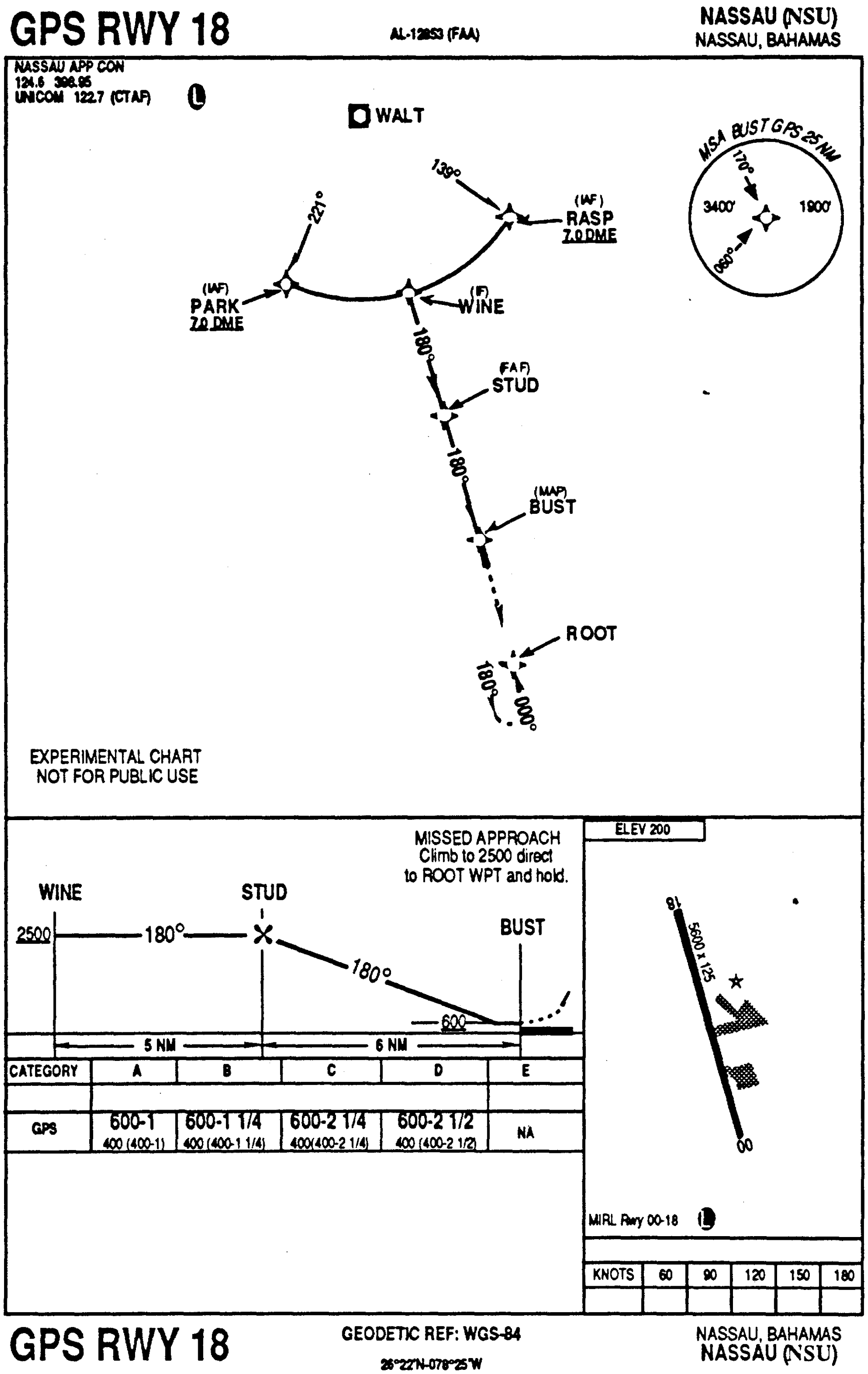
1-P1

SAMR/96/8/24/0307

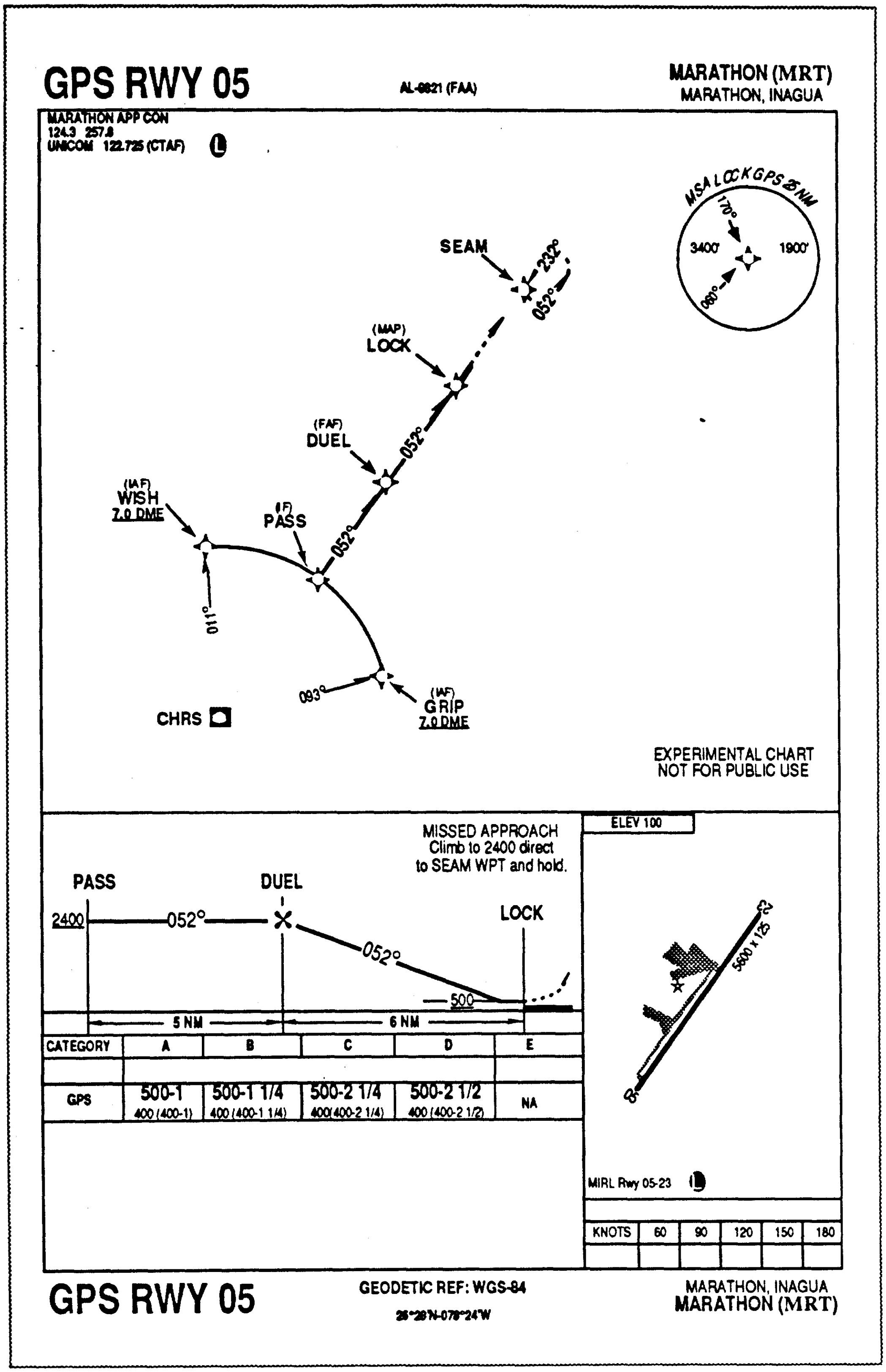
Task		State	Comments	(distance to end)
Pilot's Initials				
Run Number				
Flight Plan		2p		
Pilot has Approach Plate		NSU		
Display Type		disp T		
HSI Button		OFF		
Initial Waypoint		WINE		
Pop Quiz				
Start Time				
DME Performance				
First Turn Performance				
Initial Approach Performance				
Radio Announcement				@ nm
Frequency		122.7		
Landing Gear		100%		@ nm
Flaps		50%		@ nm
Boost Pumps		ON		@ nm
Carb Heat		ON		@ nm
Long Final Performance				
Side Step				
Short Final Performance				
Minimum Descent Altitude		600		
Missed Approach Decision				@ nm
Stop Time				
Landing Gear		0%		
Flaps		0%		
Boost Pumps		OFF		
Carb Heat		OFF		
Initiate Level Flight				
Turn On Auto Pilot				
Initial Workload Score				
Final Workload Score				
Relative Workload Rank				

D.2. Approach Plates

D.2.1. Nassau



D.2.2. Marathon



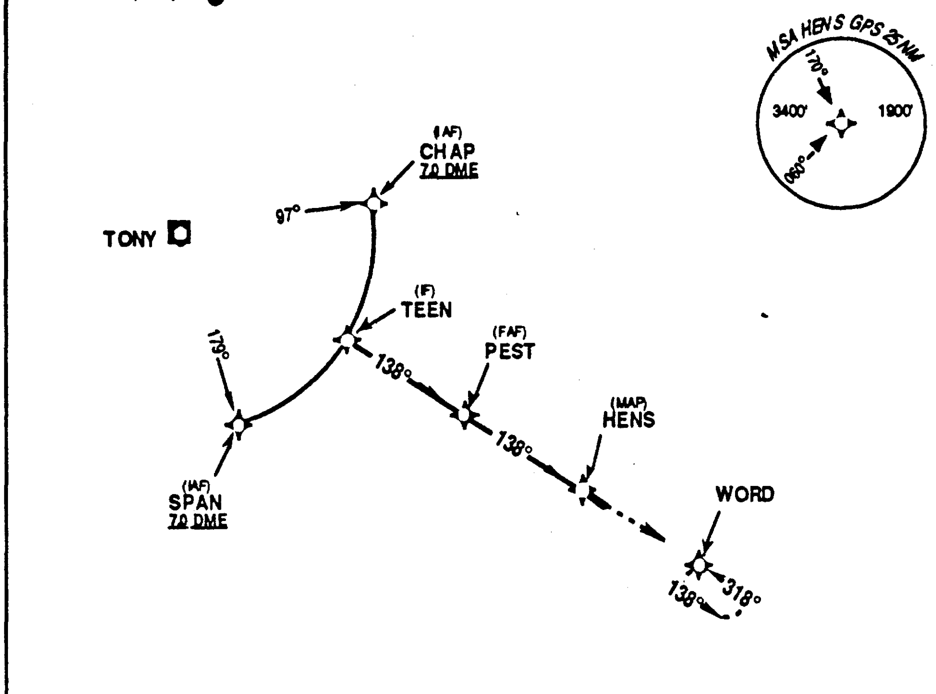
D.2.3. Tavernier

GPS RWY 14

AL-29515 (FAA)

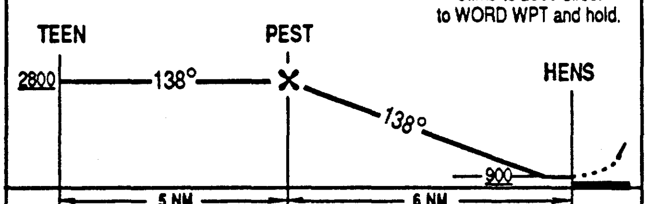
TAVERNIER (TVN)
TAVERNIER, BIMINIS

TAVERNIER APP CON
127.4 269.0
UNICOM 122.975 (CTAF) L

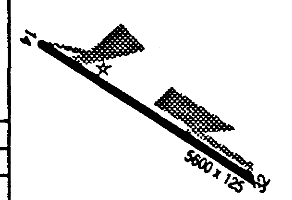


EXPERIMENTAL CHART
NOT FOR PUBLIC USE

MISSED APPROACH
Climb to 2800 direct
to WORD WPT and hold.



ELEV 500



CATEGORY	A	B	C	D	E
GPS	900-1 400 (400-1)	900-1 1/4 400 (400-1 1/4)	900-2 1/4 400(400-2 1/4)	900-2 1/2 400 (400-2 1/2)	NA

MIRL Rwy 14-32 L

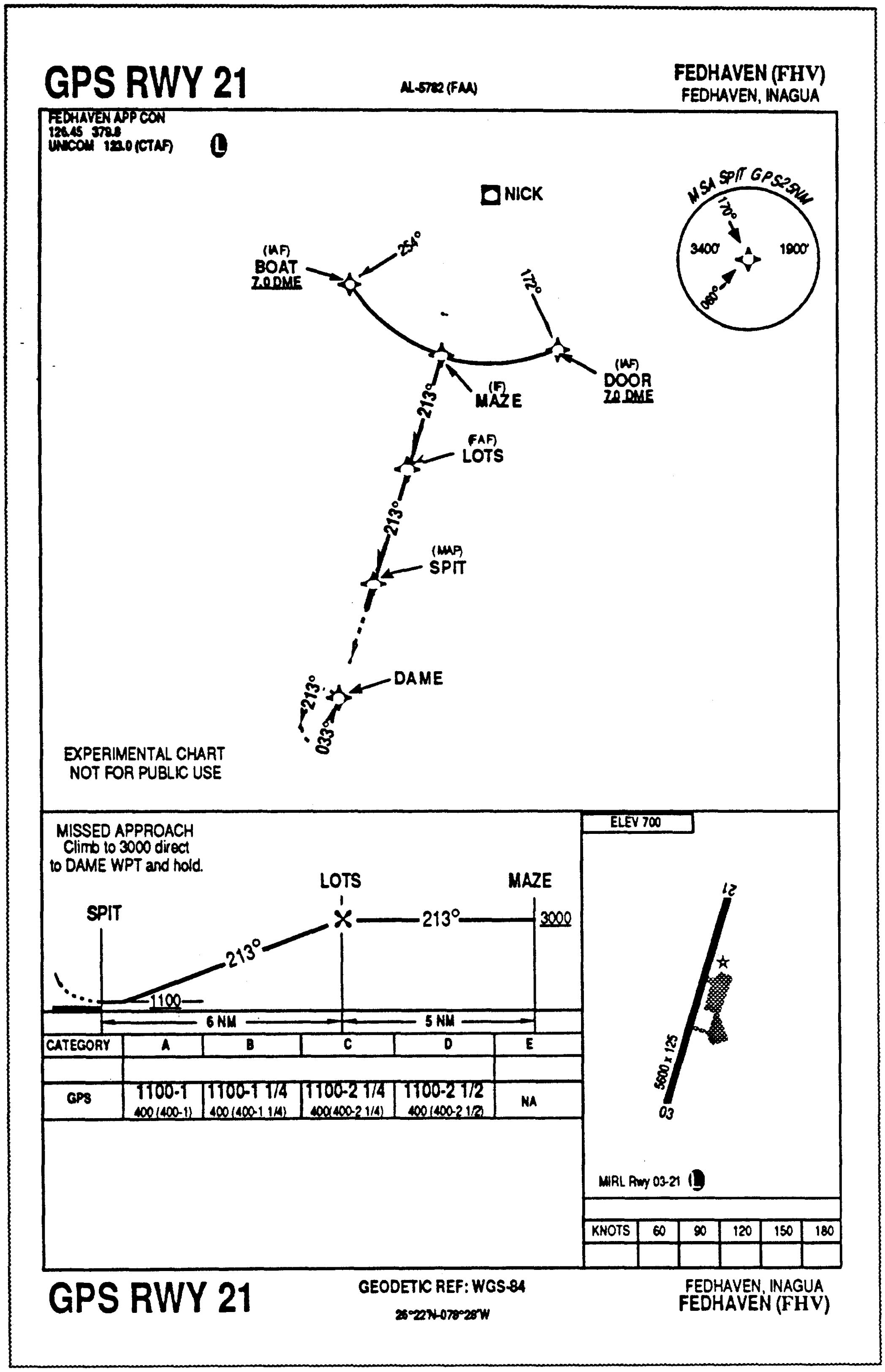
KNOTS	60	90	120	150	180

GPS RWY 14

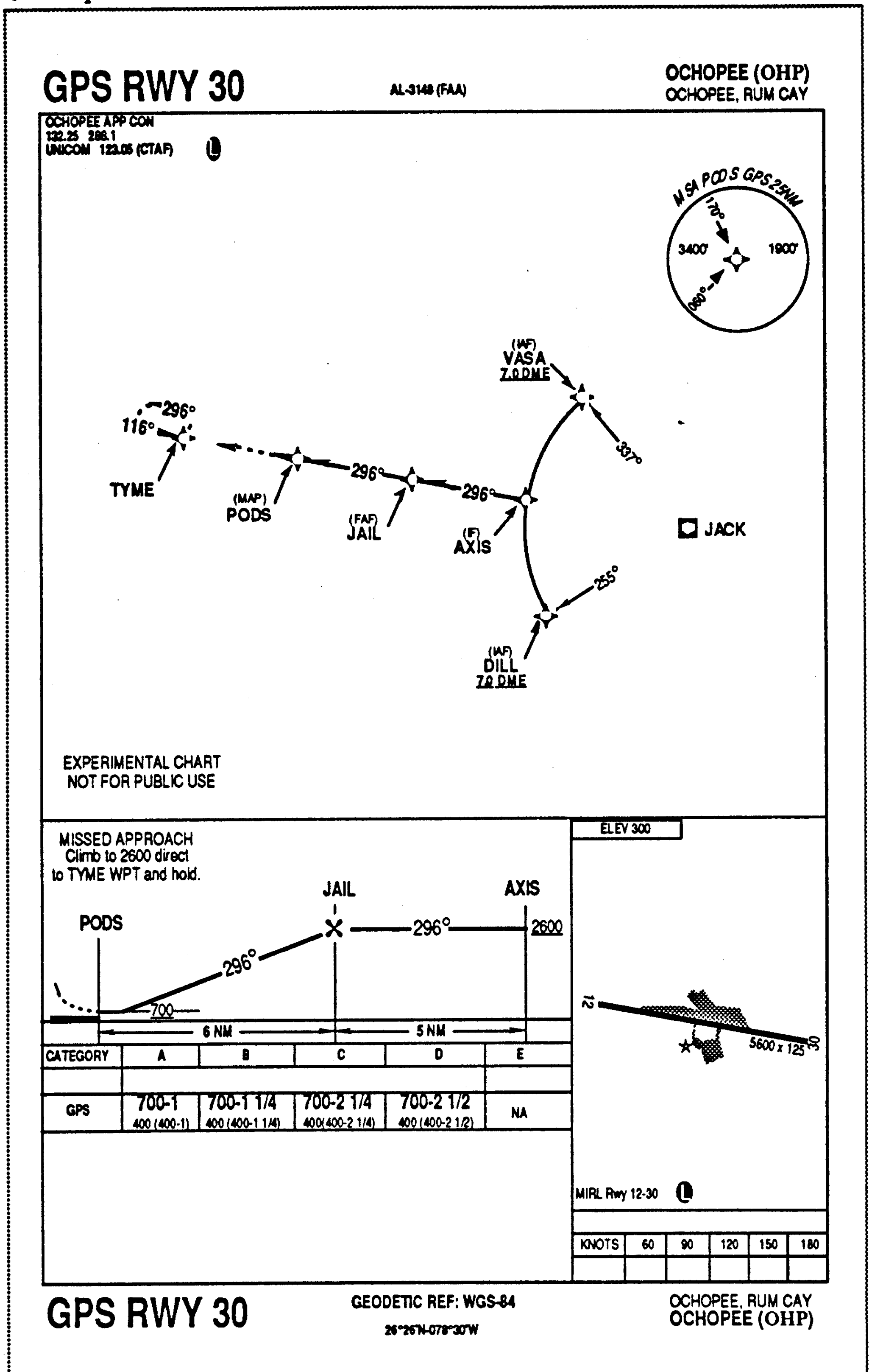
GEODETIC REF: WGS-84
26°23'N-078°23'W

TAVERNIER, BIMINIS
TAVERNIER (TVN)

D.2.4. Fedhaven



D.2.5. Ochopee


GPS RWY 30
AL-3148 (FAA)
OCHOPEE (OHP)
OCHOPEE, RUM CAY
OCHOPEE APP CON
132.25 288.1
UNICOM 123.05 (CTAF)
MSA PODS GPS 25NM
170°
3400'
1900'
060°
(IAF)
VASA
7.0 DME
296°
116°
TYME
(MAP)
PODS
296°
(FAF)
JAIL
296°
(IF)
AXIS
337°
JACK
255°
(IAF)
DILL
7.0 DME
EXPERIMENTAL CHART
NOT FOR PUBLIC USE
MISSED APPROACH
Climb to 2600 direct
to TYME WPT and hold.
JAIL
AXIS
PODS
296°
296°
2600
700
6 NM
5 NM
ELEV 300
12
5600 x 125
30
CATEGORY | A | B | C | D | E
GPS | 700-1 400 (400-1) | 700-1 1/4 400 (400-1 1/4) | 700-2 1/4 400(400-2 1/4) | 700-2 1/2 400 (400-2 1/2) | NA
MIRL Rwy 12-30
KNOTS | 60 | 90 | 120 | 150 | 180
GPS RWY 30
GEODETIC REF: WGS-84
26°26'N-078°30'W
OCHOPEE, RUM CAY
OCHOPEE (OHP)

APPENDIX E: SUBJECT DATA AND COMMENTS

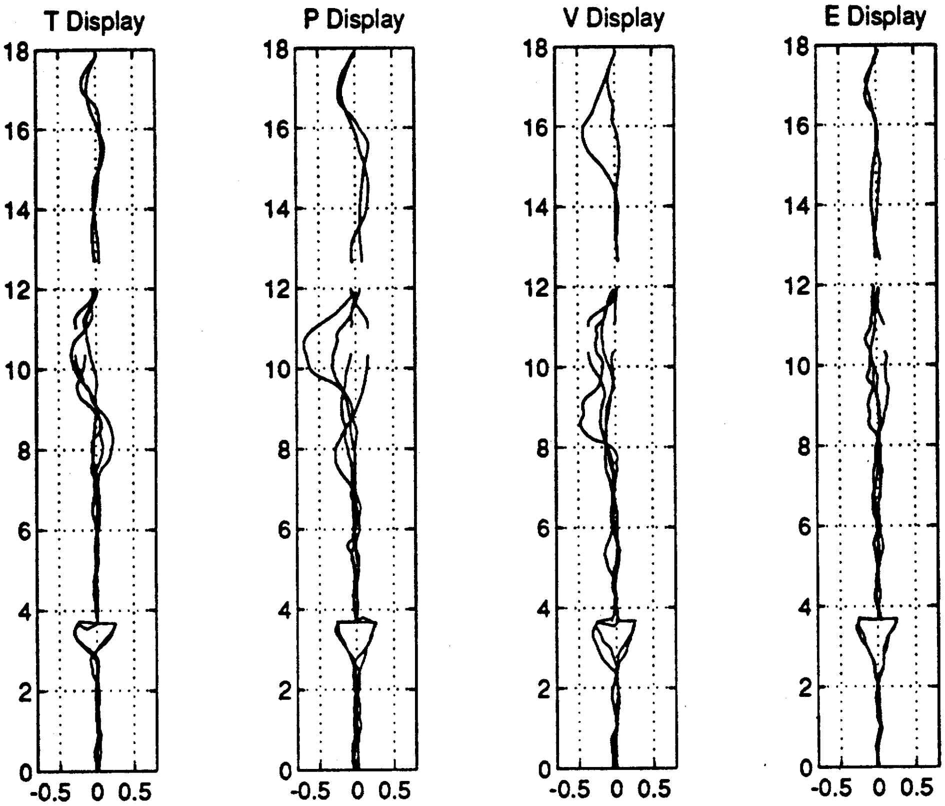
E.1. Subject One

Date Tested: 8 December 1995 and 11 December 1995
Subject Type: 1

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	T		001.dat	11:36	11:45		4.0	6.5
P2	4p	V						4.0	5.0
P3	3p	P		003.dat	12:13	12:24		6.0	7.0
P4	1p	E		004.dat	12:31	12:40		4.0	5.0
1	1a	E		005.dat				2.0	2.5
2	3d	V		006.dat	14:52	15:07		3.5	3.5
3	4b	P		007.dat	15:13	15:20		3.0	3.5
4	2c	T		008.dat	15:22	15:30		3.0	3.5
5	3c	P		009.dat	15:34	15:44		3.0	3.0
6	2a	E		010.dat	15:52	16:00		4.5	5.0
7	4d	T		011.dat	16:02	16:09		2.5	3.5
8	1b	V		012.dat	16:11	16:20		4.0	5.0
9	4a	V		013.dat	15:55	16:04		8.0	5.0
10	1c	T		014.dat	16:06	16:14		6.0	6.0
11	3b	E		015.dat	16:16	16:28		6.5	4.5
12	2d	P		016.dat				5.0	5.0
13	3a	T		017.dat	16:46	16:55		4.5	3.5
14	1d	P		018.dat	16:58	17:08		4.5	4.0
15	2b	V		021.dat				2.5	4.0
16	4c	E		020.dat	17:26	17:33		2.5	3.0

```
Comments:
Not willing to give workload ratings as low as 2
P2: data lost
 2: stayed at 1700 because mis-read altimeter
 9: turbulence probably at 5.0
    high workload rating because aircraft seemed different
    very easy for last mile
11: higher workload getting squared away
    plan not set up when it was given to me
    takes a while to get initial heading
16: pretty straightforward
 1: data lost, redone 11 Dec 1995 -> 005.dat
    pretty well under control
 7: data lost, redone 11 Dec 1995 -> 011.dat
15: sidestep did not work, redone 11 Dec 1995 -> 021.dat
    getting tired - cause of altitude excursion

Questionnaire:
V vs T - 10/4; X vs T - 11/3; E vs T - 10/4
X vs V -  9/5; E vs V -  9/5; E vs X -  7/7
EOI: V - 3; T - 4; P - 2; E - 1
   : The electronic HSI and Predictor are a toss-up
   : At times the Predictor is easier to use
FPA: V - 3; T - 4; P - 1; E - 2
   : Many times I would get confused with Triangle indicator
OP : V - 3; T - 4; P - 1; E - 2
```



T Display
P Display
V Display
E Display
18
16
14
12
10
8
6
4
2
0
-0.5 0 0.5

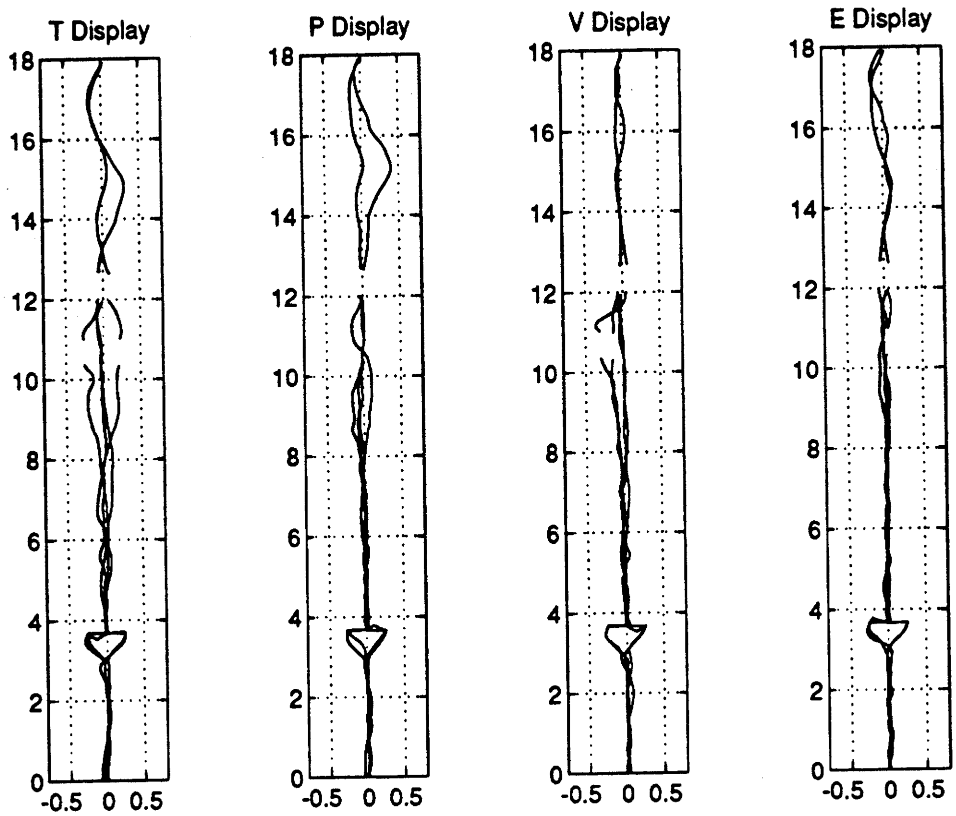
E.2. Subject Two

Date Tested: 5 January 1996
Subject Type: 1

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	V	1	000.dat		10:00		4.0	5.0
P2	4p	T	3	001.dat	10:13				
P3	3p	P	2	002.dat	10:23	10:33		3.0	4.0
P4	1p	E	4	003.dat	10:36	10:45		4.0	3.5
1	1a	E	4	002.dat	11:26	11:36		4.0	3.0
1	1a	E	4	004.dat	11:45	12:00		4.0	4.0
2	3d	V	1	005.dat	12:05	12:15		3.0	3.0
3	4b	P	2	006.dat	12:21	12:28		3.0	3.5
4	2c	T	3	007.dat	12:32	12:41		3.0	3.5
5	3c	P	2	008.dat	13:32	13:41		3.0	3.5
6	2a	E	4	009.dat	13:43	13:55		3.0	3.0
7	4d	T	3	010.dat	13:59	14:06		3.0	3.0
8	1b	V	1	011.dat	14:10	14:20		3.0	3.5
9	4a	V		012.dat	14:36	14:43		3.0	3.0
9	4a	V	1	013.dat	14:46	14:53		3.0	3.0
10	1c	T	3	014.dat	14:58	15:07		3.0	4.0
11	3b	E	4	015.dat	15:12	15:21		3.0	3.0
12	2d	P	2	016.dat	15:26	15:33		3.0	3.0
13	3a	T		017.dat	16:00	16:09		3.5	3.0
14	1d	P		018.dat	16:14	16:23		3.0	2.5
15	2b	V		019.dat	16:28	16:33			
15	2b	V		020.dat	16:39	16:46		2.5	2.5
16	4c	E		021.dat	16:49	16:57		2.5	2.5

Comments:
P3: Turbulence probably at 5.0. Turbulence set to 4.3 at end of run.
P4: Don't like this display because following it too much
It has a lag but is suckering me into following it like a HSI
1: sidestep did not work, redone
7: All the runs seem the same, both parts seem the same
9: sidestep did not work, redone
15: sidestep did not work, redone

Questionnaire:
V vs T - 12/2; X vs T - 11/ 3; E vs T - 6/ 8
X vs V - 5/9; E vs V - 0/14; E vs X - 3/11
EOI: V - 1; T - 4; P - 3; E - 2
FPA: V - 1; T - 3; P - 2; E - 4
OP: V - 1; T - 3; P - 2; E - 4
OW: V - 1; T - 4; P - 3; E - 2
Enough practise approaches; V - 1; T - 2; P - 2; E - 1
Did not use numeric TAE; L/R symbol was not useful
Did not have trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Predictor movement OK
Almost never used DTK information
Turbulence same on all legs
Most recent aircraft: PA38-112 and Tomahawk; no HSI
10 hours with GPS; King GPS-Com; no moving map


T Display
P Display
V Display
E Display
18
16
14
12
10
8
6
4
2
0
-0.5 0 0.5

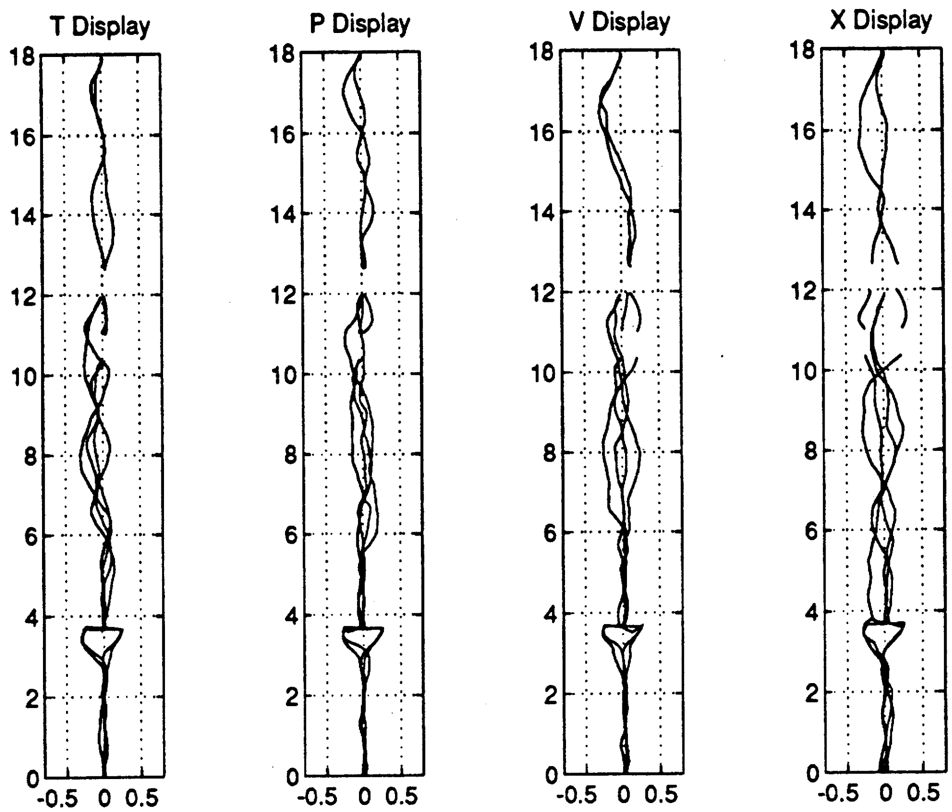
E.3. Subject Three

Date Tested: 9 February 1996
Subject Type: 1

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	T		000.dat	10:52				
P2	4p	V		001.dat	11:20	11:30		3.0	2.5
P3	3p	P		002.dat	11:37	11:51		3.0	4.0
P4	1p	X		003.dat	11:56	12:07		5.0	3.0
1	1a	X		004.dat	13:10	13:20	2 3 4	5.0	4.5
2	3d	V		005.dat	13:25	13:35	1 2 3	3.0	5.0
3	4b	P		006.dat	13:42	13:50	1 2	2.0	2.0
4	2c	T		007.dat	13:54	14:03	1	2.0	3.0
5	3c	P		008.dat	14:09	14:19	2 3 4	4.0	3.0
6	2a	X		009.dat	14:21	14:30	1 2 3	2.0	3.0
7	4d	T		010.dat	14:34	14:42	1 2	2.0	3.0
8	1b	V		011.dat	14:45	14:54	1	2.0	2.0
9	4a	V		012.dat	15:04	15:12	1 1 1	5.0	4.0
10	1c	T		013.dat	15:15	15:26	2 2 2	3.0	3.0
11	3b	X		014.dat	15:29	15:39	3 4	5.0	2.0
12	2d	P		015.dat	15:54	16:00	3	4.0	3.0
13	3a	T		016.dat	16:05	16:15	2 3 4	5.0	3.0
14	1d	P		017.dat	16:20	16:29	1 2 3	3.0	2.0
15	2b	V		018.dat	16:35	16:43	1 2	3.0	3.0
16	4c	X		019.dat	16:48	16:55	1	2.0	2.0

Comments:
8: Incorrect judgement of arrow direction at start
15: Not looking at OTW, unaware of cloud level

Questionnaire:
V vs T - 14/ 0; P vs T - 10/ 4; X vs T - 5/9
P vs V - 4/10; X vs V - 1/13; X vs P - 14/0
EOI: V - 2; T - 4; P - 3; X - 1
: Overall I like the track vector the best
FPA: V - 1; T - 3; P - 2; X - 4
OP: V - 1; T - 3; P - 2; X - 4
: Close call between 3 and 4
OW: V - 1; T - 3; P - 2; X - 4
Enough practise approaches; V - 1; T - 1; P - 1; X - 0
Did not use numeric TAE; L/R symbol was not useful
Had trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Predictor movement OK
Never used DTK information
Never used GS information
Turbulence same on all legs
Most recent aircraft: Aztec and Aerostar; has HSI
50 hours with GPS; KLN 90B; has moving map which I use



T Display
P Display
V Display
X Display
18
16
14
12
10
8
6
4
2
0
-0.5 0 0.5

E.4. Subject Four

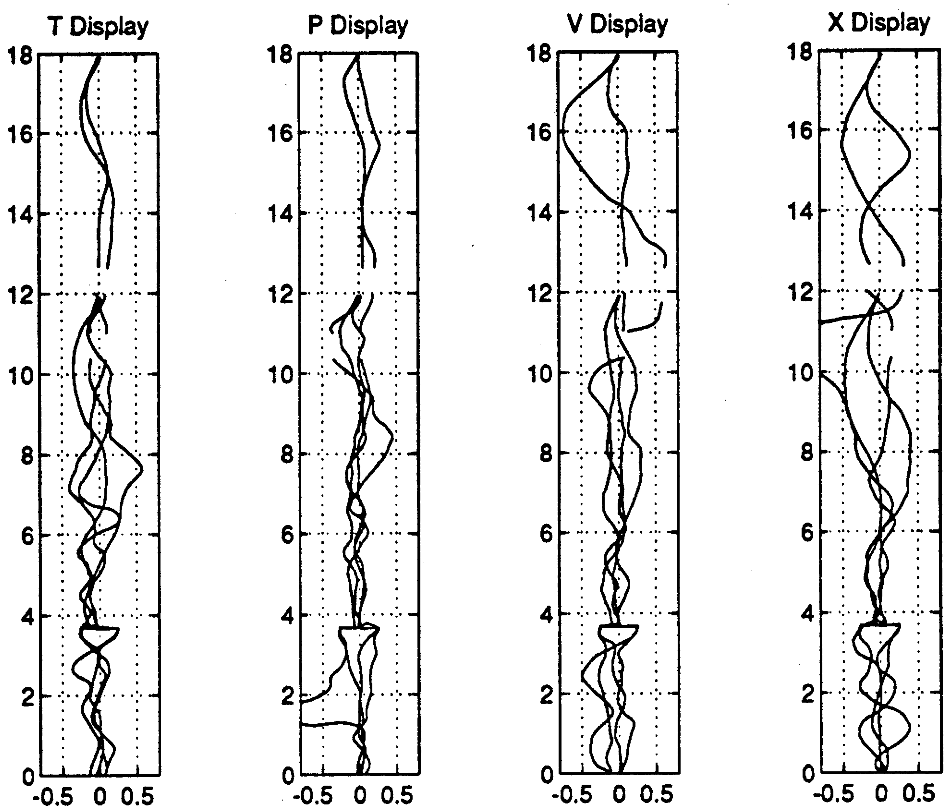
Date Tested: 28 February 1996
Subject Type: 2

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X		000.dat	11:15	11:30		4.0	7.0
P2	4p	V		001.dat	11:32	11:41			
P3	4p	V		002.dat	11:43	11:53		2.0	5.0
P4	2p	V		003.dat		12:08		3.0	6.0
P5	3p	V		004.dat	12:15	12:23		2.0	4.0
P6	1p	P		005.dat	13:20	13:30		2.0	4.5
P7	1p	T		006.dat	13:35			3.0	5.0
1	2d	P	1	007.dat	14:00	14:10	1 1 1	2.0	5.0
2	3b	X	4	008.dat	14:13	14:20	2 3 4	2.0	6.0
3	1c	T	3	009.dat	14:31	14:43	2 3	3.0	6.0
4	4a	V	2	010.dat	14:46	14:53	2	2.0	5.0
5	2c	T	4	011.dat	15:04	15:13	2 3 4	3.0	5.0
6	4b	P	1	012.dat	15:16	15:21	1 1 1	3.0	3.0
7	3d	V	2	013.dat	15:24	15:40	2 2	2.0	3.0
8	1a	X	3	014.dat	15:45	15:55	3	2.0	3.5
9	4c	X	1	015.dat	16:08	16:16	1 1 1	2.0	4.0
10	2b	V	3	016.dat	16:20	16:29	2 2 3	3.0	4.0
11	1d	P	4	017.dat	16:34	16:43	3 4	2.0	4.0
12	3a	T	2	018.dat	16:48	17:00	2	3.0	3.0
13	1b	V	4	019.dat	17:08	17:20	2 3 4	2.0	4.0
14	4d	T	3	020.dat	17:23	17:30	1 2 3	2.0	4.0
15	2a	X	2	021.dat	17:34	17:42	1 2	2.0	4.0
16	3c	P	1	022.dat	17:45	17:56	1	2.0	2.0

Comments:
Didn't use the triangle, instead used the TAE number
It was a little hard initially figuring out what was going on
1: sidestep didn't work, redone
2: Turned away from bar instead of towards, didn't see large TAE
Corrected and then over corrected, forgot to look at TAE
10: A little sidetracked in the wind
Spent too much time reading the needle
11: Couldn't tell correct direction when needles were off scale
Made wrong turn after sidestep
14: Concentrating too much on TAE

Questionnaire:
V vs T - 13/ 1; P vs T - 13/ 1; X vs T - 7/ 7
P vs V - 3/11; X vs V - 0/14; X vs P - 0/14
EOI: V - 1; T - 4; P - 2; X - 2
: Saw little difference between Triangle and XTE Only because he
: ignored triangle and only used numeric TAE
FPA: V - 1; T - 4; P - 2; X - 3
OP: V - 1; T - 4; P - 2; X - 3
OW: V - 1; T - 3; P - 1; X - 2
Enough practise approaches; V - 2; T - 2; P - 2; X - 2
Used numeric TAE; L/R symbol was not useful
Had trouble remembering Triangle direction - didn't use it
Did not have trouble remembering Predictor direction
Predictor movement OK

Used DTK information when checking TAE
Only looked at GS information about 12 times
Turbulence strongest on Intermediate Leg
Most recent aircraft: Mooney 201; has HSI
0 hours with GPS



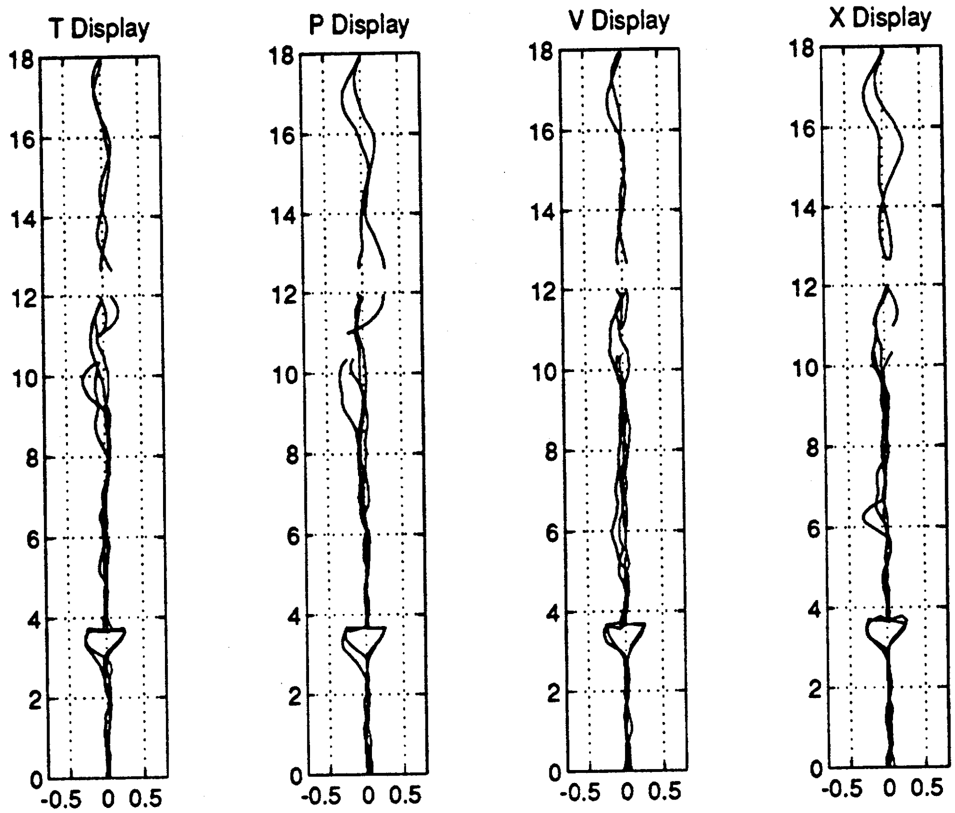
E.5. Subject Five

Date Tested: 1 March 1996
Subject Type: 3

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	T	4	000.dat	11:08	11:19		7.0	6.0
P2	4p	V	2	001.dat	11:22	11:31		5.0	5.0
P3	3p	P	1	002.dat	11:35	11:46		4.0	3.0
P4	1p	X	3	003.dat	11:50	12:00		4.0	3.0
P5	1p	T		004.dat	12:05	12:09			
1	3a	T	2	005.dat	13:00	13:12	1 1 2	3.0	3.0
2	1d	P	4	006.dat	13:17	13:23	2 3 4	3.0	4.0
3	2b	V	3	007.dat	13:34	13:41	2 3	3.0	3.0
4	4c	X	1	008.dat	13:45	13:53	1	3.0	3.0
5	4a	V	4	009.dat	13:55	14:04	2 3 4	2.0	3.0
6	1c	T	3	010.dat	14:07	14:17	1 2 3	2.0	2.0
7	3b	X		011.dat	14:20	14:30			
7	3b	X	2	012.dat	14:32	14:41	1 2	2.0	2.0
8	2d	P	1	013.dat	14:46	14:56	1	2.0	3.0
9	3c	P	3	014.dat	15:07	15:16	2 2 3	2.0	2.0
10	2a	X	1	015.dat	15:20	15:29	1 1 1	2.0	2.0
11	4d	T	4	016.dat	15:31	15:37	3 4	2.0	2.5
12	1b	V		017.dat	16:02	16:12			
12	1b	V	2	018.dat	16:14		2	2.0	2.0
13	1a	X	1	019.dat	16:29	16:39	1 1 1	3.5	2.0
14	3d	V	4	020.dat	16:43	16:52	2 3 4	2.5	2.5
15	4b	P	2	021.dat	16:55	17:05	2 2	2.0	2.0
16	2c	T	3	022.dat	17:06	17:15	3	2.0	2.5

Comments:
7: sidestep didn't work, redone
12: sidestep didn't work, redone
13: looked down to check runway and discovered was off-course

Questionnaire:
V vs T - 13/ 1; P vs T - 10/4; X vs T - 14/0
P vs V - 2/12; X vs V - 12/2; X vs P - 12/2
EOI: V - 2; T - 4; P - 3; X - 1
FPA: V - 1; T - 4; P - 3; X - 2
OP: V - 1; T - 4; P - 3; X - 2
OW: V - 1; T - 4; P - 3; X - 2
Enough practise approaches; 1 to 2 runs based on pilot preference
Used numeric TAE; <> symbols were useful
Did not have trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Predictor movement OK
Used DTK information on all arcs and most intermediate segments
Used GS information from the FAF inbound
Turbulence strongest on Final Approach
Most recent aircraft: C172; has HSI
5 hours with GPS; does have a moving map; uses map about half the time
5 hours with LORAN or RNAV; NorthStar; does not have a moving map


T Display
P Display
V Display
X Display
18
16
14
12
10
8
6
4
2
0
-0.5 0 0.5

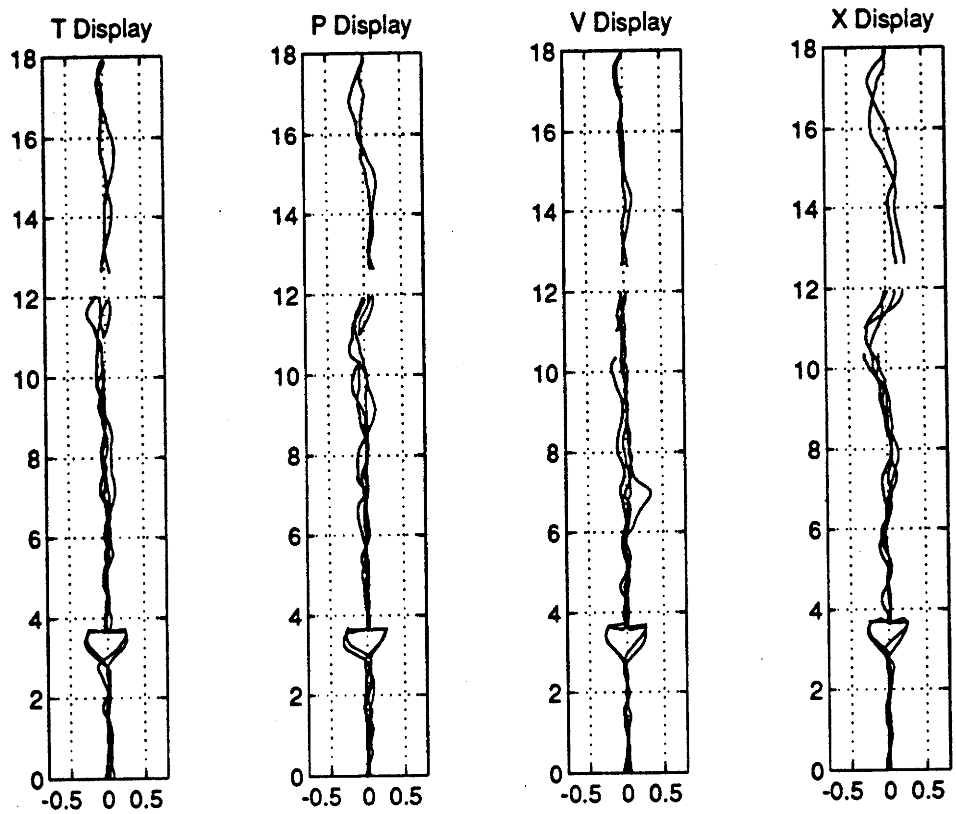
E.6. Subject Six

Date Tested: 8 March 1996
Subject Type: 4

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X		000.dat	11:30	11:38	1	9.0	9.0
P2	4p	V		001.dat	11:41	11:48	2	7.0	6.5
P3	3p	P		002.dat	11:52	12:01		5.0	4.0
P4	1p	T		003.dat	12:08	12:20			
1	1b	V	4	004.dat	13:30	13:38	2 3 4	6.5	7.0
2	4d	T	3	005.dat	13:44	13:52	1 2 3	4.0	4.5
3	2a	X	1	006.dat	13:56	14:03	1 1	3.0	3.0
4	3c	P	2	007.dat	14:06	14:17	2	2.5	3.0
5	4c	X	1	008.dat	14:25	14:33	1 1 1	2.5	2.5
6	2b	V	4	009.dat	14:36	14:46	2 3 4	2.0	2.0
7	1d	P	3	010.dat	14:48	15:00	2 3	3.0	3.0
8	3a	T	2	011.dat	15:01	15:13	2	3.0	2.0
9	2c	T	1	012.dat	15:41	15:50	1 1 1	2.0	3.0
10	4b	P	3	013.dat	15:56	16:04	2 2 3	3.0	3.0
11	3d	V	4	014.dat	16:09	16:18	3 4	3.0	4.0
12	1a	X	2	015.dat	16:22	16:33	2	3.0	3.0
13	2d	P	4	016.dat	16:38	16:46	2 3 4	3.0	3.0
14	3b	X	2	017.dat	16:49	17:00	1 2 2	2.5	2.5
15	1c	T	1	018.dat	17:02	17:13	1 1	2.0	2.0
16	4a	V	3	019.dat	17:17	17:25	3	3.0	2.0

Comments:
Noticed that he hadn't used the X display in a long while at run 12
Used mumble mode extensively
7: Reset bug to match DTK on arc repeatedly
14: Getting very tired

Questionnaire:
V vs T - 2/12; P vs T - 5/9; X vs T - 7/7
P vs V - 9/ 5; X vs V - 13/1; X vs P - 9/5
EOI: V - 4; T - 2; P - 3; X - 1
FPA: V - 4; T - 1; P - 3; X - 2
OP: V - 4; T - 2; P - 3; X - 1
OW: V - 4; T - 2; P - 3; X - 1
Enough practise approaches; V - 1; T - 1; P - 1; X - 1
Used numeric TAE; <> symbols were useful
Had trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Tried to visualise the Predictor in 3D; Found this useful
Predictor movement Too Sensitive
Used DTK information to find winds and during arcs
Rarely used GS information
Turbulence same on all legs
Most recent aircraft: Beech Duchess; has HSI
Has used a GPS; Garmin; has a moving map; uses the map
20 hours with LORAN or RNAV; NorthStar; no moving map



T Display
P Display
V Display
X Display
18
16
14
12
10
8
6
4
2
0
-0.5 0 0.5

E.7. Subject Seven

Date Tested: 6 March 1996
Subject Type: 1

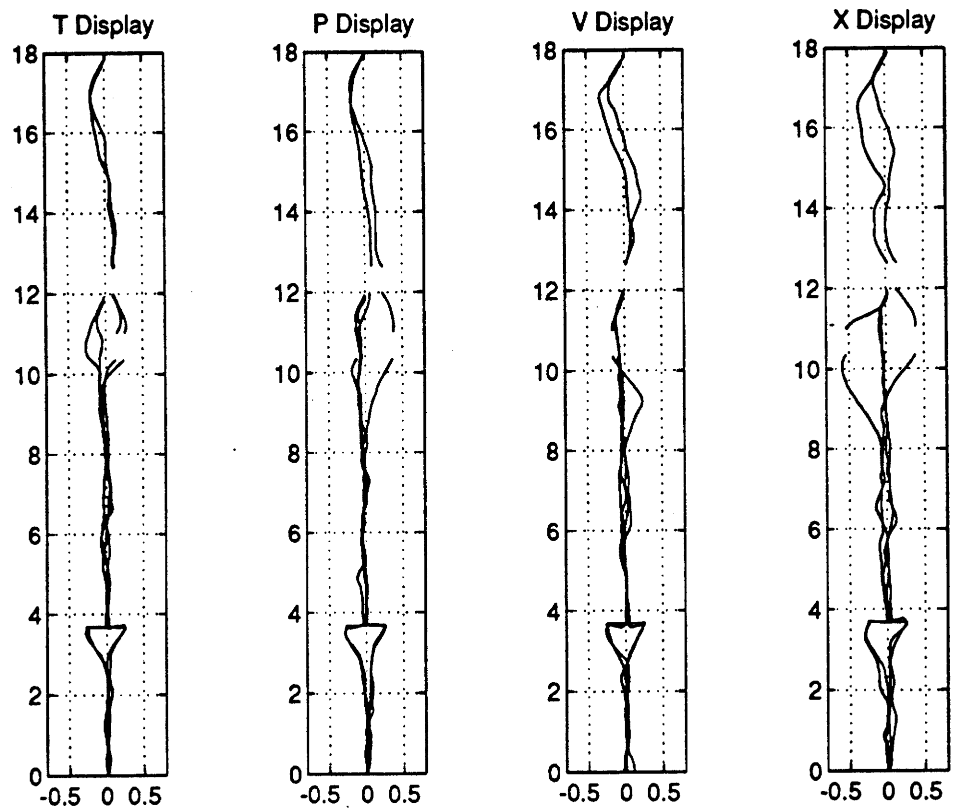
Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	1	000.dat	11:40	11:50	1 1	5.0	5.0
P2	4p	V	3	001.dat	11:55	12:05	2 3	4.0	5.0
P3	3p	P	4	002.dat	12:07	12:15	4	5.0	6.0
P4	1p	T	2	003.dat	12:23	12:35	2	5.0	6.0
1	1a	X	2	004.dat	13:22	13:32	1 2	5.0	4.0
2	3d	V	3	005.dat	13:34	13:45	2 3	6.0	5.0
3	4b	P		006.dat	13:48	13:58		3.0	7.5
3	4b	P	4	008.dat	14:12		4	5.0	6.0
4	2c	T	1	007.dat	14:00	14:07	1	4.0	5.0
5	3c	P	4	008.dat	14:29	14:38	3 4	6.0	5.0
6	2a	X	3	009.dat	14:42	14:50	2 3	5.0	5.0
7	4d	T	2	010.dat	14:52	15:00	1 2	3.0	4.0
8	1b	V	1	011.dat	15:02	15:14	1	4.0	4.0
9	4a	V	1	012.dat	15:21	15:28	1 1 1	3.0	4.0
10	1c	T	4	013.dat	15:30	15:40	2 3 4	5.0	4.0
11	3b	X	2	014.dat	15:42	15:52	2 2	4.0	5.0
12	2d	P	3	015.dat	15:55	16:03	3	4.0	4.0
13	3a	T	2	016.dat	16:10	16:19	1 1 2	3.0	4.0
14	1d	P	4	017.dat	16:21	16:30	2 3 4	5.0	4.0
15	2b	V	3	018.dat	16:33	16:41	2 3	4.0	5.0
16	4c	X	1	019.dat	16:43	16:50	1	3.0	4.0

Comments:
P3: Found predictor tough to visualise
P4: Likes the triangle display best
3: Sidestep didn't work, redone
: Sometimes got confused between the two needles
10: Likes XTE Only display best
14: Getting tired

Questionnaire:
V vs T - 7/ 7; P vs T - 5/9; X vs T - 10/4
P vs V - 4/10; X vs V - 9/5; X vs P - 13/1
EOI: V - 2; T - 4; P - 3; X - 1
FPA: V - 3; T - 2; P - 4; X - 1
OP: V - 3; T - 2; P - 4; X - 1
OW: V - 3; T - 2; P - 4; X - 1
Enough practise approaches; V - 1; T - 1; P - 1; X - 1
Initially thought 2 practise per display but reconsidered
Used numeric TAE; <> symbols were useful
Used numeric TAE all the time, on almost all approaches
Did not have trouble remembering Triangle direction
Had trouble remembering Predictor direction at first
Tried to visualise the Predictor in 3D; Did not find this useful
Thought 3D was too much information
Predictor movement Too Sensitive
Used DTK information occasionally
Never used GS information but thought it was a good thing to have, just didn't use it this day
Turbulence strongest on Final Approach

Most recent aircraft: Cessna; does not have HSI
Has flown a Warrior which does have an HSI
0 hours with GPS
12 hours with LORAN; Tomorrow; no moving map

Tape Comments:
Had difficulty seeing the Predictor in 3D
For someone who is inexperienced, the predictor complicates matters where they don't need to be complicated
XTE Only is easiest to use, doesn't clutter you with information you don't need
I originally thought the triangle might be better
Flying the plane in turbulence you don't have time to picture looking down, looking at this, etc



E.8. Subject Eight

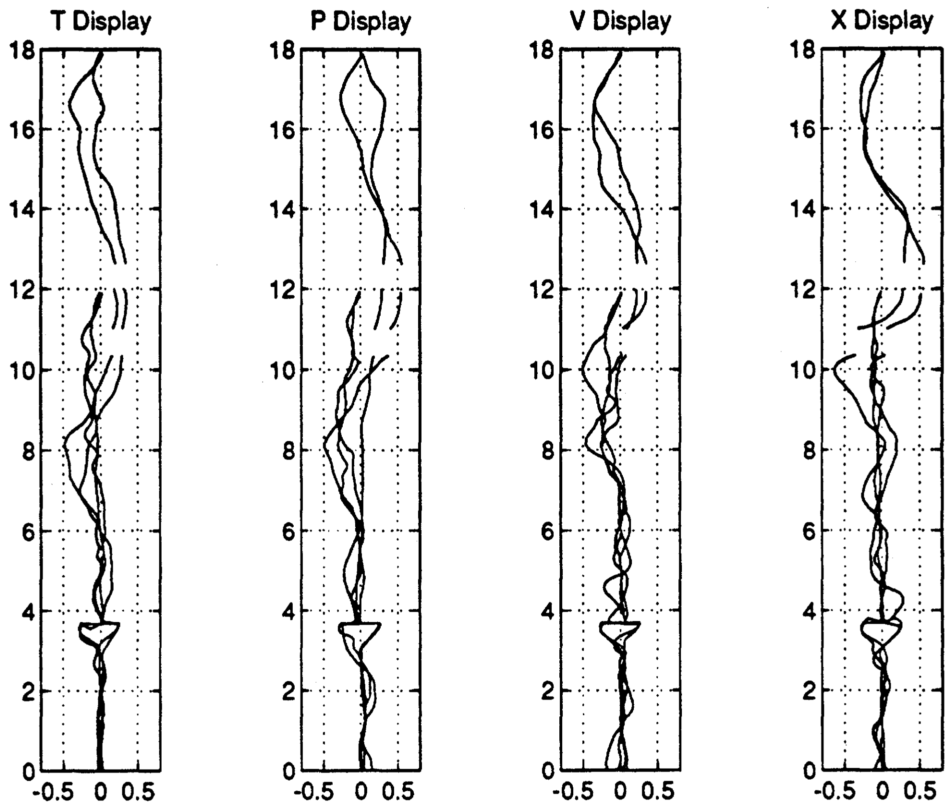
Date Tested: 13 March 1996
Subject Type: 2

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	1	000.dat	10:55	11:06		3.0	4.0
P2	4p	V	2	001.dat	11:07	11:15		2.0	2.0
P3	3p	P	4	002.dat	11:16	11:27		4.0	4.0
P4	1p	T	3	003.dat	11:30	11:41		5.0	4.0
1	2d	P	3	004.dat	12:36	12:44	2 3 3	3.0	3.0
2	3b	X	2	005.dat	12:46	12:56	1 2 2	3.0	4.0
3	1c	T	1	006.dat	13:01	13:11	1 1	3.0	2.0
4	4a	V	4	007.dat	13:15	13:24	4	5.0	7.0
5	2c	T	3	008.dat	13:31	13:38	2 2 3	2.0	2.0
6	4b	P	1	009.dat	13:41	13:50	1 1 1	3.0	2.0
7	3d	V	4	010.dat	13:53	14:03	3 4	5.0	7.0
8	1a	X	2	011.dat	14:07	14:16	2	3.0	3.0
9	4c	X	1	012.dat	14:23	14:31	1 1 1	2.0	2.0
10	2b	V	4	013.dat	14:31	14:42	2 3 4	5.0	9.0
11	1d	P	2	014.dat	14:45	14:55	2 2	4.0	4.0
12	3a	T	3	015.dat	14:59	15:08	3	4.0	5.0
13	1b	V	1	016.dat	15:24	15:33	1 1 1	3.0	3.0
14	4d	T	3	017.dat	15:38	15:46	2 3 3	2.0	2.0
15	2a	X	2	018.dat	15:48	15:56	2 2	2.0	2.0
16	3c	P	4	019.dat	15:59	16:20	4	3.0	2.0

Comments:
12: OTW display is really confusing, would have gone miss.

Questionnaire:
V vs T - 4/10; P vs T - 4/10; X vs T - 4/10
P vs V - 5/ 9; X vs V - 9/ 5; X vs P - 9/ 5
EOI: V - 4; T - 1; P - 2; X - 3
: Once I finally got used to the track vector, I liked it
FPA: V - 3; T - 2; P - 4; X - 1
: It seems as if more information was given, it takes more time away
: from scanning attitude control
OP: V - 4; T - 1; P - 3; X - 2
: the more simple the instrument display the easier to fly
: eg. ADF - easier to see
OW: V - 1; T - 2; P - 4; X - 3
: rating goes from most simple to most information
Enough practise approaches; V - 1; T - 1; P - 1; X - 1
Used numeric TAE from the FAF inbound; <> symbols were not useful
Used numeric TAE all the time, on almost all approaches
Did not have trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Did not try to visualise the Predictor in 3D; Did not find this useful
Found the Predictor to not be useful
Used DTK information from IAF to FAF
Used GS information at IAF only
Turbulence strongest on Final Approach
Most recent aircraft: Piper Warrior; has HSI
7 hours with GPS; Garmin; has moving map; uses map
30 hours with RNAV; KLN 88; no moving map

Tape Comments:
Wanted analog information in primary field of view; OK and useful to have numerics available to the side, put nav unit as close as possible.
He trains students never to look away from primary instruments for more than 3 seconds.
TAE information is useful; felt he knew well how to use it by midway through the experiment.



E.9. Subject Nine

Date Tested: 18 March 1996
Subject Type: 3

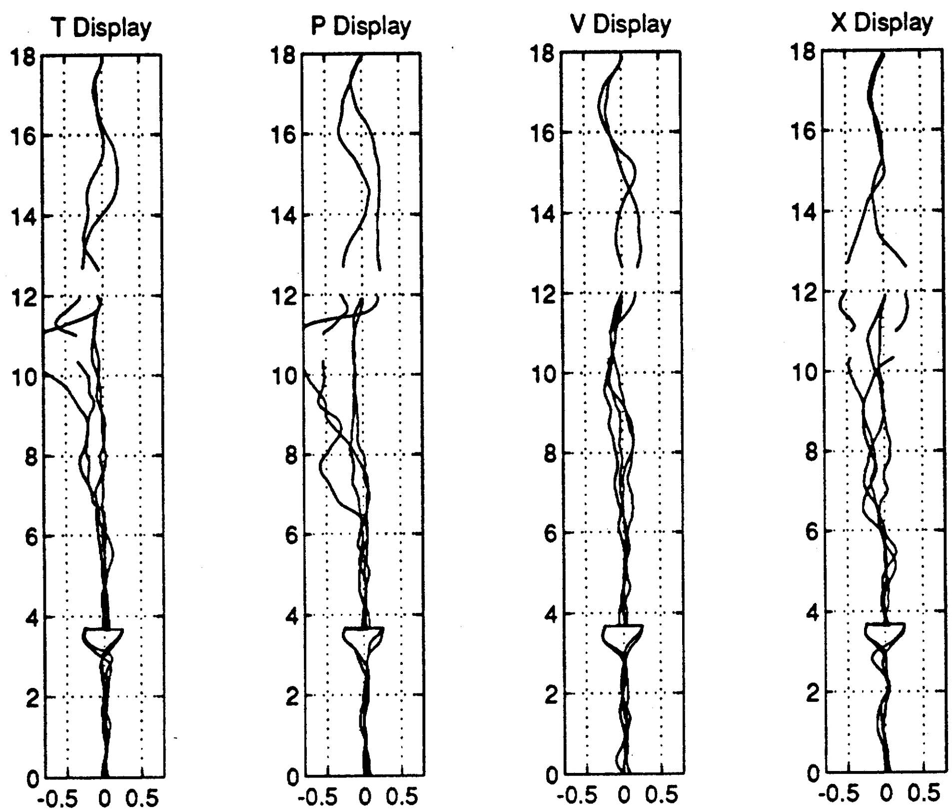
Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	1	000.dat	11:05	11:18	1 1	2.2	3.0
P2	4p	V	2	001.dat	12:01	12:10	2 2	2.4	3.3
P3	3p	P	4	002.dat	12:13	12:23	4	2.8	3.7
P4	1p	T	3	003.dat	12:27	12:38	3	2.7	3.0
1	3a	T	3	004.dat	13:34	13:45	1 2 3	4.2	3.5
2	1d	P	4	005.dat	13:49	13:59	2 3 4	3.3	4.2
3	2b	V	2	006.dat	14:10	14:17	1 2	2.0	3.6
4	4c	X	1	007.dat	14:23	14:31	1	2.1	2.6
5	4a	V	2	008.dat	14:34	14:42	1 1 2	1.7	2.6
6	1c	T	4	009.dat	14:46	14:56	2 3 4	3.2	4.1
7	3b	X	3	010.dat	15:00	15:09	2 3	2.8	4.2
8	2d	P	1	011.dat	15:14	15:22	1	2.4	3.4
9	3c	P	4	012.dat	15:36	15:48	2 3 4	2.8	3.8
10	2a	X	1	013.dat	15:51	15:58	1 1 1	2.6	3.2
11	4d	T	3	014.dat	16:00	16:07	2 3	2.4	3.1
12	1b	V	2	015.dat	16:12	16:22	2	2.2	2.4
13	1a	X	3	016.dat	16:34	16:44	1 2 3	2.1	3.1
14	3d	V	4	017.dat	16:46	16:55	2 3 4	2.4	2.4
15	4b	P	2	018.dat	17:00	17:06	1 2	1.8	2.2
16	2c	T	1	019.dat	17:08	17:16	1	1.8	1.8

Comments:
3: Restarted because display not working correctly

Questionnaire:
V vs T - 2/12; P vs T - 14/ 0; X vs T - 0/14
P vs V - 14/ 0; X vs V - 2/12; X vs P - 0/14
EOI: V - 4; T - 1; P - 3; X - 2
FPA: V - 4; T - 2; P - 1; X - 3
OP: V - 4; T - 2; P - 1; X - 3
OW: V - 4; T - 3; P - 2; X - 1
Enough practise approaches; V - 1; T - 1; P - 1; X - 1
Used numeric TAE; <> symbols were not useful
Was able to visualise TAE direction without <> symbols
Did not have trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Did try to visualise the Predictor in 3D; Found this useful
Predictor movement was OK
Used DTK information for initial along the arc
Used GS information periodically
Turbulence strongest on Final Approach
Most recent aircraft: CE-550, PA-317; has HSI
500+ hours with GPS; Garmin & Trimble; no moving map
800+ hours with LORAN; no moving map

Tape Comments:
TAE helped. Flew one intercept at 90°. Put it into a bank and reintercepted. Used on arcs also.
Choice of intercept angles: chose 45° or less usually
Didn't use GS much

His current aircraft have GPS, but not IFR approved. Use them enroute. Doesn't use TAE - mostly uses with autopilot. Will look at manuals and see if it has TAE
Liked the Track Vector but preferred Predictor and Triangle for simplicity



E.10. Subject Ten

Date Tested: 18 March 1996
Subject Type: 4

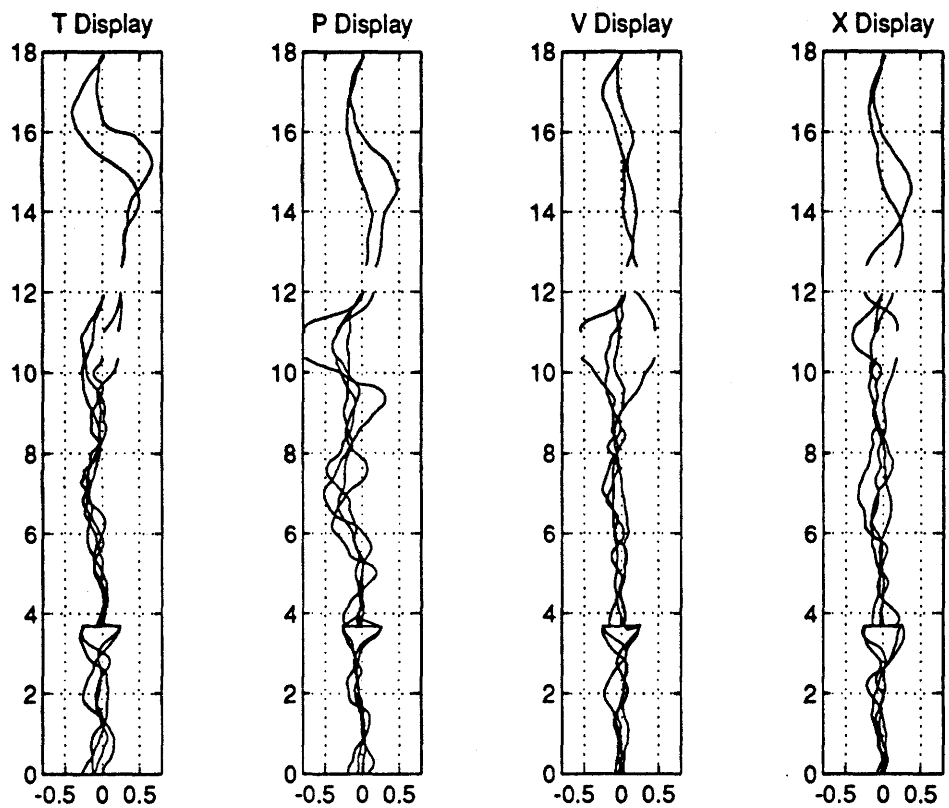
Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	3	000.dat	11:00	11:08	1 2 3	7.0	6.0
P2	4p	V	4	001.dat	11:10	11:18	2 3 4	7.0	8.0
P3	3p	P	1	002.dat	11:24	11:36	1 1	3.0	4.0
P4	1p	T	2	003.dat	11:51	12:02	2	4.0	5.0
P5	3p	P		004.dat	12:07	12:16		3.0	3.0
P6	1p	V		005.dat	12:22	12:35		3.0	3.0
1	1b	V	2	006.dat	13:36	13:47	1 1 2	3.0	3.0
2	4d	T	4	007.dat	13:50	13:58	2 3 4	5.0	4.0
3	2a	X	3	008.dat	14:03	14:12	2 3	3.0	4.0
4	3c	P	1	009.dat	14:15	14:26	1	6.0	3.0
5	4c	X	2	010.dat	14:48	14:56	1 1 2	2.0	3.0
6	2b	V	3	011.dat	15:00	15:07	2 2 3	4.0	4.0
7	1d	P	4	012.dat	15:10	15:20	3 4	3.0	4.0
8	3a	T	1	013.dat	15:25	15:35	1	3.0	3.0
9	2c	T	2	014.dat	15:47	15:55	1 1 2	3.0	3.0
10	4b	P	4	015.dat	15:59	16:06	2 3 4	5.0	6.0
11	3d	V	3	016.dat	16:10	16:24	2 3	4.0	4.0
12	1a	X	1	017.dat	16:37	16:46	1	3.0	3.0
13	2d	P	1	018.dat	16:54	17:02	1 1 1	3.0	3.0
14	3b	X	3	019.dat	17:04	17:15	2 3 3	3.0	3.0
15	1c	T	2	020.dat	17:17	17:29	2 2	3.0	3.0
16	4a	V	4	021.dat	17:32	17:40	4	3.0	3.0

Comments:
4: Felt like I was disoriented at the turn
: Looked at other instruments for the first time
5: Sidestep didn't work, used for engine failure and redone

Questionnaire:
V vs T - 14/0; P vs T - 11/3; X vs T - 14/0
P vs V - 14/0; X vs V - 14/0; X vs P - 14/0
EOI: V - 4; T - 3; P - 2; X - 1
: Track Vector least understood
FPA: V - 4; T - 2; P - 3; X - 1
: Triangle was difficult to interpret, translate to aircraft flight
: path, XTE was best
OP: V - 4; T - 3; P - 2; X - 1
: XTE was simple, required less thought
OW: V - 4; T - 3; P - 2; X - 1
: Track Vector, Triangle required too much "projection" ahead of the
: aircraft for this pilot
Enough practise approaches; V - 3; T - 2; P - 1; X - 1
Used numeric TAE mostly from IAF to MAP; <> symbols were useful
Was able to visualise TAE direction without <> symbols
Had trouble remembering Triangle direction; too weird
Did not have trouble remembering Predictor direction; but had trouble
with amount required to stop correction
Did not try to visualise the Predictor in 3D
Predictor movement was OK; disregarded it on short final

Used DTK information very little, during initial turns to intermediate approach
Never used GS information
Turbulence was Same on all legs
Most recent aircraft: PA-28-200; does not have HSI
20 hours with GPS; Trimble, hand held; no moving map
0 hours with LORAN or RNAV

Tape Comments:
Pilot was 6/6/6 current but had not flown actual instruments since November - he said he felt rusty
Had not read Pilot Brief. He felt he needed to practice approaches.
Pilot commented that he over controlled; neglected attitude; felt nervous.



E.11. Subject Eleven

```
Date Tested:      27 March 1996
Subject Type:     1
```

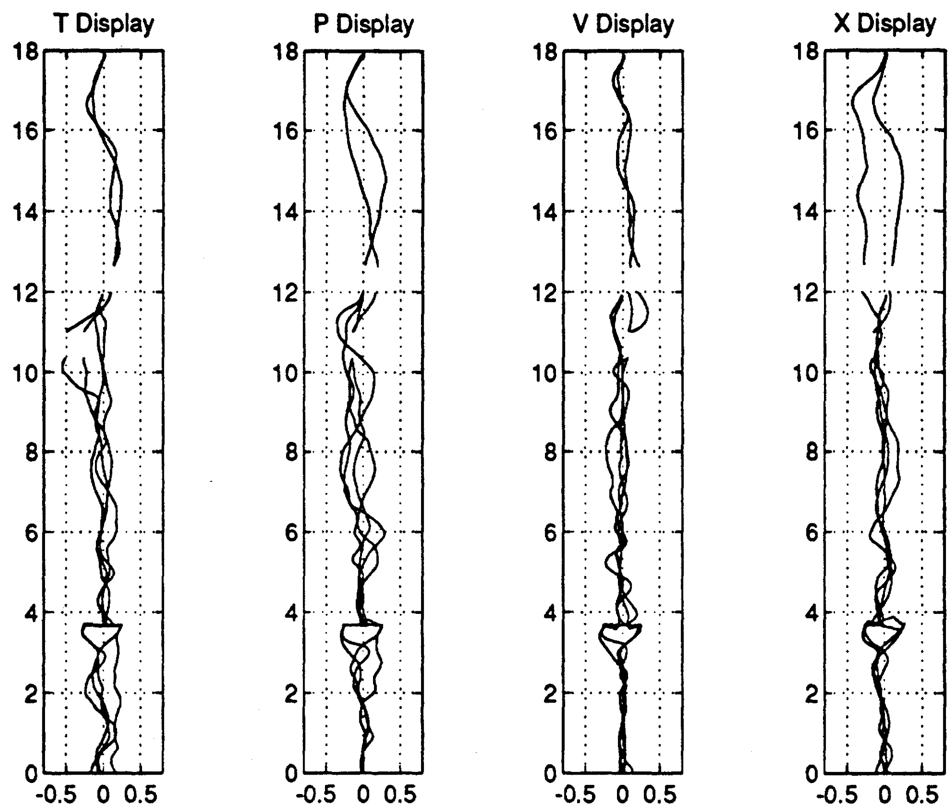
Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	3	000.dat	11:51	12:00	3 3	3.0	6.0
P2	4p	V	2	001.dat	12:12	12:22	2 2	3.0	6.0
P3	3p	P	1	002.dat	12:25	12:35	1 1	3.0	3.0
P4	1p	T	4	003.dat	12:42	12:52	4	3.0	4.0
1	1a	X	2	004.dat	13:43	13:53	1 2	3.0	4.0
2	3d	V	1	005.dat	13:59	14:07	2 1	3.0	5.0
3	4b	P	3	006.dat			3	3.0	5.0
4	2c	T	4	007.dat	14:28	14:36	4	5.0	5.0
5	3c	P	4	008.dat	14:45	14:58	2 3 4	3.0	3.0
6	2a	X	3	009.dat	15:01	15:07	1 2 3	3.0	3.0
7	4d	T	1	010.dat	15:09	15:20	1 1	2.0	2.0
8	1b	V	2	011.dat	15:25	15:34	2	2.0	2.0
9	4a	V	2	012.dat	15:52	16:04	2 2 2	2.0	2.0
10	1c	T	1	013.dat	16:08	16:18	1 1 1	2.0	2.0
11	3b	X	3	014.dat	16:23	16:35	3 3	2.0	2.0
12	2d	P	4	015.dat	16:37	16:50	4	3.0	3.0
13	3a	T	4	016.dat	17:05	17:16	2 3 4	4.0	4.0
14	1d	P	2	017.dat	17:19	17:29	1 1 2	3.0	3.0
15	2b	V	3	018.dat	17:32	17:41	2 3	3.0	3.0
16	4c	X	1	019.dat	17:45	17:53	1	1.5	1.5

```
Comments:
 2: Hit by severe turbulence at the FAF
12: Getting tired, pitch is very stiff

Questionnaire:
V vs T - 2/12; P vs T -  7/7; X vs T - 12/2
P vs V - 7/ 7; X vs V - 11/3; X vs P - 10/4
EOI: V - 4; T - 2; P - 3; X - 1
FPA: V - 3; T - 2; P - 4; X - 1
   : Felt she was constantly chasing the Predictor
 OP: V - 4; T - 2; P - 3; X - 1
   : Liked the simple line of the XTE Only
 OW: V - 4; T - 2; P - 3; X - 1
Enough practise approaches; V - 1; T - 1; P - 1; X - 1
Used numeric TAE; <> symbols were useful
Was able to visualise TAE direction without <> symbols
Did not have trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Did try to visualise the Predictor in 3D; Found this useful
Predictor movement was OK
Rarely used DTK information; didn't recall.  Might have used it more
enroute and on arcs
Used GS information sometimes on final approach (only glanced as ref.);
focused on other things
Turbulence was Same on all legs; "hard to say"
Most recent aircraft: ; has HSI
Mother's aircraft has HSI - this pilot has flown that aircraft a lot
0 hours with GPS
```

up to 100 hours with LORAN and RNAV; LORAN - NorthStar, RNAV - KNS 81; has used Argus 5000 moving map.

Tape Comments:
She was very tired. Worked till 11 the previous night. Got up early to come in.
Felt flying the simulator in turbulence depended on her arm strength.
Used numeric TAE sometimes too much at first. Found freeze heading then flew either side. Used it on arcs - found bank that stabilised TAE.
Has no instrument students.
Glasses broke during the experiment.
Needed back bolster to reach rudder pedals
Rusty on instruments.



E.12. Subject Twelve

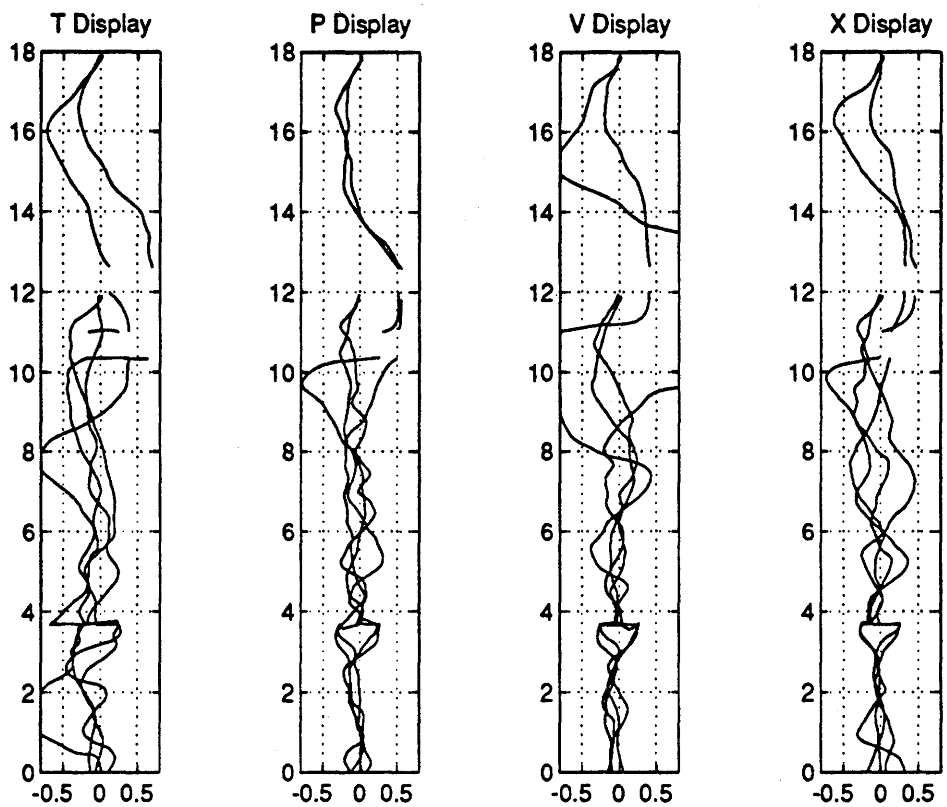
Date Tested: 3 April 1996
Subject Type: 2

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	2	000.dat	10:05	10:15	2 2 2	5.0	7.0
P2	4p	V	1	001.dat	10:21	10:34	1 1 1	3.0	4.0
P3	3p	P	3	002.dat	10:37	10:46	3 3	3.0	5.0
P4	1p	T	4	003.dat	10:50	11:04	4	5.0	6.0
P5	3p	T		004.dat	11:09	11:20			
P6	1p	V		005.dat				5.0	6.0
1	2d	P	3	006.dat	12:24	12:35	2 3 3	6.0	6.0
2	3b	X	2	007.dat	12:35	12:45	1 2 2	4.0	4.0
3	1c	T	1	008.dat	12:47	12:57	1 1	4.0	5.0
4	4a	V	4	009.dat	13:04	13:10	4	6.0	7.0
5	2c	T	1	010.dat	13:18	13:20	1 1 1	7.0	9.0
6	4b	P	2	011.dat	13:28	13:36	2 2 2	5.0	4.0
7	3d	V	4	012.dat	13:39	13:50	3 4	5.0	9.0
8	1a	X	3	013.dat	13:52	14:01	3	7.0	6.0
9	4c	X	2	014.dat	14:02	14:20	2 2 2	3.0	4.0
10	2b	V	1	015.dat	14:30	14:38	1 1 1	3.0	4.0
11	1d	P	4	016.dat	14:40	14:50	3 4	4.0	6.0
12	3a	T	3	017.dat	14:55	15:05	3	7.0	9.0
13	1b	V	2	018.dat	15:14	15:24	1 2 2	5.0	7.0
14	4d	T	3	019.dat	15:22	15:35	2 3 3	7.0	8.0
15	2a	X	1	020.dat	15:39	15:46	1 1	5.0	7.0
16	3c	P	4	021.dat	15:50	16:00	4	8.0	9.0

Comments:
P4: didn't pay attention to display
: triangle reminds of CDI on LORAN
5: took off too early after hitting ground, thought MDA was 500ft
: pilot was just not with it on final
12: thought DUEL waypoint was actually the MAP

Questionnaire:
V vs T - 4/10; P vs T - 2/12; X vs T - 7/7
P vs V - 11/ 3; X vs V - 9/ 5; X vs P - 9/5
EOI: V - 4; T - 2; P - 3; X - 1
: I'm used to seeing a CDI. Just a matter of familiarity and training
FPA: V - 4; T - 3; P - 2; X - 1
: Tie between Triangle and Predictor broken
OP: V - 4; T - 3; P - 2; X - 1
OW: V - 4; T - 3; P - 2; X - 1
Enough practise approaches; V - 1; T - 1; P - 1; X - 1
"Eight is kind of a lot, maybe 2 extras"
Did not used numeric TAE; <> symbols were not useful
Symbols and <> changed so fast. If I'd had heading under control, maybe it would have been more useful
Did not have trouble remembering Triangle direction; confused at the beginning, but worked it out towards the end
Did not have trouble remembering Predictor direction
Did not try to visualise the Predictor in 3D; Did not find this useful
Predictor movement Too sensitive; Can't remember too well what it was like

Used DTK information outside the FAF
Rarely used GS information; didn't use it for much
Turbulence was Same on all legs
Most recent aircraft: PA28-181; does not have HSI
250 hours with GPS; KLN90A; has moving map; uses map
100 hours with LORAN; NorthStar; no moving map
300 hours with RNAV; King?; no moving map



E.13. Subject Thirteen

Date Tested: 8 April 1996
Subject Type: 3

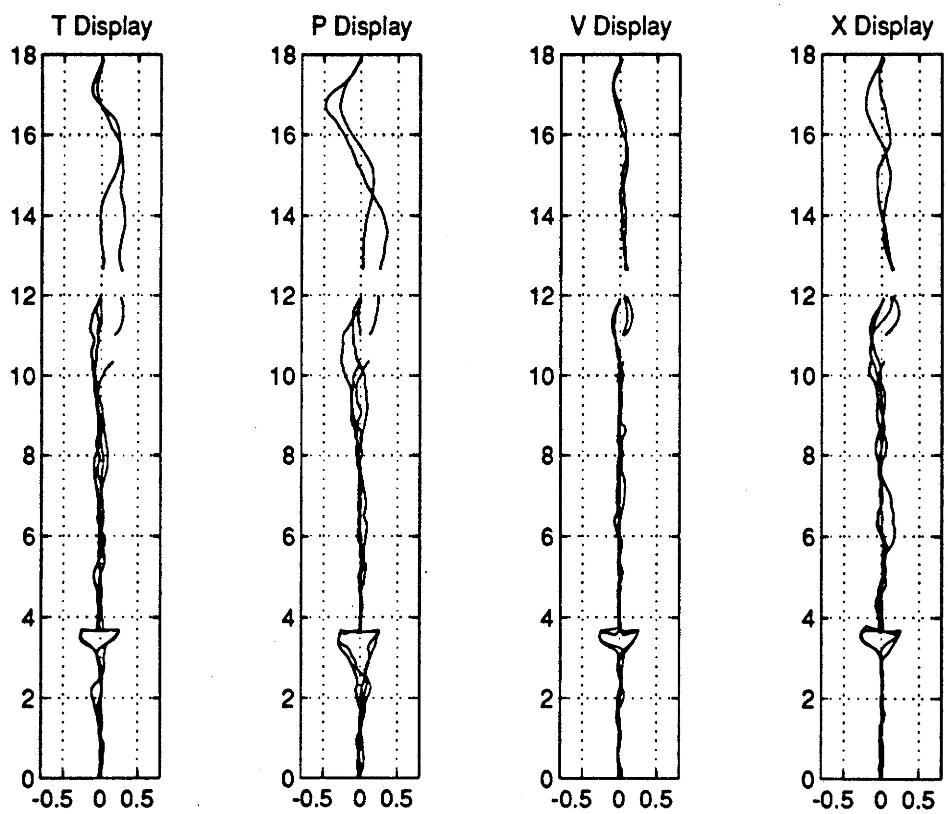
Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	2	000.dat	10:54	11:02	1 2 2	3.0	3.0
P2	4p	V	1	001.dat	11:05	11:12	2 1 1	3.0	3.0
P3	3p	P	4	002.dat	11:16	11:30	3 4	4.0	5.5
P4	1p	T	3	003.dat	11:32		3	7.0	5.0
1	3a	T	3	004.dat	12:48	12:58	1 2 3	3.0	4.5
2	1d	P	4	005.dat	13:02	13:11	2 3 4	8.0	9.0
3	2b	V	2	006.dat	13:25	13:33	1 2	3.5	3.5
4	4c	X	1	007.dat	13:35	13:44	1	2.0	3.5
5	4a	V	1	008.dat	13:51	13:59	1 1 1	2.0	3.0
6	1c	T	2	009.dat	14:02	14:10	2 2 2	3.5	3.0
7	3b	X	3	010.dat	14:13	14:23	3 3	2.0	3.0
8	2d	P	4	011.dat	14:27	14:34	4	4.0	4.5
9	3c	P	2	012.dat	14:48	14:58	2 2 2	4.0	4.0
10	2a	X	1	013.dat	15:00	15:08	1 1 1	2.0	3.0
11	4d	T	3	014.dat	15:11	15:19	3 3	3.0	4.0
12	1b	V	4	015.dat	15:22	15:31	4	4.0	5.0
13	1a	X	1	016.dat	15:50	16:00	1 1 1	4.0	3.0
14	3d	V	2	017.dat	16:03	16:12	2 2 2	4.0	4.0
15	4b	P	3	018.dat	16:16	16:23	3 3	5.0	5.5
16	2c	T	4	019.dat	16:26	16:35	4	4.0	6.0

Comments:
Averages 100 approaches per month
P3: Hardest of three
2: turned wrong way because second guessing
: turbulence seemed very severe
: ended up just chasing the needle
15: turbulence seemed to increase on this run

Questionnaire:
V vs T - 13/ 1; P vs T - 11/ 3; X vs T - 2/12
P vs V - 2/12; X vs V - 0/14; X vs P - 0/14
EOI: V - 1; T - 4; P - 3; X - 2
: Triangle takes longer to interpret. You have to experiment.
: Wasn't able to fly Triangle around with bank angle.
: Track Vector looks like a map, easy to interpret.
: XTE Only numeric TAE information didn't help
FPA: V - 1; T - 3; P - 2; X - 4
OP: V - 1; T - 3; P - 2; X - 4
OW: V - 1; T - 4; P - 3; X - 2
Enough practise approaches; V - 1; T - 2; P - 1; X - 1
Two approaches with Triangle to practise with wind corrections
Did not use numeric TAE; <> symbols were not useful
When he got stabilised he'd look at it to find freeze heading, especially with XTE Only and Triangle. It wasn't obvious what the arrow was telling him (command vs status ambiguity).
Had trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Did try to visualise the Predictor in 3D; Found this useful
Predictor movement was OK

Never used DTK information; not even on arcs
Used GS information to determine rate of descent
Turbulence was Same on all legs; "Hard to say"
Most recent aircraft: Beech 1900; has HSI
75 hours with GPS; Apollo; has moving map; uses map
30 hours with RNAV; King; no moving map
75-100 & 75 hours with LORAN; NorthStar, Apollo; no moving map

Tape Comments:
Had flown a great deal with EFIS, which have a map mode with vector.
He prefers to fly with EHSI mode.
Flies contact with sectional folded track up.
Additional TAE information helped on Track Vector.
Flies many IFR approaches each week. Flies two pilot BE-02.
All hand flying, little or no autopilot because of short legs.
Impressive pilot. He trained up very fast. Instrument proficiency was very apparent. Gave him full turbulence very early. Probably the lowest FTE, smoothest pilot so far.



E.14. Subject Fourteen

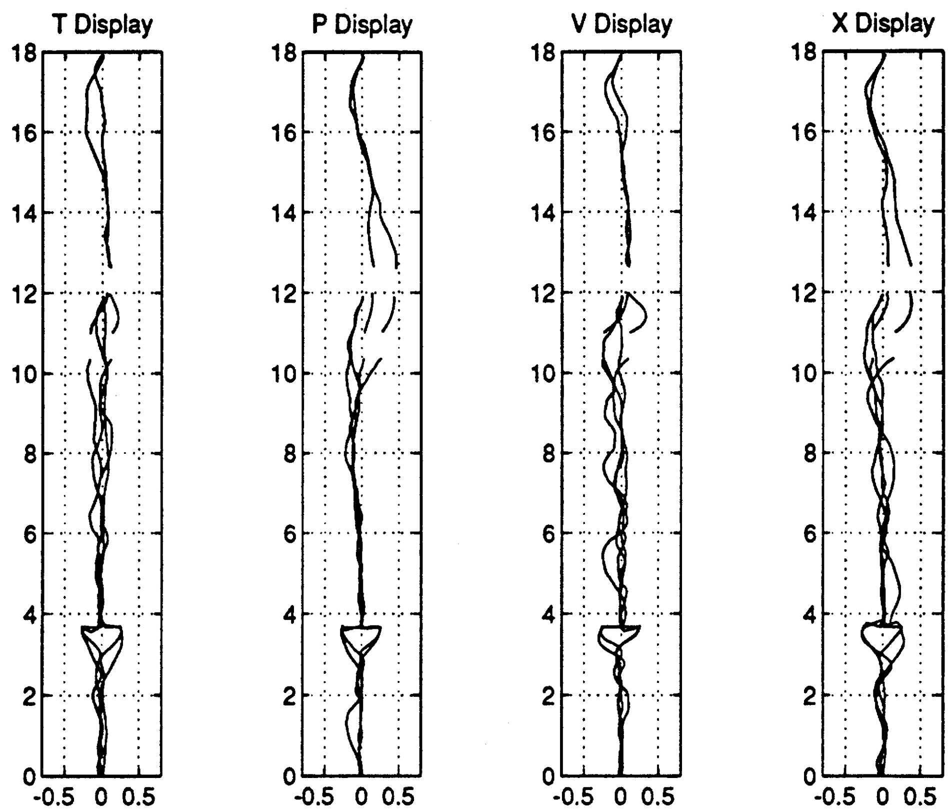
Date Tested: 18 March 1996
Subject Type: 4

Test #	Flight Plan	Display	Workload Ranking	File Number	Start Time	Stop Time	Progressive Rankings	Initial Workload	Final Workload
P1	2p	X	4	000.dat	10:46	10:56	2 3 4	3.0	4.0
P2	4p	V	2	001.dat	10:59	11:08	1 1 2	4.5	3.5
P3	3p	P	3	002.dat	11:12	11:22	2 3	3.0	5.0
P4	1p	T	1	003.dat	11:28	11:38	1	2.5	4.5
1	1b	V	4	004.dat	12:26	12:36	2 3 4	4.0	6.0
2	4d	T	2	005.dat	12:42	12:49	1 2 2	3.5	4.0
3	2a	X	1	006.dat	12:53	13:01	1 1	4.0	4.0
4	3c	P	3	007.dat	13:12	13:22	3	3.5	4.5
5	4c	X	1	008.dat	13:37	13:45	1 1 1	2.0	2.5
6	2b	V	4	009.dat	13:49	13:53	2 3 4	3.0	4.5
7	1d	P	3	010.dat	14:08	14:17	2 3	3.5	5.5
8	3a	T	2	011.dat	14:20	14:31	2	2.0	3.0
9	2c	T	2	012.dat	14:40	14:46	1 1 2	3.0	4.5
10	4b	P	3	013.dat	14:53	15:02	2 2 3	2.5	5.0
11	3d	V	4	014.dat	15:06	15:15	3 4	3.5	5.5
12	1a	X	1	015.dat	15:20	15:29	1	2.5	5.5
13	2d	P	3	016.dat	15:38	15:46	2 3 3	3.5	4.5
14	3b	X	2	017.dat	15:49	15:59	1 2 2	2.0	2.0
15	1c	T	1	018.dat	16:03	16:13	1 1	2.0	3.0
16	4a	V	4	019.dat	16:17		4	3.5	5.0

Comments:
P2: At first was reading Vector backwards
P3: Finds DTW number extremely useful especially for picturing where
  : aircraft is
  : Busy on final segment
P4: Likes Triangle display because it shows where I am and what I am
  : doing. Sometimes had to remind self of which way.
 2: Had refined technique and felt more comfortable
 3: Didn't get much information from display
 8: It is important to declutter information. Triangle display and XTE
  : Only are good. But Vector and Predictor have too many lines. When
  : things are busy you need to be able to just get the information that
  : you need.
10: Read display in reverse for a minute before the MAP.

Questionnaire:
V vs T - 3/11; P vs T - 5/9; X vs T - 7/7
P vs V - 9/ 5; X vs V - 12/2; X vs P - 8/6
EOI: V - 4; T - 2; P - 3; X - 1
  : XTE has least information, therefore easiest to see and understand
  : Best for interpretation at a glance
FPA: V - 4; T - 1; P - 3; X - 2
  : Simple basic presentation; the Triangle helped get back on track a
  : little faster
OP: V - 4; T - 2; P - 3; X - 1
  : XTE and Triangle very similar, would find either one very
  : satisfactory in ??? weather
OW: V - 4; T - 1; P - 3; X - 2
  : Triangle helped calculate an display returns to course very well.

Enough practise approaches; V - 2+; T - 1 to 2; P - 2; X - 1
Got better during the session and workload decreased
Did not use numeric TAE; <> symbols were not useful
Did not have trouble remembering Triangle direction
Did not have trouble remembering Predictor direction
Did try to visualise the Predictor in 3D; Found this useful
Predictor movement was OK
Never used DTK information
Used GS information not very much today but don't lose it
Turbulence was Same on all legs
Most recent aircraft: SAAB 340B; has HSI (EHSI)
0 hours with GPS
450 hours with LORAN; Apollo; no moving map
250 hours with OMEGA; Marconi; no moving map



APPENDIX F: SIMULATOR PROGRAMS

DATALINK.BAS

DDELINKS.BAS

EXPT2.BAS

EXPT2.FRM

EXPT2.VBP

FLGTPLAN.BAS

GPSSERIA.FRM

GPSSERIA.MAK

GPS_NAV.BAS

GPS_PROC.BAS

OTW.BAS

OTW.FRM

OTW.MAK

SERIAL.BAS

SERIAL.FRM

SYSSET.BAS

SYSSET.FRM

THREED.BAS

TRIGCODE.BAS

```
Option Explicit

Global Const MAXSOCKBUFFER = 2048

Type SocketBuffer
    Buffer As String * MAXSOCKBUFFER
End Type

Type IDSockBuffer
    ID As Long
    Buffer As SocketBuffer
End Type

Global Const IDMSG_REQUEST = 1
Global Const IDMSG_REPLY = 2
Global Const DSCMD_GETDATA = 3
Global Const DSCMD_SETLATLONG = 4
Global Const DSCMD_SETALT = 5
Global Const DSCMD_SETPITCH = 6
Global Const DSCMD_SETROLL = 7
Global Const DSCMD_SETYAW = 8
Global Const DSCMD_SETAIRSPEED = 9
Global Const DSMSG_DATA = 10
Global Const GPSCMD_GETDATA = 11
Global Const GPSMSG_DATA = 12

Type DataSrvrRec
  Latitude As Double
  Longitude As Double
  Airspeed As Single
  CompassHdg As Single
  Altitude As Double
  VertSpd As Single
  Pitch As Single
  Roll As Single
  TrueHdg As Single
  Nav1Stat As Integer
  Nav1Freq As Integer
  Nav1Radial As Single
  Nav1Dist As Single
  Nav2Stat As Integer
  Nav2Freq As Integer
  Nav2Radial As Single
  Nave2Dist As Single
  ADFStat As Integer
  ADFFreq As Integer
  ADFBrg As Single
  Com1Stat As Integer
  Com1Freq As Integer
  LeftEngine As Single
  RightEngine As Single
  Gear As Single
  Flap As Single
  Elevator As Single
  Rudder As Single
  Aileron As Single
  PitchQ As Single
  PitchR As Single
  PitchKf As Single
  Pitchtau As Single
  PitchTurbulence As Single
  RollQ As Single
  RollR As Single
  RollKf As Single
  Rolltau As Single
  RollTurbulence As Single
  YawQ As Single
  YawR As Single
  YawKf As Single
  Yawtau As Single
  YawTurbulence As Single
  VSQ As Single
  VSR As Single
  VSKf As Single
  VStau As Single
  VSTurbulence As Single
  TimeStamp As Long
End Type

Type GPSRec
  LastWPIdent As String * 10
  CurWPIdent As String * 10
  CurWPType As String * 10
  CurWPLat As Double
  CurWPLong As Double
  CurRadial As Double
  CurLegLen As Double
  NextWPIdent As String * 10
  NextWPType As String * 10
  NextWPLat As Double
  NextWPLong As Double
  NextRadial As Double
  NextLegLen As Double
  XAE As Double
  XTE As Double
  GAE As Double
  AltErr As Double
  DisttoWP As Double
  TrkDistToWP As Double
  HdgtoWP As Double
  DistToEnd As Double
  Track As Double
  CGA As Double
  TAE As Double
  GrndSpeed As Double
  ETW As Double
  CDISensitivity As Single
  CDIVal As Single
  TimeStamp As Long
End Type

Type PrePositionRec
  Latitude As Double
  Longitude As Double
  Altitude As Double
  Pitch As Single
  Roll As Single
  Heading As Single
  Airspeed As Single
End Type

Type TurbulenceRec
  PitchQ As Single
  PitchR As Single
  PitchKf As Single
  Pitchtau As Single
  RollQ As Single
  RollR As Single
  RollKf As Single
  Rolltau As Single
  YawQ As Single
  YawR As Single
  YawKf As Single
  Yawtau As Single
  VSQ As Single
```

```
  VSR As Single
  VSKf As Single
  VStau As Single
  OverAllLevel As Single
End Type

Function MakeDataSrvrStr (DataSrvrData As
DataSrvrRec) As String
  Dim TempDataStr As String

  TempDataStr = Format$(DataSrvrData.Latitude,
"00.000000")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Longitude, "000.000000")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Altitude, "00000.0")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Pitch, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Roll, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.TrueHdg, "000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Airspeed, "000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.LeftEngine, "000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.RightEngine, "000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Elevator, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Aileron, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Rudder, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Flap, "000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Gear, "000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.PitchQ, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.PitchR, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.PitchKf, "+0.000;-0.000")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Pitchtau, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.PitchTurbulence, "+00.00;-
00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.RollQ, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.RollR, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.RollKf, "+0.000;-0.000")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Rolltau, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.RollTurbulence, "+00.00;-
00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.YawQ, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.YawR, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.YawKf, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.Yawtau, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.YawTurbulence, "+000.00;-
000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.VSQ, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.VSR, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.VSKf, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.VStau, "+00.00;-00.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.VSTurbulence, "+0000.00;-
0000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.VertSpd, "+00000.00;-
00000.00")
  TempDataStr = TempDataStr +
Format$(DataSrvrData.TimeStamp, "+0000000000;-
0000000000")
  MakeDataSrvrStr = TempDataStr

End Function

Function MakeGPSStr (GPSData As GPSRec) As String
  Dim TempDataStr As String

  TempDataStr = Format$(GPSData.CurWPIdent,
"@@@@@@@@@@;          ")
  TempDataStr = TempDataStr +
Format$(GPSData.CurWPType, "@@@@@@@@@@;          ")
  If GPSData.CurRadial = -1 Then
    TempDataStr = TempDataStr + "000.00"
  Else
    TempDataStr = TempDataStr +
Format$(GPSData.CurRadial, "+000.00;-000.00")
  End If
  TempDataStr = TempDataStr +
Format$(GPSData.NextWPIdent, "@@@@@@@@@@;
")
  TempDataStr = TempDataStr +
Format$(GPSData.NextWPType, "@@@@@@@@@@;
")
  If GPSData.NextRadial = -1 Then
    TempDataStr = TempDataStr + "000.00"
  Else
    TempDataStr = TempDataStr +
Format$(GPSData.NextRadial, "000.00")
  End If
  If (GPSData.XAE > 999.99) Then GPSData.XAE =
999.99
  If (GPSData.XAE < -999.99) Then GPSData.XAE =
-999.99
  TempDataStr = TempDataStr + Format$(GPSData.XAE,
"+000.00;-000.00")
  If (GPSData.XTE > 999.9999) Then GPSData.XTE =
999.9999
  If (GPSData.XTE < -999.9999) Then GPSData.XTE =
-999.9999
  TempDataStr = TempDataStr + Format$(GPSData.XTE,
"+000.0000;-000.0000")
  If (GPSData.GAE > 999.99) Then GPSData.GAE =
999.99
  If (GPSData.GAE < -999.99) Then GPSData.GAE =
-999.99
  TempDataStr = TempDataStr + Format$(GPSData.GAE,
"+000.00;-000.00")
  If (GPSData.AltErr > 99999.99) Then
GPSData.AltErr = 99999.99
  If (GPSData.AltErr < -99999.99) Then
GPSData.AltErr = -99999.99
  TempDataStr = TempDataStr +
Format$(GPSData.AltErr, "+00000.00;-00000.00")
  If (GPSData.DisttoWP > 9999.99) Then
GPSData.DisttoWP = 9999.99
```

```
  If (GPSData.DisttoWP < 0) Then GPSData.DisttoWP =
0
  TempDataStr = TempDataStr +
Format$(GPSData.DisttoWP, "0000.00")
  If (GPSData.TrkDistToWP > 9999.99) Then
GPSData.TrkDistToWP = 9999.99
  If (GPSData.TrkDistToWP < 0) Then
GPSData.TrkDistToWP = 0
  TempDataStr = TempDataStr +
Format$(GPSData.TrkDistToWP, "0000.00")
  If (GPSData.DistToEnd > 9999.99) Then
GPSData.DistToEnd = 9999.99
  If (GPSData.DistToEnd < 0) Then GPSData.DistToEnd
= 0
  TempDataStr = TempDataStr +
Format$(GPSData.DistToEnd, "0000.00")
  If (GPSData.DistToEnd > 9999.99) Then
GPSData.DistToEnd = 9999.99
  If (GPSData.DistToEnd < 0) Then GPSData.DistToEnd
= 0
  TempDataStr = TempDataStr +
Format$(GPSData.HdgtoWP, "000.00")
  TempDataStr = TempDataStr +
Format$(GPSData.Track, "000.00")
  TempDataStr = TempDataStr + Format$(GPSData.TAE,
"+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(GPSData.GrndSpeed, "000.00")
  TempDataStr = TempDataStr + Format$(GPSData.ETW,
"000.00")
  TempDataStr = TempDataStr +
Format$(GPSData.LastWPIdent, "@@@@@@@@@@;
")
  TempDataStr = TempDataStr +
Format$(GPSData.CDISensitivity, "+00.0000;-
00.0000")
  TempDataStr = TempDataStr +
Format$(GPSData.CDIVal, "+0000.00;-0000.00")
  TempDataStr = TempDataStr +
Format$(GPSData.TimeStamp, "+0000000000;-
0000000000")
  TempDataStr = TempDataStr + Format$(GPSData.CGA,
"+000.00;-000.00")
  MakeGPSStr = TempDataStr
End Function

Function MakePrePositionStr (PrePositionData As
PrePositionRec) As String
  Dim TempDataStr As String

  TempDataStr = Format$(PrePositionData.Latitude,
"00.000000")
  TempDataStr = TempDataStr +
Format$(PrePositionData.Longitude, "000.000000")
  TempDataStr = TempDataStr +
Format$(PrePositionData.Altitude, "00000.0")
  TempDataStr = TempDataStr +
Format$(PrePositionData.Pitch, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(PrePositionData.Roll, "+000.00;-000.00")
  TempDataStr = TempDataStr +
Format$(PrePositionData.Heading, "000.00")
  TempDataStr = TempDataStr +
Format$(PrePositionData.Airspeed, "000.00")
  MakePrePositionStr = TempDataStr
End Function

Function MakeTurbulenceStr (TurbulenceData As
TurbulenceRec) As String
  Dim TempDataStr As String

  TempDataStr = Format$(TurbulenceData.PitchQ,
"00.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.PitchR, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.PitchKf, "0.000")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.Pitchtau, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.RollQ, "00.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.RollR, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.RollKf, "0.000")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.Rolltau, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.YawQ, "00.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.YawR, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.YawKf, "00.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.Yawtau, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.VSQ, "00.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.VSR, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.VSKf, "0.00")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.VStau, "0.0")
  TempDataStr = TempDataStr +
Format$(TurbulenceData.OverAllLevel, "00.0")
  MakeTurbulenceStr = TempDataStr


End Function

Function NumofParams (ParamList As String) As
Integer
  Dim i As Integer
  Dim char As Integer

  If Len(ParamList) = 0 Then
    NumofParams = 0
  Else
    i = 1
    ParamList = Trim$(ParamList)
    For char = 1 To Len(ParamList)
      If (Mid$(ParamList, char, 1) = " ") And
(Mid$(ParamList, char, 2) <> "  ") Then
        i = i + 1
      End If
    Next char
    NumofParams = i
  End If
End Function

Sub ParseDataSrvrRec (DataSrvrData As DataSrvrRec,
DataSrvrStr As String)
  If (Len(RTrim$(DataSrvrStr)) >= 240) Then    '277
    DataSrvrData.Latitude = Val(Left$(DataSrvrStr,
9))
    DataSrvrData.Longitude = Val(Mid$(DataSrvrStr,
10, 10))
    DataSrvrData.Altitude = Val(Mid$(DataSrvrStr,
20, 7))
    DataSrvrData.Airspeed = Val(Mid$(DataSrvrStr,
47, 6))
    DataSrvrData.TrueHdg = Val(Mid$(DataSrvrStr,
41, 6))
```

```
    DataSrvrData.Pitch = Val(Mid$(DataSrvrStr, 27,
7))
    DataSrvrData.Roll = Val(Mid$(DataSrvrStr, 34,
7))
    DataSrvrData.LeftEngine = Val(Mid$(DataSrvrStr,
53, 6))
    DataSrvrData.RightEngine =
Val(Mid$(DataSrvrStr, 59, 6))
    DataSrvrData.Elevator = Val(Mid$(DataSrvrStr,
65, 7))
    DataSrvrData.Rudder = Val(Mid$(DataSrvrStr, 72,
7))
    DataSrvrData.Aileron = Val(Mid$(DataSrvrStr,
79, 7))
    DataSrvrData.Flap = Val(Mid$(DataSrvrStr, 86,
6))
    DataSrvrData.Gear = Val(Mid$(DataSrvrStr, 92,
6))
    DataSrvrData.PitchQ = Val(Mid$(DataSrvrStr, 98,
6))
    DataSrvrData.PitchR = Val(Mid$(DataSrvrStr,
104, 6))
    DataSrvrData.PitchKf = Val(Mid$(DataSrvrStr,
110, 6))
    DataSrvrData.Pitchtau = Val(Mid$(DataSrvrStr,
116, 6))
    DataSrvrData.PitchTurbulence =
Val(Mid$(DataSrvrStr, 122, 6))
    DataSrvrData.RollQ = Val(Mid$(DataSrvrStr, 128,
6))
    DataSrvrData.RollR = Val(Mid$(DataSrvrStr, 134,
6))
    DataSrvrData.RollKf = Val(Mid$(DataSrvrStr,
140, 6))
    DataSrvrData.Rolltau = Val(Mid$(DataSrvrStr,
146, 6))
    DataSrvrData.RollTurbulence =
Val(Mid$(DataSrvrStr, 152, 6))
    DataSrvrData.YawQ = Val(Mid$(DataSrvrStr, 158,
6))
    DataSrvrData.YawR = Val(Mid$(DataSrvrStr, 164,
6))
    DataSrvrData.YawKf = Val(Mid$(DataSrvrStr, 170,
6))
    DataSrvrData.Yawtau = Val(Mid$(DataSrvrStr,
176, 6))
    DataSrvrData.YawTurbulence =
Val(Mid$(DataSrvrStr, 182, 7))
    DataSrvrData.VSQ = Val(Mid$(DataSrvrStr, 189,
6))
    DataSrvrData.VSR = Val(Mid$(DataSrvrStr, 195,
6))
    DataSrvrData.VSKf = Val(Mid$(DataSrvrStr, 201,
6))
    DataSrvrData.VStau = Val(Mid$(DataSrvrStr, 207,
6))
    DataSrvrData.VSTurbulence =
Val(Mid$(DataSrvrStr, 213, 8))
    DataSrvrData.VertSpd = Val(Mid$(DataSrvrStr,
221, 9))
    DataSrvrData.TimeStamp = Val(Mid$(DataSrvrStr,
230, 11))
  End If
End Sub

Sub ParseGPSRec (GPSData As GPSRec, GPSStr As
String)
  If Len(RTrim$(GPSStr)) >= 156 Then
    GPSData.CurWPIdent = Mid$(GPSStr, 1, 10)
    GPSData.CurWPType = Mid$(GPSStr, 11, 10)
    GPSData.CurRadial = Val(Mid$(GPSStr, 21, 7))
    GPSData.NextWPIdent = Mid$(GPSStr, 28, 10)
    GPSData.NextWPType = Mid$(GPSStr, 38, 10)
    GPSData.NextRadial = Val(Mid$(GPSStr, 48, 6))
    GPSData.XAE = Val(Mid$(GPSStr, 54, 7))
    GPSData.XTE = Val(Mid$(GPSStr, 61, 9))
    GPSData.GAE = Val(Mid$(GPSStr, 70, 7))
    GPSData.AltErr = Val(Mid$(GPSStr, 77, 9))
    GPSData.DisttoWP = Val(Mid$(GPSStr, 86, 7))
    GPSData.TrkDistToWP = Val(Mid$(GPSStr, 93, 7))
    GPSData.DistToEnd = Val(Mid$(GPSStr, 100, 7))
    GPSData.HdgtoWP = Val(Mid$(GPSStr, 107, 6))
    GPSData.Track = Val(Mid$(GPSStr, 113, 6))
    GPSData.TAE = Val(Mid$(GPSStr, 119, 7))
    GPSData.GrndSpeed = Val(Mid$(GPSStr, 126, 6))
    GPSData.ETW = Val(Mid$(GPSStr, 132, 6))
    GPSData.LastWPIdent = Mid$(GPSStr, 138, 10)
    GPSData.CDISensitivity = Val(Mid$(GPSStr, 148,
8))
    GPSData.CDIVal = Val(Mid$(GPSStr, 156, 8))
    GPSData.TimeStamp = Val(Mid$(GPSStr, 164, 11))
    GPSData.CGA = Val(Mid$(GPSStr, 175, 7))
  End If
End Sub

Sub ParsePrePositionRec (PrePositionData As
PrePositionRec, PrePositionStr As String)
    PrePositionData.Latitude =
Val(Left$(PrePositionStr, 9))
    PrePositionData.Longitude =
Val(Mid$(PrePositionStr, 10, 10))
    PrePositionData.Altitude =
Val(Mid$(PrePositionStr, 20, 7))
    PrePositionData.Pitch =
Val(Mid$(PrePositionStr, 27, 7))
    PrePositionData.Roll = Val(Mid$(PrePositionStr,
34, 7))
    PrePositionData.Heading =
Val(Mid$(PrePositionStr, 41, 6))
    PrePositionData.Airspeed =
Val(Mid$(PrePositionStr, 47, 6))
End Sub

Sub ParseTurbulenceRec (TurbulenceData As
TurbulenceRec, TurbulenceStr As String)
  If Len(TurbulenceStr) >= 62 Then
    TurbulenceData.PitchQ =
Val(Left$(TurbulenceStr, 4))
    TurbulenceData.PitchR = Val(Mid$(TurbulenceStr,
5, 3))
    TurbulenceData.PitchKf =
Val(Mid$(TurbulenceStr, 8, 5))
    TurbulenceData.Pitchtau =
Val(Mid$(TurbulenceStr, 13, 3))
    TurbulenceData.RollQ = Val(Mid$(TurbulenceStr,
16, 4))
    TurbulenceData.RollR = Val(Mid$(TurbulenceStr,
20, 3))
    TurbulenceData.RollKf = Val(Mid$(TurbulenceStr,
23, 5))
    TurbulenceData.Rolltau =
Val(Mid$(TurbulenceStr, 28, 3))
    TurbulenceData.YawQ = Val(Mid$(TurbulenceStr,
31, 4))
    TurbulenceData.YawR = Val(Mid$(TurbulenceStr,
35, 3))
    TurbulenceData.YawKf = Val(Mid$(TurbulenceStr,
38, 4))
    TurbulenceData.Yawtau = Val(Mid$(TurbulenceStr,
42, 3))
    TurbulenceData.VSQ = Val(Mid$(TurbulenceStr,
45, 4))
    TurbulenceData.VSR = Val(Mid$(TurbulenceStr,
49, 3))
```

```
    TurbulenceData.VSKf = Val(Mid$(TurbulenceStr,
52, 4))
    TurbulenceData.VStau = Val(Mid$(TurbulenceStr,
56, 3))
    TurbulenceData.OverAllLevel =
Val(Mid$(TurbulenceStr, 59, 4))
  Else
    TurbulenceData.PitchQ = 0
    TurbulenceData.PitchR = 0
    TurbulenceData.PitchKf = 0
    TurbulenceData.Pitchtau = 0
    TurbulenceData.RollQ = 0
    TurbulenceData.RollR = 0
    TurbulenceData.RollKf = 0
    TurbulenceData.Rolltau = 0
    TurbulenceData.YawQ = 0
    TurbulenceData.YawR = 0
    TurbulenceData.YawKf = 0
    TurbulenceData.Yawtau = 0
    TurbulenceData.VSQ = 0
    TurbulenceData.VSR = 0
    TurbulenceData.VSKf = 0
    TurbulenceData.VStau = 0
    TurbulenceData.OverAllLevel = 0
  End If
End Sub

Function ReturnParam (ParamList As String, x As
Integer) As String
  Dim i As Integer
  Dim j As Integer
  Dim char As Integer

  ParamList = Trim$(ParamList)

  If Len(ParamList) = 0 Then
    ReturnParam = ""
  Else
    i = 1
    j = 1
    If x = 1 Then
      While (Mid$(ParamList, j, 1) <> " ") And j <=
Len(ParamList)
        j = j + 1
      Wend
      ReturnParam = Left$(ParamList, j - 1)
    Else
      For char = 1 To Len(ParamList)
        If (Mid$(ParamList, char, 1) = " ") And
(Mid$(ParamList, char, 2) <> "  ") Then
          i = i + 1
          If x = i Then
            j = char + 1
            While (Mid$(ParamList, j, 1) <> " ")
And j <= Len(ParamList)
              j = j + 1
            Wend
            ReturnParam = Mid$(ParamList, char + 1,
j - char - 1)
          End If
        End If
      Next char
    End If
  End If

End Function
```

DDELINKS.BAS

```
Attribute VB_Name = "ddelinks"
Option Explicit

Global LinkedtoGPS As Integer
Global LinkedtoDS As Integer
Global LinkedtoCC As Integer

Sub CheckCCLinks()
  If Not LinkedtoCC Then
    OpenCCLinks
  End If
End Sub

Sub CheckDSLinks()
  If Not LinkedtoDS Then
    OpenDSLinks
  End If
End Sub

Sub CheckGPSLinks()
  If Not LinkedtoGPS Then
    OpenGPSLinks
  End If
End Sub

Sub ClearCCLinks()
  LinkedtoCC = False
  Main.CCLinkLight.FillColor = &HFF&
' Link to get current Panel Parameters
  Main.PanelParam.LinkMode = LINK_NONE
  Main.PanelParam.LinkTimeout = 2    'Two tenths of
a second
  Main.PanelParam.LinkItem = "PanelParam"
  Main.PanelParam.LinkTopic =
SystemSettings.CtrlConPath
' Link to get Control Console Status
  Main.CntrlConStatus.LinkMode = LINK_NONE
  Main.CntrlConStatus.LinkTimeout = 2    'Two
tenths of a second
  Main.CntrlConStatus.LinkItem = "CtrlConStatus"
  Main.CntrlConStatus.LinkTopic =
SystemSettings.CtrlConPath
' Link to get Experiment Name
  Main.ExpName.LinkMode = LINK_NONE
  Main.ExpName.LinkTimeout = 2    'Two tenths of a
second
  Main.ExpName.LinkItem = "ExpName"
  Main.ExpName.LinkTopic =
SystemSettings.CtrlConPath
' Link to get Profile Name
  Main.ProfileName.LinkMode = LINK_NONE
  Main.ProfileName.LinkTimeout = 2    'Two tenths
of a second
  Main.ProfileName.LinkItem = "ProfileName"
  Main.ProfileName.LinkTopic =
SystemSettings.CtrlConPath
' Link to get Flight Plan Name
  Main.FltPlnName.LinkMode = LINK_NONE
  Main.FltPlnName.LinkTimeout = 2    'Two tenths of
a second
  Main.FltPlnName.LinkItem = "FltPlnName"
  Main.FltPlnName.LinkTopic =
SystemSettings.CtrlConPath
  Main.ConnectTimer.Enabled = True
End Sub

Sub ClearDSLinks()
```

```
  LinkedtoDS = False
  Main.DSLinkLight.FillColor = &HFF&
' Link to Pre-Position field to do total position
  Main.PrePosition.LinkMode = LINK_NONE
  Main.PrePosition.LinkTimeout = 2    'Two tenths
of a second
  Main.PrePosition.LinkItem = "PrePosition"
  Main.PrePosition.LinkTopic =
SystemSettings.DataSrvrPath
' Link to Manual Command field to do our own
commands
  Main.ManualCmd.LinkMode = LINK_NONE
  Main.ManualCmd.LinkTimeout = 2    'Two tenths of
a second
  Main.ManualCmd.LinkItem = "ManualCmd"
  Main.ManualCmd.LinkTopic =
SystemSettings.DataSrvrPath
  Main.ConnectTimer.Enabled = True
End Sub

Sub ClearGPSLinks()
  LinkedtoGPS = False
  Main.GPSLinkLight.FillColor = &HFF&
  Main.GPSData.LinkMode = LINK_NONE
  Main.GPSData.LinkTimeout = 2    'Two tenths of a
second
  Main.GPSData.LinkItem = "GPSData"
  Main.GPSData.LinkTopic = SystemSettings.GPSPath
  Main.GPSStatus.LinkMode = LINK_NONE
  Main.GPSStatus.LinkTimeout = 2  'Two tenths of a
second
  Main.GPSStatus.LinkItem = "GPSStatus"
  Main.GPSStatus.LinkTopic = SystemSettings.GPSPath
  Main.ConnectTimer.Enabled = True
End Sub

Sub OpenCCLinks()
  On Error Resume Next
  Main.PanelParam.LinkMode = LINK_AUTOMATIC
  If Err = 0 Then
    LinkedtoCC = True
    Main.ConnectTimer.Enabled = Not (LinkedtoDS And
LinkedtoGPS)
    Main.CCLinkLight.FillColor = &HFF00&
    Main.CntrlConStatus.LinkMode = LINK_AUTOMATIC
    Main.ExpName.LinkMode = LINK_AUTOMATIC
    Main.ProfileName.LinkMode = LINK_AUTOMATIC
    Main.FltPlnName.LinkMode = LINK_AUTOMATIC
  Else
    Main.PanelParam.LinkMode = LINK_NONE
  End If
  On Error GoTo 0
End Sub

Sub OpenDSLinks()
  On Error Resume Next
  Main.PrePosition.LinkMode = LINK_AUTOMATIC
  If Err = 0 Then
    LinkedtoDS = True
    Main.ConnectTimer.Enabled = Not (LinkedtoCC And
LinkedtoGPS)
    Main.DSLinkLight.FillColor = &HFF00&
    Main.ManualCmd.LinkMode = LINK_AUTOMATIC
  Else
    Main.PrePosition.LinkMode = LINK_NONE
  End If
  On Error GoTo 0
End Sub

Sub OpenGPSLinks()
  On Error Resume Next
  Main.GPSData.LinkMode = LINK_AUTOMATIC
  If Err = 0 Then
    LinkedtoGPS = True
    Main.ConnectTimer.Enabled = Not (LinkedtoCC And
LinkedtoDS)
    Main.GPSLinkLight.FillColor = &HFF00&
    Main.GPSStatus.LinkMode = LINK_AUTOMATIC
  Else
    Main.GPSData.LinkMode = LINK_NONE
    Main.GPSStatus.LinkMode = LINK_NONE
  End If
  On Error GoTo 0
End Sub
```

EXPT2.BAS

```
Attribute VB_Name = "EXPT21"
Option Explicit

#If Win16 Then
    Declare Function GetTickCount Lib "user" () As
Long
    Declare Function FloodFill Lib "gdi" (ByVal hdc
As Integer, ByVal x As Integer, ByVal y As Integer,
ByVal crColor As Long) As Integer
    Declare Function GetProfileString Lib "Kernel"
(ByVal lpAppName As String, ByVal lpKeyName As
String, ByVal lpDefault As String, ByVal
lpReturnedString As String, ByVal nSize As Integer)
As Integer
    Declare Function GetProfileInt Lib "Kernel"
(ByVal lpAppName As String, ByVal lpKeyName As
String, ByVal nDefault As Integer) As Integer
    Declare Function WriteProfileString Lib
"Kernel" (ByVal lpApplicationName As String, ByVal
lpKeyName As String, ByVal lpString As String) As
Integer
#ElseIf Win32 Then
    Declare Function GetProfileString Lib
"kernel32" Alias "GetProfileStringA" (ByVal
lpAppName As String, ByVal lpKeyName As String,
ByVal lpDefault As String, ByVal lpReturnedString
As String, ByVal nSize As Long) As Long
    Declare Function GetProfileInt Lib "kernel32"
Alias "GetProfileIntA" (ByVal lpAppName As String,
ByVal lpKeyName As String, ByVal nDefault As Long)
As Long
    Declare Function WriteProfileString Lib
"kernel32" Alias "WriteProfileStringA" (ByVal
lpszSection As String, ByVal lpszKeyName As String,
ByVal lpszString As String) As Long
    Declare Function FloodFill Lib "gdi32" (ByVal
hdc As Long, ByVal x As Long, ByVal y As Long,
ByVal crColor As Long) As Long
    Declare Function GetTickCount Lib "kernel32" ()
As Long
#End If


Global DisplayType As Integer
Global VHold As Single
Global TAEHold As Single
Global GAEHold As Single
Global TimeHold As Single

Global Const VectorType = 1
```

```
Global Const PredType = 2
Global Const TriangleType = 3
Global Const EHSIType = 4
Global Const XTEType = 5
Global Const TextForeColor = &HFFFF& ' Yellow
Global Const TextBackColor = &H0&    ' Black
Global Const Pi = 3.14159
Global Const OnForeColor = &H0&      ' Black
Global Const OnBackColor = &HFFFFFF  ' White
Global Const OffForeColor = &H808080 ' Light Grey
Global Const OffBackColor = &H808080 ' Mid Grey

Function Arcsin(x As Double) As Double
  Arcsin = Atn(x / Sqr(-x * x + 1))
End Function

Sub ConnectCtrlCon()
End Sub

Sub ConnectDataSrvr()
  Dim RetryCntr As Integer

'  Main.DataSrvrData.LinkItem = "DataSrvrData"
'  Main.DataSrvrData.LinkTopic =
SystemSettings.DataSrvrPath
'  Main.DataSrvrStatus.LinkItem = "DataSrvrStatus"
'  Main.DataSrvrStatus.LinkTopic =
SystemSettings.DataSrvrPath

'  LinkedtoDataSrvr = False
'  RetryCntr = 0
'  Do
'    On Error GoTo 0
'    On Error Resume Next
'    Main.DataSrvrData.LinkMode = LINK_AUTOMATIC
'    Main.DataSrvrStatus.LinkMode = LINK_AUTOMATIC
'    RetryCntr = RetryCntr + 1
'    DoEvents
'  Loop Until (Err = 0) Or (RetryCntr >=
MaxDDERetry)
'  DoEvents
'  If Err <> 0 Then
'    i = MsgBox("Cannot connect to Data Server.
Check Path and status", MB_OK + MB_ICONSTOP,
"Control Console")
'    LinkedtoDataSrvr = False
'    Main.DataSrvrData.LinkMode = LINK_NONE
'    Main.DataSrvrStatus.LinkMode = LINK_NONE
'  Else
''    Main.DataSrvrData.LinkMode = LINK_MANUAL
'    DoEvents
'    LinkedtoDataSrvr = True
''    Main.DataSrvrInd.FillColor = &HFF00&
'  End If
'  On Error GoTo 0
End Sub

Sub ConnectGPS()
'  Dim RetryCntr As Integer
'  Dim i As Integer
'
'  Main.GPSData.LinkItem = "GPSData"
'  Main.GPSData.LinkTopic = SystemSettings.GPSPath
'
'  LinkedtoGPS = False
'  RetryCntr = 0
'  Do
'    On Error GoTo 0
'    On Error Resume Next
'    Main.GPSData.LinkMode = LINK_AUTOMATIC
'    RetryCntr = RetryCntr + 1
'    DoEvents
'  Loop Until (Err = 0) Or (RetryCntr >=
MaxDDERetry)
'  DoEvents
'  If Err <> 0 Then
'    i = MsgBox("Cannot connect to GPS Machine.
Check Path and status", MB_OK + MB_ICONSTOP, "Two
CDI v2")
'    LinkedtoGPS = False
'    Main.GPSData.LinkMode = LINK_NONE
'  Else
'    Main.DataSrvrData.LinkMode = LINK_MANUAL
'    DoEvents
'    LinkedtoGPS = True
'    Main.DataSrvrInd.FillColor = &HFF00&
'  End If
'  On Error GoTo 0
End Sub

Sub ConnectInstSrvr()
End Sub

Sub ConnectLight()

End Sub

Sub ConnectPanel()

End Sub

Sub ConnectWind()

End Sub

Function DegtoRad(Degrees As Double) As Double
  DegtoRad = Degrees * (Pi / 180)
End Function

Function FindListIndex(List As ComboBox, SearchStr
As String) As Integer
  Dim i As Integer

  FindListIndex = -1
  If List.ListCount > 0 Then
    i = 0
    While (i < List.ListCount) And
(Trim$(List.List(i)) <> Trim$(SearchStr))
      i = i + 1
    Wend
    If i < List.ListCount Then
      FindListIndex = i
    End If
  End If
End Function

Function FindListIndexValue(List As ComboBox,
DataValue As Long) As Integer
  Dim i As Integer

  FindListIndexValue = -1
  If List.ListCount > 0 Then
    i = 0
    While (i < List.ListCount) And
(List.ItemData(i) <> DataValue)
      i = i + 1
    Wend
    If i < List.ListCount Then
      FindListIndexValue = i
    End If
  End If

End Function
```

```
Function LatFormat$(Latitude As Double)
  Dim Deg As Long
  Dim Min As Double

  Min = Latitude * 60
  Deg = Int(Min) - (Int(Min) Mod 60)
  Deg = Deg / 60
  Min = Min - (Deg * 60)
  LatFormat$ = Format$(Deg, "00") + " " +
Format$(Min, "00.00")
End Function

Function LatVal(Latitude As String) As Double
  LatVal = ((Val(Left$(Latitude, 2)) * 60) +
Val(Right$(Latitude, 5))) / 60
End Function

Function LongFormat$(Longitude As Double)
  Dim Deg As Long
  Dim Min As Double

  Min = Longitude * 60
  Deg = Int(Min) - (Int(Min) Mod 60)
  Deg = Deg / 60
  Min = Min - (Deg * 60)
  LongFormat$ = Format$(Deg, "000") + " " +
Format$(Min, "00.00")
End Function

Function LongVal(Longitude As String) As Double
  LongVal = ((Val(Left$(Longitude, 3)) * 60) +
Val(Right$(Longitude, 5))) / 60
End Function

Function NullCheck(DataVal As Variant) As Variant
  If IsNull(DataVal) Then
    NullCheck = ""
  Else
    NullCheck = DataVal
  End If
End Function

Function RadtoDeg(Radians As Double) As Double
  RadtoDeg = Radians * (180 / Pi)
End Function

Function Reciprocal(Angle As Double) As Double
  If Angle < 180 Then
    Reciprocal = 180 + Angle
  Else
    Reciprocal = Angle - 180
  End If
End Function

Sub RefreshList(UpdateList As ComboBox, UpdateTbl
As Dynaset, TextFieldName As String, IDFieldName As
String)
  If Not (UpdateTbl.EOF And UpdateTbl.BOF) Then
    UpdateTbl.MoveFirst
  End If
  UpdateList.Clear
  While Not UpdateTbl.EOF
    UpdateList.AddItem
NullCheck(UpdateTbl(TextFieldName))
    UpdateList.ItemData(UpdateList.NewIndex) =
Val(UpdateTbl(IDFieldName))
    UpdateTbl.MoveNext
  Wend
End Sub

Function TranslateX(XCoord As Single, YCoord As
Single, BEARING As Double)
  Dim Quadrant As Integer
  Dim DISTANCE As Double
  Dim CurBearing As Single
  Dim NewBearing As Double

  If (XCoord <= 0) And (YCoord <= 0) Then
    Quadrant = 0
  ElseIf (XCoord <= 0) And (YCoord > 0) Then
    Quadrant = 1
  ElseIf (XCoord > 0) And (YCoord > 0) Then
    Quadrant = 2
  Else
    Quadrant = 3
  End If
  DISTANCE = Sqr(XCoord ^ 2 + YCoord ^ 2)
  If (Abs(DISTANCE) > 0) And ((Abs(XCoord) /
Abs(DISTANCE)) < 1) Then
    CurBearing = RadtoDeg(Arcsin(Abs(XCoord) /
Abs(DISTANCE)))
  ElseIf ((Abs(XCoord) / Abs(DISTANCE)) = 1) Then
    CurBearing = 90
  Else
    CurBearing = 0
  End If
  Select Case Quadrant
    Case 1
      CurBearing = 180 - CurBearing
    Case 2
      CurBearing = CurBearing + 180
    Case 3
      CurBearing = 360 - CurBearing
  End Select
  NewBearing = CurBearing - Reciprocal(BEARING)
  If NewBearing < 0 Then
    NewBearing = 360 + NewBearing
  End If
  'TranslateX = XOffset(NewBearing, Distance)
End Function

Function TranslateY(XCoord As Single, YCoord As
Single, BEARING As Double)
  Dim Quadrant As Integer
  Dim DISTANCE As Double
  Dim CurBearing As Single
  Dim NewBearing As Double

  If (XCoord <= 0) And (YCoord <= 0) Then
    Quadrant = 0
  ElseIf (XCoord <= 0) And (YCoord > 0) Then
    Quadrant = 1
  ElseIf (XCoord > 0) And (YCoord > 0) Then
    Quadrant = 2
  Else
    Quadrant = 3
  End If
  DISTANCE = Sqr(XCoord ^ 2 + YCoord ^ 2)
  If (Abs(DISTANCE) > 0) And ((Abs(XCoord) /
Abs(DISTANCE)) < 1) Then
    CurBearing = RadtoDeg(Arcsin(Abs(XCoord) /
Abs(DISTANCE)))
  ElseIf ((Abs(XCoord) / Abs(DISTANCE)) = 1) Then
    CurBearing = 90
  Else
    CurBearing = 0
  End If
  Select Case Quadrant
    Case 1
      CurBearing = 180 - CurBearing
    Case 2
      CurBearing = CurBearing + 180
    Case 3
      CurBearing = 360 - CurBearing
```

```
  End Select
  NewBearing = CurBearing - Reciprocal(BEARING)
  If NewBearing < 0 Then
    NewBearing = 360 + NewBearing
  End If
  'TranslateY = YOffset(NewBearing, Distance)

End Function
```

EXPT2.FRM

```
VERSION 4.00
Begin VB.Form Main
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   BorderStyle     =   0  'None
   ClientHeight    =   6915
   ClientLeft      =   1140
   ClientTop       =   1425
   ClientWidth     =   9660
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   13.5
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Height          =   7320
   KeyPreview      =   -1  'True
   Left            =   1080
   LinkTopic       =   "Form1"
   ScaleHeight     =   461
   ScaleMode       =   3  'Pixel
   ScaleWidth      =   644
   Top             =   1080
   Width           =   9780
   WindowState     =   2  'Maximized
   Begin Threed.SSPanel GoAround
      Height          =   435
      Left            =   720
      TabIndex        =   108
      Top             =   6420
      Width           =   1350
      _Version        =   65536
      _ExtentX        =   2381
      _ExtentY        =   767
      _StockProps     =   15
      Caption         =   "Go Around"
      ForeColor       =   8421504
      BackColor       =   8421504
      BeginProperty Font {0BE35203-8F91-11CE-9DE3-
00AA004BB851}
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   9.75
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      BevelWidth      =   2
      BorderWidth     =   0
      BevelInner      =   1
      Font3D          =   3
      Enabled         =   0   'False
   End
   Begin VB.PictureBox WPArrowI
      Appearance      =   0  'Flat
      BackColor       =   &H0000FFFF&
      BorderStyle     =   0  'None
      ForeColor       =   &H80000008&
      Height          =   360
      Left            =   1620
      Picture         =   "EXPT2.frx":0000
      ScaleHeight     =   360
      ScaleWidth      =   450
      TabIndex        =   107
      Top             =   900
      Visible         =   0   'False
      Width           =   450
   End
   Begin VB.PictureBox WPArrow
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      BorderStyle     =   0  'None
      ForeColor       =   &H80000008&
      Height          =   360
      Left            =   1620
      Picture         =   "EXPT2.frx":0202
      ScaleHeight     =   360
      ScaleWidth      =   450
      TabIndex        =   106
      Top             =   900
      Width           =   450
   End
   Begin Threed.SSPanel TurnLight
      Height          =   435
      Left            =   4080
      TabIndex        =   64
      Top             =   3420
      Width           =   1350
      _Version        =   65536
      _ExtentX        =   2381
      _ExtentY        =   767
      _StockProps     =   15
      Caption         =   "TURN"
      ForeColor       =   8421504
      BackColor       =   8421504
      BevelWidth      =   2
      BorderWidth     =   0
      BevelInner      =   1
      Font3D          =   3
   End
   Begin Threed.SSPanel ActLight
      Height          =   435
      Left            =   2400
      TabIndex        =   63
      Top             =   3420
      Width           =   1350
      _Version        =   65536
      _ExtentX        =   2381
      _ExtentY        =   767
      _StockProps     =   15
      Caption         =   "ACTV"
      ForeColor       =   8421504
      BackColor       =   8421504
      BevelWidth      =   2
      BorderWidth     =   0
      BevelInner      =   1
      Font3D          =   3
      Enabled         =   0   'False
   End
   Begin Threed.SSPanel ArmLight
      Height          =   435
      Left            =   720
      TabIndex        =   62
```

```
      Top             =   3420
      Width           =   1350
      _Version        =   65536
      _ExtentX        =   2381
      _ExtentY        =   767
      _StockProps     =   15
      Caption         =   "ARMD"
      ForeColor       =   8421504
      BackColor       =   8421504
      BevelWidth      =   2
      BorderWidth     =   0
      BevelInner      =   1
      Font3D          =   3
      Enabled         =   0   'False
   End
   Begin Threed.SSPanel AnalogControls
      Height          =   2445
      Left            =   6450
      TabIndex        =   44
      Top             =   1950
      Visible         =   0   'False
      Width           =   2295
      _Version        =   65536
      _ExtentX        =   4048
      _ExtentY        =   4313
      _StockProps     =   15
      Caption         =   "Analog Display"
      ForeColor       =   -2147483640
      BackColor       =   -2147483633
      BeginProperty Font {0BE35203-8F91-11CE-9DE3-
00AA004BB851}
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      BevelWidth      =   2
      BevelInner      =   1
      Alignment       =   6
      Begin Threed.SSFrame Frame3D1
         Height          =   795
         Left            =   180
         TabIndex        =   45
         Top             =   360
         Width           =   1935
         _Version        =   65536
         _ExtentX        =   3413
         _ExtentY        =   1402
         _StockProps     =   14
         Caption         =   "Altitude"
         BeginProperty Font {0BE35203-8F91-11CE-
9DE3-00AA004BB851}
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Begin VB.TextBox AltSensitivity
            Appearance      =   0   'Flat
            BeginProperty Font
               name            =   "MS Sans Serif"
               charset         =   0
               weight          =   700
               size            =   8.25
               underline       =   0   'False
               italic          =   0   'False
               strikethrough   =   0   'False
            EndProperty
            Height          =   285
            Left            =   930
            TabIndex        =   46
            Text            =   "500"
            Top             =   300
            Width           =   495
         End
         Begin Spin.SpinButton AltFSSpin
            Height          =   495
            Left            =   1530
            TabIndex        =   73
            Top             =   180
            Width           =   345
            _Version        =   65536
            _ExtentX        =   609
            _ExtentY        =   873
            _StockProps     =   73
            ForeColor       =   -2147483640
            BackColor       =   -2147483643
            TdThickness     =   2
         End
         Begin VB.Label AltSensLabel
            Alignment       =   2  'Center
            Appearance      =   0  'Flat
            BackColor       =   &H00C0C0C0&
            Caption         =   "Full Scale"
            BeginProperty Font
               name            =   "MS Sans Serif"
               charset         =   0
               weight          =   700
               size            =   8.25
               underline       =   0   'False
               italic          =   0   'False
               strikethrough   =   0   'False
            EndProperty
            ForeColor       =   &H80000008&
            Height          =   375
            Left            =   180
            TabIndex        =   47
            Top             =   270
            Width           =   615
         End
      End
      Begin Threed.SSFrame PredictorFrame
         Height          =   1125
         Left            =   180
         TabIndex        =   48
         Top             =   1170
         Width           =   1935
         _Version        =   65536
         _ExtentX        =   3413
         _ExtentY        =   1984
         _StockProps     =   14
         Caption         =   "Predictor"
         BeginProperty Font {0BE35203-8F91-11CE-
9DE3-00AA004BB851}
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Begin VB.TextBox PredTime
            Appearance      =   0  'Flat
            BeginProperty Font
               name            =   "MS Sans Serif"
               charset         =   0
               weight          =   700
```

```
         size            =   8.25
         underline       =   0    'False
         italic          =   0    'False
         strikethrough   =   0    'False
      EndProperty
      Height          =   285
      Left            =   990
      TabIndex        =   54
      Text            =   "15"
      Top             =   330
      Width           =   495
   End
   Begin Threed.SSCheck PredOrder
      Height          =   255
      Left            =   240
      TabIndex        =   53
      Top             =   810
      Width           =   1515
      _Version        =   65536
      _ExtentX        =   2672
      _ExtentY        =   450
      _StockProps     =   78
      Caption         =   "Second Order"
      BeginProperty Font {0BE35203-8F91-11CE-
9DE3-00AA004BB851}
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0    'False
         italic          =   0    'False
         strikethrough   =   0    'False
      EndProperty
   End
   Begin Threed.SSRibbon AddPredictor
      Height          =   492
      Left            =   360
      TabIndex        =   74
      Top             =   240
      Width           =   492
      _Version        =   65536
      _ExtentX        =   873
      _ExtentY        =   873
      _StockProps     =   65
      BackColor       =   -2147483633
   End
   Begin Spin.SpinButton PredTimeSpin
      Height          =   495
      Left            =   1530
      TabIndex        =   75
      Top             =   240
      Width           =   345
      _Version        =   65536
      _ExtentX        =   609
      _ExtentY        =   873
      _StockProps     =   73
      ForeColor       =   -2147483640
      BackColor       =   -2147483643
      TdThickness     =   2
   End
   Begin VB.Label TimeLabel
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "Time"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0    'False
         italic          =   0    'False
            strikethrough   =   0    'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   255
         Left            =   510
         TabIndex        =   55
         Top             =   1260
         Width           =   495
      End
   End
End
Begin Threed.SSPanel DigitalControls
   Height          =   1935
   Left            =   0
   TabIndex        =   27
   Top             =   4770
   Visible         =   0   'False
   Width           =   6435
   _Version        =   65536
   _ExtentX        =   11351
   _ExtentY        =   3413
   _StockProps     =   15
   Caption         =   "Digital Displays"
   ForeColor       =   -2147483640
   BackColor       =   -2147483633
   BeginProperty Font {0BE35203-8F91-11CE-9DE3-
00AA004BB851}
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0    'False
      italic          =   0    'False
      strikethrough   =   0    'False
   EndProperty
   BevelWidth      =   2
   BevelInner      =   1
   Alignment       =   6
   Begin VB.PictureBox ShowFromWPControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0    'False
         italic          =   0    'False
         strikethrough   =   0    'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   900
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   51
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowFromWP
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   76
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
```

```
      Begin Threed.SSRibbon ShowFromWP
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   77
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         Value           =   -1  'True
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowFromWPLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "From WP"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   52
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowToWPControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   1440
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   49
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowToWP
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   78
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowToWP
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   79
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         Value           =   -1  'True
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowToWPLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "To WP"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   50
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowDTKControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   1980
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   42
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowDTK
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   80
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowDTK
         Height          =   492
```

```
         Index           =   0
         Left            =   0
         TabIndex        =   81
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         Value           =   -1  'True
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowDTKLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "DTK"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   195
         Left            =   0
         TabIndex        =   43
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowTRKControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   5220
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   40
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowTRK
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   82
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowTRK
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   83
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowTRKLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "TRK"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   41
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowBRGControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   5760
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   38
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowBRG
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   84
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowBRG
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   85
```

```
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowBRGLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "BRG"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   39
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowGSControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   4680
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   32
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowGS
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   86
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowGS
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   87
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         Value           =   -1  'True
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowGSLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "GS"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   33
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowETEControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   4140
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   31
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowETE
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   88
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowETE
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   89
         Top             =   420
         Width           =   492
         _Version        =   65536
```

```
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowETELabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "ETE"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   34
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowDTWControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   3600
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   30
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowDTW
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   90
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         Value           =   -1  'True
      End
      Begin Threed.SSRibbon ShowDTW
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   91
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowDTWLabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "DTW"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   35
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowTAEControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   3060
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   29
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowTAE
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   92
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowTAE
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   93
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
```

```
         BackColor       =   -2147483633
         Value           =   -1  'True
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowTAELabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "TAE"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   36
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.PictureBox ShowXTEControl
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   1395
      Left            =   2520
      ScaleHeight     =   1395
      ScaleWidth      =   495
      TabIndex        =   28
      Top             =   360
      Width           =   495
      Begin Threed.SSRibbon ShowXTE
         Height          =   492
         Index           =   1
         Left            =   0
         TabIndex        =   94
         Top             =   900
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
      End
      Begin Threed.SSRibbon ShowXTE
         Height          =   492
         Index           =   0
         Left            =   0
         TabIndex        =   95
         Top             =   420
         Width           =   492
         _Version        =   65536
         _ExtentX        =   873
         _ExtentY        =   873
         _StockProps     =   65
         BackColor       =   -2147483633
         Value           =   -1  'True
         GroupAllowAllUp =   -1  'True
      End
      Begin VB.Label ShowXTELabel
         Alignment       =   2  'Center
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "XTE"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   375
         Left            =   0
         TabIndex        =   37
         Top             =   0
         Width           =   495
      End
   End
   Begin VB.Label ShowNormalLabel
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H00FFFFFF&
      Caption         =   "Show Normal"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   375
      Left            =   120
      TabIndex        =   57
      Top             =   840
      Width           =   735
   End
   Begin VB.Label ShowReverseLabel
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "Show Reverse "
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00FFFFFF&
      Height          =   375
      Left            =   120
      TabIndex        =   56
      Top             =   1320
      Width           =   735
   End
End
Begin Threed.SSPanel TestControls
   Height          =   2295
   Left            =   6450
   TabIndex        =   66
```

```
      Top             =   4410
      Visible         =   0   'False
      Width           =   2295
      _Version        =   65536
      _ExtentX        =   4048
      _ExtentY        =   4048
      _StockProps     =   15
      ForeColor       =   -2147483640
      BackColor       =   -2147483633
      BeginProperty Font {0BE35203-8F91-11CE-9DE3-
00AA004BB851}
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      BevelWidth      =   2
      BevelInner      =   1
      Alignment       =   6
      Begin VB.TextBox TestGS
         Appearance      =   0  'Flat
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Height          =   288
         Left            =   660
         TabIndex        =   72
         Text            =   "120"
         Top             =   1380
         Width           =   1212
      End
      Begin VB.TextBox TestTAE
         Appearance      =   0  'Flat
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Height          =   288
         Left            =   660
         TabIndex        =   71
         Text            =   "0"
         Top             =   480
         Width           =   1212
      End
      Begin VB.TextBox TestXTE
         Appearance      =   0  'Flat
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Height          =   288
         Left            =   660
         TabIndex        =   70
```

```
         Text            =   "0"
         Top             =   180
         Width           =   1212
      End
      Begin VB.CommandButton Test
         Appearance      =   0  'Flat
         BackColor       =   &H80000005&
         Caption         =   "Test"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Height          =   495
         Left            =   840
         TabIndex        =   69
         Top             =   1680
         Width           =   915
      End
      Begin VB.TextBox TestGAE
         Appearance      =   0  'Flat
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Height          =   288
         Left            =   660
         TabIndex        =   68
         Text            =   "0"
         Top             =   1080
         Width           =   1212
      End
      Begin VB.TextBox TestALE
         Appearance      =   0  'Flat
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         Height          =   288
         Left            =   660
         TabIndex        =   67
         Text            =   "0"
         Top             =   780
         Width           =   1212
      End
      Begin VB.Label TestGNSLabel
         Alignment       =   1  'Right Justify
         Appearance      =   0  'Flat
         BackColor       =   &H80000005&
         Caption         =   "GS"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
```

```
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   195
   Left            =   180
   TabIndex        =   7
   Top             =   1440
   Width           =   435
End
Begin VB.Label TestTAELabel
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "TAE"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   195
   Left            =   180
   TabIndex        =   8
   Top             =   540
   Width           =   435
End
Begin VB.Label TestXTELabel
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "XTE"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   195
   Left            =   180
   TabIndex        =   9
   Top             =   240
   Width           =   435
End
Begin VB.Label TestGAELabel
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "GAE"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   195
   Left            =   180
   TabIndex        =   10
   Top             =   1140
   Width           =   435
End
Begin VB.Label TestALELabel
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "ALE"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   195
   Left            =   180
   TabIndex        =   26
   Top             =   840
   Width           =   435
  End
End
Begin VB.Timer FlashTimer
   Enabled         =   0   'False
   Interval        =   250
   Left            =   6030
   Top             =   4320
End
Begin VB.CommandButton SysSetCommand
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "S&ys Settings"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   Height          =   420
   Left            =   7740
   TabIndex        =   25
   Top             =   840
   Visible         =   0   'False
   Width           =   1425
End
Begin VB.CommandButton QuitButton
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "&Quit"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   Height          =   420
   Left            =   7740
   TabIndex        =   4
   Top             =   0
   Visible         =   0   'False
   Width           =   1425
End
Begin VB.CommandButton SetupButton
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "Display &Setup"
   BeginProperty Font
```

```
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      Height          =   420
      Left            =   7740
      TabIndex        =   1
      Top             =   420
      Visible         =   0   'False
      Width           =   1425
   End
   Begin VB.CommandButton TypeButton
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "Display &Type"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      Height          =   420
      Left            =   6300
      TabIndex        =   5
      Top             =   0
      Visible         =   0   'False
      Width           =   1425
   End
   Begin VB.Timer ConnectTimer
      Enabled         =   0   'False
      Interval        =   1000
      Left            =   6030
      Top             =   3690
   End
   Begin VB.CommandButton ReconnectButton
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "&Reconnect"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      Height          =   420
      Left            =   6300
      TabIndex        =   0
      Top             =   420
      Visible         =   0   'False
      Width           =   1425
   End
   Begin VB.Frame DDEFrame
      Caption         =   "DDE Links"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      Height          =   1065
      Left            =   0
      TabIndex        =   96
      Top             =   3690
      Visible         =   0   'False
      Width           =   6045
      Begin VB.Label CntrlConStatus
         Appearance      =   0  'Flat
         AutoSize        =   -1  'True
         BackColor       =   &H80000005&
         Caption         =   "CntrlConStatus"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   195
         Left            =   60
         TabIndex        =   105
         Top             =   600
         Width           =   1230
      End
      Begin VB.Label ManualCmd
         Appearance      =   0  'Flat
         AutoSize        =   -1  'True
         BackColor       =   &H80000005&
         Caption         =   "ManualCmd"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   195
         Left            =   690
         TabIndex        =   104
         Top             =   810
         Width           =   990
      End
      Begin VB.Label PanelParam
         Appearance      =   0  'Flat
         AutoSize        =   -1  'True
         BackColor       =   &H80000005&
         Caption         =   "PanelParam"
         BeginProperty Font
            name            =   "MS Sans Serif"
            charset         =   0
            weight          =   700
            size            =   8.25
            underline       =   0   'False
            italic          =   0   'False
            strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H80000008&
         Height          =   195
         Left            =   1320
         LinkItem        =   "PanelParam"
         TabIndex        =   103
         Top             =   600
         Width           =   1995
      End
      Begin VB.Label GPSStatus
         Appearance      =   0  'Flat
```

```
      AutoSize         =   -1  'True
      BackColor        =   &H80000005&
      Caption          =   "GPSStatus"
      BeginProperty Font
         name             =   "MS Sans Serif"
         charset          =   0
         weight           =   700
         size             =   8.25
         underline        =   0   'False
         italic           =   0   'False
         strikethrough    =   0   'False
      EndProperty
      ForeColor        =   &H80000008&
      Height           =   195
      Left             =   3330
      TabIndex         =   102
      Top              =   600
      Width            =   2655
   End
   Begin VB.Label PrePosition
      Appearance       =   0  'Flat
      AutoSize         =   -1  'True
      BackColor        =   &H80000005&
      Caption          =   "PrePosition"
      BeginProperty Font
         name             =   "MS Sans Serif"
         charset          =   0
         weight           =   700
         size             =   8.25
         underline        =   0   'False
         italic           =   0   'False
         strikethrough    =   0   'False
      EndProperty
      ForeColor        =   &H80000008&
      Height           =   195
      Left             =   60
      TabIndex         =   101
      Top              =   810
      Width            =   615
   End
   Begin VB.Label GPSData
      Appearance       =   0  'Flat
      AutoSize         =   -1  'True
      BackColor        =   &H80000005&
      Caption          =   "GPSData"
      BeginProperty Font
         name             =   "MS Sans Serif"
         charset          =   0
         weight           =   700
         size             =   8.25
         underline        =   0   'False
         italic           =   0   'False
         strikethrough    =   0   'False
      EndProperty
      ForeColor        =   &H80000008&
      Height           =   195
      Left             =   60
      TabIndex         =   100
      Top              =   180
      Width            =   5925
   End
   Begin VB.Label ExpName
      Appearance       =   0  'Flat
      AutoSize         =   -1  'True
      BackColor        =   &H00C0C0C0&
      Caption          =   "ExpName"
      BeginProperty Font
         name             =   "MS Sans Serif"
         charset          =   0
         weight           =   700
         size             =   8.25
         underline        =   0   'False
         italic           =   0   'False
         strikethrough    =   0   'False
      EndProperty
      ForeColor        =   &H80000008&
      Height           =   195
      Left             =   60
      LinkItem         =   "AllData"
      LinkTopic        =   
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex         =   99
      Top              =   390
      Width            =   1965
   End
   Begin VB.Label ProfileName
      Appearance       =   0  'Flat
      AutoSize         =   -1  'True
      BackColor        =   &H00C0C0C0&
      Caption          =   "ProfileName"
      BeginProperty Font
         name             =   "MS Sans Serif"
         charset          =   0
         weight           =   700
         size             =   8.25
         underline        =   0   'False
         italic           =   0   'False
         strikethrough    =   0   'False
      EndProperty
      ForeColor        =   &H80000008&
      Height           =   195
      Left             =   2040
      LinkItem         =   "AllData"
      LinkTopic        =   
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex         =   98
      Top              =   390
      Width            =   1965
   End
   Begin VB.Label FltPlnName
      Appearance       =   0  'Flat
      AutoSize         =   -1  'True
      BackColor        =   &H00C0C0C0&
      Caption          =   "FltPlnName"
      BeginProperty Font
         name             =   "MS Sans Serif"
         charset          =   0
         weight           =   700
         size             =   8.25
         underline        =   0   'False
         italic           =   0   'False
         strikethrough    =   0   'False
      EndProperty
      ForeColor        =   &H80000008&
      Height           =   195
      Left             =   4020
      LinkItem         =   "AllData"
      LinkTopic        =   
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex         =   97
      Top              =   390
      Width            =   1965
   End
End
Begin VB.PictureBox CDIScale
   Appearance       =   0  'Flat
   AutoRedraw       =   -1  'True
   BackColor        =   &H00000000&
   ClipControls     =   0   'False
   DrawWidth        =   2
   FillColor        =   &H0000FFFF&
   BeginProperty Font
      name             =   "MS Sans Serif"
      charset          =   0
```

```
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   3240
      Left            =   3660
      ScaleHeight     =   214
      ScaleMode       =   3  'Pixel
      ScaleWidth      =   214
      TabIndex        =   65
      Top             =   30
      Width           =   3240
   End
   Begin VB.Label TAEDirection
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "<"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Index           =   1
      Left            =   1050
      TabIndex        =   109
      Top             =   1680
      Width           =   165
   End
   Begin Threed.SSCheck StepIsOn
      Height          =   255
      Left            =   6960
      TabIndex        =   58
      Top             =   900
      Visible         =   0   'False
      Width           =   255
      _Version        =   65536
      _ExtentX        =   450
      _ExtentY        =   450
      _StockProps     =   78
      BeginProperty Font {0BE35203-8F91-11CE-9DE3-
00AA004BB851}
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
   End
   Begin VB.Label TAEDirection
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   ">"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Index           =   0
      Left            =   1590
      TabIndex        =   2
      Top             =   1680
      Width           =   195
   End
   Begin VB.Label TimeToWP
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "00:00"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   990
      TabIndex        =   6
      Top             =   2340
      Visible         =   0   'False
      Width           =   945
   End
   Begin VB.Label GrndSpeed
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "000"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   3000
      TabIndex        =   11
      Top             =   1680
      Width           =   570
   End
   Begin VB.Label DesiredTrack
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "000"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   3000
      TabIndex        =   12
      Top             =   1350
      Width           =   570
```

```
End
Begin VB.Label CurTrack
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "000"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   3000
   TabIndex        =   13
   Top             =   2340
   Visible         =   0   'False
   Width           =   570
End
Begin VB.Label CurBearing
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "000"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   3000
   TabIndex        =   14
   Top             =   2010
   Visible         =   0   'False
   Width           =   570
End
Begin VB.Label DTKLabel
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "DTK"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   2340
   TabIndex        =   15
   Top             =   1350
   Width           =   675
End
Begin VB.Label TRKLabel
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "TRK"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   2340
   TabIndex        =   16
   Top             =   2340
   Visible         =   0   'False
   Width           =   675
End
Begin VB.Label BRGLabel
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "BRG"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   2340
   TabIndex        =   17
   Top             =   2010
   Visible         =   0   'False
   Width           =   675
End
Begin VB.Label GSLabel
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "GS"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   2340
   TabIndex        =   18
   Top             =   1680
   Width           =   675
End
Begin VB.Label ETELabel
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   Caption         =   "ETE"
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   0
      weight          =   700
      size            =   12
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H0000FFFF&
   Height          =   300
   Left            =   330
   TabIndex        =   19
```

```
      Top             =   2340
      Visible         =   0   'False
      Width           =   675
   End
   Begin VB.Label DistLabel
      Appearance      =   0  'Flat
      BackColor       =   &H0000FFFF&
      Caption         =   "DTW"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   300
      Left            =   330
      TabIndex        =   20
      Top             =   2010
      Width           =   675
   End
   Begin VB.Label DistToWP
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H0000FFFF&
      Caption         =   "000.0"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   300
      Left            =   990
      TabIndex        =   21
      Top             =   2010
      Width           =   795
   End
   Begin VB.Label XTEVal
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "00.0"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   990
      TabIndex        =   22
      Top             =   1350
      Width           =   795
   End
   Begin VB.Label XTELabel
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "XTE"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   330
      TabIndex        =   23
      Top             =   1350
      Width           =   675
   End
   Begin VB.Label TAEVal
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "00"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   1155
      TabIndex        =   24
      Top             =   1680
      Width           =   405
   End
   Begin VB.Label TypeLabel
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "H"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   8.25
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   195
      Left            =   7350
      TabIndex        =   59
      Top             =   930
      Visible         =   0   'False
      Width           =   135
   End
   Begin VB.Label NextWP
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "XXXXX"
      ForeColor       =   &H0000FFFF&
      Height          =   360
      Left            =   2070
      TabIndex        =   61
      Top             =   900
      Width           =   1305
   End
   Begin VB.Label LastWP
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "XXXXX"
```

```
      ForeColor       =   &H0000FFFF&
      Height          =   360
      Left            =   330
      TabIndex        =   60
      Top             =   900
      Width           =   1305
   End
   Begin VB.Shape GPSLinkLight
      FillColor       =   &H000000FF&
      FillStyle       =   0  'Solid
      Height          =   255
      Left            =   9330
      Shape           =   3  'Circle
      Top             =   60
      Visible         =   0   'False
      Width           =   315
   End
   Begin VB.Shape DSLinkLight
      FillColor       =   &H000000FF&
      FillStyle       =   0  'Solid
      Height          =   255
      Left            =   9330
      Shape           =   3  'Circle
      Top             =   360
      Visible         =   0   'False
      Width           =   315
   End
   Begin VB.Shape CCLinkLight
      FillColor       =   &H000000FF&
      FillStyle       =   0  'Solid
      Height          =   255
      Left            =   9330
      Shape           =   3  'Circle
      Top             =   660
      Visible         =   0   'False
      Width           =   315
   End
   Begin VB.Label TAELabel
      Appearance      =   0  'Flat
      BackColor       =   &H00000000&
      Caption         =   "TAE"
      BeginProperty Font
         name            =   "MS Sans Serif"
         charset         =   0
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H0000FFFF&
      Height          =   300
      Left            =   330
      TabIndex        =   3
      Top             =   1680
      Width           =   1455
   End
End
Attribute VB_Name = "Main"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit
Option Base 1

Dim curFlightPlan As FlightPlanRec
Dim GPSRecord As GPSRec
Dim SendParam As Integer
Dim Stepped As Integer
Dim DoStep As Integer
Dim StepPos As PrePositionRec
Dim StepAt As Single
Dim saveBRG As Single
Dim wpPos() As Double
Dim orgLong As Double
Dim orgLat As Double
Dim curWPind As Integer
Dim curWPIdent As String
Dim curLong As Double
Dim curLat As Double
Dim nextWPind As Integer
Dim nextWPIdent As String
Dim FAFdist As Single
Dim MAPdist As Single

Dim FlashActLight As Integer
Dim FlashArmLight As Integer
Dim FlashTurnLight As Integer

' Scaling parameters
Dim Border As Double
Dim CDIPicWidth As Double
Dim CDIPicHeight As Double
Dim CDIPicLeft As Double
Dim CDIPicRight As Double
Dim CDIPicTop As Double
Dim CDIPicBottom As Double
Dim CDIUnit As Single
Dim ACLeft As Double
Dim ACTop As Double
Dim XPeg As Single

Private Sub DoGoAround()
  Dim distToMAP As Single

  distToMAP = GPSRecord.DistToEnd - MAPdist
  GoAround.BackColor = &H4000&
  If distToMAP < 0 Then
    GoAround.Tag = distToMAP
    GoAround.Enabled = False
  End If
End Sub

Private Sub AddPredictor_Click(Value As Integer)
' Clear any old values off the CDI Display
  SetDefaultCDI
  If Value Then
    If PredOrder Then
      TypeLabel = "S"
    Else
      TypeLabel = "F"
    End If
  Else
    TypeLabel = "N"
  End If
End Sub

Private Sub AltFSSpin_SpinDown()
  AltSensitivity = AltSensitivity - 50
End Sub

Private Sub AltFSSpin_SpinUp()
  AltSensitivity = AltSensitivity + 50
End Sub

Private Sub AltSensitivity_Change()
  SetDefaultCDI
End Sub

Private Sub ClearCDI()
  CDIScale.Cls
End Sub

Private Sub CntrlConStatus_Change()
  Stepped = True
```

```
  If (CntrlConStatus = "RUNNING") Then
    Stepped = False
  End If
End Sub

Private Sub ConnectTimer_Timer()
  CheckGPSLinks
  CheckDSLinks
  CheckCCLinks
End Sub

Private Sub DrawCDIFrame()
  Dim temp As Single
  Dim i As Integer

' Draw a box for the main frame
  temp = CDIPicWidth / 10
  CDIScale.Line (CDIPicLeft - temp, CDIPicTop)-
(CDIPicRight + temp, CDIPicBottom), , B

' Draw aircraft symbol and the dots
  temp = 0.01 * CDIPicWidth
  CDIScale.Line (ACLeft - 4 * temp, ACTop)-(ACLeft
+ 4 * temp, ACTop)
  CDIScale.Line (ACLeft, ACTop - 3 * temp)-(ACLeft,
ACTop + 6 * temp)
  CDIScale.Line (ACLeft - 2 * temp, ACTop + 4 *
temp)-(ACLeft + 2 * temp, ACTop + 4 * temp)
  temp = CDIPicWidth / 20
  For i = 1 To 4
    CDIScale.Circle (ACLeft - i * 2 * temp, ACTop),
temp / 10
    CDIScale.Circle (ACLeft + i * 2 * temp, ACTop),
temp / 10
  Next i
  CDIScale.Line (ACLeft - 10 * temp, ACTop + temp)-
(ACLeft - 10 * temp, ACTop - temp)
  CDIScale.Line (ACLeft + 10 * temp, ACTop + temp)-
(ACLeft + 10 * temp, ACTop - temp)
End Sub

Private Sub DrawHSI(ByVal XTEValue As Single, ByVal
TAEValue As Single, ByVal PredValue As Single,
ByVal CTrack As Single)
  Dim DialCX As Double
  Dim DialCY As Double
  Dim tick() As Double
  Dim M() As Double
  Dim XTEOffset As Single
  Dim PredOffsetX As Single

  Dim temp As Single
  Dim tick2() As Double
  Dim i As Integer
  Dim j As Integer

' Limit the scale indication
  If (XTEValue < -XPeg) Then
    XTEValue = -XPeg
  ElseIf (XTEValue > XPeg) Then
    XTEValue = XPeg
  End If

  XTEOffset = (CDIPicWidth * XTEValue) / 2

' Limit the scale indication
  If (PredValue < -XPeg) Then
    PredValue = -XPeg
  ElseIf (PredValue > XPeg) Then
    PredValue = XPeg
  End If

  PredOffsetX = (CDIPicWidth * PredValue) / 2

' Draw the compass dial
  DialCX = CDIScale.ScaleWidth / 2
  DialCY = CDIScale.ScaleWidth / 2
  CDIScale.Circle (DialCX, DialCY), 1.3 *
CDIPicWidth / 2

' Draw triangle indicator at the top
  ReDim M(4, 4)
  EyeMat M()
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(4, 4)
  tick(1, 1) = 0: tick(1, 2) = -1.3 * CDIPicWidth /
2: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = 0.025 * CDIPicWidth: tick(2, 2) =
-0.72 * CDIPicWidth: tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = -0.025 * CDIPicWidth: tick(3, 2) =
-0.72 * CDIPicWidth: tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = 0: tick(4, 2) = -0.2 * CDIPicWidth:
tick(4, 3) = 0: tick(4, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  CDIScale.Line (tick(2, 1), tick(2, 2))-(tick(3,
1), tick(3, 2))
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(3,
1), tick(3, 2))
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(4,
1), tick(4, 2))

' Draw large tick marks around the compass
  ReDim M(4, 4)
  EyeMat M()
  RotZMat M(), -CTrack * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(2, 4)
  tick(1, 1) = 0: tick(1, 2) = -1.28 * CDIPicWidth
/ 2: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = 0: tick(2, 2) = -0.52 * CDIPicWidth:
tick(2, 3) = 0: tick(2, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  EyeMat M()
  TransMat M(), -DialCX, -DialCY, 0
  RotZMat M(), 90 * DtoR
  TransMat M(), DialCX, DialCY, 0
  For i = 1 To 3
    Transform3D tick(), M()
    CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  Next i

' Draw a N for North
  EyeMat M()
  RotZMat M(), -CTrack * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(4, 4)
  tick(1, 1) = -0.025 * CDIPicWidth: tick(1, 2) =
-0.4 * CDIPicWidth: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = -0.025 * CDIPicWidth: tick(2, 2) =
-0.48 * CDIPicWidth: tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = 0.025 * CDIPicWidth: tick(3, 2) =
-0.4 * CDIPicWidth: tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = 0.025 * CDIPicWidth: tick(4, 2) =
-0.48 * CDIPicWidth: tick(4, 3) = 0: tick(4, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  CDIScale.Line (tick(2, 1), tick(2, 2))-(tick(3,
1), tick(3, 2))
```

```
  CDIScale.Line (tick(3, 1), tick(3, 2))-(tick(4,
1), tick(4, 2))

' Draw an E for East
  EyeMat M()
  RotZMat M(), (-CTrack + 90) * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(6, 4)
  tick(1, 1) = -0.025 * CDIPicWidth: tick(1, 2) =
-0.4 * CDIPicWidth: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = -0.025 * CDIPicWidth: tick(2, 2) =
-0.48 * CDIPicWidth: tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = 0.025 * CDIPicWidth: tick(3, 2) =
-0.4 * CDIPicWidth: tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = 0.025 * CDIPicWidth: tick(4, 2) =
-0.48 * CDIPicWidth: tick(4, 3) = 0: tick(4, 4) = 1
  tick(5, 1) = -0.025 * CDIPicWidth: tick(5, 2) =
-0.44 * CDIPicWidth: tick(5, 3) = 0: tick(5, 4) = 1
  tick(6, 1) = 0.0125 * CDIPicWidth: tick(6, 2) =
-0.44 * CDIPicWidth: tick(6, 3) = 0: tick(6, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(3, 1), tick(3, 2))-(tick(1,
1), tick(1, 2))
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(5,
1), tick(5, 2))
  CDIScale.Line (tick(5, 1), tick(5, 2))-(tick(6,
1), tick(6, 2))
  CDIScale.Line (tick(5, 1), tick(5, 2))-(tick(2,
1), tick(2, 2))
  CDIScale.Line (tick(2, 1), tick(2, 2))-(tick(4,
1), tick(4, 2))

' Draw a S for South
  EyeMat M()
  RotZMat M(), (-CTrack + 180) * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(6, 4)
  tick(1, 1) = -0.025 * CDIPicWidth: tick(1, 2) =
-0.42 * CDIPicWidth: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = 0: tick(2, 2) = -0.4 * CDIPicWidth:
tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = 0.025 * CDIPicWidth: tick(3, 2) =
-0.42 * CDIPicWidth: tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = -0.025 * CDIPicWidth: tick(4, 2) =
-0.46 * CDIPicWidth: tick(4, 3) = 0: tick(4, 4) = 1
  tick(5, 1) = 0: tick(5, 2) = -0.48 * CDIPicWidth:
tick(5, 3) = 0: tick(5, 4) = 1
  tick(6, 1) = 0.025 * CDIPicWidth: tick(6, 2) =
-0.46 * CDIPicWidth: tick(6, 3) = 0: tick(6, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  CDIScale.Line (tick(2, 1), tick(2, 2))-(tick(3,
1), tick(3, 2))
  CDIScale.Line (tick(3, 1), tick(3, 2))-(tick(4,
1), tick(4, 2))
  CDIScale.Line (tick(4, 1), tick(4, 2))-(tick(5,
1), tick(5, 2))
  CDIScale.Line (tick(5, 1), tick(5, 2))-(tick(6,
1), tick(6, 2))

' Draw a W for West
  EyeMat M()
  RotZMat M(), (-CTrack - 90) * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(5, 4)
  tick(1, 1) = -0.025 * CDIPicWidth: tick(1, 2) =
-0.4 * CDIPicWidth: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = -0.025 * CDIPicWidth: tick(2, 2) =
-0.48 * CDIPicWidth: tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = 0.025 * CDIPicWidth: tick(3, 2) =
-0.4 * CDIPicWidth: tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = 0.025 * CDIPicWidth: tick(4, 2) =
-0.48 * CDIPicWidth: tick(4, 3) = 0: tick(4, 4) = 1
  tick(5, 1) = 0: tick(5, 2) = -0.44 * CDIPicWidth:
tick(5, 3) = 0: tick(5, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(2, 1), tick(2, 2))-(tick(1,
1), tick(1, 2))
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(5,
1), tick(5, 2))
  CDIScale.Line (tick(5, 1), tick(5, 2))-(tick(3,
1), tick(3, 2))
  CDIScale.Line (tick(3, 1), tick(3, 2))-(tick(4,
1), tick(4, 2))

' Draw small tick marks around the compass
  EyeMat M()
  RotZMat M(), -CTrack * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(2, 4)
  tick(1, 1) = 0: tick(1, 2) = -1.28 * CDIPicWidth
/ 2: tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = 0: tick(2, 2) = -0.585 *
CDIPicWidth: tick(2, 3) = 0: tick(2, 4) = 1
  Transform3D tick(), M()
  EyeMat M()
  TransMat M(), -DialCX, -DialCY, 0
  RotZMat M(), 10 * DtoR
  TransMat M(), DialCX, DialCY, 0
  For i = 1 To 4
    Transform3D tick(), M()
    For j = 1 To 8
      CDIScale.Line (tick(1, 1), tick(1, 2))-
(tick(2, 1), tick(2, 2))
      Transform3D tick(), M()
    Next j
  Next i

' Set up matrix for drawing the obs and cdi
  EyeMat M()
  RotZMat M(), -GPSRecord.TAE * DtoR
  TransMat M(), DialCX, DialCY, 0
  ReDim tick(8, 4)
  tick(1, 1) = 0: tick(1, 2) = -0.52 * CDIPicWidth:
tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = 0: tick(2, 2) = -0.2 * CDIPicWidth:
tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = 0: tick(3, 2) = -0.585 *
CDIPicWidth: tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = 0.025 * CDIPicWidth: tick(4, 2) =
-0.52 * CDIPicWidth: tick(4, 3) = 0: tick(4, 4) = 1
  tick(5, 1) = -0.025 * CDIPicWidth: tick(5, 2) =
-0.52 * CDIPicWidth: tick(5, 3) = 0: tick(5, 4) = 1
  tick(6, 1) = XTEOffset: tick(6, 2) = -0.2 *
CDIPicWidth: tick(6, 3) = 0: tick(6, 4) = 1
  tick(7, 1) = XTEOffset: tick(7, 2) = 0.2 *
CDIPicWidth: tick(7, 3) = 0: tick(7, 4) = 1
  tick(8, 1) = CDIPicWidth / 20: tick(8, 2) = 0:
tick(8, 3) = 0: tick(8, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  CDIScale.Line (-tick(3, 1) + 2 * DialCX, -tick(3,
2) + 2 * DialCY)-(-tick(2, 1) + 2 * DialCX,
-tick(2, 2) + 2 * DialCY)
  CDIScale.Line (tick(3, 1), tick(3, 2))-(tick(4,
1), tick(4, 2))
  CDIScale.Line (tick(3, 1), tick(3, 2))-(tick(5,
1), tick(5, 2))
  CDIScale.Line (tick(4, 1), tick(4, 2))-(tick(5,
1), tick(5, 2))
  CDIScale.Line (tick(6, 1), tick(6, 2))-(tick(7,
1), tick(7, 2))
```

```
' Draw aircraft symbol and the dots
  temp = 0.01 * CDIPicWidth
  CDIScale.Line (ACLeft - 4 * temp, ACTop)-(ACLeft
+ 4 * temp, ACTop)
  CDIScale.Line (ACLeft, ACTop - 3 * temp)-(ACLeft,
ACTop + 6 * temp)
  CDIScale.Line (ACLeft - 2 * temp, ACTop + 4 *
temp)-(ACLeft + 2 * temp, ACTop + 4 * temp)
  tick(8, 1) = tick(8, 1) - DialCX: tick(8, 2) =
tick(8, 2) - DialCY
  For i = 1 To 4
    CDIScale.Circle (DialCX + i * 2 * tick(8, 1),
DialCY + i * 2 * tick(8, 2)), temp / 2
    CDIScale.Circle (DialCX - i * 2 * tick(8, 1),
DialCY - i * 2 * tick(8, 2)), temp / 2
  Next i
' Add a line for max. scale
  ReDim tick2(6, 2)
  tick2(1, 1) = DialCX - tick(8, 2) + 10 * tick(8,
1): tick2(1, 2) = DialCY + tick(8, 1) + 10 *
tick(8, 2)
  tick2(2, 1) = DialCX + tick(8, 2) + 10 * tick(8,
1): tick2(2, 2) = DialCY - tick(8, 1) + 10 *
tick(8, 2)
  tick2(3, 1) = DialCX + 11 * tick(8, 1): tick2(3,
2) = DialCY + 11 * tick(8, 2)
  tick2(4, 1) = DialCX - tick(8, 2) - 10 * tick(8,
1): tick2(4, 2) = DialCY + tick(8, 1) - 10 *
tick(8, 2)
  tick2(5, 1) = DialCX + tick(8, 2) - 10 * tick(8,
1): tick2(5, 2) = DialCY - tick(8, 1) - 10 *
tick(8, 2)
  tick2(6, 1) = DialCX - 11 * tick(8, 1): tick2(6,
2) = DialCY - 11 * tick(8, 2)
  CDIScale.Line (tick2(1, 1), tick2(1, 2))-
(tick2(2, 1), tick2(2, 2))
  CDIScale.Line (tick2(4, 1), tick2(4, 2))-
(tick2(5, 1), tick2(5, 2))
' Include an offscale indication if appropriate
  If XTEValue = -XPeg Then
    CDIScale.Line (tick2(4, 1), tick2(4, 2))-
(tick2(6, 1), tick2(6, 2))
    CDIScale.Line (tick2(5, 1), tick2(5, 2))-
(tick2(6, 1), tick2(6, 2))
  ElseIf XTEValue = XPeg Then
    CDIScale.Line (tick2(1, 1), tick2(1, 2))-
(tick2(3, 1), tick2(3, 2))
    CDIScale.Line (tick2(2, 1), tick2(2, 2))-
(tick2(3, 1), tick2(3, 2))
  End If
' Add the Predictor
  If AddPredictor Then
    ReDim tick(4, 4)
    tick(1, 1) = PredOffsetX + 0.01 * CDIPicWidth:
tick(1, 2) = -0.07 * CDIPicWidth: tick(1, 3) = 0:
tick(1, 4) = 1
    tick(2, 1) = PredOffsetX - 0.01 * CDIPicWidth:
tick(2, 2) = -0.07 * CDIPicWidth: tick(2, 3) = 0:
tick(2, 4) = 1
    tick(3, 1) = PredOffsetX + 0.01 * CDIPicWidth:
tick(3, 2) = 0.07 * CDIPicWidth: tick(3, 3) = 0:
tick(3, 4) = 1
    tick(4, 1) = PredOffsetX - 0.01 * CDIPicWidth:
tick(4, 2) = 0.07 * CDIPicWidth: tick(4, 3) = 0:
tick(4, 4) = 1
    Transform3D tick(), M()
    CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
    CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(3,
1), tick(3, 2))
    CDIScale.Line (tick(4, 1), tick(4, 2))-(tick(2,
1), tick(2, 2))
    CDIScale.Line (tick(4, 1), tick(4, 2))-(tick(3,
1), tick(3, 2))
  End If
End Sub

Private Sub DrawPredictor(ByVal PredValue As
Single)
  Dim PredOffset As Single

  PredOffset = ScaleIndication(PredValue)

' Draw the Prediction indicator
'   Box format
'  CDIScale.Line (PredOffset - CDIUnit, ACTop - 4 *
CDIUnit)-(PredOffset + CDIUnit, ACTop + 4 *
CDIUnit), , B
'   Short line format
'  CDIScale.Line (PredOffset, ACTop - 6 * CDIUnit)-
(PredOffset, ACTop + 2 * CDIUnit)
'   Dotted line format
  CDIScale.Line (PredOffset, ACTop - 8 * CDIUnit)-
(PredOffset, ACTop - 7.5 * CDIUnit)
  CDIScale.Line (PredOffset, ACTop - 5.5 *
CDIUnit)-(PredOffset, ACTop - 4 * CDIUnit)
  CDIScale.Line (PredOffset, ACTop - 2 * CDIUnit)-
(PredOffset, ACTop - 0.5 * CDIUnit)
  CDIScale.Line (PredOffset, ACTop + 1.5 *
CDIUnit)-(PredOffset, ACTop + 2 * CDIUnit)
End Sub

Private Sub DrawText(MagDeviation As Single, CTrack
As Single)
  Dim temp As Single

' Insert Waypoint names
  If Main.LastWP.Visible Then
    Main.LastWP = Trim(GPSRecord.LastWPIdent)
  End If
  If Main.NextWP.Visible Then
    Main.NextWP = Trim(GPSRecord.curWPIdent)
  End If

' Display the Distance to Waypoint
  If Main.DisttoWP.Visible Then
    Main.DisttoWP = Format(GPSRecord.DisttoWP,
"00.0")
  End If

' Display the Time to Waypoint
  If Main.TimeToWP.Visible Then
    Main.TimeToWP = Format(Int(GPSRecord.ETW / 60)
Mod 60, "00") + ":" + Format(GPSRecord.ETW Mod 60,
"00")
  End If

' Display the Ground Speed
  If Main.GrndSpeed.Visible Then
    Main.GrndSpeed = Format(GPSRecord.GrndSpeed,
"000")
  End If

' Display the Desired Track in magnetic degrees
  If Main.DesiredTrack.Visible Then
'     temp = Reciprocal(CDbl(GPSRecord.CurRadial)) +
MagDeviation
    temp = CTrack - GPSRecord.TAE
    If temp < 0 Then temp = temp + 360
    If temp > 360 Then temp = temp - 360
    Main.DesiredTrack = Format(temp, "000")
  End If

' Display the Current Track in magnetic degrees
```

```
  If Main.CurTrack.Visible Then
    Main.CurTrack = Format(CTrack, "000")
  End If

' Display the Current Bearing in magnetic degrees
  If Main.CurBearing.Visible Then
    temp = GPSRecord.HdgtoWP + MagDeviation
    If temp < 0 Then temp = temp + 360
    If temp > 360 Then temp = temp - 360
    Main.CurBearing = Format(temp, "000")
  End If

' Display the Cross Track Error digitally
  If Main.XTEVal.Visible Then
    Main.XTEVal = Format(Abs(GPSRecord.XTE),
"00.0")
  End If

' Display the Track Angle Error digitally
  If Main.TAEVal.Visible Then
    temp = GPSRecord.TAE
    If temp > 90 Then temp = temp - 180
    If temp < -90 Then temp = temp + 180
    Main.TAEVal = Format(Abs(temp), "00")
'   Display the Direction of the Track Angle Error
    Main.TAEDirection(0).Visible = temp <= 0
    Main.TAEDirection(1).Visible = temp >= 0
  End If


End Sub

Private Sub DrawTriangle(ByVal TAEValue As Single)
  Dim TAEOffset As Single

  TAEOffset = ScaleIndication(TAEValue)

' Draw the TAE indicator
  CDIScale.Line (TAEOffset, ACTop + 5 * CDIUnit)-
(TAEOffset + 1.5 * CDIUnit, ACTop + 8 * CDIUnit)
  CDIScale.Line (TAEOffset, ACTop + 5 * CDIUnit)-
(TAEOffset - 1.5 * CDIUnit, ACTop + 8 * CDIUnit)
  CDIScale.Line (TAEOffset - 1.5 * CDIUnit, ACTop +
8 * CDIUnit)-(TAEOffset + CDIUnit, ACTop + 8 *
CDIUnit)
End Sub

Private Sub DrawVector(ByVal XTEValue As Single,
ByVal TAEValue As Single)
  Dim tick() As Double
  Dim M() As Double
  Dim XTEOffset As Single

  XTEOffset = ScaleIndication(XTEValue)

' Draw the XTE indicators
  CDIScale.Line (XTEOffset, ACTop - 10 * CDIUnit)-
(XTEOffset, ACTop - 8 * CDIUnit)
  CDIScale.Line (XTEOffset, ACTop + 10 * CDIUnit)-
(XTEOffset, ACTop + 8 * CDIUnit)

  ReDim M(4, 4)

' Calculate transformation matrix to draw the track
vector
  EyeMat M()
  RotZMat M(), -TAEValue * DtoR
  TransMat M(), CDbl(XTEOffset), ACTop, 0

' Transform the track vector and draw it
  ReDim tick(7, 4)
  tick(1, 1) = 0: tick(1, 2) = -8 * CDIUnit:
tick(1, 3) = 0: tick(1, 4) = 1
  tick(2, 1) = -CDIUnit: tick(2, 2) = -6 * CDIUnit:
tick(2, 3) = 0: tick(2, 4) = 1
  tick(3, 1) = CDIUnit: tick(3, 2) = -6 * CDIUnit:
tick(3, 3) = 0: tick(3, 4) = 1
  tick(4, 1) = 0: tick(4, 2) = -6 * CDIUnit:
tick(4, 3) = 0: tick(4, 4) = 1
  tick(5, 1) = 0: tick(5, 2) = 8 * CDIUnit: tick(5,
3) = 0: tick(5, 4) = 1
  tick(6, 1) = 0: tick(6, 2) = -CDIPicWidth:
tick(6, 3) = 0: tick(6, 4) = 1
  tick(7, 1) = 0: tick(7, 2) = CDIPicWidth: tick(7,
3) = 0: tick(7, 4) = 1
  Transform3D tick(), M()
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(2,
1), tick(2, 2))
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(3,
1), tick(3, 2))
  CDIScale.Line (tick(2, 1), tick(2, 2))-(tick(3,
1), tick(3, 2))
  CDIScale.Line (tick(4, 1), tick(4, 2))-(tick(5,
1), tick(5, 2))

' Clip the vector to the frame
  If (tick(6, 2) < CDIPicTop) Then
    tick(6, 1) = tick(7, 1) + (tick(6, 1) - tick(7,
1)) * (CDIPicTop - tick(7, 2)) / (tick(6, 2) -
tick(7, 2))
    tick(6, 2) = CDIPicTop
  ElseIf (tick(6, 2) > CDIPicBottom) Then
    tick(6, 1) = tick(7, 1) + (tick(6, 1) - tick(7,
1)) * (CDIPicBottom - tick(7, 2)) / (tick(6, 2) -
tick(7, 2))
    tick(6, 2) = CDIPicBottom
  End If
  If (tick(6, 1) < CDIPicLeft) Then
    tick(6, 2) = tick(7, 2) + (tick(6, 2) - tick(7,
2)) * (CDIPicLeft - tick(7, 1)) / (tick(6, 1) -
tick(7, 1))
    tick(6, 1) = CDIPicLeft
  ElseIf (tick(6, 1) > CDIPicRight) Then
    tick(6, 2) = tick(7, 2) + (tick(6, 2) - tick(7,
2)) * (CDIPicRight - tick(7, 1)) / (tick(6, 1) -
tick(7, 1))
    tick(6, 1) = CDIPicRight
  End If
  If (tick(7, 2) < CDIPicTop) Then
    tick(7, 1) = tick(6, 1) + (tick(7, 1) - tick(6,
1)) * (CDIPicTop - tick(6, 2)) / (tick(7, 2) -
tick(6, 2))
    tick(7, 2) = CDIPicTop
  ElseIf (tick(7, 2) > CDIPicBottom) Then
    tick(7, 1) = tick(6, 1) + (tick(7, 1) - tick(6,
1)) * (CDIPicBottom - tick(6, 2)) / (tick(7, 2) -
tick(6, 2))
    tick(7, 2) = CDIPicBottom
  End If
  If (tick(7, 1) < CDIPicLeft) Then
    tick(7, 2) = tick(6, 2) + (tick(7, 2) - tick(6,
2)) * (CDIPicLeft - tick(6, 1)) / (tick(7, 1) -
tick(6, 1))
    tick(7, 1) = CDIPicLeft
  ElseIf (tick(7, 1) > CDIPicRight) Then
    tick(7, 2) = tick(6, 2) + (tick(7, 2) - tick(6,
2)) * (CDIPicRight - tick(6, 1)) / (tick(7, 1) -
tick(6, 1))
    tick(7, 1) = CDIPicRight
  End If

' Draw the clipped lines
```

```
  CDIScale.Line (tick(1, 1), tick(1, 2))-(tick(6,
1), tick(6, 2))
  CDIScale.Line (tick(5, 1), tick(5, 2))-(tick(7,
1), tick(7, 2))
End Sub

Private Sub DrawXTE(ByVal XTEValue As Single)
  Dim XTEOffset As Single

  XTEOffset = ScaleIndication(XTEValue)

' Draw the XTE indicator
  CDIScale.Line (XTEOffset, ACTop - 6 * CDIUnit)-
(XTEOffset, ACTop + 6 * CDIUnit)
End Sub

Private Sub FlashTimer_Timer()
  Dim OneOn As Integer

  OneOn = 0
  If FlashArmLight Then
    If ArmLight.BackColor = OnBackColor Then
      ArmLight.BackColor = OffBackColor     ' Light
Grey
      ArmLight.ForeColor = OffForeColor     ' Mid
Grey
      OneOn = 1
    Else
      ArmLight.BackColor = OnBackColor      ' White
      ArmLight.ForeColor = OnForeColor      ' Black
      OneOn = 2
    End If
  End If
  If FlashActLight Then
    If ActLight.BackColor = OnBackColor And OneOn
<> 2 Then
      ActLight.BackColor = OffBackColor     ' Light
Grey
      ActLight.ForeColor = OffForeColor     ' Mid
Grey
      OneOn = 1
    Else
      ActLight.BackColor = OnBackColor      ' White
      ActLight.ForeColor = OnForeColor      ' Black
      OneOn = 2
    End If
  End If
  If FlashTurnLight Then
    If TurnLight.BackColor = OnBackColor And OneOn
<> 2 Then
      TurnLight.BackColor = OffBackColor    ' Light
Grey
      TurnLight.ForeColor = OffForeColor    ' Mid
Grey
      OneOn = 1
    Else
      TurnLight.BackColor = OnBackColor     ' White
      TurnLight.ForeColor = OnForeColor     ' Black
      OneOn = 2
    End If
  End If

  If OneOn = 0 Then
'   If all of the flashing lights are disabled then
turn off the flash timer
    If FlashTimer.Enabled Then FlashTimer.Enabled =
False
  End If
End Sub

Private Sub FltPlnName_Change()
  Dim i As Integer
```

```
  Dim planLen As Single

' Open the new flight plan
  openFlightPlan curFlightPlan,
Format$(ExpName.Caption),
Format$(ProfileName.Caption),
Format$(FltPlnName.Caption)

  If curFlightPlan.NumWayPnts < 1 Then
      ' We don't really have a new flight plan
      Exit Sub
  End If
  ReDim wpPos(curFlightPlan.NumWayPnts, 4)

' Compile waypoints into a table of homogeneous co-
ordinates
' Also find the FAF and MAP and calculate the
distance to them from the start
  orgLong = curFlightPlan.WayPnts(0).Longitude
  orgLat = curFlightPlan.WayPnts(0).Latitude
  planLen = 0
  For i = 0 To curFlightPlan.NumWayPnts - 1
    wpPos(i + 1, 1) = curFlightPlan.LongCoef *
(curFlightPlan.WayPnts(i).Longitude - orgLong) * 60
* 6080
    wpPos(i + 1, 2) =
(curFlightPlan.WayPnts(i).Latitude - orgLat) * 60 *
6080
    wpPos(i + 1, 3) =
curFlightPlan.WayPnts(i).Altitude
    wpPos(i + 1, 4) = 1
    planLen = planLen +
curFlightPlan.WayPnts(i).LegLen
    If
UCase$(Trim$(curFlightPlan.WayPnts(i).WPType)) =
"FAF" Then FAFdist = planLen
    If
UCase$(Trim$(curFlightPlan.WayPnts(i).WPType)) =
"MAP" Then MAPdist = planLen
  Next i

' Convert FAF and MAP coordinates to distance to go
  FAFdist = planLen - FAFdist
  MAPdist = planLen - MAPdist

' Reset the GoAround button
  GoAround.ForeColor = OffForeColor
  GoAround.BackColor = OffBackColor
  GoAround.Enabled = True
  GoAround.Tag = 0
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
  Dim c As String * 1

  If Not TestControls.Visible Then
    c = UCase$(Chr$(KeyAscii))
    Select Case c
      Case "H", "2"
        ' Hide/Show all of the system controls
        QuitButton.Visible = Not QuitButton.Visible
        SetupButton.Visible = Not
SetupButton.Visible
        SysSetCommand.Visible = Not
SysSetCommand.Visible
        ReconnectButton.Visible = Not
ReconnectButton.Visible
        TypeButton.Visible = Not TypeButton.Visible
        StepIsOn.Visible = Not StepIsOn.Visible
        TypeLabel.Visible = Not TypeLabel.Visible
        CCLinkLight.Visible = Not
CCLinkLight.Visible
```

```
        DSLinkLight.Visible = Not
DSLinkLight.Visible
        GPSLinkLight.Visible = Not
GPSLinkLight.Visible
      Case "Q"
        End
      Case "D"
        ' Turn on/off the DDE links
        ConnectTimer.Enabled = Not
ConnectTimer.Enabled
      Case "S", "3"
        ' Show/Hide the DDE link data
        DDEFrame.Visible = Not DDEFrame.Visible
      Case "T", "1"
        TypeButton = True
      Case "0"
        DoGoAround
    End Select
  End If
End Sub

Private Sub Form_Load()

  Load SystemSettings
  initTrig
  initFlightPlan
Format$(SystemSettings.CtrlConDB.Text)
  curWPind = 0
  curWPIdent = ""
  curLat = 0
  curLong = 0
  nextWPind = 1
  nextWPIdent = ""
  orgLat = 0
  orgLong = 0

' Set the background to black
  Main.BackColor = &H0&

' Clear the Waypoint Captions
  LastWP.Caption = ""
  NextWP.Caption = ""

' Set up some useful variables
  Border = 0.2
  CDIPicWidth = CDIScale.ScaleWidth * (1 - 2 *
Border)
  CDIPicHeight = CDIScale.ScaleHeight * (1 - 2 *
Border)
  CDIPicLeft = CDIScale.ScaleWidth * Border
  CDIPicRight = CDIScale.ScaleWidth * (1 - Border)
  CDIPicTop = 3 * CDIScale.ScaleHeight / 8
  CDIPicBottom = 5 * CDIScale.ScaleHeight / 8
  CDIUnit = CDIScale.ScaleHeight / 80
  ACLeft = CDIScale.ScaleWidth / 2
  ACTop = CDIScale.ScaleHeight / 2
  XPeg = 0.98

' Use the Blank Display as the default
  SetDisplayType XTEType

' Clear off the analog CDI display
  SetDefaultCDI

' Zero the Second order predictor hold variables
  VHold = 0
  TAEHold = 0
  GAEHold = 0
  TimeHold = 0

  MAPdist = 0
  FAFdist = 0

' Set up the DDE connections
  ClearGPSLinks
  ClearDSLinks
  ClearCCLinks
  ConnectTimer.Enabled = True
  Stepped = True
  DoStep = False

' Reset Flashing lights
  FlashActLight = False
  FlashArmLight = False
  FlashTurnLight = False

  saveBRG = Int(360 * Rnd)

End Sub

Private Sub Form_Unload(Cancel As Integer)
  closeFlightPlan
End Sub

Private Sub GoAround_Click()
  DoGoAround
End Sub

Private Sub GPSData_Change()

  If Trim$(UCase$(CntrlConStatus.Caption)) =
"RUNNING" Then
'   Parse the incoming data from the GPS module
    ParseGPSRec GPSRecord, Format$(GPSData.Caption)
    ProcessData
  Else
    CntrlConStatus_Change
  End If

End Sub

Private Sub GPSStatus_Change()
  Dim Tpos As Integer
  Dim Tchar As String

' Set turn annuciator to indicate turn is happening
  Tpos = InStr(GPSStatus, "T")
  If Tpos > 0 Then
    Tchar = Mid$(GPSStatus, Tpos + 1, 1)
  Else
    Tchar = "N"
  End If

  Select Case Tchar
    Case "A"  ' Turn is happening and aircraft is
after the waypoint
      If FlashTurnLight Then FlashTurnLight = False
      If TurnLight.BackColor <> OnBackColor Then
        TurnLight.BackColor = OnBackColor
        TurnLight.ForeColor = OnForeColor
      End If
    Case "B"  ' Turn is happening and aircraft is
before the waypoint
      If FlashTurnLight Then FlashTurnLight = False
      If TurnLight.BackColor <> OnBackColor Then
        TurnLight.BackColor = OnBackColor
        TurnLight.ForeColor = OnForeColor
      End If
    Case "C"  ' Turn is close - within 1 nm
      If Not FlashTimer.Enabled Then
FlashTimer.Enabled = True
      If Not FlashTurnLight Then
        TurnLight.BackColor = OnBackColor
        TurnLight.ForeColor = OnForeColor
```

```
      FlashTurnLight = True     ' Flash on Turn
    End If
  Case "N"  ' no Turn imminent
    If FlashTurnLight Then FlashTurnLight = False
    If TurnLight.BackColor <> OffBackColor Then
      TurnLight.BackColor = OffBackColor
      TurnLight.ForeColor = OffForeColor
    End If
  End Select
End Sub


Private Sub PanelParam_Change()
  Dim temp As String
  Dim comPos As Integer
  Dim comStr As String
  Dim spcPos As Integer
  Dim dispType As String

' Check to see if this programme made the change
' If so then send the new data to the control con
  If SendParam Then
    If LinkedtoCC Then
      On Error Resume Next
      PanelParam.LinkPoke
      If Err <> 0 Then
        ClearCCLinks
      End If
      On Error GoTo 0
    End If
    SendParam = False
    Return
  End If

' Otherwise parse the new values ...

' Find the "step" command if it exists
  comPos = InStr(1, PanelParam, "step", 1)
  If comPos <> 0 Then
'   Turn on the step
    DoStep = True
    comStr = LTrim$(Right$(PanelParam,
Len(PanelParam) - comPos - 3))
'   Set up step data
    spcPos = InStr(1, comStr, " ", 0)
    StepAt = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    StepPos.Latitude = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    StepPos.Longitude = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    StepPos.Altitude = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    StepPos.Pitch = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    StepPos.Roll = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    StepPos.Heading = Left(comStr, spcPos)
    comStr = LTrim$(Right$(comStr, Len(comStr) -
spcPos))
    spcPos = InStr(1, comStr, " ", 0)
    If spcPos = 0 Then spcPos = Len(comStr)
    StepPos.Airspeed = Left(comStr, spcPos)
  Else
'   Turn off the step
    DoStep = False
  End If

' Find the "pred" command if it exists
  comPos = InStr(1, PanelParam, "pred", 1)
  If comPos <> 0 Then
    comStr = LTrim$(Right$(PanelParam,
Len(PanelParam) - comPos - 3))
  End If

' Find the "disp" command if it exists
  comPos = InStr(1, PanelParam, "disp", 1)
  If comPos <> 0 Then
    comStr = LTrim$(Right$(PanelParam,
Len(PanelParam) - comPos - 3))
    dispType = Left$(comStr, 1)
    Select Case dispType
      Case "X"
        SetDisplayType XTEType
      Case "E"
        SetDisplayType EHSIType
      Case "V"
        SetDisplayType VectorType
      Case "T"
        SetDisplayType TriangleType
      Case "P"
        SetDisplayType PredType
    End Select
  End If

End Sub

Private Sub PredOrder_Click(Value As Integer)
  If AddPredictor Then
    If Value Then
      TypeLabel = "S"
    Else
      TypeLabel = "F"
    End If
  Else
    TypeLabel = "N"
  End If

End Sub

Private Sub PredTime_Change()
  SetDefaultCDI
End Sub

Private Sub PredTimeSpin_SpinDown()
  PredTime = PredTime - 5
End Sub

Private Sub PredTimeSpin_SpinUp()
  PredTime = PredTime + 5
End Sub

Private Sub ProcessData()
  Dim XTrackError As Single
  Dim AltitudeError As Single
  Dim DGA As Single
  Dim DistPred As Single
  Dim DistPredA As Single
  Dim AltFact As Single
  Dim V As Single
  Dim dV As Single
  Dim dTAE As Single
```

```
  Dim dGAE As Single
  Dim dTime As Single
  Dim XTEPred As Single
  Dim ALEPred As Single
  Dim MagDeviation As Single
  Dim CTrack As Single

  Dim XTEOffset As Single
  Dim ALEOffset As Single
  Dim PredOffsetX As Single
  Dim PredOffsetA As Single
  Dim TAEOffset As Single

  Dim distToFAF As Single
  Dim distToMAP As Single
  Dim hnavSens As Single

' Calculate Desired Glide Angle
  DGA = GPSRecord.CGA - GPSRecord.GAE

' Calculate the current magnetic deviation
  MagDeviation = 16

' Calculate the current track
  CTrack = GPSRecord.Track + MagDeviation
  If CTrack < 0 Then CTrack = CTrack + 360
  If CTrack > 360 Then CTrack = CTrack - 360

' Redraw the digital displays
  DrawText MagDeviation, CTrack

' Quit if we don't have a flight plan yet
'  If (curFlightPlan.NumWayPnts < 2) Then Exit Sub

' Go to the next waypoint if necessary
  TransitionWP

' Calculate full scale defelctions
  distToFAF = GPSRecord.DistToEnd - FAFdist
  distToMAP = GPSRecord.DistToEnd - MAPdist
  hnavSens = 0.3 + 0.7 * distToFAF / 2
  If hnavSens > 1 Then hnavSens = 1
  If hnavSens < 0.3 Then hnavSens = 0.3
  If distToMAP < 0 Then
    GoAround.ForeColor = OnForeColor           '
Black
    GoAround.BackColor = &HC000&       ' Green
    If Not GoAround.Enabled Then
      hnavSens = 0.3 + 0.7 * (GoAround.Tag -
distToMAP) / 2
      If hnavSens > 1 Then hnavSens = 1
      If hnavSens < 0.3 Then hnavSens = 0.3
    End If
  End If

' Set the annunciators to indicate the sensitivity
change
' Arming for approach indication
  If (distToFAF < 3) And (distToFAF > 0) Then
    If Not FlashTimer.Enabled Then
FlashTimer.Enabled = True
    If Not FlashArmLight Then
      ArmLight.BackColor = OnBackColor     ' White
      ArmLight.ForeColor = OnForeColor           '
Black
      FlashArmLight = True              ' Flash on
ARM
      ArmLight.Enabled = True
    End If
  ElseIf (distToFAF < 0) And ((distToMAP > 0) Or
GoAround.Enabled) Then
    If ArmLight.BackColor <> OffBackColor Then
      ArmLight.BackColor = OffBackColor    ' Light
Grey
      ArmLight.ForeColor = OffForeColor    ' Mid
Grey
    End If
    If FlashArmLight Then FlashArmLight = False
    If ArmLight.Enabled Then ArmLight.Enabled =
False
  Else
    If ArmLight.BackColor <> OnBackColor Then
      ArmLight.BackColor = OnBackColor     ' White
      ArmLight.ForeColor = OnForeColor           '
Black
    End If
    If FlashArmLight Then FlashArmLight = False
    If Not ArmLight.Enabled Then ArmLight.Enabled =
True
  End If
' Active approach indication
  If (distToFAF < 2) And (distToFAF > 0) Then
    If Not FlashTimer.Enabled Then
FlashTimer.Enabled = True
    If Not FlashActLight Then
      ActLight.BackColor = OnBackColor     ' White
      ActLight.ForeColor = OnForeColor           '
Black
      FlashActLight = True              ' Flash on
ACT
      ActLight.Enabled = True
    End If
  ElseIf (distToFAF < 0) And ((distToMAP > 0) Or
GoAround.Enabled) Then
    If ActLight.BackColor <> OnBackColor Then
      ActLight.BackColor = OnBackColor     ' White
      ActLight.ForeColor = OnForeColor           '
Black
    End If
    If FlashActLight Then FlashActLight = False
' Steady on ACT
    If Not ActLight.Enabled Then ActLight.Enabled =
True
  Else
    If ActLight.BackColor <> OffBackColor Then
      ActLight.BackColor = OffBackColor    ' Light
Grey
      ActLight.ForeColor = OffForeColor    ' Mid
Grey
    End If
    If FlashActLight Then FlashActLight = False
    If ActLight.Enabled Then ActLight.Enabled =
False
  End If

' Calculate first and second order predictor if
required
  XTEPred = GPSRecord.XTE
  If AddPredictor Then
'   Do an initial set of the time hold
    If TimeHold <= 0 Then
      TimeHold = GPSRecord.TimeStamp
    End If

'   Calculate Distance predictions based on current
Ground speed
    V = (GPSRecord.GrndSpeed / 3600)
    DistPred = PredTime * V
    DistPredA = DistPred * AltFact
    If PredOrder Then
      dTime = GPSRecord.TimeStamp - TimeHold
      dV = V - VHold
      If dTime > 0 Then
```

```
        DistPred = DistPred + (PredTime ^ 2) * dV /
(2 * dTime)
        DistPredA = DistPredA + ((PredTime *
AltFact) ^ 2) * dV / (2 * dTime)
      End If
      VHold = V
      TimeHold = GPSRecord.TimeStamp
    End If

'   Calculate XTE prediction based on TAE and
current Ground speed
    XTEPred = XTEPred + DistPred *
Sin(GPSRecord.TAE * DtoR)
    If PredOrder Then
      dTAE = GPSRecord.TAE - TAEHold
      If dTime > 0 Then
        XTEPred = XTEPred + (PredTime ^ 2) * V *
dTAE * Cos(GPSRecord.TAE * DtoR) / (2 * dTime)
      End If
      TAEHold = GPSRecord.TAE
    End If
  End If

' Scale the XTE Indication to +/- 1.0
  XTrackError = -GPSRecord.XTE / hnavSens

' Scale the Predictor Indication to +/- 1.0
  XTEPred = -XTEPred / hnavSens

' Scale TAE Indication to +/- 90 full scale
deflection
  TAEOffset = CSng(GPSRecord.TAE) / 90

' Clear the old indication and redraw everything
  ClearCDI
  If (DisplayType = EHSIType) Then
    DrawHSI XTrackError, CSng(GPSRecord.TAE),
XTEPred, CTrack
  Else
    DrawCDIFrame
    If (DisplayType = VectorType) Then
      DrawVector XTrackError, CSng(GPSRecord.TAE)
    Else
      DrawXTE XTrackError
      If (DisplayType = TriangleType) Then
        DrawTriangle TAEOffset
      ElseIf (DisplayType = PredType) Then
        DrawPredictor XTEPred
      End If
    End If
  End If

  StepIsOn = DoStep And Not Stepped
' Check to see if the step condition has been met.
  If DoStep And (GPSRecord.DistToEnd < StepAt) And
(GPSRecord.DistToEnd > (StepAt - 1)) And Not
Stepped Then
    If StepPos.Altitude < 0 Then StepPos.Altitude =
-StepPos.Altitude + GPSRecord.AltErr
    PrePosition = MakePrePositionStr(StepPos)
    If LinkedtoDS Then
      On Error Resume Next
      PrePosition.LinkPoke
      If Err = 0 Then
        Stepped = True
      Else
        ClearDSLinks
      End If
      On Error GoTo 0
    End If
  End If
End Sub

Private Sub QuitButton_Click()
  End
End Sub

Private Sub ReconnectButton_Click()
  ClearDSLinks
  ClearCCLinks
  ClearGPSLinks
End Sub

Private Function ScaleIndication(ByVal ScaleValue
As Single) As Single
  Dim ScaleOffset As Single
  Dim temp As Single

' Limit the scale indication
  If (ScaleValue < -XPeg) Then
    ScaleValue = -XPeg
  ElseIf (ScaleValue > XPeg) Then
    ScaleValue = XPeg
  End If

' Calculate the offset for drawing this indication
  ScaleOffset = (CDIScale.ScaleWidth + CDIPicWidth
* ScaleValue) / 2
  ScaleIndication = ScaleOffset

' Draw offscale indicators if necessary
  If Abs(ScaleValue) = XPeg Then
    ScaleValue = Sgn(ScaleValue)
    ScaleOffset = (CDIScale.ScaleWidth +
CDIPicWidth * ScaleValue) / 2
    temp = CDIPicWidth / 20
    CDIScale.Line (ScaleOffset, ACTop - temp)-
(ScaleOffset + ScaleValue * temp, ACTop)
    CDIScale.Line (ScaleOffset, ACTop + temp)-
(ScaleOffset + ScaleValue * temp, ACTop)
    CDIScale.Line (ScaleOffset, ACTop - temp)-
(ScaleOffset, ACTop + temp)
  End If
End Function

Private Sub SetDefaultCDI()
  Test = True
End Sub

Private Sub SetDisplayType(Value As Integer)
  DisplayType = Value

  AddPredictor = False
  PredOrder = False

  Select Case Value
    Case XTEType
      TypeLabel = "X"
    Case VectorType
      TypeLabel = "V"
    Case PredType
      AddPredictor = True
      TypeLabel = "P"
    Case TriangleType
      TypeLabel = "T"
    Case EHSIType
'       AddPredictor = True
      TypeLabel = "E"
  End Select
  SetDefaultCDI
End Sub

Private Sub SetupButton_Click()
  AnalogControls.Visible = Not
AnalogControls.Visible
```

```
  TestControls.Visible = Not TestControls.Visible
  DigitalControls.Visible = Not
DigitalControls.Visible
End Sub

Private Sub ShowBRG_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    CurBearing.ForeColor = TextBackColor
    CurBearing.BackColor = TextForeColor
    BRGLabel.ForeColor = TextBackColor
    BRGLabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    CurBearing.ForeColor = TextForeColor
    CurBearing.BackColor = TextBackColor
    BRGLabel.ForeColor = TextForeColor
    BRGLabel.BackColor = TextBackColor
  End If
  CurBearing.Visible = Value
  BRGLabel.Visible = Value
End Sub

Private Sub ShowDTK_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    DesiredTrack.ForeColor = TextBackColor
    DesiredTrack.BackColor = TextForeColor
    DTKLabel.ForeColor = TextBackColor
    DTKLabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    DesiredTrack.ForeColor = TextForeColor
    DesiredTrack.BackColor = TextBackColor
    DTKLabel.ForeColor = TextForeColor
    DTKLabel.BackColor = TextBackColor
  End If
  DesiredTrack.Visible = Value
  DTKLabel.Visible = Value
End Sub

Private Sub ShowDTW_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    DisttoWP.ForeColor = TextBackColor
    DisttoWP.BackColor = TextForeColor
    DistLabel.ForeColor = TextBackColor
    DistLabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    DisttoWP.ForeColor = TextForeColor
    DisttoWP.BackColor = TextBackColor
    DistLabel.ForeColor = TextForeColor
    DistLabel.BackColor = TextBackColor
  End If
  DisttoWP.Visible = Value
  DistLabel.Visible = Value
End Sub

Private Sub ShowETE_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    TimeToWP.ForeColor = TextBackColor
    TimeToWP.BackColor = TextForeColor
    ETELabel.ForeColor = TextBackColor
    ETELabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    TimeToWP.ForeColor = TextForeColor
    TimeToWP.BackColor = TextBackColor
    ETELabel.ForeColor = TextForeColor
    ETELabel.BackColor = TextBackColor
  End If
  TimeToWP.Visible = Value
  ETELabel.Visible = Value
End Sub

Private Sub ShowFromWP_Click(Index As Integer,
Value As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    LastWP.ForeColor = TextBackColor
    LastWP.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    LastWP.ForeColor = TextForeColor
    LastWP.BackColor = TextBackColor
  End If
  LastWP.Visible = Value
End Sub

Private Sub ShowGS_Click(Index As Integer, Value As
Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    GrndSpeed.ForeColor = TextBackColor
    GrndSpeed.BackColor = TextForeColor
    GSLabel.ForeColor = TextBackColor
    GSLabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    GrndSpeed.ForeColor = TextForeColor
    GrndSpeed.BackColor = TextBackColor
    GSLabel.ForeColor = TextForeColor
    GSLabel.BackColor = TextBackColor
  End If
  GrndSpeed.Visible = Value
  GSLabel.Visible = Value
End Sub

Private Sub ShowTAE_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    TAEVal.ForeColor = TextBackColor
    TAEVal.BackColor = TextForeColor
    TAELabel.ForeColor = TextBackColor
    TAELabel.BackColor = TextForeColor
    TAEDirection(0).ForeColor = TextBackColor
    TAEDirection(0).BackColor = TextForeColor
    TAEDirection(1).ForeColor = TextBackColor
    TAEDirection(1).BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    TAEVal.ForeColor = TextForeColor
```

```
    TAEVal.BackColor = TextBackColor
    TAELabel.ForeColor = TextForeColor
    TAELabel.BackColor = TextBackColor
    TAEDirection(0).ForeColor = TextForeColor
    TAEDirection(0).BackColor = TextBackColor
    TAEDirection(1).ForeColor = TextForeColor
    TAEDirection(1).BackColor = TextBackColor
  End If
  TAEVal.Visible = Value
  TAELabel.Visible = Value
  TAEDirection(0).Visible = Value
  TAEDirection(1).Visible = Value
End Sub

Private Sub ShowToWP_Click(Index As Integer, Value
As Integer)
  NextWP.Visible = Value
  WPArrow.Visible = Value
  WPArrowI.Visible = Value
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    NextWP.ForeColor = TextBackColor
    NextWP.BackColor = TextForeColor
    If WPArrow.Visible Then WPArrow.Visible = False
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    NextWP.ForeColor = TextForeColor
    NextWP.BackColor = TextBackColor
    If WPArrowI.Visible Then WPArrowI.Visible =
False
  End If
End Sub

Private Sub ShowTRK_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    CurTrack.ForeColor = TextBackColor
    CurTrack.BackColor = TextForeColor
    TRKLabel.ForeColor = TextBackColor
    TRKLabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    CurTrack.ForeColor = TextForeColor
    CurTrack.BackColor = TextBackColor
    TRKLabel.ForeColor = TextForeColor
    TRKLabel.BackColor = TextBackColor
  End If
  CurTrack.Visible = Value
  TRKLabel.Visible = Value
End Sub

Private Sub ShowXTE_Click(Index As Integer, Value
As Integer)
  If (Index = 1 And Value) Then
'   Set Fore Color to Black and Back Color to
Yellow
    XTEVal.ForeColor = TextBackColor
    XTEVal.BackColor = TextForeColor
    XTELabel.ForeColor = TextBackColor
    XTELabel.BackColor = TextForeColor
  Else
'   Set Fore Color to Yellow and Back Color to
Black
    XTEVal.ForeColor = TextForeColor
    XTEVal.BackColor = TextBackColor
    XTELabel.ForeColor = TextForeColor
    XTELabel.BackColor = TextBackColor
  End If
  XTEVal.Visible = Value
  XTELabel.Visible = Value
End Sub

Private Sub SysSetCommand_Click()
  SystemSettings.Show MODELESS
End Sub

Private Sub Test_Click()
  Dim curR As Single
  Dim temp As GPSRec

  temp.curWPIdent = "CURN"
  temp.CurWPType = ""
  temp.CurWPLong = orgLong
  temp.CurWPLat = orgLat
  saveBRG = saveBRG + 1
  If saveBRG > 360 Then saveBRG = 0
  temp.Track = saveBRG
  temp.CurRadial = temp.Track - testTAE
  If temp.CurRadial < 0 Then temp.CurRadial =
temp.CurRadial + 360
  If temp.CurRadial > 360 Then temp.CurRadial =
temp.CurRadial - 360
  curR = Reciprocal(CDbl(temp.CurRadial))
  temp.nextWPIdent = "NEXT"
  temp.NextWPType = ""
  temp.NextRadial = curR
  temp.XAE = 0
  temp.XTE = testXTE
  temp.GAE = testGAE
  temp.CGA = RadtoDeg(Atn(-1600# / (5# * 6080#)))
  temp.AltErr = testALE
  temp.TrkDistToWP = 0.001
  temp.DisttoWP = Sqr(temp.TrkDistToWP ^ 2 +
testXTE ^ 2)
  temp.DistToEnd = temp.TrkDistToWP + FAFdist
  temp.TAE = testTAE
  temp.HdgtoWP = curR + RadtoDeg(Atn(testXTE /
temp.TrkDistToWP))
  If temp.HdgtoWP < 0 Then temp.HdgtoWP =
temp.HdgtoWP + 360
  If temp.HdgtoWP > 360 Then temp.HdgtoWP =
temp.HdgtoWP - 360
  temp.GrndSpeed = testGS
  If (temp.GrndSpeed > 0) Then
    temp.ETW = temp.DisttoWP * 3600 / testGS
  Else
    temp.ETW = 0
  End If
  temp.LastWPIdent = "LAST"
  temp.CDISensitivity = 1
  temp.CDIVal = testXTE * 100

  GPSRecord = temp
  ProcessData
End Sub

Private Sub TestALE_Change()
  SetDefaultCDI
End Sub

Private Sub TestGAE_Change()
  SetDefaultCDI
End Sub

Private Sub TestGS_Change()
  SetDefaultCDI
End Sub

Private Sub TestTAE_Change()
```

```
  SetDefaultCDI
End Sub

Private Sub TestXTE_Change()
  SetDefaultCDI
End Sub

Private Sub TransitionWP()
  Dim newWPIdent As String
  Dim resetOnce As Integer
  Dim startWP As Integer

' Transition using the next waypoint name because
the current name might be faked by
' the DME processing.

  newWPIdent = UCase$(Trim$(GPSRecord.nextWPIdent))
  resetOnce = False
  startWP = nextWPind
  nextWPIdent =
UCase$(Trim$(curFlightPlan.WayPnts(nextWPind).Ident
))
  Do While True
'   Check if this is the correct waypoint
    If (newWPIdent = nextWPIdent) Then
      Exit Do
    End If

'   Try the next waypoint
    nextWPind = nextWPind + 1
    If (nextWPind >= curFlightPlan.NumWayPnts) Then
      nextWPind = 1  ' WP = 0 is the first waypoint
      resetOnce = True
    End If

'   Can't find the correct waypoint so just stay
with the current one
    If (nextWPind = startWP) And resetOnce Then
Exit Do

'   Update the store of information about the
current waypoint
    curWPind = nextWPind - 1
    curWPIdent =
UCase$(Trim$(curFlightPlan.WayPnts(curWPind).Ident)
)
    curLong =
curFlightPlan.WayPnts(curWPind).Longitude
    curLat =
curFlightPlan.WayPnts(curWPind).Latitude
    nextWPIdent =
UCase$(Trim$(curFlightPlan.WayPnts(nextWPind).Ident
))
  Loop

End Sub

Private Sub TypeButton_Click()
  DisplayType = DisplayType + 1
  If DisplayType > 5 Then
    DisplayType = 1
  End If
  SetDisplayType DisplayType
End Sub
```

EXPT2.VBP

```
Form=EXPT2.FRM
Module=DATALINK; ..\COMMON\DATALINK.BAS
Module=VBDEF; ..\COMMON\VBDEF.BAS
Module=THREED1; ..\COMMON\THREED.BAS
Module=FLGTPLAN; ..\common\flgtplan.bas
Module=TRIGCODE; ..\COMMON\TRIGCODE.BAS
Form=..\COMMON\SYSSET.FRM
Module=ddelinks; Ddelinks.bas
Module=EXPT21; Expt2.bas
Object={F9043C88-F6F2-101A-A3C9-
08002B2F49FB}#1.0#0; COMDLG32.OCX
Object={B16553C3-06DB-101B-85B2-
0000C009BE81}#1.0#0; SPIN32.OCX
Object={0BA686C6-F7D3-101A-993E-
0000C0EF6F5E}#1.0#0; THREED32.OCX
Reference=*\G{00025E04-0000-0000-C000-
000000000046}#2.5#0#C:\PROGRAM FILES\COMMON
FILES\MICROSOFT SHARED\DAO\dao2532.tlb#Microsoft
DAO 2.5/3.0 Compatibility Library
ProjWinSize=27,482,236,397
ProjWinShow=2
IconForm="Main"
Title="Experiment Two"
ExeName32="EXPT2.exe"
ExeName="EXPT2.EXE"
Name="expt2"
HelpContextID="0"
StartMode=0
VersionCompatible32="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName="USDOT Volpe Center"
```

FLGTPLAN.BAS

```
Option Explicit

Global Const MaxFltPlnWayPnts = 30

Type WayPntRec
  Ident As String
  WPType As String
  Latitude As Double
  Longitude As Double
  Altitude As Double
  Airspeed As Single
  Radial As Double
  LegLen As Double
  NextRadial As Double
  ArcRadius As Single
End Type

Type FlightPlanRec
  NumWayPnts As Integer
  WayPnts(0 To MaxFltPlnWayPnts) As WayPntRec
  FltPlnLen As Double
  LongCoef As Double
  MagVariation As Double
End Type

Dim CtrlConDB As Database
Dim ExpSet As SnapShot
Dim ProfileSet As SnapShot
Dim FltPlnSet As SnapShot
Dim FltPlnWayPntSet As SnapShot
```

```
Dim WayPntTbl As SnapShot

Private Sub addWayPnt (FlightPlan As FlightPlanRec,
Ident As String, WPType As String, Latitude As
Double, Longitude As Double, Altitude As Double)
  Dim DeltaX As Double
  Dim DeltaY As Double
  Dim nWayPts As Integer
  Dim DMEpos As Integer

  nWayPts = FlightPlan.NumWayPnts

  FlightPlan.WayPnts(nWayPts).Ident = Trim$(Ident)
  FlightPlan.WayPnts(nWayPts).WPType =
Trim$(WPType)
  FlightPlan.WayPnts(nWayPts).Latitude = Latitude
  FlightPlan.WayPnts(nWayPts).Longitude = Longitude
  FlightPlan.WayPnts(nWayPts).Altitude = Altitude
  DMEpos = InStr(1, Trim$(WPType), "DME", 1)
  If DMEpos <> 0 Then
    FlightPlan.WayPnts(nWayPts).ArcRadius =
Val(Mid$(Trim$(WPType), DMEpos + 3))
  Else
    FlightPlan.WayPnts(nWayPts).ArcRadius = 0
  End If
  If nWayPts > 0 Then
    DeltaY = (FlightPlan.WayPnts(nWayPts).Latitude
- FlightPlan.WayPnts(nWayPts - 1).Latitude) * 60
    DeltaX = (FlightPlan.LongCoef *
(FlightPlan.WayPnts(nWayPts - 1).Longitude -
FlightPlan.WayPnts(nWayPts).Longitude)) * 60
    FlightPlan.WayPnts(nWayPts).Radial =
radialFromWP(DeltaY, DeltaX) / DtoR
    FlightPlan.WayPnts(nWayPts).LegLen = Sqr(DeltaY
* DeltaY + DeltaX * DeltaX)
    FlightPlan.FltPlnLen = FlightPlan.FltPlnLen +
FlightPlan.WayPnts(nWayPts).LegLen
  Else
    FlightPlan.LongCoef =
Cos(FlightPlan.WayPnts(nWayPts).Latitude * DtoR)
    FlightPlan.WayPnts(nWayPts).Radial = 0
    FlightPlan.WayPnts(nWayPts).LegLen = 0
    FlightPlan.FltPlnLen = 0
  End If
  FlightPlan.NumWayPnts = FlightPlan.NumWayPnts + 1
End Sub

Function checkNull (DataVal As Variant) As Variant
  If IsNull(DataVal) Then
    checkNull = ""
  Else
    checkNull = DataVal
  End If
End Function

Sub closeFlightPlan ()
  ExpSet.Close
  ProfileSet.Close
  FltPlnSet.Close
  FltPlnWayPntSet.Close
  WayPntTbl.Close
  CtrlConDB.Close
End Sub

Sub initFlightPlan (DBName As String)
  ' Open the tables

  Set CtrlConDB = OpenDatabase(DBName, False, True)
  Set ExpSet = CtrlConDB.CreateSnapshot("Select *
From Experiment")
  Set ProfileSet = CtrlConDB.CreateSnapshot("Select
* From Profile")
  Set FltPlnSet = CtrlConDB.CreateSnapshot("Select
* From FlightPlan")
  Set WayPntTbl = CtrlConDB.CreateSnapshot("Select
* From WayPoint")

End Sub

Sub openFlightPlan (FlightPlan As FlightPlanRec,
ExptName As String, ProfName As String, FlgtName As
String)
  Dim CurFltPlnID As Long

  CurFltPlnID = -1
  ExpSet.FindFirst "ExperimentName = """ + ExptName
+ """"
  If Not ExpSet.NoMatch Then
    ProfileSet.FindFirst "ExperimentID = " +
Format$(ExpSet("ExperimentID")) + " And ProfileName
= """ + ProfName + """"
    If Not ProfileSet.NoMatch Then
      FltPlnSet.FindFirst "ProfileID = " +
Format$(ProfileSet("ProfileID")) + " And
FlightPlanName = """ + FlgtName + """"
      If Not FltPlnSet.NoMatch Then
        CurFltPlnID = FltPlnSet("FlightPlanID")
      End If
    End If
  End If

  ' Load the magnetic variation from the database
  On Error Resume Next
  FlightPlan.MagVariation =
FltPlnSet("MagVariation")
  If (Err = 3018) Or (Err = 3265) Then
    FlightPlan.MagVariation = -16   ' 16 deg W
  Else
    MsgBox Error, MB_ICONSTOP
    Stop
  End If
  On Error GoTo 0

  Set FltPlnWayPntSet =
CtrlConDB.CreateSnapshot("Select * From
FltPlnWayPnt Where FlightPlanID = " +
Format$(CurFltPlnID) + " Order by Sequence")
  If Not (FltPlnWayPntSet.EOF And
FltPlnWayPntSet.BOF) Then
    FltPlnWayPntSet.MoveFirst
    FlightPlan.NumWayPnts = 0
    While Not FltPlnWayPntSet.EOF
      WayPntTbl.FindFirst "WayPntName = """ +
FltPlnWayPntSet("WayPntName") + """"
      If Not WayPntTbl.NoMatch Then
        addWayPnt FlightPlan,
CStr(checkNull(FltPlnWayPntSet("WayPntName"))),
CStr(checkNull(FltPlnWayPntSet("WayPntType"))),
Val(WayPntTbl("Latitude")),
Val(WayPntTbl("Longitude")),
Val(FltPlnWayPntSet("Altitude"))
      End If
      FltPlnWayPntSet.MoveNext
    Wend
  End If
End Sub
```

```
VERSION 2.00
Begin Form Main
   BackColor       =   &H00C0C0C0&
   Caption         =   "FRASCA Control Console
Version 1.20"
   ClientHeight    =   7260
   ClientLeft      =   75
   ClientTop       =   465
   ClientWidth     =   12255
   Height          =   7665
   Icon            =   GPSSERIA.FRX:0000
   KeyPreview      =   -1  'True
   Left            =   15
   LinkMode        =   1  'Source
   LinkTopic       =   "Main"
   ScaleHeight     =   7260
   ScaleWidth      =   12255
   Top             =   120
   Width           =   12375
   Begin dsSocket GPSDataSock
      DataSize        =   2048
      EOLChar         =   0
      Index           =   0
      Left            =   9150
      LineMode        =   0   'False
      Linger          =   0   'False
      LocalPort       =   0
      RemoteDotAddr   =   ""
      RemoteHost      =   ""
      RemotePort      =   0
      ServiceName     =   ""
      Timeout         =   10
      Top             =   6570
   End
   Begin dsSocket ListenSock
      DataSize        =   2048
      EOLChar         =   0
      Left            =   8670
      LineMode        =   0   'False
      Linger          =   0   'False
      LocalPort       =   15001
      RemoteDotAddr   =   ""
      RemoteHost      =   ""
      RemotePort      =   0
      ServiceName     =   ""
      Timeout         =   10
      Top             =   6570
   End
   Begin TextBox TimerValue
      Height          =   285
      Left            =   5130
      TabIndex        =   55
      Text            =   "1000"
      Top             =   6300
      Width           =   735
   End
   Begin dsSocket DasDataSock
      DataSize        =   2048
      EOLChar         =   0
      Left            =   8190
      LineMode        =   0   'False
      Linger          =   0   'False
      LocalPort       =   0
      RemoteDotAddr   =   "152.122.8.167"
      RemoteHost      =   ""
      RemotePort      =   15000
      ServiceName     =   ""
      Timeout         =   10
      Top             =   6570
   End
   Begin PictureBox MyDebug
      FontBold        =   0   'False
      FontItalic      =   0   'False
      FontName        =   "MS Serif"
      FontSize        =   6.75
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   828
      Left            =   9750
      ScaleHeight     =   795
      ScaleWidth      =   1545
      TabIndex        =   70
      Top             =   6360
      Visible         =   0   'False
      Width           =   1572
   End
   Begin CommandButton Reconnect
      Caption         =   "&Reconnect"
      Height          =   372
      Left            =   120
      TabIndex        =   68
      Top             =   6360
      Width           =   1452
   End
   Begin Timer ConnectTimer
      Enabled         =   0   'False
      Interval        =   1000
      Left            =   5520
      Top             =   6600
   End
   Begin SSCommand SysSet
      Caption         =   "S&ystem Settings"
      Height          =   372
      Left            =   1680
      TabIndex        =   54
      Top             =   6360
      Width           =   1452
   End
   Begin SSCommand Quit
      Caption         =   "&Quit"
      Height          =   372
      Left            =   3240
      TabIndex        =   53
      Top             =   6360
      Width           =   1452
   End
   Begin SSPanel TitlePanel
      BevelOuter      =   0  'None
      Caption         =   "GPS Module"
      FontBold        =   -1  'True
      FontItalic      =   0   'False
      FontName        =   "MS Sans Serif"
      FontSize        =   12
      FontStrikethru  =   0   'False
      FontUnderline   =   0   'False
      Height          =   1815
      Left            =   7200
      TabIndex        =   0
      Top             =   120
      Width           =   6495
   End
   Begin SSPanel DataSrvrPanel
      Alignment       =   0  'Left Justify - TOP
      BevelInner      =   1  'Inset
      BevelOuter      =   0  'None
      Caption         =   "Data Server"
```

```
      Height          =   852
      Left            =   0
      TabIndex        =   1
      Top             =   2040
      Width           =   13692
      Begin Label Label18
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Data Server Status:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   2
         Top             =   360
         Width           =   1695
      End
      Begin Label Label17
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Data String:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   3
         Top             =   600
         Width           =   1035
      End
      Begin Label DataSrvrData
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "DataSrvrData"
         Height          =   195
         Left            =   1920
         LinkItem        =   "DataSrvrData"
         LinkTopic       =
"\\DATASERVER\NDDE$|DATASRVR$"
         TabIndex        =   4
         Top             =   600
         Width           =   1215
      End
      Begin Label DataSrvrStatus
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "DataSrvrStatus"
         Height          =   195
         Left            =   1920
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   5
         Top             =   360
         Width           =   1305
      End
   End
   Begin SSPanel GPSPanel
      Alignment       =   0  'Left Justify - TOP
      BevelInner      =   1  'Inset
      BevelOuter      =   0  'None
      Caption         =   "GPS Module"
      Height          =   1332
      Left            =   0
      TabIndex        =   6
      Top             =   2880
      Width           =   13692
      Begin Label NumTCPConnections
         Alignment       =   1  'Right Justify
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "0"
         Height          =   195
         Left            =   5940
         TabIndex        =   73
         Top             =   60
         Width           =   120
      End
      Begin Label TCPLinkID
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Unidentified"
         FontBold        =   0   'False
         FontItalic      =   0   'False
         FontName        =   "MS Serif"
         FontSize        =   6.75
         FontStrikethru  =   0   'False
         FontUnderline   =   0   'False
         Height          =   168
         Index           =   0
         Left            =   4536
         TabIndex        =   72
         Top             =   276
         Visible         =   0   'False
         Width           =   876
      End
      Begin Label Label28
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "TCP Links"
         Height          =   195
         Left            =   4080
         TabIndex        =   71
         Top             =   60
         Width           =   885
      End
      Begin Label Label22
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Data String:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   7
         Top             =   1080
         Width           =   1035
      End
      Begin Label Label21
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Status:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   8
         Top             =   840
         Width           =   615
      End
      Begin Label Label20
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Parameters:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   9
         Top             =   600
```

```
      Width           =   1020
   End
   Begin Label Label19
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Name:"
      Height          =   195
      Left            =   120
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   10
      Top             =   360
      Width           =   555
   End
   Begin Label GPSData
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "GPSData"
      Height          =   195
      Left            =   1320
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   11
      Top             =   1080
      Width           =   795
   End
   Begin Label GPSStatus
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "GPSStatus"
      Height          =   195
      Left            =   1320
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   12
      Top             =   840
      Width           =   930
   End
   Begin Label GPSParam
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "GPSParam"
      Height          =   195
      Left            =   1320
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   13
      Top             =   600
      Width           =   915
   End
   Begin Label GPSName
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "GPSName"
      Height          =   195
      Left            =   1320
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   14
      Top             =   360
      Width           =   870
   End
End
Begin SSPanel NavPanel
   Alignment       =   0  'Left Justify - TOP
   BevelInner      =   1  'Inset
   BevelOuter      =   0  'None
   Caption         =   "Navigation Data"
   Height          =   2052
   Left            =   0
   TabIndex        =   15
   Top             =   4200
   Width           =   6372
   Begin SSCheck TurnAnticipation
      Caption         =   "Use Turn
Anticipation"
      Height          =   312
      Left            =   3180
      TabIndex        =   69
      Top             =   60
      Value           =   -1  'True
      Width           =   2832
   End
   Begin Label Label29
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "ETW:"
      Height          =   192
      Left            =   3840
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   67
      Top             =   1680
      Width           =   492
   End
   Begin Label ETW
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "ETW"
      Height          =   192
      Left            =   5160
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   66
      Top             =   1680
      Width           =   432
   End
   Begin Label GrndSpeed
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "GrndSpeed"
      Height          =   192
      Left            =   5160
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   65
      Top             =   1440
      Width           =   960
   End
   Begin Label Label25
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Ground Speed:"
      Height          =   192
      Left            =   3840
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   64
      Top             =   1440
      Width           =   1296
   End
   Begin Label Label27
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Current Track:"
```

```
         Height          =   192
         Left            =   3840
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   52
         Top             =   960
         Width           =   1248
      End
      Begin Label Label26
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "TAE:"
         Height          =   192
         Left            =   3840
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   58
         Top             =   1200
         Width           =   432
      End
      Begin Label TAE
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "TAE"
         Height          =   192
         Left            =   5160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   57
         Top             =   1200
         Width           =   372
      End
      Begin Label Track
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "Track"
         Height          =   192
         Left            =   5160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   56
         Top             =   960
         Width           =   516
      End
      Begin Label WayPnt
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "WayPnt"
         Height          =   195
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   16
         Top             =   360
         Width           =   675
      End
      Begin Label Radial
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "Radial"
         Height          =   195
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   17
         Top             =   840
         Width           =   555
      End
      Begin Label HdgToWayPnt
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "HdgToWayPnt"
         Height          =   195
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   18
         Top             =   1080
         Width           =   1245
      End
      Begin Label DistToWayPnt
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "DistToWayPnt"
         Height          =   195
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   19
         Top             =   1320
         Width           =   1230
      End
      Begin Label XAE
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "XAE"
         Height          =   192
         Left            =   5160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   20
         Top             =   480
         Width           =   372
      End
      Begin Label XTE
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "XTE"
         Height          =   192
         Left            =   5160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   21
         Top             =   720
         Width           =   372
      End
      Begin Label DistToEnd
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "DistToEnd"
         Height          =   195
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   22
         Top             =   1800
         Width           =   900
      End
      Begin Label WayPntType
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "WayPntType"
         Height          =   195
```

```
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   23
         Top             =   600
         Width           =   1095
      End
      Begin Label TrkDistToWayPnt
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "TrkDistToWayPnt"
         Height          =   195
         Left            =   2160
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   24
         Top             =   1560
         Width           =   1515
      End
      Begin Label Label1
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Current Waypoint
Type:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   25
         Top             =   600
         Width           =   2025
      End
      Begin Label Label2
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Distance to End:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   26
         Top             =   1800
         Width           =   1440
      End
      Begin Label Label3
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "XTE:"
         Height          =   192
         Left            =   3840
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   27
         Top             =   720
         Width           =   432
      End
      Begin Label Label4
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "XAE:"
         Height          =   192
         Left            =   3840
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   28
         Top             =   480
         Width           =   432
      End
      Begin Label Label5
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Dist. to Waypoint:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   29
         Top             =   1320
         Width           =   1545
      End
      Begin Label Label6
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Heading to Waypoint:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   30
         Top             =   1080
         Width           =   1860
      End
      Begin Label Label7
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Current Radial:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   31
         Top             =   840
         Width           =   1290
      End
      Begin Label Label8
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Current Waypoint:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   32
         Top             =   360
         Width           =   1545
      End
      Begin Label Label9
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "Track Dist.to
Waypoint:"
         Height          =   195
         Left            =   120
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   33
         Top             =   1560
         Width           =   2040
      End
   End
   Begin SSPanel InstSrvrPanel
      Alignment       =   0  'Left Justify - TOP
      BevelInner      =   1  'Inset
```

```
      BevelOuter      =   0  'None
      Caption         =   "Instrument Server"
      Height          =   2052
      Left            =   6360
      TabIndex        =   34
      Top             =   4200
      Width           =   3972
      Begin TextBox CDISens
         BackColor       =   &H00C0C0C0&
         Height          =   285
         Left            =   2280
         TabIndex        =   35
         Text            =   "CDISens"
         Top             =   600
         Width           =   1575
      End
      Begin TextBox CDIVal
         BackColor       =   &H00C0C0C0&
         Height          =   285
         Left            =   2280
         TabIndex        =   36
         Text            =   "CDIVal"
         Top             =   960
         Width           =   1575
      End
      Begin Label Label23
         AutoSize        =   -1  'True
         BackColor       =   &H00000000&
         BackStyle       =   0  'Transparent
         Caption         =   "LCD Sub-panel Data"
         Height          =   195
         Left            =   240
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   37
         Top             =   1320
         Width           =   1755
      End
      Begin Label Label10
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "CDI Value"
         Height          =   195
         Left            =   240
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   38
         Top             =   960
         Width           =   870
      End
      Begin Label Label11
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "CDI Sensitivity"
         Height          =   195
         Left            =   240
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   39
         Top             =   600
         Width           =   1260
      End
      Begin Label LCDDisplayString
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "LCDDisplayString"
         Height          =   195
         Left            =   2280
         TabIndex        =   40
         Top             =   1320
         Width           =   1485
      End
      Begin Label Label12
         AutoSize        =   -1  'True
         BackStyle       =   0  'Transparent
         Caption         =   "HSI To/From Flag"
         Height          =   195
         Left            =   240
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   41
         Top             =   360
         Width           =   1530
      End
      Begin Label HSIToFromFlag
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "HSIToFromFlag"
         Height          =   195
         Left            =   2280
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   42
         Top             =   360
         Width           =   1320
      End
   End
   Begin SSPanel CtrlConPanel
      Alignment       =   0  'Left Justify - TOP
      BevelInner      =   1  'Inset
      BevelOuter      =   0  'None
      Caption         =   "Control Console"
      Height          =   2055
      Left            =   0
      TabIndex        =   43
      Top             =   0
      Width           =   7215
      Begin SSPanel Panel3D3
         Alignment       =   1  'Left Justify -
MIDDLE
         BevelInner      =   1  'Inset
         BevelOuter      =   0  'None
         Caption         =   " Instrument Server"
         Height          =   615
         Left            =   4560
         TabIndex        =   63
         Top             =   1320
         Width           =   2415
         Begin Shape InstSrvrInd
            FillColor       =   &H000000FF&
            FillStyle       =   0  'Solid
            Height          =   375
            Left            =   1920
            Shape           =   3  'Circle
            Top             =   120
            Width           =   375
         End
      End
      Begin SSPanel Panel3D2
         Alignment       =   1  'Left Justify -
MIDDLE
         BevelInner      =   1  'Inset
         BevelOuter      =   0  'None
         Caption         =   " Control Console"
         Height          =   615
         Left            =   4560
         TabIndex        =   62
         Top             =   720
         Width           =   2415
```

```
      Begin Shape CtrlConInd
         FillColor       =   &H000000FF&
         FillStyle       =   0  'Solid
         Height          =   375
         Left            =   1920
         Shape           =   3  'Circle
         Top             =   120
         Width           =   375
      End
   End
   Begin SSPanel Panel3D1
      Alignment       =   1  'Left Justify -
MIDDLE
      BevelInner      =   1  'Inset
      BevelOuter      =   0  'None
      Caption         =   " Data Server"
      Height          =   615
      Left            =   4560
      TabIndex        =   61
      Top             =   120
      Width           =   2415
      Begin Shape DasTCPInd
         FillColor       =   &H000000FF&
         FillStyle       =   0  'Solid
         Height          =   375
         Left            =   1440
         Shape           =   3  'Circle
         Top             =   120
         Width           =   375
      End
      Begin Shape DataSrvrInd
         FillColor       =   &H000000FF&
         FillStyle       =   0  'Solid
         Height          =   375
         Left            =   1920
         Shape           =   3  'Circle
         Top             =   120
         Width           =   375
      End
   End
   Begin Label Mode
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "Mode"
      Height          =   195
      Left            =   2280
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   60
      Top             =   480
      Width           =   480
   End
   Begin Label Label24
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Mode:"
      Height          =   195
      Left            =   120
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   59
      Top             =   480
      Width           =   540
   End
   Begin Label Label16
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Control Console
Status:"
      Height          =   195
      Left            =   120
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   44
      Top             =   1440
      Width           =   2010
   End
   Begin Label Label15
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Flight Plan Name:"
      Height          =   195
      Left            =   120
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   45
      Top             =   1200
      Width           =   1515
   End
   Begin Label Label14
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Profile Name:"
      Height          =   195
      Left            =   120
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   46
      Top             =   960
      Width           =   1155
   End
   Begin Label Label13
      AutoSize        =   -1  'True
      BackStyle       =   0  'Transparent
      Caption         =   "Experiment Name:"
      Height          =   195
      Left            =   120
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   47
      Top             =   720
      Width           =   1545
   End
   Begin Label ExpName
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "ExpName"
      Height          =   195
      Left            =   2280
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   48
      Top             =   720
      Width           =   810
   End
   Begin Label ProfileName
      AutoSize        =   -1  'True
      BackColor       =   &H00C0C0C0&
      Caption         =   "ProfileName"
      Height          =   195
      Left            =   2280
      LinkItem        =   "AllData"
      LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
      TabIndex        =   49
      Top             =   960
      Width           =   1035
```

```
      End
      Begin Label FltPlnName
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "FltPlnName"
         Height          =   195
         Left            =   2280
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   50
         Top             =   1200
         Width           =   975
      End
      Begin Label CtrlConStatus
         AutoSize        =   -1  'True
         BackColor       =   &H00C0C0C0&
         Caption         =   "CtrlConStatus"
         Height          =   195
         Left            =   2280
         LinkItem        =   "AllData"
         LinkTopic       =
"\\FRASCA_CONSOLE\NDDE$|FDCS$"
         TabIndex        =   51
         Top             =   1440
         Width           =   1170
      End
   End
   Begin Timer GPSTimer
      Enabled         =   0   'False
      Interval        =   1000
      Left            =   5040
      Top             =   6600
   End
   Begin Shape Shape2
      FillColor       =   &H0000C000&
      FillStyle       =   0  'Solid
      Height          =   372
      Left            =   7440
      Top             =   6480
      Width           =   372
   End
   Begin Shape Shape1
      FillColor       =   &H0000C000&
      FillStyle       =   0  'Solid
      Height          =   372
      Left            =   6960
      Top             =   6480
      Width           =   372
   End
End
Option Explicit

Dim CurNavData As NavRec
Dim CurDataSrvrData As DataSrvrRec
Dim nNumTCPConnections As Integer
Dim DasOutBuffer As IDSockBuffer
Dim GPSOutBuffer() As IDSockBuffer

Sub ConnectTimer_Timer ()
  If Not DasDDELinkOpen Then
    ConnectDataSrvr
  End If
  DoEvents

  If Not InsDDELinkOpen Then
    ConnectInstSrvr
  End If
  DoEvents

  If Not ConDDELinkOpen Then
    ConnectCtrlCon
    If ConDDELinkOpen Then
      FltPlnName_Change
      GPSParam_Change
      CtrlConStatus_Change
    End If
  End If
  DoEvents

  ConnectTimer.Enabled = False'Not (ConDDELinkOpen
And INSDDELinkOpen And DasDDELinkOpen)

End Sub

Sub CtrlConStatus_Change ()
  Dim i As Integer

  Select Case Trim$(UCase$(CtrlConStatus.Caption))
    Case "RUNNING"
      ResetNav CurNavData
'       If DasDDELinkOpen Then
'         Main.DataSrvrData.LinkMode =
LINK_AUTOMATIC
'       End If
'       GPSTimer.Enabled = True
    Case "STOPPED"
'       GPSTimer.Enabled = False
'       If DasDDELinkOpen Then
'         Do
'           On Error GoTo 0
'           On Error Resume Next
'           Main.DataSrvrData.LinkMode = LINK_MANUAL
'         Loop Until Err = 0
'       End If
      ResetNav CurNavData
    Case "SHUTDOWN"
    ' i = ExitWindows(0, 0)
  End Select

End Sub

Sub CtrlConStatus_LinkError (LinkErr As Integer)
  ConDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub DasDataSock_Close (ErrorCode As Integer,
ErrorDesc As String)
'  Debug.Print "Close" + " " +
Format$(DasDataSock.State)
  DasTCPLinkOpen = False
  DasTCPInd.FillColor = &HFF&
  ConnectTimer.Enabled = True
End Sub

Sub DasDataSock_Connect ()
'  Debug.Print "Connect" + " " +
Format$(DasDataSock.State)
  DasTCPLinkOpen = True
'  DasTCPSendReady = False
  DasTCPInd.FillColor = &HFF00&
  DasOutBuffer.ID = 0
End Sub

Sub DasDataSock_Exception (ErrorCode As Integer,
ErrorDesc As String)
  MsgBox "FDS Socket Error - " + ErrorDesc, (MB_OK
+ MB_ICONSTOP), "GPS Error"

' Close the link as a dumb default thing to do
  If (DasDataSock.State <> 1) Then
DasDataSock.Action = 1
  DasTCPLinkOpen = False
```

```
  DasTCPInd.FillColor = &HFF&
  ConnectTimer.Enabled = True
End Sub

Sub DasDataSock_Receive (ReceiveData As String)
  Dim RawBuffer As SocketBuffer
  Dim MsgBuffer As IDSockBuffer

  RawBuffer.buffer = ReceiveData
  LSet MsgBuffer = RawBuffer
  Select Case MsgBuffer.ID
    Case DSMSG_DATA
      LSet CurDataSrvrData = MsgBuffer.buffer
      ProcessData
    Case IDMSG_REQUEST
      RawBuffer.buffer = "GPS Module"
      DasOutBuffer.ID = IDMSG_REPLY
      DasOutBuffer.buffer = RawBuffer
      DasDataSock_SendReady
'      MyDebug.Print "ID sent"
  End Select
End Sub

Sub DasDataSock_SendReady ()
  Dim RawBuffer As SocketBuffer

  If DasOutBuffer.ID <> 0 Then
    On Error Resume Next
    LSet RawBuffer = DasOutBuffer
    DasDataSock.Send = Trim$(RawBuffer.buffer)
    If Err = 0 Then
      DasOutBuffer.ID = 0
    End If
    On Error GoTo 0
  End If
End Sub

Sub DataSrvrData_LinkError (LinkErr As Integer)
'  DasDDELinkOpen = False
'  ConnectTimer.Enabled = True
End Sub

Sub DataSrvrStatus_LinkError (LinkErr As Integer)
  DasDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub ExpName_LinkError (LinkErr As Integer)
  InsDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub FltPlnName_Change ()
'  Dim Lat1 As Double
'  Dim Long1 As Double
'  Dim Lat2 As Double
'  Dim Long2 As Double
'  Dim DeltaX As Double
'  Dim DeltaY As Double

  InitNav CurNavData
  openFlightPlan CurNavData.FlightPlan,
Format$(ExpName.Caption),
Format$(ProfileName.Caption),
Format$(FltPlnName.Caption)
  ResetNav CurNavData

  CurNavData.NavMode = NAV_FLTPLN
End Sub

Sub FltPlnName_LinkError (LinkErr As Integer)
  InsDDELinkOpen = False
```

```
  ConnectTimer.Enabled = True
End Sub

Sub Form_Load ()
' Initialise the trigonometry package
  initTrig

' Load the System settings form to grab the comms
info
  Load SystemSettings

' Setup the geometry of the main form
  Main.Move ((Screen.Width - Main.Width) / 2),
((Screen.Height - Main.Height) / 2), Main.Width,
Main.Height
  TitlePanel.Width = Main.ScaleWidth -
CtrlConPanel.Width
  DataSrvrPanel.Width = Main.ScaleWidth
  GPSPanel.Width = Main.ScaleWidth
  NavPanel.Width = Main.ScaleWidth \ 2
  InstSrvrPanel.Move (NavPanel.Width + 1),
InstSrvrPanel.Top, NavPanel.Width,
InstSrvrPanel.Height
  Main.Caption = App.EXEName
  CDISens.Text = "1.0"
  Main.Show 0
  DoEvents

' Open the Flight plan database
  initFlightPlan
Format$(SystemSettings.CtrlConDB.Text)

' Open the serial port
  Serial.Show 1

  ConDDELinkOpen = False
  DasDDELinkOpen = False
  InsDDELinkOpen = False
  ConnectTimer.Enabled = True
  GPSTimer.Enabled = True

  DoEvents

' Init current flight plan
  FltPlnName_Change
  GPSParam_Change
  CtrlConStatus_Change

' Init current profile


'This is just to test timer val
'TimerValue = GPSTimer.Interval

' Open the TCP connection
  DasDataSock.RemoteDotAddr =
SystemSettings.DataSrvrIP
  DasDataSock.RemotePort =
Val(SystemSettings.DataSrvrPort)
  If (DasDataSock.State <> 2) Then
DasDataSock.Action = 2
  If Err Then
    MsgBox "Socket Error - " + Str(Err), (MB_OK +
MB_ICONSTOP + MB_SYSTEMMODAL), "GPS Error"
  End If

' Start listening for connections from others
  ListenSock.Action = 3
  If Err Then
    MsgBox "Socket Error - " + Str(Err), (MB_OK +
MB_ICONSTOP + MB_SYSTEMMODAL), "GPS Error"
  End If
```

```
End Sub


Sub Form_Resize ()
  TitlePanel.Width = Main.ScaleWidth -
CtrlConPanel.Width
  DataSrvrPanel.Width = Main.ScaleWidth
  GPSPanel.Width = Main.ScaleWidth
  NavPanel.Width = Main.ScaleWidth \ 2
  InstSrvrPanel.Move (NavPanel.Width + 1),
InstSrvrPanel.Top, NavPanel.Width,
InstSrvrPanel.Height
End Sub

Sub Form_Unload (Cancel As Integer)
' Ignore any errors
  On Error Resume Next

' Disconnect the TCP connection
  If (DasDataSock.State <> 1) Then
DasDataSock.Action = 1

  If ConDDELinkOpen Then
    GPSStatus.Caption = ""
  End If

  DoEvents

  closeFlightPlan
  End
End Sub

Sub GPSData_LinkError (LinkErr As Integer)
  ConDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub GPSDataSock_Close (Index As Integer, ErrorCode
As Integer, ErrorDesc As String)
   If Index <> 0 Then
    Unload GPSDataSock(Index)
    Unload TCPLinkID(Index)
   Else
     TCPLinkID(Index).Visible = False
   End If
   nNumTCPConnections = nNumTCPConnections - 1
   NumTCPConnections = Str$(nNumTCPConnections)
End Sub

Sub GPSDataSock_Exception (Index As Integer,
ErrorCode As Integer, ErrorDesc As String)
  MsgBox "GPS Socket Error - " + ErrorDesc, (MB_OK
+ MB_ICONSTOP), "GPS Error"
End Sub

Sub GPSDataSock_Receive (Index As Integer,
ReceiveData As String)
  Dim RawBuffer As SocketBuffer
  Dim MsgBuffer As IDSockBuffer

  RawBuffer.buffer = ReceiveData
  LSet MsgBuffer = RawBuffer
  Select Case MsgBuffer.ID
    Case GPSCMD_GETDATA
'     Send the current data in response to this
request
      GPSOutBuffer(Index).ID = GPSMSG_DATA
      LSet GPSOutBuffer(Index).buffer =
CurNavData.GPSData
      GPSDataSock_SendReady Index
    Case IDMSG_REPLY
'     Save the ID String to the link identifier
      RawBuffer = MsgBuffer.buffer
      TCPLinkID(Index) = RawBuffer.buffer
      'MyDebug.Print "ID received"
  End Select
End Sub

Sub GPSDataSock_SendReady (Index As Integer)
  Dim RawBuffer As SocketBuffer

  If GPSOutBuffer(Index).ID <> 0 Then
    On Error Resume Next
    LSet RawBuffer = GPSOutBuffer(Index)
    GPSDataSock(Index).Send =
Trim$(RawBuffer.buffer)
    If Err = 0 Then
      GPSOutBuffer(Index).ID = 0
    End If
    On Error GoTo 0
  End If
End Sub

Sub GPSName_LinkError (LinkErr As Integer)
  ConDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub GPSParam_Change ()
  GPSStatus.Caption = "Running-Sens. " +
Format$(Val(Mid$(GPSParam.Caption, 7)))
End Sub

Sub GPSParam_LinkError (LinkErr As Integer)
  ConDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub GPSStatus_LinkError (LinkErr As Integer)
  ConDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub GPSTimer_Timer ()
  Dim RawBuffer As SocketBuffer

  If Trim$(UCase$(CtrlConStatus.Caption)) =
"RUNNING" Then
    RawBuffer.buffer = ""
    DasOutBuffer.ID = DSCMD_GETDATA
    DasOutBuffer.buffer = RawBuffer
    DasDataSock_SendReady
  Else
    CtrlConStatus_Change
  End If
End Sub


Sub LCDDisplayString_LinkError (LinkErr As Integer)
  InsDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub ListenSock_Accept (SocketID As Integer)
  Dim NewSocketIndex As Integer
  Dim RawBuffer As SocketBuffer
  Dim MsgBuffer As IDSockBuffer

  NewSocketIndex = 0
  If GPSDataSock(NewSocketIndex).State <> 1 Then
    On Error Resume Next
    Do
      NewSocketIndex = NewSocketIndex + 1
```

```
      Load GPSDataSock(NewSocketIndex)
    Loop Until Err = 0
    On Error GoTo 0
    Load TCPLinkID(NewSocketIndex)
  End If
  If (NewSocketIndex = 0) Then
    TCPLinkID(NewSocketIndex).Top = 270
  Else
    TCPLinkID(NewSocketIndex).Top =
TCPLinkID(NewSocketIndex - 1).Top +
TCPLinkID(NewSocketIndex - 1).Height + 5
  End If
  TCPLinkID(NewSocketIndex).Visible = True
  TCPLinkID(NewSocketIndex).Caption =
"Unidentified"
  GPSDataSock(NewSocketIndex).Socket = SocketID
  nNumTCPConnections = nNumTCPConnections + 1
  NumTCPConnections = Str$(nNumTCPConnections)
  ReDim Preserve GPSOutBuffer(nNumTCPConnections)
  DoEvents

  If (GPSDataSock(NewSocketIndex).State = 2) Then
'   Send a request for the ID of this link
    RawBuffer.buffer = ""
    GPSOutBuffer(NewSocketIndex).ID = IDMSG_REQUEST
    GPSOutBuffer(NewSocketIndex).buffer = RawBuffer
    GPSDataSock_SendReady NewSocketIndex
    MyDebug.Print "ID requested"
  End If
End Sub

Sub Mode_LinkError (LinkErr As Integer)
  InsDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub ProcessData ()
  Dim i As Integer
  Dim j As Integer
  Dim k As Integer
  Dim TempData As String
  Dim DirIndicator As String
  Dim CurTime As Long

  DataSrvrData.Caption =
Format$(CurDataSrvrData.TimeStamp) + " " +
Format$(CurDataSrvrData.TimeStamp -
CurNavData.TimeStamp)
  UpdateNav CurNavData, CurDataSrvrData.Latitude,
CurDataSrvrData.Longitude,
CurDataSrvrData.Altitude, CurDataSrvrData.TimeStamp

  WayPnt.Caption = CurNavData.GPSData.CurWPIdent
  WayPntType.Caption = CurNavData.GPSData.CurWPType
  Radial.Caption =
Format$(CurNavData.GPSData.CurRadial, "0.00")
  HdgToWayPnt.Caption =
Format$(CurNavData.GPSData.HdgtoWP, "0.00")
  DistToWayPnt.Caption =
Format$(CurNavData.GPSData.DisttoWP, "0.00")
  TrkDistToWayPnt.Caption =
Format$(CurNavData.GPSData.TrkDistToWP, "0.00")
  DistToEnd.Caption =
Format$(CurNavData.GPSData.DistToEnd, "0.00")
  XAE.Caption = Format$(CurNavData.GPSData.XAE,
"0.00")
  XTE.Caption = Format$(CurNavData.GPSData.XTE,
"0.00")
  Track.Caption = Format$(CurNavData.GPSData.Track,
"0.00")
  TAE.Caption = Format$(CurNavData.GPSData.TAE,
"0.00")
  GrndSpeed.Caption =
Format$(CurNavData.GPSData.GrndSpeed, "0.00")
  ETW.Caption = Format$(CurNavData.GPSData.ETW,
"0.00")
  TempData = MakeGPSStr(CurNavData.GPSData)
  GPSData.Caption = TempData
  If ConDDELinkOpen Then
    On Error Resume Next
    GPSData.LinkPoke
    GPSStatus.LinkPoke
    If Err <> 0 Then
      ConDDELinkOpen = False
      ConnectTimer.Enabled = True
    End If
    On Error GoTo 0
  End If
  CDIVal.Text = Format$(CurNavData.GPSData.CDIVal,
"0.00")
  CDISens.Text =
Format$(CurNavData.GPSData.CDISensitivity, "0.00")
  LCDDisplayString.Caption = TempData
  If InsDDELinkOpen Then
    On Error Resume Next
    CDIVal.LinkPoke
    LCDDisplayString.LinkPoke
    If Err <> 0 Then
      InsDDELinkOpen = False
      ConnectTimer.Enabled = True
    End If
    On Error GoTo 0
  End If

' Write the serial data
  WriteSerialData CurNavData
End Sub

Sub ProfileName_LinkError (LinkErr As Integer)
  InsDDELinkOpen = False
  ConnectTimer.Enabled = True
End Sub

Sub Quit_Click ()
  Unload Main
End Sub

Sub Reconnect_Click ()
  Main.CtrlConInd.FillColor = &HFF&
  Main.DataSrvrInd.FillColor = &HFF&
  Main.InstSrvrInd.FillColor = &HFF&
  ConDDELinkOpen = False
  DasDDELinkOpen = False
  InsDDELinkOpen = False
  ConnectTimer.Enabled = True

  DoEvents
End Sub

Sub SysSet_Click ()
  SystemSettings.Show MODELESS
End Sub

Sub TimerValue_Change ()
  GPSTimer.Interval = TimerValue
End Sub
```

```
GPSSERIA.FRM
GPS_PROC.BAS
C:\WINDOWS\SYSTEM\CMDIALOG.VBX
GPS_NAV.BAS
C:\WINDOWS\SYSTEM\THREED.VBX
..\COMMON\VBDEF.BAS
C:\WINDOWS\SYSTEM\DSSOCK.VBX
..\COMMON\FLGTPLAN.BAS
..\COMMON\TRIGCODE.BAS
..\COMMON\DATALINK.BAS
C:\WINDOWS\SYSTEM\MSCOMM.VBX
SERIAL.BAS
..\COMMON\SYSSET.FRM
..\COMMON\SERIAL.FRM
ProjWinSize=22,482,236,397
ProjWinShow=2
IconForm="Main"
Title="GPS Serial"
ExeName="GPSSERIA.EXE"
```

GPS_NAV.BAS

```
Option Explicit

Const ZERO = 1E-20

Global Const NAV_OFF = 0
Global Const NAV_FLTPLN = 1

Type NavRec
  TimeStamp As Long
  NavMode As Integer
  FlightPlan As FlightPlanRec
  CurWayPntIndex As Integer
  Latitude As Double
  Longitude As Double
  Altitude As Double
  DeltaX As Double
  DeltaY As Double
  DeltaAlt As Double
  TrkDeltaX As Double
  TrkDeltaY As Double
  TrkDeltaAlt As Double
'  FltPlnLen As Double
  CurFltPlnOffset As Double
  GPSData As GPSRec
  LastRadial As Double
  LCRadial As Double
  CLRadial As Double
  NextRadial As Double
  LastRadius As Double
  LastBisect As Double
  LastWPLat As Double
  LastWPLong As Double
  LastWPAlt As Double
  CurBisect As Double
  CurWPAlt As Double
  DistHist(5) As Double
  TimeHist(5) As Double
End Type

Dim TurnRate As Double

Function CDISensitivity (NavData As NavRec) As
Single
  Static LastWPType As String
  Static CurWPType As String
  Static CurWPIndex As Integer
  Dim TempSens As Single

  If (NavData.GPSData.CDISensitivity = 0) Then
NavData.GPSData.CDISensitivity = 1
  If UCase$(Trim$(LastWPType)) = "MAP" Then
    CDISensitivity = 1#
  ElseIf (UCase$(Trim$(NavData.GPSData.CurWPType))
= "FAF") And (NavData.GPSData.TrkDistToWP < 2#)
Then
    TempSens = Val(Mid$(Main.GPSParam.Caption, 7))
    If (TempSens > 0) Then
      CDISensitivity = TempSens
    Else
      CDISensitivity = .3
    End If
  Else
    CDISensitivity = NavData.GPSData.CDISensitivity
  End If
  If Not (NavData.CurWayPntIndex = CurWPIndex) Then
    CurWPIndex = NavData.CurWayPntIndex
    LastWPType = CurWPType
    CurWPType = NavData.GPSData.CurWPType
  End If
End Function

Sub CheckDME (NavData As NavRec)
  Dim NextWPType As String
  Dim nextWPind As Integer
  Dim prevWPind As Integer

End Sub

Function DMERadius (WPType As String) As Double
  Dim DMEpos As Integer

  DMEpos = InStr(1, WPType, "DME", 1)
  If DMEpos <> 0 Then
    DMERadius = Val(Mid$(Trim$(WPType), DMEpos +
3))
  Else
    DMERadius = 0
  End If
End Function

Sub InitNav (NavData As NavRec)
  Dim i As Integer

  NavData.TimeStamp = 0
  NavData.NavMode = NAV_OFF
  NavData.FlightPlan.NumWayPnts = 0
  NavData.FlightPlan.LongCoef = 0
  NavData.FlightPlan.FltPlnLen = 0
  NavData.CurWayPntIndex = -1
  NavData.Latitude = 0
  NavData.Longitude = 0
  NavData.Altitude = 0
  NavData.DeltaX = 0
  NavData.DeltaY = 0
  NavData.DeltaAlt = 0
  NavData.TrkDeltaX = 0
  NavData.TrkDeltaY = 0
  NavData.TrkDeltaAlt = 0
  NavData.CurFltPlnOffset = 0
  NavData.GPSData.LastWPIdent = ""
```

```
  NavData.GPSData.CurWPIdent = ""
  NavData.GPSData.CurWPType = ""
  NavData.GPSData.CurWPLat = 0
  NavData.GPSData.CurWPLong = 0
  NavData.GPSData.CurRadial = 0
  NavData.GPSData.NextWPIdent = ""
  NavData.GPSData.NextWPType = ""
  NavData.GPSData.NextWPLat = 0
  NavData.GPSData.NextWPLong = 0
  NavData.GPSData.NextRadial = 0
  NavData.GPSData.XAE = 0
  NavData.GPSData.XTE = 0
  NavData.GPSData.GAE = 0
  NavData.GPSData.CGA = 0
  NavData.GPSData.AltErr = 0
  NavData.GPSData.DisttoWP = 0
  NavData.GPSData.TrkDistToWP = 0
  NavData.GPSData.HdgtoWP = 0
  NavData.GPSData.DistToEnd = 0
  NavData.GPSData.Track = 0
  NavData.GPSData.TAE = 0
  NavData.GPSData.GrndSpeed = 0
  NavData.GPSData.ETW = 0

  NavData.LastRadial = 0
  NavData.LastBisect = 0
  NavData.LastWPLat = 0
  NavData.LastWPLong = 0
  NavData.LastWPAlt = 0
  NavData.CurBisect = 0
  NavData.CurWPAlt = 0
  NavData.LCRadial = 0
  NavData.CLRadial = 0
  NavData.NextRadial = 0
  NavData.CurBisect = 0
  For i = 1 To 5
    NavData.DistHist(i) = 0
    NavData.TimeHist(i) = 0
  Next i

  TurnRate = (360 * Deg2Rad / (2 * 60)) * 60 * 60
'full turn in 2 minutes - units are rad/hour
End Sub

Sub ResetNav (NavData As NavRec)
  Dim curWPind As Integer

  If NavData.FlightPlan.NumWayPnts > 1 Then
    NavData.TimeStamp = -1
    NavData.NavMode = NAV_FLTPLN
    NavData.CurWayPntIndex = 0
    curWPind = NavData.CurWayPntIndex
    NavData.Latitude =
NavData.FlightPlan.WayPnts(curWPind).Latitude
    NavData.Longitude =
NavData.FlightPlan.WayPnts(curWPind).Longitude
    NavData.Altitude =
NavData.FlightPlan.WayPnts(curWPind).Altitude
    NavData.LastWPAlt = NavData.Altitude
    NavData.DeltaX = 0
    NavData.DeltaY = 0
    NavData.DeltaAlt = 0
    NavData.TrkDeltaX = 0
    NavData.TrkDeltaY = 0
    NavData.TrkDeltaAlt = 0
    NavData.GPSData.LastWPIdent = ""
    NavData.GPSData.CDISensitivity = 1#
    NavData.GPSData.CurWPIdent =
NavData.FlightPlan.WayPnts(curWPind).Ident
    NavData.GPSData.CurWPType =
NavData.FlightPlan.WayPnts(curWPind).WPType
    NavData.GPSData.CurWPLat =
NavData.FlightPlan.WayPnts(curWPind).Latitude
    NavData.GPSData.CurWPLong =
NavData.FlightPlan.WayPnts(curWPind).Longitude
    NavData.GPSData.CurRadial =
NavData.FlightPlan.WayPnts(curWPind + 1).Radial
    NavData.GPSData.CurLegLen =
NavData.FlightPlan.WayPnts(curWPind).LegLen
    NavData.CurWPAlt =
NavData.FlightPlan.WayPnts(curWPind).Altitude
    NavData.LastRadius =
NavData.FlightPlan.WayPnts(curWPind).ArcRadius
'DMERadius(NavData.GPSData.CurWPType)
    NavData.LastRadial = NavData.GPSData.CurRadial
* Deg2Rad
'   Fake a waypoint a 'good' distance from the
first
    NavData.LastWPLat = NavData.GPSData.CurWPLat +
Cos(NavData.LastRadial) * 2 / 60
    NavData.LastWPLong = NavData.GPSData.CurWPLong
- (Sin(NavData.LastRadial) * 2 / 60) /
NavData.FlightPlan.LongCoef
    NavData.LastWPAlt = NavData.CurWPAlt
    NavData.CLRadial = NavData.LastRadial
    NavData.LCRadial = recipAng(NavData.CLRadial)
    NavData.LastBisect =
smallBisect(NavData.LastRadial, NavData.LCRadial)
    NavData.LastWPAlt = NavData.Altitude
    If NavData.FlightPlan.NumWayPnts > 1 Then
      NavData.GPSData.NextWPIdent =
NavData.FlightPlan.WayPnts(curWPind + 1).Ident
      NavData.GPSData.NextWPType =
NavData.FlightPlan.WayPnts(curWPind + 1).WPType
      NavData.GPSData.NextWPLat =
NavData.FlightPlan.WayPnts(curWPind + 1).Latitude
      NavData.GPSData.NextWPLong =
NavData.FlightPlan.WayPnts(curWPind + 1).Longitude
      NavData.GPSData.NextRadial =
NavData.FlightPlan.WayPnts(curWPind + 1).Radial
      NavData.GPSData.NextLegLen =
NavData.FlightPlan.WayPnts(curWPind).LegLen
      NavData.NextRadial =
Reciprocal(NavData.GPSData.NextRadial) * Deg2Rad
      NavData.CurBisect =
smallBisect(NavData.CLRadial, NavData.NextRadial)
    Else
      NavData.GPSData.NextWPIdent = ""
      NavData.GPSData.NextWPType = ""
      NavData.GPSData.NextWPLat = 0
      NavData.GPSData.NextWPLong = 0
      NavData.GPSData.NextRadial = 0
      NavData.GPSData.NextLegLen = 0
    End If
    NavData.CurFltPlnOffset =
NavData.FlightPlan.FltPlnLen -
NavData.GPSData.CurLegLen
  Else
    NavData.CurWayPntIndex = -1
    NavData.Latitude = 0
    NavData.Longitude = 0
    NavData.DeltaX = 0
    NavData.DeltaY = 0
    NavData.TrkDeltaX = 0
    NavData.TrkDeltaY = 0
    NavData.GPSData.LastWPIdent = ""
    NavData.GPSData.CurWPIdent = ""
    NavData.GPSData.CurWPType = ""
    NavData.GPSData.CurWPLat = 0
    NavData.GPSData.CurWPLong = 0
    NavData.GPSData.CurRadial = 0
    NavData.GPSData.NextWPIdent = ""
    NavData.GPSData.NextWPType = ""
```

```
    NavData.GPSData.NextWPLat = 0
    NavData.GPSData.NextWPLong = 0
    NavData.GPSData.NextRadial = 0
  End If
  NavData.GPSData.XAE = 0
  NavData.GPSData.XTE = 0
  NavData.GPSData.GAE = 0
  NavData.GPSData.CGA = 0
  NavData.GPSData.AltErr = 0
  NavData.GPSData.DisttoWP = 0
  NavData.GPSData.TrkDistToWP = 0
  NavData.GPSData.HdgtoWP = 0
  NavData.GPSData.DistToEnd = 0
  NavData.GPSData.Track = 0
  NavData.GPSData.TAE = 0
  NavData.GPSData.GrndSpeed = 0
  NavData.GPSData.ETW = 0
End Sub

Function TAntDistance (ByVal r1 As Double, ByVal r2
As Double, ByVal TRadius As Double) As Double
  Dim alpha As Double
  Dim tempTan As Double

  alpha = Abs(sgnDiff(r1, r2))
  If Abs(alpha - 2 * PIo2) < ZERO Then
    TAntDistance = 0
  Else
    If (alpha < 10 * Deg2Rad) Then
      tempTan = Tan(5 * Deg2Rad)
    Else
      tempTan = Tan(alpha / 2)
    End If
    TAntDistance = TRadius / tempTan
  End If
End Function

Function TurnChar (StatusStr As String) As String
  Dim charPos As Integer

  charPos = InStr(StatusStr, "T")
  If charPos = 0 Then
    TurnChar = "N"
  Else
    TurnChar = Mid$(StatusStr, charPos + 1, 1)
  End If
End Function

Function TurnRadius (ByVal V As Double, ByVal TRate
As Double) As Double
  If (Abs(TRate) > ZERO) Then
    TurnRadius = V / TRate
  Else
    TurnRadius = 0
  End If
End Function

Sub UpdateNav (NavData As NavRec, NewLatitude As
Double, NewLongitude As Double, NewAltitude As
Double, TimeStamp As Long)
  Dim curWPind As Integer
  Dim nextWPind As Integer
  Dim prevWPind As Integer
  Dim CurTime As Long
  Dim ACRadial As Double
  Dim ACAngle As Double
  Dim ACDist As Double
  Dim ACTrack As Double
  Dim TurnRad As Double
  Dim StandardTurnRad As Double
  Dim CurTurnRad As Double
  Dim LastTurnRad As Double
  Dim CircDist As Double
  Dim TurnDist As Double
  Dim Bisect As Double
  Dim LastTurnDist As Double
  Dim i As Integer
  Dim DeltaTime As Double
  Dim TotTime As Double
  Dim TotDist As Double
  Dim RedoCalc As Integer
  Dim DTK As Double
  Dim DGA As Double         ' Desired Glide Angle
  Dim GAtoWP As Double      ' Glide Angle to Waypoint
  Dim dmeDeltaX As Double
  Dim dmeDeltaY As Double
  Dim oldTurnChar As String

  CurTime = TimeStamp
  NavData.TrkDeltaY = (NavData.Latitude -
NewLatitude) * 60
  NavData.TrkDeltaX = (NavData.FlightPlan.LongCoef
* (NewLongitude - NavData.Longitude)) * 60
  NavData.TrkDeltaAlt = NewAltitude -
NavData.Altitude
  NavData.Latitude = NewLatitude
  NavData.Longitude = NewLongitude
  NavData.Altitude = NewAltitude
  ACTrack = radialFromWP(NavData.TrkDeltaY,
NavData.TrkDeltaX)
  NavData.GPSData.Track = ACTrack / Deg2Rad
  ACDist = Sqr(NavData.TrkDeltaY *
NavData.TrkDeltaY + NavData.TrkDeltaX *
NavData.TrkDeltaX)
  NavData.GPSData.CGA = Atn2(NavData.TrkDeltaAlt,
ACDist * 6080) / Deg2Rad

  If (NavData.TimeStamp > 0) Then
' Calculate an average ground speed based on the
last six time steps
    DeltaTime = CurTime - NavData.TimeStamp
    TotTime = DeltaTime
    TotDist = ACDist
    For i = 1 To 5
      TotTime = TotTime + NavData.TimeHist(i)
      TotDist = TotDist + NavData.DistHist(i)
    Next i

    If (TotTime > 0) Then
      NavData.GPSData.GrndSpeed = TotDist * 3600000
/ TotTime
    Else
      NavData.GPSData.GrndSpeed = 0
    End If

    For i = 5 To 2 Step -1
      NavData.DistHist(i) = NavData.DistHist(i - 1)
      NavData.TimeHist(i) = NavData.TimeHist(i - 1)
    Next i
    NavData.DistHist(1) = ACDist
    NavData.TimeHist(1) = DeltaTime
  Else
    NavData.GPSData.GrndSpeed = 120 'Arbitrary
reasonable number that will be fixed on the next
time step.
  End If

  NavData.TimeStamp = CurTime
  NavData.GPSData.TimeStamp = CurTime
  If NavData.NavMode <> NAV_OFF Then
    NavData.DeltaY = (NavData.GPSData.CurWPLat -
NavData.Latitude) * 60
```

```
    NavData.DeltaX = (NavData.FlightPlan.LongCoef *
(NavData.Longitude - NavData.GPSData.CurWPLong)) *
60
    NavData.DeltaAlt = NavData.CurWPAlt -
NavData.Altitude
    ACRadial = radialFromWP(NavData.DeltaY,
NavData.DeltaX)
    ACAngle = -Sgn(NavData.CurBisect) *
sgnDiff(ACRadial, NavData.CurBisect)
    curWPind = NavData.CurWayPntIndex
'   If crossed the bisector...
    If (ACAngle < 0) Then
      If NavData.CurWayPntIndex <
(NavData.FlightPlan.NumWayPnts - 1) Then
        NavData.GPSData.LastWPIdent =
NavData.GPSData.CurWPIdent
        NavData.LastRadius =
NavData.FlightPlan.WayPnts(curWPind).ArcRadius
'DMERadius(NavData.GPSData.CurWPType)
        NavData.LastRadial = NavData.CLRadial
        NavData.LastBisect = NavData.CurBisect
        NavData.LastWPLat =
NavData.GPSData.CurWPLat
        NavData.LastWPLong =
NavData.GPSData.CurWPLong
        NavData.LastWPAlt = NavData.CurWPAlt
        NavData.CurWayPntIndex =
NavData.CurWayPntIndex + 1
        curWPind = NavData.CurWayPntIndex
        NavData.GPSData.CurWPIdent =
NavData.FlightPlan.WayPnts(curWPind).Ident
        NavData.GPSData.CurWPType =
NavData.FlightPlan.WayPnts(curWPind).WPType
        NavData.GPSData.CurWPLat =
NavData.FlightPlan.WayPnts(curWPind).Latitude
        NavData.GPSData.CurWPLong =
NavData.FlightPlan.WayPnts(curWPind).Longitude
        NavData.GPSData.CurRadial =
NavData.FlightPlan.WayPnts(curWPind).Radial
        NavData.GPSData.CurLegLen =
NavData.FlightPlan.WayPnts(curWPind).LegLen
        NavData.CurWPAlt =
NavData.FlightPlan.WayPnts(curWPind).Altitude
        If curWPind <
(NavData.FlightPlan.NumWayPnts - 1) Then
          NavData.GPSData.NextWPIdent =
NavData.FlightPlan.WayPnts(curWPind + 1).Ident
          NavData.GPSData.NextWPType =
NavData.FlightPlan.WayPnts(curWPind + 1).WPType
          NavData.GPSData.NextWPLat =
NavData.FlightPlan.WayPnts(curWPind + 1).Latitude
          NavData.GPSData.NextWPLong =
NavData.FlightPlan.WayPnts(curWPind + 1).Longitude
          NavData.GPSData.NextRadial =
NavData.FlightPlan.WayPnts(curWPind + 1).Radial
          NavData.GPSData.NextLegLen =
NavData.FlightPlan.WayPnts(curWPind + 1).LegLen
        Else
          NavData.GPSData.NextRadial =
NavData.GPSData.CurRadial
        End If
        NavData.CLRadial =
NavData.GPSData.CurRadial * Deg2Rad
        NavData.LCRadial =
recipAng(NavData.CLRadial)
        NavData.NextRadial =
Reciprocal(NavData.GPSData.NextRadial) * Deg2Rad
        NavData.CurFltPlnOffset =
NavData.CurFltPlnOffset - NavData.GPSData.CurLegLen
        NavData.DeltaY = (NavData.GPSData.CurWPLat
- NavData.Latitude) * 60
        NavData.DeltaX =
(NavData.FlightPlan.LongCoef * (NavData.Longitude -
NavData.GPSData.CurWPLong)) * 60
        NavData.DeltaAlt = NavData.CurWPAlt -
NavData.Altitude
        ACRadial = radialFromWP(NavData.DeltaY,
NavData.DeltaX)
        NavData.CurBisect =
smallBisect(NavData.CLRadial, NavData.NextRadial)
        'ACAngle = -Sgn(NavData.CurBisect) *
SgnDiff(ACRadial, NavData.CurBisect)
      End If
    End If
    NavData.GPSData.DisttoWP = Sqr(NavData.DeltaY *
NavData.DeltaY + NavData.DeltaX * NavData.DeltaX)
    ACDist = NavData.GPSData.DisttoWP
    NavData.GPSData.XAE = sgnDiff(NavData.CLRadial,
ACRadial) / Deg2Rad
    NavData.GPSData.XTE = ACDist *
Sin(NavData.GPSData.XAE * Deg2Rad)
    DGA = Atn2((NavData.CurWPAlt -
NavData.LastWPAlt), (NavData.GPSData.CurLegLen *
6080#))
    GAtoWP = Atn2(NavData.DeltaAlt, (ACDist *
6080#))
    NavData.GPSData.GAE = NavData.GPSData.CGA - DGA
/ Deg2Rad
    NavData.GPSData.AltErr = 6080# * ACDist *
(Sin(DGA) - Sin(GAtoWP))
    NavData.GPSData.TrkDistToWP = Sqr(ACDist *
ACDist - NavData.GPSData.XTE * NavData.GPSData.XTE)
    NavData.GPSData.HdgtoWP = recipAng(ACRadial) /
Deg2Rad
    DTK = NavData.LCRadial
    If (Main.TurnAnticipation) Then
      StandardTurnRad =
TurnRadius(NavData.GPSData.GrndSpeed, TurnRate)
      LastTurnRad = NavData.LastRadius
      If (LastTurnRad = 0) Then
        LastTurnRad = StandardTurnRad
      End If
      CurTurnRad =
NavData.FlightPlan.WayPnts(curWPind).ArcRadius
'DMERadius(NavData.GPSData.CurWPType)
      If (CurTurnRad = 0) Then
        CurTurnRad = StandardTurnRad
      End If
      LastTurnDist =
TAntDistance(NavData.LastRadial, NavData.LCRadial,
LastTurnRad)
      TurnDist = TAntDistance(NavData.CLRadial,
NavData.NextRadial, CurTurnRad)
      RedoCalc = False
'       Main.MyDebug.Print LastTurnDist, TurnDist
      oldTurnChar = TurnChar(CStr(Main.GPSStatus))
      If (TurnDist > .01) And
(NavData.GPSData.TrkDistToWP < (TurnDist + .2)) And
(NavData.GPSData.TrkDistToWP >= TurnDist) Then
'       Indicate aircraft is close to a turn
        If (oldTurnChar <> "C") And
(NavData.FlightPlan.WayPnts(curWPind).ArcRadius =
0) Then
          Main.GPSStatus = "Running-Sens. " +
Format$(Val(Mid$(Main.GPSParam.Caption, 7))) + "
TC"
        End If
      ElseIf (NavData.GPSData.TrkDistToWP <
TurnDist) And (TurnDist > .01) Then
'       Indicate aircraft is before the waypoint in
a turn
```

```
        If (oldTurnChar <> "B") And
(NavData.FlightPlan.WayPnts(curWPind).ArcRadius =
0) Then
          Main.GPSStatus = "Running-Sens. " +
Format$(Val(Mid$(Main.GPSParam.Caption, 7))) + "
TB"
        End If
        If Main.Shape1.FillColor <> RGB(255, 0, 0)
Then
          Main.Shape1.FillColor = RGB(255, 0, 0)
        End If
        If Main.Shape2.FillColor <> RGB(0, 255, 0)
Then
          Main.Shape2.FillColor = RGB(0, 255, 0)
        End If
      ElseIf ((NavData.GPSData.CurLegLen -
NavData.GPSData.TrkDistToWP) < LastTurnDist) And
(LastTurnDist > .01) Then
'       Indicate aircraft is after the waypoint in
a turn
        If (oldTurnChar <> "A") And
(NavData.LastRadius = 0) Then
          Main.GPSStatus = "Running-Sens. " +
Format$(Val(Mid$(Main.GPSParam.Caption, 7))) + "
TA"
        End If
        If Main.Shape2.FillColor <> RGB(255, 0, 0)
Then
          Main.Shape2.FillColor = RGB(255, 0, 0)
        End If
        If Main.Shape1.FillColor <> RGB(0, 255, 0)
Then
          Main.Shape1.FillColor = RGB(0, 255, 0)
        End If
      Else
'       Indicate not in any turn
        If (oldTurnChar <> "N") Then
          Main.GPSStatus = "Running-Sens. " +
Format$(Val(Mid$(Main.GPSParam.Caption, 7)))
        End If
        If Main.Shape1.FillColor <> RGB(0, 255, 0)
Then
          Main.Shape1.FillColor = RGB(0, 255, 0)
        End If
        If Main.Shape2.FillColor <> RGB(0, 255, 0)
Then
          Main.Shape2.FillColor = RGB(0, 255, 0)
        End If
      End If
      If (NavData.GPSData.TrkDistToWP < TurnDist)
And (TurnDist > .01) Then
'       in next turn so set up to calculate next
turn TAE and XTE
        Bisect = NavData.CurBisect
        TurnRad = CurTurnRad
        RedoCalc = True
      ElseIf ((NavData.GPSData.CurLegLen -
NavData.GPSData.TrkDistToWP) < LastTurnDist) And
(LastTurnDist > .01) Then
'       still in previous turn so set up to
calculate previous turn TAE and XTE
        TurnDist = LastTurnDist
        NavData.DeltaY = (NavData.LastWPLat -
NavData.Latitude) * 60
        NavData.DeltaX =
(NavData.FlightPlan.LongCoef * (NavData.Longitude -
NavData.LastWPLong)) * 60
        ACRadial = radialFromWP(NavData.DeltaY,
NavData.DeltaX)
        ACDist = Sqr(NavData.DeltaY *
NavData.DeltaY + NavData.DeltaX * NavData.DeltaX)
        Bisect = NavData.LastBisect
        TurnRad = LastTurnRad
        RedoCalc = True
      End If
      If RedoCalc Then
        CircDist = Sqr(TurnRad * TurnRad + TurnDist
* TurnDist)
        XTEAndDTK NavData.GPSData.XTE, DTK,
TurnRad, ACDist, ACRadial, CircDist, Bisect
      End If
    End If
    NavData.GPSData.TAE = sgnDiff(ACTrack, DTK) /
Deg2Rad
    If NavData.GPSData.GrndSpeed > 0 Then
      NavData.GPSData.ETW =
(NavData.GPSData.DisttoWP /
NavData.GPSData.GrndSpeed) * 60
    Else
      NavData.GPSData.ETW = 0
    End If
    If (NavData.CurFltPlnOffset <=
-NavData.GPSData.CurLegLen) Then
      NavData.GPSData.DistToEnd =
-NavData.GPSData.TrkDistToWP
    Else
      NavData.GPSData.DistToEnd =
NavData.GPSData.TrkDistToWP +
NavData.CurFltPlnOffset
    End If
    NavData.GPSData.CDISensitivity =
CDISensitivity(NavData)
    NavData.GPSData.CDIVal = (NavData.GPSData.XTE /
NavData.GPSData.CDISensitivity) * 100

'   Fake some information if currently in a DME arc
    nextWPind = NavData.CurWayPntIndex
    Do While (nextWPind <
(NavData.FlightPlan.NumWayPnts))
      If
(NavData.FlightPlan.WayPnts(nextWPind).ArcRadius =
0) Then Exit
Do'(Left$(NavData.FlightPlan.WayPnts(nextWPind).WPT
ype, 1) <> ")") Then Exit Do
      nextWPind = nextWPind + 1
    Loop
    NavData.GPSData.CurWPIdent =
NavData.FlightPlan.WayPnts(nextWPind).Ident
    prevWPind = NavData.CurWayPntIndex - 1
    If (prevWPind >= 0) Then
      Do While (prevWPind > 0)
        If
(NavData.FlightPlan.WayPnts(prevWPind).ArcRadius =
0) Then Exit
Do'(Left$(NavData.FlightPlan.WayPnts(prevWPind).WPT
ype, 1) <> ")") Then Exit Do
        prevWPind = prevWPind - 1
      Loop
      NavData.GPSData.LastWPIdent =
NavData.FlightPlan.WayPnts(prevWPind).Ident
    End If
    If ((nextWPind - prevWPind) > 1) Then
      dmeDeltaY =
(NavData.FlightPlan.WayPnts(nextWPind).Latitude -
NavData.Latitude) * 60
      dmeDeltaX = (NavData.FlightPlan.LongCoef *
(NavData.Longitude -
NavData.FlightPlan.WayPnts(nextWPind).Longitude)) *
60
      NavData.GPSData.DisttoWP = Sqr(dmeDeltaY *
dmeDeltaY + dmeDeltaX * dmeDeltaX)
    End If
  Else
  End If
```

GPS_NAV.BAS

```
End Sub

Sub XTEAndDTK (XTEout As Double, DTKout As Double,
ByVal TRadius As Double, ByVal ACd As Double, ByVal
ACrad As Double, ByVal Cd As Double, ByVal Crad As
Double)
  Dim longXTE As Double
  Dim wideDTK As Double
  Dim longXTE2 As Double
  Dim ACang As Double
  Dim test As Double
  Dim ACd2 As Double
  Dim Cd2 As Double

  ACang = -Sgn(Crad) * sgnDiff(ACrad, Crad)
  ACd2 = ACd * ACd
  Cd2 = Cd * Cd
  longXTE2 = ACd2 + Cd2 - 2 * ACd * Cd * Cos(ACang)
  test = longXTE2 + ACd2 - Cd2
  longXTE = Sqr(longXTE2)
  XTEout = -Sgn(Crad) * (TRadius - longXTE)
  If (longXTE > ZERO) Then
    wideDTK = asin(Cd * Sin(ACang) / longXTE)
  Else
    wideDTK = 2 * PIo2
  End If
  If (Abs(test) < ZERO) Then
    DTKout = Crad + Sgn(Crad) * PIo2
  Else
    DTKout = ACrad - Sgn(test) * Sgn(Crad) * (PIo2
+ wideDTK)
  End If
End Sub
```

GPS_PROC.BAS

```
Option Explicit

Declare Function ExitWindows Lib "User" (ByVal
dwReturnCode As Long, ByVal wReserved As Integer)
As Integer
Declare Function GetProfileString Lib "Kernel"
(ByVal lpAppName As String, ByVal lpKeyName As
String, ByVal lpDefault As String, ByVal
lpReturnedString As String, ByVal nSize As Integer)
As Integer
Declare Function WriteProfileString Lib "Kernel"
(ByVal lpApplicationName As String, ByVal lpKeyName
As String, ByVal lpString As String) As Integer
Declare Function GetTickCount Lib "User" () As Long


Const MaxDDERetry = 10  'Maximum number of DDE
retries

Global InsDDELinkOpen As Integer     'Instrument
Server DDE Link Flag
Global DasDDELinkOpen As Integer     'Data Server
DDE Link Flag
Global DasTCPLinkOpen As Integer     'Data Server
TCP Link Flag
'Global DasTCPSendReady As Integer     'Data Server
TCP is ready to send
Global ConDDELinkOpen As Integer     'Control
Console TCP Link Flag
Global ConTCPLinkOpen As Integer     'Control
Console TCP Link Flag

Sub ConnectCtrlCon ()
  Dim RetryCntr As Integer

  Main.CtrlConInd.FillColor = &HFF&
  DoEvents
  Main.ExpName.LinkItem = "ExpName"
  Main.ProfileName.LinkItem = "ProfileName"
  Main.FltPlnName.LinkItem = "FltPlnName"
  Main.CtrlConStatus.LinkItem = "CtrlConStatus"
  Main.GPSData.LinkItem = "GPSData"
  Main.GPSStatus.LinkItem = "GPSStatus"
  Main.GPSName.LinkItem = "GPSName"
  Main.GPSParam.LinkItem = "GPSParam"
  Main.ExpName.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.ProfileName.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.FltPlnName.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.CtrlConStatus.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.GPSData.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.GPSStatus.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.GPSName.LinkTopic =
SystemSettings.CtrlConPath.Text
  Main.GPSParam.LinkTopic =
SystemSettings.CtrlConPath.Text

  RetryCntr = 0
'  Loop added because could not connect on first
try when no wind module was loaded
  On Error Resume Next
  Main.ExpName.LinkMode = LINK_AUTOMATIC
  Main.ProfileName.LinkMode = LINK_AUTOMATIC
  Main.FltPlnName.LinkMode = LINK_AUTOMATIC
  Main.CtrlConStatus.LinkMode = LINK_AUTOMATIC
  Main.GPSData.LinkMode = LINK_AUTOMATIC
  Main.GPSStatus.LinkMode = LINK_AUTOMATIC
  Main.GPSName.LinkMode = LINK_AUTOMATIC
  Main.GPSParam.LinkMode = LINK_AUTOMATIC
  RetryCntr = RetryCntr + 1
  DoEvents
  If Err <> 0 Then
'    i = MsgBox("Cannot connect to Control Console.
Check Path and status", MB_OK + MB_ICONSTOP, "GPS
Error")
    ConDDELinkOpen = False
    Main.ExpName.LinkMode = LINK_NONE
    Main.ProfileName.LinkMode = LINK_NONE
    Main.FltPlnName.LinkMode = LINK_NONE
    Main.CtrlConStatus.LinkMode = LINK_NONE
    Main.GPSData.LinkMode = LINK_NONE
    Main.GPSStatus.LinkMode = LINK_NONE
    Main.GPSName.LinkMode = LINK_NONE
    Main.GPSParam.LinkMode = LINK_NONE
  Else
    ConDDELinkOpen = True
    Main.CtrlConInd.FillColor = &HFF00&
  End If
  On Error GoTo 0

End Sub

Sub ConnectDataSrvr ()
  Dim i As Integer
  Dim RetryCntr As Integer

  DoEvents
```

```
'  Main.DataSrvrData.LinkItem = "DataSrvrData"
'  Main.DataSrvrData.LinkTopic =
SystemSettings.DataSrvrPath
  Main.DataSrvrStatus.LinkItem = "DataSrvrStatus"
  Main.DataSrvrStatus.LinkTopic =
SystemSettings.DataSrvrPath

  DasDDELinkOpen = False
  RetryCntr = 0
  On Error Resume Next
  Do
'     Main.DataSrvrData.LinkMode = LINK_AUTOMATIC
    Main.DataSrvrStatus.LinkMode = LINK_AUTOMATIC
    RetryCntr = RetryCntr + 1
    DoEvents
  Loop Until (Err = 0) Or (RetryCntr >=
MaxDDERetry)
  DoEvents
  If Err Then
    i = MsgBox("Cannot connect to Data Server.
Check Path and status", MB_OK + MB_ICONSTOP, "GPS
Error")
    DasDDELinkOpen = False
    Main.DataSrvrStatus.LinkMode = LINK_NONE
'     Main.DataSrvrData.LinkMode = LINK_NONE
  Else
'     Main.DataSrvrData.LinkMode = LINK_MANUAL
    DoEvents
    DasDDELinkOpen = True
    Main.DataSrvrInd.FillColor = &HFF00&
  End If
  On Error GoTo 0

  If (Not DasTCPLinkOpen) Then
    Main.DasDataSock.RemoteDotAddr =
SystemSettings.DataSrvrIP
    Main.DasDataSock.RemotePort =
Val(SystemSettings.DataSrvrPort)
    If (Main.DasDataSock.State <> 2) Then
Main.DasDataSock.Action = 2
    If Err Then
      MsgBox "Socket Error - " + Error,
MB_ICONSTOP, "GPS Error"
    End If
  End If
End Sub

Sub ConnectGPS ()

End Sub

Sub ConnectInstSrvr ()
  Dim i As Integer
  Dim RetryCntr As Integer

  Main.CDIVal.LinkItem = "HSICDI"
  Main.CDIVal.LinkTopic =
SystemSettings.InstSrvrPath
  Main.LCDDisplayString.LinkItem = "LCDDisplay"
  Main.LCDDisplayString.LinkTopic =
SystemSettings.InstSrvrPath
  Main.HSIToFromFlag.LinkItem = "HSIToFromFlag"
  Main.HSIToFromFlag.LinkTopic =
SystemSettings.InstSrvrPath

  InsDDELinkOpen = True
  RetryCntr = 0
  On Error Resume Next
  Do
    Main.CDIVal.LinkMode = LINK_AUTOMATIC
    Main.LCDDisplayString.LinkMode = LINK_AUTOMATIC
    Main.HSIToFromFlag.LinkMode = LINK_AUTOMATIC
    RetryCntr = RetryCntr + 1
    DoEvents
  Loop Until (Err = 0) Or (RetryCntr >=
MaxDDERetry)
  DoEvents
  If Err <> 0 Then
    i = MsgBox("Cannot connect to Instrument
Server.  Check Path and status", MB_OK +
MB_ICONSTOP, "GPS Error")
    InsDDELinkOpen = False
    Main.CDIVal.LinkMode = LINK_NONE
    Main.LCDDisplayString.LinkMode = LINK_NONE
    Main.HSIToFromFlag.LinkMode = LINK_NONE
  Else
    Main.CDIVal.LinkMode = LINK_MANUAL
    Main.LCDDisplayString.LinkMode = LINK_MANUAL
    Main.HSIToFromFlag.LinkMode = LINK_MANUAL
    Main.InstSrvrInd.FillColor = &HFF00&
  End If
  On Error GoTo 0
End Sub

Sub ConnectLight ()

End Sub

Sub ConnectPanel ()

End Sub

Sub ConnectWind ()

End Sub

Function FindListIndex (List As ComboBox, SearchStr
As String) As Integer
  Dim i As Integer

  FindListIndex = -1
  If List.ListCount > 0 Then
    i = 0
    While (i < List.ListCount) And
(Trim$(List.List(i)) <> Trim$(SearchStr))
      i = i + 1
    Wend
    If i < List.ListCount Then
      FindListIndex = i
    End If
  End If
End Function

Function LatFormat$ (Latitude As Double)
  Dim Deg As Long
  Dim Min As Double

  Min = Latitude * 60
  Deg = Int(Min) - (Int(Min) Mod 60)
  Deg = Deg / 60
  Min = Min - (Deg * 60)
  LatFormat$ = Format$(Deg, "00") + " " +
Format$(Min, "00.00")
End Function

Function LatVal (Latitude As String) As Double
  LatVal = ((Val(Left$(Latitude, 2)) * 60) +
Val(Right$(Latitude, 5))) / 60
End Function

Function LongFormat$ (Longitude As Double)
  Dim Deg As Long
  Dim Min As Double
```

```
  Min = Longitude * 60
  Deg = Int(Min) - (Int(Min) Mod 60)
  Deg = Deg / 60
  Min = Min - (Deg * 60)
  LongFormat$ = Format$(Deg, "000") + " " +
Format$(Min, "00.00")
End Function

Function LongVal (Longitude As String) As Double
  LongVal = ((Val(Left$(Longitude, 3)) * 60) +
Val(Right$(Longitude, 5))) / 60
End Function

Function NullCheck (DataVal As Variant) As Variant
  If IsNull(DataVal) Then
    NullCheck = ""
  Else
    NullCheck = DataVal
  End If
End Function

Sub ProcessSerial (Message As String)

End Sub

Function Reciprocal (Angle As Double) As Double
  If Angle < 180 Then
    Reciprocal = 180 + Angle
  Else
    Reciprocal = Angle - 180
  End If
End Function

Sub RefreshList (UpdateList As ComboBox, UpdateTbl
As Snapshot, FieldName As String)
  If Not (UpdateTbl.EOF And UpdateTbl.BOF) Then
    UpdateTbl.MoveFirst
  End If
  UpdateList.Clear
  While Not UpdateTbl.EOF
    UpdateList.AddItem UpdateTbl(FieldName)
    UpdateTbl.MoveNext
  Wend
End Sub
```

OTW.BAS

```
Option Explicit

Declare Function CreateSolidBrush Lib "GDI" (ByVal
crColor As Long) As Integer
Declare Function ExtFloodFill Lib "GDI" (ByVal hDC
As Integer, ByVal X As Integer, ByVal Y As Integer,
ByVal crColor As Long, ByVal wFillType As Integer)
As Integer
Declare Function SelectObject Lib "GDI" (ByVal hDC
As Integer, ByVal hObject As Integer) As Integer
```

OTW.FRM

```
VERSION 2.00
Begin Form Main
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   0  'None
   ClientHeight    =   8535
   ClientLeft      =   1230
   ClientTop       =   570
   ClientWidth     =   9600
   ForeColor       =   &H00000000&
   Height          =   8940
   KeyPreview      =   -1  'True
   Left            =   1170
   LinkTopic       =   "Form1"
   ScaleHeight     =   569
   ScaleMode       =   3  'Pixel
   ScaleWidth      =   640
   Top             =   225
   Width           =   9720
   WindowState     =   2  'Maximized
   Begin CommandButton BreakOut
      BackColor       =   &H00000000&
      Caption         =   "Break Out"
      Height          =   345
      Left            =   2550
      TabIndex        =   14
      Top             =   6780
      Width           =   1245
   End
   Begin SSPanel TestControls
      Alignment       =   6  'Center - TOP
      BevelInner      =   1  'Inset
      BevelWidth      =   2
      Height          =   1575
      Left            =   30
      TabIndex        =   1
      Top             =   4620
      Visible         =   0   'False
      Width           =   1485
      Begin TextBox TestWPDist
         Height          =   285
         Left            =   660
         TabIndex        =   11
         Text            =   "3"
         Top             =   810
         Width           =   645
      End
      Begin TextBox testRoll
         Height          =   285
         Left            =   660
         TabIndex        =   2
         Text            =   "0"
         Top             =   480
         Width           =   645
      End
      Begin TextBox testPitch
         Height          =   285
         Left            =   660
         TabIndex        =   4
         Text            =   "0"
         Top             =   180
         Width           =   645
      End
      Begin CommandButton Test
         Caption         =   "Test"
         Height          =   285
         Left            =   180
         TabIndex        =   5
         Top             =   1140
         Width           =   1125
      End
      Begin Label TestDistLabel
         Alignment       =   1  'Right Justify
         Caption         =   "Dist"
         Height          =   195
```

```
      Left            =   180
      TabIndex        =   12
      Top             =   870
      Width           =   435
   End
   Begin Label TestRollLabel
      Alignment       =   1  'Right Justify
      Caption         =   "Roll"
      Height          =   195
      Left            =   180
      TabIndex        =   7
      Top             =   540
      Width           =   435
   End
   Begin Label TestPitchLabel
      Alignment       =   1  'Right Justify
      Caption         =   "Pitch"
      Height          =   195
      Left            =   180
      TabIndex        =   8
      Top             =   240
      Width           =   435
   End
End
Begin dsSocket GPSDataSock
   DataSize        =   2048
   EOLChar         =   0
   Left            =   90
   LineMode        =   0   'False
   Linger          =   0   'False
   LocalPort       =   0
   RemoteDotAddr   =   "152.122.10.105"
   RemoteHost      =   ""
   RemotePort      =   15001
   ServiceName     =   ""
   Timeout         =   10
   Top             =   1440
End
Begin Timer FDSTimer
   Interval        =   100
   Left            =   90
   Top             =   60
End
Begin Timer ConnectTimer
   Enabled         =   0   'False
   Interval        =   500
   Left            =   90
   Top             =   540
End
Begin PictureBox OTWDisp
   AutoRedraw      =   -1  'True
   BackColor       =   &H00FFFFFF&
   ClipControls    =   0   'False
   DrawWidth       =   2
   FillColor       =   &H0000C0C0&
   ForeColor       =   &H00C0C0C0&
   Height          =   6075
   Left            =   30
   ScaleHeight     =   403
   ScaleMode       =   3  'Pixel
   ScaleWidth      =   625
   TabIndex        =   6
   Top             =   90
   Width           =   9405
End
Begin CommandButton SysSetCommand
   BackColor       =   &H00000000&
   Cancel          =   -1  'True
   Caption         =   "S&ys Settings"
   Height          =   345
   Left            =   2550
   TabIndex        =   9
   Top             =   6210
   Width           =   1245
End
Begin dsSocket FDSDataSock
   DataSize        =   2048
   EOLChar         =   0
   Left            =   90
   LineMode        =   0   'False
   Linger          =   0   'False
   LocalPort       =   0
   RemoteDotAddr   =   "152.122.10.104"
   RemoteHost      =   ""
   RemotePort      =   15000
   ServiceName     =   ""
   Timeout         =   10
   Top             =   990
End
Begin CommandButton QuitButton
   BackColor       =   &H00000000&
   Caption         =   "&Quit"
   Height          =   345
   Left            =   30
   TabIndex        =   3
   Top             =   6210
   Width           =   1245
End
Begin CommandButton SetupButton
   BackColor       =   &H00000000&
   Caption         =   "Display &Setup"
   Height          =   345
   Left            =   1290
   TabIndex        =   0
   Top             =   6210
   Width           =   1245
End
Begin SSPanel Panel3D1
   BevelInner      =   1  'Inset
   BevelWidth      =   2
   Height          =   495
   Left            =   4590
   TabIndex        =   10
   Top             =   6300
   Width           =   1185
   Begin Shape DiceInd
      BackColor       =   &H000000FF&
      BackStyle       =   1  'Opaque
      FillColor       =   &H00FFC0C0&
      FillStyle       =   0  'Solid
      Height          =   285
      Left            =   90
      Shape           =   1  'Square
      Top             =   90
      Visible         =   0   'False
      Width           =   285
   End
   Begin Shape AltInd
      BackColor       =   &H000000FF&
      BackStyle       =   1  'Opaque
      FillColor       =   &H00FF8080&
      FillStyle       =   0  'Solid
      Height          =   288
      Left            =   480
      Shape           =   1  'Square
      Top             =   96
      Visible         =   0   'False
      Width           =   288
   End
   Begin Shape WPInd
      BackColor       =   &H000000FF&
      BackStyle       =   1  'Opaque
      FillColor       =   &H00800000&
      FillStyle       =   0  'Solid
```

```
      Height          =   285
      Left            =   810
      Shape           =   1  'Square
      Top             =   90
      Visible         =   0    'False
      Width           =   285
   End
  End
  Begin Label Flag
     Caption        =   "3.4"
     FontBold       =   -1  'True
     FontItalic     =   0   'False
     FontName       =   "Courier New"
     FontSize       =   60
     FontStrikethru =   0   'False
     FontUnderline  =   0   'False
     Height         =   1305
     Left           =   90
     TabIndex       =   13
     Top            =   6720
     Width          =   2235
  End
  Begin Shape GPSTCPInd
     BackColor      =   &H000000FF&
     BackStyle      =   1  'Opaque
     FillColor      =   &H000000FF&
     FillStyle      =   0  'Solid
     Height         =   285
     Left           =   4200
     Shape          =   3  'Circle
     Top            =   6240
     Width          =   285
  End
  Begin Shape FDSTCPInd
     BackColor      =   &H000000FF&
     BackStyle      =   1  'Opaque
     FillColor      =   &H000000FF&
     FillStyle      =   0  'Solid
     Height         =   285
     Left           =   3870
     Shape          =   3  'Circle
     Top            =   6240
     Width          =   285
  End
End
Option Explicit

Dim FDSRecord As DataSrvrRec
Dim FDSTCPLinkOpen As Integer
'Dim FDSTCPSendReady As Integer
Dim GPSRecord As GPSRec
Dim GPSTCPLinkOpen As Integer
'Dim GPSTCPSendReady As Integer

Dim greenBrush As Integer
Dim blueBrush As Integer
Dim FDSOutBuffer As IDSockBuffer
Dim GPSOutBuffer As IDSockBuffer
Dim dice As Single

Sub BreakOut_Click ()
    dice = 0
End Sub

Sub ConnectTimer_Timer ()
  Dim OneDead As Integer

  OneDead = False
  If FDSDataSock.State = 1 Then
    FDSDataSock.RemoteDotAddr =
SystemSettings.DataSrvrIP
    FDSDataSock.RemotePort =
SystemSettings.DataSrvrPort
    FDSDataSock.Action = 2
    OneDead = True
  End If
  If GPSDataSock.State = 1 Then
    GPSDataSock.RemoteDotAddr =
SystemSettings.GPSIP
    GPSDataSock.RemotePort = SystemSettings.GPSPort
    GPSDataSock.Action = 2
    OneDead = True
  End If
  If Not OneDead Then
    ConnectTimer.Enabled = False
  End If
End Sub

Sub FDSDataSock_Close (ErrorCode As Integer,
ErrorDesc As String)
  FDSTCPLinkOpen = False
  FDSTCPInd.FillColor = &HFF&
  ConnectTimer.Enabled = True
End Sub

Sub FDSDataSock_Connect ()
  FDSTCPLinkOpen = True
'  FDSTCPSendReady = False
  FDSTCPInd.FillColor = &HFF00&
  FDSOutBuffer.ID = 0
End Sub

Sub FDSDataSock_Exception (ErrorCode As Integer,
ErrorDesc As String)
  MsgBox "FDS Socket Error - " + ErrorDesc, (MB_OK
+ MB_ICONSTOP), "OTW Error"

' Close the link as a dumb default thing to do
  If (FDSDataSock.State <> 1) Then
FDSDataSock.Action = 1
  FDSTCPLinkOpen = False
  FDSTCPInd.FillColor = &HFF&
  ConnectTimer.Enabled = True
End Sub

Sub FDSDataSock_Receive (ReceiveData As String)
  Dim RawBuffer As SocketBuffer
  Dim MsgBuffer As IDSockBuffer

  RawBuffer.buffer = ReceiveData
  LSet MsgBuffer = RawBuffer
  Select Case MsgBuffer.ID
    Case DSMSG_DATA
      LSet FDSRecord = MsgBuffer.buffer
      ProcessData
    Case IDMSG_REQUEST
      RawBuffer.buffer = "OTW Module"
      FDSOutBuffer.ID = IDMSG_REPLY
      FDSOutBuffer.buffer = RawBuffer
      FDSDataSock_SendReady
  End Select
End Sub

Sub FDSDataSock_SendReady ()
  Dim RawBuffer As SocketBuffer

  If FDSOutBuffer.ID <> 0 Then
    On Error Resume Next
    LSet RawBuffer = FDSOutBuffer
    FDSDataSock.Send = Trim$(RawBuffer.buffer)
    If Err = 0 Then
      FDSOutBuffer.ID = 0
    End If
```

```
    On Error GoTo 0
  End If
End Sub

Sub FDSTimer_Timer ()
  Dim RawBuffer As SocketBuffer

  RawBuffer.buffer = ""

  FDSOutBuffer.ID = DSCMD_GETDATA
  FDSOutBuffer.buffer = RawBuffer
  FDSDataSock_SendReady

  GPSOutBuffer.ID = GPSCMD_GETDATA
  GPSOutBuffer.buffer = RawBuffer
  GPSDataSock_SendReady
End Sub

Sub Form_KeyPress (KeyAscii As Integer)
  Dim c As String * 1

  c = UCase$(Chr$(KeyAscii))
  Select Case c
    Case "H"
      ' Hide all of the system controls
      QuitButton.Visible = Not QuitButton.Visible
      SetupButton.Visible = Not SetupButton.Visible
      SysSetCommand.Visible = Not
SysSetCommand.Visible
      FDSTCPInd.Visible = Not FDSTCPInd.Visible
    Case "Q"
      End
  End Select
End Sub

Sub Form_Load ()
  Dim tempFDS As DataSrvrRec
  Dim tempGPS As GPSRec

  initTrig

  Load SystemSettings

' Connect to the Data Server using TCP
  FDSTCPLinkOpen = False
  GPSTCPLinkOpen = False
  ConnectTimer.Enabled = True

' Set the background to black
  Main.BackColor = &H0&

' Create brushes for doing the color fills
'  greenBrush = CreateSolidBrush(RGB(96, 192, 0))
'  blueBrush = CreateSolidBrush(RGB(96, 255, 255))
'  greenBrush = CreateSolidBrush(RGB(192, 208, 0))
  greenBrush = CreateSolidBrush(RGB(224, 224, 224))
  blueBrush = CreateSolidBrush(RGB(208, 255, 255))

' Run through the process loop once to initialise
the dice
  tempFDS.Pitch = 0
  tempFDS.Roll = 0

  tempGPS.CurWPType = "FAF"
  tempGPS.AltErr = 0
  tempGPS.DisttoWP = 1

  FDSRecord = tempFDS
  GPSRecord = tempGPS
  ProcessData
End Sub

Sub Form_Unload (Cancel As Integer)
' Disconnect the TCP connection
  If (FDSDataSock.State <> 1) Then
FDSDataSock.Action = 1
  If (GPSDataSock.State <> 1) Then
GPSDataSock.Action = 1
  DoEvents

  End
End Sub

Sub GPSDataSock_Close (ErrorCode As Integer,
ErrorDesc As String)
  GPSTCPLinkOpen = False
  GPSTCPInd.FillColor = &HFF&
  ConnectTimer.Enabled = True
End Sub

Sub GPSDataSock_Connect ()
  GPSTCPLinkOpen = True
'  GPSTCPSendReady = False
  GPSTCPInd.FillColor = &HFF00&
  GPSOutBuffer.ID = 0
End Sub

Sub GPSDataSock_Exception (ErrorCode As Integer,
ErrorDesc As String)
  MsgBox "GPS Socket Error - " + ErrorDesc, (MB_OK
+ MB_ICONSTOP), "OTW Error"

' Close the link as a dumb default thing to do
  If (GPSDataSock.State <> 1) Then
GPSDataSock.Action = 1
  GPSTCPLinkOpen = False
  GPSTCPInd.FillColor = &HFF&
  ConnectTimer.Enabled = True
End Sub

Sub GPSDataSock_Receive (ReceiveData As String)
  Dim RawBuffer As SocketBuffer
  Dim MsgBuffer As IDSockBuffer

  RawBuffer.buffer = ReceiveData
  LSet MsgBuffer = RawBuffer
  Select Case MsgBuffer.ID
    Case GPSMSG_DATA
      LSet GPSRecord = MsgBuffer.buffer
    Case IDMSG_REQUEST
      RawBuffer.buffer = "OTW Module"
      GPSOutBuffer.ID = IDMSG_REPLY
      GPSOutBuffer.buffer = RawBuffer
      GPSDataSock_SendReady
  End Select
End Sub

Sub GPSDataSock_SendReady ()
  Dim RawBuffer As SocketBuffer

  If GPSOutBuffer.ID <> 0 Then
    On Error Resume Next
    LSet RawBuffer = GPSOutBuffer
    GPSDataSock.Send = Trim$(RawBuffer.buffer)
    If Err = 0 Then
      GPSOutBuffer.ID = 0
    End If
    On Error GoTo 0
  End If
End Sub

Sub ProcessData ()
  Dim oldBrush As Integer
  Dim leftX As Single
```

```
  Dim leftY As Single
  Dim rightX As Single
  Dim rightY As Single
  Dim result As Integer
  Dim WPdist As Single
  Dim MAPAlt As Single
  Dim MAPwp As Integer
  Dim MAHPwp As Integer
'  Static dice As Single
'  Static diceUsed As Integer
  Static OTWOn As Integer

  MAPwp = (UCase$(Trim$(GPSRecord.CurWPType)) =
"MAP")
  MAHPwp = (UCase$(Trim$(GPSRecord.CurWPType)) =
"MAHP")

  AltInd.Visible = GPSRecord.AltErr < 200
  DiceInd.Visible = dice <= .75
  WPInd.Visible = MAPwp Or MAHPwp

' Clear the old display
  OTWDisp.Cls

  Flag = 2 - 8 * dice / 3

' Determine if horizon should be shown or not
  If MAPwp Then
    MAPAlt = 1900 * GPSRecord.DisttoWP / 6
  ElseIf MAHPwp Then
    MAPAlt = 1900 * (1 - GPSRecord.DisttoWP / 5)
  Else
    OTWOn = False
    dice = Rnd
    Exit Sub
  End If

  If ((MAPAlt + GPSRecord.AltErr) >= 70) Then
    Exit Sub
  End If

  WPdist = (2 - GPSRecord.DisttoWP) * 3 / 8' less
than 2 miles, 25%/75% no-go/go

  If Not OTWOn And (dice > WPdist) Then
    Exit Sub
  End If

  OTWOn = True

' Calculate left and right edge points for the
horizon
  leftX = OTWDisp.ScaleLeft - 10
  leftY = OTWDisp.ScaleHeight / 2 * (1 +
Tan(FDSRecord.Pitch * DtoR) / Tan(30 * DtoR)) +
(OTWDisp.ScaleWidth + 20) * Sin(FDSRecord.Roll *
DtoR) / 2
  rightX = OTWDisp.ScaleLeft + OTWDisp.ScaleWidth +
10
  rightY = OTWDisp.ScaleHeight / 2 * (1 +
Tan(FDSRecord.Pitch * DtoR) / Tan(30 * DtoR)) -
(OTWDisp.ScaleWidth + 20) * Sin(FDSRecord.Roll *
DtoR) / 2

' Draw a black line for the horizon
  OTWDisp.Line (leftX, leftY)-(rightX, rightY),
&H606060

' Fill the ground
  oldBrush = SelectObject(OTWDisp.hDC, greenBrush)
  leftX = OTWDisp.ScaleLeft + OTWDisp.ScaleWidth /
2
  leftY = (leftY + rightY) / 2 +
OTWDisp.ScaleHeight / 10
  result = ExtFloodFill(OTWDisp.hDC, leftX, leftY,
&HFFFFFF, 1)
  oldBrush = SelectObject(OTWDisp.hDC, oldBrush)
' Fill the sky
  oldBrush = SelectObject(OTWDisp.hDC, blueBrush)
  result = ExtFloodFill(OTWDisp.hDC, leftX, leftY -
OTWDisp.ScaleHeight / 5, &HFFFFFF, 1)
  oldBrush = SelectObject(OTWDisp.hDC, oldBrush)
End Sub

Sub QuitButton_Click ()
  Unload Main
End Sub

Sub SetupButton_Click ()
  TestControls.Visible = Not TestControls.Visible
  ConnectTimer.Enabled = False
  FDSTimer.Enabled = False
  FDSDataSock.Action = 1
  GPSDataSock.Action = 1
End Sub

Sub SysSetCommand_Click ()
  SystemSettings.Show 1
End Sub

Sub Test_Click ()
  Dim tempFDS As DataSrvrRec
  Dim tempGPS As GPSRec

  tempFDS.Pitch = testPitch
  tempFDS.Roll = testRoll

  If TestWPdist <= -2 Then
    TestWPdist = 7
  Else
    TestWPdist = TestWPdist - .5
  End If
  tempGPS.DisttoWP = TestWPdist
  tempGPS.AltErr = -1900
  If TestWPdist < 0 Then
    tempGPS.CurWPType = "MAHP"
    tempGPS.DisttoWP = 5 + TestWPdist
  ElseIf TestWPdist < 6 Then
    tempGPS.CurWPType = "MAP"
  Else
    tempGPS.CurWPType = "FAF"
  End If

  FDSRecord = tempFDS
  GPSRecord = tempGPS
  ProcessData
End Sub

Sub testPitch_Change ()
  Test = True
End Sub

Sub testRoll_Change ()
  Test = True
End Sub
```

```
OTW.FRM
..\COMMON\THREED.BAS
C:\WINDOWS\SYSTEM\DSSOCK.VBX
C:\WINDOWS\SYSTEM\CMDIALOG.VBX
C:\WINDOWS\SYSTEM\THREED.VBX
..\COMMON\SYSSET.BAS
..\COMMON\DATALINK.BAS
..\COMMON\VBDEF.BAS
..\COMMON\TRIGCODE.BAS
OTW.BAS
..\COMMON\SYSSET.FRM
ProjWinSize=25,481,236,397
ProjWinShow=2
IconForm="Main"
Title="Simple OTW"
ExeName="OTW.EXE"
```

SERIAL.BAS

```
Option Explicit

Declare Function ComOutput Lib "MSComm" (ByVal hWnd
As Integer, lpData As Any, ByVal cbData As Integer)
As Integer
Declare Function ComInput Lib "MSComm" (ByVal hWnd
As Integer, lpData As Any, ByVal cbData As Integer)
As Integer

Global GPSBox As Integer
Global Const NONEType = 0
Global Const KLNType = 1
Global Const M3Type = 2
Global Const G155Type = 3
Global Const FrascaType = 4

Global InSerialData As String

Global STX As String
Global ETX As String
Global CRLF As String
Global CR As String


Sub InitSerial ()
  STX = Chr$(2)
  ETX = Chr$(3)
  CRLF = Chr$(13) + Chr$(10)
  CR = Chr$(13)
End Sub

Function M3CheckSum (OutString As String) As String
  Dim i As Integer
  Dim csum As Integer
  Dim temp As Integer
  Dim hbyte As String
  Dim lbyte As String

  csum = 0
  For i = 1 To Len(OutString)
    csum = csum Xor Asc(Mid$(OutString, i, 1))
  Next i
  temp = Fix(csum / 16)
  If (temp < 10) Then
    hbyte = Chr$(Asc("0") + temp)
  Else
    hbyte = Chr$(Asc("A") + temp - 10)
  End If
  temp = Fix(csum Mod 16)
  If (temp < 10) Then
    lbyte = Chr$(Asc("0") + temp)
  Else
    lbyte = Chr$(Asc("A") + temp - 10)
  End If
  M3CheckSum = "*" + hbyte + lbyte
End Function

Sub Parse155Data (Message As String, GPSData As
GPSRec, DSData As DataSrvrRec)

End Sub

Sub ParseKLNData (Message As String, GPSData As
GPSRec, DSData As DataSrvrRec)
' NOTE: This data is intended to be read *FROM*
'       the KLN 90B.
  Static CurWPIdent As String
  Dim msgitem As String
  Dim msgdata As String
  Dim pos As Integer
  Dim temp As Double
  Dim DTK As Double
  Dim LMV As Double

' Read in the data items
  msgitem = Left$(Message, 1)
  While (msgitem <> ETX)
'   Grab the data for this item
    pos = InStr(Message, CR)
    msgdata = Mid$(Message, 2, pos - 2)
    Message = Mid$(Message, pos + 1)
'   Switch on the item designator
    Select Case msgitem
      Case "A"
'       Present Latitude
        msgitem = Left$(msgdata, 1)
        If (msgitem <> "-") Then
          temp = Val(Mid$(msgdata, 3, 2)) +
Val(Mid$(msgdata, 6, 4)) * .6
          If msgitem = "S" Then
            temp = -temp
          End If
          DSData.Latitude = temp
        End If
      Case "B"
'       Present Longitude
        msgitem = Left$(msgdata, 1)
        If (msgitem <> "-") Then
          temp = Val(Mid$(msgdata, 3, 3)) +
Val(Mid$(msgdata, 7, 4)) * .6
          If msgitem = "E" Then
            temp = -temp
          End If
          DSData.Longitude = temp
        End If
      Case "C"
'       Track (magnetic)
        GPSData.Track = Val(msgdata)
      Case "D"
'       Ground Speed (knots)
        GPSData.GrndSpeed = Val(msgdata)
      Case "E"
'       Distance to active Waypoint
        GPSData.DisttoWP = Val(msgdata) / 10
```

```
      Case "G"
'       Cross Track Error
        msgitem = Left$(msgdata, 1)
        If (msgitem <> "-") Then
          temp = Val(Mid$(msgdata, 2)) / 100
          If (msgitem = "L") Then
            temp = -temp
          End If
          GPSData.XTE = temp
        End If
      Case "I"
'       Desired Track (magnetic)
        DTK = Val(msgdata) / 10
      Case "K"
'       Active Waypoint
        GPSData.CurWPIdent = msgdata
      Case "L"
'       Bearing to active waypoint
        GPSData.HdgtoWP = Val(msgdata) / 10
      Case "Q"
'       Magnetic Variation
        msgitem = Left$(msgdata, 1)
        If (msgitem <> "-") Then
          temp = Val(Mid$(msgdata, 2)) / 10
          If (msgitem = "W") Then
            temp = -temp
          End If
          LMV = temp
        End If
      Case "T"
'       Dashes - ignore
      Case "l"
'       Distance to destination
        GPSData.DistToEnd = Val(msgdata) / 10
      Case "u"
'       Self-test data
      Case "w"
'       Flight Plan Data
    End Select
    msgitem = Left$(Message, 1)
  Wend

' Convert magnetic values to true
  temp = GPSData.Track + LMV
  If (temp > 360) Then temp = temp - 360
  If (temp < 0) Then temp = temp + 360
  GPSData.Track = temp
  temp = GPSData.HdgtoWP + LMV
  If (temp > 360) Then temp = temp - 360
  If (temp < 0) Then temp = temp + 360
  GPSData.HdgtoWP = temp
  temp = DTK + LMV
  If (temp > 360) Then temp = temp - 360
  If (temp < 0) Then temp = temp + 360
  DTK = temp
  GPSData.TAE = DTK - GPSData.HdgtoWP

' If TAE is greater than 90 then reverse XTE
  If (GPSData.TAE > 90) Or (GPSData.TAE < -90) Then
    GPSData.XTE = -GPSData.XTE
  End If

' Calculate as much of the missing data as possible
  If (GPSData.CurWPIdent <> CurWPIdent) Then
    GPSData.LastWPIdent = CurWPIdent
    CurWPIdent = GPSData.CurWPIdent
  End If
  GPSData.XAE = DTK - GPSData.Track
  GPSData.TrkDistToWP = Sqr(GPSData.DisttoWP ^ 2 -
GPSData.XTE ^ 2)
  If GPSData.GrndSpeed > 0 Then
    GPSData.ETW = (GPSData.DisttoWP /
GPSData.GrndSpeed) * 60
  Else
    GPSData.ETW = 0
  End If
  GPSData.TimeStamp = GetTickCount()
  DSData.TimeStamp = GPSData.TimeStamp
End Sub

Sub ParseM3Data (Mesage As String, GPSData As
GPSRec, DSData As DataSrvrRec)
End Sub

Sub ReadSerialStr (Message As String)
  Static SerialIn As String
  Dim pos As Integer

' Put an ETX at the start of SerialIn to make sure
it is not NULL
  SerialIn = ETX + SerialIn

' Read Data until the first <STX>
  pos = InStr(SerialIn, STX)
  While (pos = 0)
    SerialIn = Serial.Comm1.Input
    pos = InStr(SerialIn, STX)
    DoEvents
  Wend

' Throw away the STX and everything before it
  SerialIn = Mid$(SerialIn, pos + 1)

' Read until the first <ETX>
  pos = 0
  While (pos = 0)
    SerialIn = SerialIn + Serial.Comm1.Input
    pos = InStr(SerialIn, ETX)
    DoEvents
  Wend

' Save the message
  Message = Left$(SerialIn, pos)

' Save anything after the ETX for the next message
  SerialIn = Mid$(SerialIn, pos + 1)
End Sub

Sub Write155Data (NavData As NavRec)
' NOTE: This data is intended to be transmitted to
'       the inflight computer and is a simulation
of the
'       data normally received *FROM* the Garmin
155

End Sub

Sub WriteFrascaData ()
  Dim SerialOut As String
  Dim numBytes As Integer

' Start of Text
  SerialOut = STX

' Experiment Name
  SerialOut = SerialOut + "A" + Trim$(Main.ExpName)
+ CR

' Profile Name
  SerialOut = SerialOut + "B" +
Trim$(Main.ProfileName) + CR

' Flight Plan Name
```

```
  SerialOut = SerialOut + "C" +
Trim$(Main.FltPlnName) + CR

' Control Console Status
  SerialOut = SerialOut + "D" +
Trim$(Main.CtrlConStatus) + CR

' Data Server Data
  SerialOut = SerialOut + "E" +
Trim$(Main.DataSrvrData) + CR

' Data Server Status
  SerialOut = SerialOut + "G" +
Trim$(Main.DataSrvrStatus) + CR

' GPS Data
  SerialOut = SerialOut + "H" + Trim$(Main.GPSData)
+ CR

' GPS Status
  SerialOut = SerialOut + "I" +
Trim$(Main.GPSStatus) + CR

' End of Text
'   numBytes = Len(SerialOut)
  SerialOut = SerialOut + ETX
'   numBytes = ComOutput(Serial.Comm1.hWnd, ETX, 1)

  'Debug.Print "------------" + CR + SerialOut
'   numBytes = ComOutput(Serial.Comm1.hWnd,
SerialOut, numBytes + 1)
  Serial.Comm1.Output = SerialOut
End Sub

Sub WriteKLNData (NavData As NavRec)
' NOTE: This data is intended to be transmitted to
'       the inflight computer and is a simulation
of the
'       data normally received *FROM* the KLN 90B
  Dim SerialOut As String
  Dim temp As Double
  Dim LMV As Double
  Dim numBytes As Integer

  LMV = NavData.FlightPlan.MagVariation

' Start of Text
  SerialOut = STX

' Present Latitude
  SerialOut = SerialOut + "A"
  If NavData.Latitude < 0 Then
    SerialOut = SerialOut + "S "
  Else
    SerialOut = SerialOut + "N "
  End If
  temp = Abs(NavData.Latitude)
  SerialOut = SerialOut + Format$(temp, "00") + " "
  temp = (temp - Fix(temp)) * 6000
  SerialOut = SerialOut + Format$(temp, "0000") +
CR

' Present Longitude
  SerialOut = SerialOut + "B"
  If NavData.Latitude < 0 Then
    SerialOut = SerialOut + "E "
  Else
    SerialOut = SerialOut + "W "
  End If
  temp = Abs(NavData.Latitude)
  SerialOut = SerialOut + Format$(temp, "00") + " "
  temp = (temp - Fix(temp)) * 6000
  SerialOut = SerialOut + Format$(temp, "0000") +
CR

' Track (magnetic)
  temp = NavData.GPSData.Track - LMV
  If (temp > 360) Then temp = temp - 360
  If (temp < 0) Then temp = temp + 360
  SerialOut = SerialOut + "C" + Format$(temp,
"000") + CR

' Ground Speed (knots)
  temp = NavData.GPSData.GrndSpeed
  If (temp < 0) Then temp = 0
  If (temp > 999) Then temp = 999
  SerialOut = SerialOut + "D" + Format$(temp,
"000") + CR

' Distance to active Waypoint
  temp = NavData.GPSData.DisttoWP * 10
  If (temp < 0) Then temp = 0
  If (temp > 99999) Then temp = 99999
  SerialOut = SerialOut + "E" + Format$(temp,
"00000") + CR

' Cross Track Error
  SerialOut = SerialOut + "G"
  If (NavData.GPSData.TAE > -90) And
(NavData.GPSData.TAE < 90) Then
    If (NavData.GPSData.XTE > 0) Then
      SerialOut = SerialOut + "L"
    Else
      SerialOut = SerialOut + "R"
    End If
  Else
    If (NavData.GPSData.XTE > 0) Then
      SerialOut = SerialOut + "R"
    Else
      SerialOut = SerialOut + "L"
    End If
  End If
  temp = Abs(NavData.GPSData.XTE) * 100
  SerialOut = SerialOut + Format$(temp, "0000") +
CR

' Desired Track (magnetic)
  temp = ((NavData.GPSData.TAE +
NavData.GPSData.HdgtoWP) - LMV) * 10
  If (temp > 3600) Then temp = temp - 3600
  If (temp < 0) Then temp = temp + 3600
  SerialOut = SerialOut + "I" + Format$(temp,
"0000") + CR

' Active Waypoint
  SerialOut = SerialOut + "K" +
Left$(NavData.GPSData.CurWPIdent, 5) + CR

' Bearing to active waypoint
  temp = (NavData.GPSData.HdgtoWP - LMV) * 10
  If (temp > 3600) Then temp = temp - 3600
  If (temp < 0) Then temp = temp + 3600
  SerialOut = SerialOut + "L" + Format$(temp,
"0000") + CR

' Magnetic Variation
  SerialOut = SerialOut + "Q"
  If LMV < 0 Then
    SerialOut = SerialOut + "W"
  Else
    SerialOut = SerialOut + "E"
  End If
  temp = Abs(LMV) * 10
  SerialOut = SerialOut + Format$(temp, "000") + CR
```

```
' Dashes
  SerialOut = SerialOut + "T---------"

' Distance to destination
  temp = Abs(NavData.GPSData.DistToEnd) * 10
  If (temp > 999999) Then temp = 999999
  SerialOut = SerialOut + "1" + Format$(temp,
"000000") + CR

' Altitude (added to help out Frank - not normally
part of
' KLN data stream)
  SerialOut = SerialOut

  numBytes = Len(SerialOut)

' Self-test data

' Flight Plan Data

' End of Text
  SerialOut = SerialOut + ETX

  numBytes = ComOutput(Serial.Comm1.hWnd,
SerialOut, numBytes + 1)
End Sub

Sub WriteM3Data (NavData As NavRec)
' NOTE:  This data is intended to be transmitted
*TO*
'        the M3 box.  It drives this box to give
'        appropriate information inside the FRASCA
cockpit.
  Dim SerialOut As String
  Dim Message As String
  Dim temp As Double
  Dim angle As Double
  Dim numBytes As Integer

  If (Not Serial.Comm1.PortOpen) Then
    Debug.Print ("Port not open")
  End If

' Checksum and Output
'  SerialOut =
"PMVXG,022,123456.89,01.0,01.0,01.0,10,11,12,13,14,
15"
  SerialOut = "PMVXG,022,123456.89,01.0,-
0.1,01.0,00.1000,00.1000,1,00.500,100"
  SerialOut = "$" + SerialOut +
M3CheckSum(SerialOut) + CRLF
  Message = SerialOut
  SerialOut = "PMVXG,030,SIMS,SIM,FRASCA,FRASCA"
  SerialOut = "$" + SerialOut +
M3CheckSum(SerialOut) + CRLF
  Message = Message + SerialOut
'  numBytes = Len(SerialOut)
'  numBytes = ComOutput(Serial.Comm1.hWnd,
SerialOut, numBytes)
'  Serial.Comm1.Output = SerialOut
'  DoEvents

  SerialOut = "PMVXG,021,123456.89,"

' Present Latitude
  temp = Abs(NavData.Latitude)
  SerialOut = SerialOut + Format$(Fix(temp), "00")
  temp = (temp - Fix(temp)) * 60
  SerialOut = SerialOut + Format$(temp, "00.0000")
  If NavData.Latitude < 0 Then
    SerialOut = SerialOut + ",S,"
  Else
    SerialOut = SerialOut + ",N,"
  End If

' Present Longitude
  temp = Abs(NavData.Longitude)
  SerialOut = SerialOut + Format$(Fix(temp), "000")
  temp = (temp - Fix(temp)) * 60
  SerialOut = SerialOut + Format$(temp, "00.0000")
  If NavData.Longitude < 0 Then
    SerialOut = SerialOut + ",E,"
  Else
    SerialOut = SerialOut + ",W,"
  End If

' Current Altitude
  temp = NavData.Altitude * .3048
  SerialOut = SerialOut + Format$(temp, "00000.0;-
0000.0") + ","

' Geoidal Height
  SerialOut = SerialOut + "0000.0,"

  angle = (90 - NavData.GPSData.Track) * DtoR

' East/West Velocity
  temp = NavData.GPSData.GrndSpeed * Cos(angle) *
6080 * .3048 / 3600
  SerialOut = SerialOut + Format$(temp, "0000.0;-
000.0") + ","

' North/South Velocity
  temp = NavData.GPSData.GrndSpeed * Sin(angle) *
6080 * .3048 / 3600
  SerialOut = SerialOut + Format$(temp, "0000.0;-
000.0") + ","

' Nav mode
  SerialOut = SerialOut + "03"

' Checksum and Output
  SerialOut = "$" + SerialOut +
M3CheckSum(SerialOut) + CRLF
  Message = Message + SerialOut
'  numBytes = Len(SerialOut)
'  numBytes = ComOutput(Serial.Comm1.hWnd,
SerialOut, numBytes)
  Serial.Comm1.Output = Message
End Sub

Sub WriteSerialData (NavData As NavRec)
  If Serial.Comm1.PortOpen Then
    If GPSBox = KLNType Then
      WriteKLNData NavData
    ElseIf GPSBox = M3Type Then
      WriteM3Data NavData
    ElseIf GPSBox = G155Type Then
      Write155Data NavData
    ElseIf GPSBox = FrascaType Then
      WriteFrascaData
    End If
  End If
End Sub
```

```
VERSION 2.00
Begin Form Serial
   BackColor       =   &H00C0C0C0&
   Caption         =   "Open Serial Port"
   ClientHeight    =   2655
   ClientLeft      =   2505
   ClientTop       =   2730
   ClientWidth     =   3630
   Height          =   3060
   Left            =   2445
   LinkTopic       =   "Form1"
   ScaleHeight     =   2655
   ScaleWidth      =   3630
   Top             =   2385
   Width           =   3750
   Begin SSPanel ChoosePort
      Alignment       =   6  'Center - TOP
      BevelInner      =   1  'Inset
      BevelOuter      =   0  'None
      Caption         =   "Serial Port"
      Height          =   1755
      Left            =   60
      TabIndex        =   7
      Top             =   90
      Width           =   2175
      Begin SSOption CommPort
         Caption         =   "1"
         Height          =   315
         Index           =   0
         Left            =   90
         TabIndex        =   12
         TabStop         =   0   'False
         Top             =   270
         Width           =   405
      End
      Begin SSOption CommPort
         Caption         =   "2"
         Height          =   315
         Index           =   1
         Left            =   540
         TabIndex        =   11
         Top             =   270
         Value           =   -1  'True
         Width           =   405
      End
      Begin SSOption CommPort
         Caption         =   "3"
         Height          =   315
         Index           =   2
         Left            =   990
         TabIndex        =   10
         TabStop         =   0   'False
         Top             =   270
         Width           =   405
      End
      Begin SSOption CommPort
         Caption         =   "4"
         Height          =   315
         Index           =   3
         Left            =   1440
         TabIndex        =   9
         TabStop         =   0   'False
         Top             =   270
         Width           =   405
      End
      Begin TextBox Settings
         Height          =   315
         Left            =   840
         TabIndex        =   8
         Text            =   "4800,n,8,1"
         Top             =   660
         Width           =   1185
      End
      Begin Label Label1
         Alignment       =   1  'Right Justify
         Caption         =   "Settings"
         Height          =   192
         Left            =   60
         TabIndex        =   14
         Top             =   720
         Width           =   732
      End
      Begin Label Label2
         Caption         =   "M3 - 4800,n,8,1
KLN - 9600,n,8,2  Frasca - 19200,n,8,1"
         Height          =   585
         Left            =   60
         TabIndex        =   13
         Top             =   1050
         Width           =   1995
         WordWrap        =   -1  'True
      End
   End
   Begin PictureBox SelectPanel
      BackColor       =   &H00C0C0C0&
      BorderStyle     =   0  'None
      Height          =   705
      Left            =   30
      ScaleHeight     =   705
      ScaleWidth      =   3525
      TabIndex        =   5
      Top             =   1590
      Visible         =   0   'False
      Width           =   3525
      Begin CommandButton ChooseKLN
         Caption         =   "&KLN 90B"
         Height          =   345
         Left            =   1170
         TabIndex        =   3
         Top             =   360
         Width           =   1185
      End
      Begin CommandButton Choose155
         Caption         =   "&Garmin 155"
         Height          =   345
         Left            =   0
         TabIndex        =   4
         Top             =   360
         Width           =   1185
      End
      Begin CommandButton ChooseM3
         Caption         =   "NStar &M3"
         Height          =   345
         Left            =   2340
         TabIndex        =   15
         Top             =   360
         Width           =   1185
      End
      Begin CommandButton ChooseFrasca
         Caption         =   "&Frasca"
         Height          =   345
         Left            =   2340
         TabIndex        =   6
         Top             =   30
         Width           =   1185
      End
   End
   Begin MSComm Comm1
```

```
      CommPort        =   2
      Interval        =   1000
      Left            =   2670
      OutBufferSize   =   2048
      RThreshold      =   1
      Settings        =   "4800,n,8,1"
      Top             =   960
   End
   Begin CommandButton DoOpen
      Caption         =   "&Open Port"
      Default         =   -1  'True
      Height          =   345
      Left            =   2310
      TabIndex        =   1
      Top             =   90
      Width           =   1275
   End
   Begin CommandButton DoCancel
      Cancel          =   -1  'True
      Caption         =   "&Cancel"
      Height          =   345
      Left            =   2310
      TabIndex        =   0
      Top             =   450
      Width           =   1275
   End
   Begin Label Message
      Alignment       =   2  'Center
      Caption         =   "Serial Link is Closed"
      Height          =   225
      Left            =   90
      TabIndex        =   2
      Top             =   2340
      Width           =   3435
   End
End
Option Explicit

Sub Choose155_Click ()
  GPSBox = G155Type
  Serial.Hide
End Sub

Sub ChooseFrasca_Click ()
  GPSBox = FrascaType
  Serial.Hide
End Sub

Sub ChooseKLN_Click ()
  GPSBox = KLNType
  Serial.Hide
End Sub

Sub ChooseM3_Click ()
  GPSBox = M3Type
  Serial.Hide
End Sub

Sub Comm1_OnComm ()
  Select Case Comm1.CommEvent
'   Errors
    Case MSCOMM_ER_BREAK    ' A Break was received
'     Code to handle a BREAK goes here.
    Case MSCOMM_ER_CDTO     ' CD (RLSD) Timeout.
    Case MSCOMM_ER_CTSTO    ' CTS Timeout.
    Case MSCOMM_ER_DSRTO    ' DSR Timeout.
    Case MSCOMM_ER_FRAME    ' Framing Error
    Case MSCOMM_ER_OVERRUN  ' Data Lost.
    Case MSCOMM_ER_RXOVER   ' Receive buffer
overflow.
    Case MSCOMM_ER_RXPARITY ' Parity Error.
    Case MSCOMM_ER_TXFULL   ' Transmit buffer full.
'   Events
    Case MSCOMM_EV_CD       ' Change in the CD
line.
    Case MSCOMM_EV_CTS      ' Change in the CTS
line.
    Case MSCOMM_EV_DSR      ' Change in the DSR
line.
    Case MSCOMM_EV_RING     ' Change in the Ring
Indicator.
    Case MSCOMM_EV_RECEIVE  ' Received RThreshold #
of chars.
      Comm1.RThreshold = 0  ' Disable further
receive events
      'DoEvents
      ReadSerialStr InSerialData
      ProcessSerial InSerialData
      Comm1.RThreshold = 1  ' Re-enable receive
events
    Case MSCOMM_EV_SEND     ' There are SThreshold
number of
                ' characters in the transmit buffer.
  End Select
End Sub

Sub DoCancel_Click ()
  If Comm1.PortOpen Then
    Comm1.PortOpen = False
    Message = "Serial Link is Closed"
    ChooseKLN.Visible = False
    Choose155.Visible = False
  End If
  Serial.Hide
  GPSBox = NONEType
End Sub

Sub DoOpen_Click ()
  Dim i As Integer
  If Comm1.PortOpen Then
    Comm1.PortOpen = False
    Message = "Serial Link is Closed"
    DoOpen.Caption = "&Open Port"
    SelectPanel.Visible = False
  Else
    For i = 0 To 3
      If CommPort(i) Then
   Comm1.CommPort = i + 1
   Exit For
      End If
    Next i
    Comm1.Settings = Settings

    On Error Resume Next
    Comm1.PortOpen = True
    If Err <> 0 Then
      Message = Error
    Else
      Message = "Serial Link is Open"
      DoOpen.Caption = "&Close Port"
      SelectPanel.Visible = True
'     Set to receive
      Comm1.RThreshold = 1
    End If
    On Error GoTo 0
  End If
End Sub

Sub Form_Load ()
  Serial.Move ((Screen.Width - Serial.Width) / 2),
((Screen.Height - Serial.Height) / 2),
Serial.Width, Serial.Height
  InitSerial
End Sub
```

SYSSET.BAS

```
Option Explicit

Declare Function GetProfileString Lib "Kernel" (ByVal lpAppName As String, ByVal lpKeyName As String,
ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Integer) As Integer
Declare Function GetProfileInt Lib "Kernel" (ByVal lpAppName As String, ByVal lpKeyName As String, ByVal
nDefault As Integer) As Integer
Declare Function WriteProfileString Lib "Kernel" (ByVal lpApplicationName As String, ByVal lpKeyName As
String, ByVal lpString As String) As Integer
Declare Function SendMessage Lib "User" (ByVal hWnd As Integer, ByVal wMsg As Integer, ByVal wParam As
Integer, lParam As Any) As Long
Declare Function PostAppMessage Lib "User" (ByVal hTask As Integer, ByVal wMsg As Integer, ByVal wParam
As Integer, lParam As Any) As Integer
Declare Function GetTickCount Lib "User" () As Long
```

SYSSET.FRM

```
VERSION 4.00
Begin VB.Form SystemSettings
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   BorderStyle     =   0  'None
   Caption         =   "System Settings"
   ClientHeight    =   6360
   ClientLeft      =   135
   ClientTop       =   825
   ClientWidth     =   9360
   ControlBox      =   0   'False
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   1
      weight          =   700
      size            =   8.25
      underline       =   0   'False
      italic          =   0   'False
      strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   6765
   HelpContextID   =   99000
   Left            =   75
   LinkTopic       =   "Form2"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   6360
   ScaleWidth      =   9360
   Top             =   480
   Width           =   9480
   Begin Threed.SSPanel Panel3D3
      Height          =   1095
      Left            =   120
      TabIndex        =   0
      Top             =   5160
      Width           =   7215
      _version        =   65536
      _extentx        =   12726
      _extenty        =   1931
      _stockprops     =   15
      caption         =   "System Settings"
      backcolor       =   -2147483633
      BeginProperty font {FB8F0823-0164-101B-84ED-
08002B2EC713}
         name            =   "MS Sans Serif"
         charset         =   1
         weight          =   700
         size            =   12
         underline       =   0   'False
         italic          =   0   'False
         strikethrough   =   0   'False
      EndProperty
      bevelouter      =   0
      font3d          =   3
   End
   Begin Threed.SSPanel Panel3D2
      Height          =   1095
      Left            =   7320
      TabIndex        =   1
      Top             =   5160
      Width           =   1935
      _version        =   65536
      _extentx        =   3413
      _extenty        =   1931
      _stockprops     =   15
      backcolor       =   -2147483633
      bevelouter      =   0
      bevelinner      =   1
      font3d          =   1
      alignment       =   0
      Begin VB.CommandButton ClosePreferences
         Appearance      =   0  'Flat
         BackColor       =   &H80000005&
         Caption         =   "Close"
         Height          =   375
         Left            =   120
         TabIndex        =   2
         Top             =   600
         Width           =   1695
      End
      Begin VB.CommandButton PrinterSetup
         Appearance      =   0  'Flat
         BackColor       =   &H80000005&
         Caption         =   "Printer Setup >>"
         Height          =   375
         Left            =   120
         TabIndex        =   3
         Top             =   120
         Width           =   1695
      End
   End
   Begin Threed.SSPanel Panel3D1
      Height          =   5055
      Left            =   120
      TabIndex        =   8
      Top             =   120
      Width           =   9135
      _version        =   65536
      _extentx        =   16113
      _extenty        =   8916
      _stockprops     =   15
      backcolor       =   -2147483633
      bevelouter      =   0
```

```
bevelinner      =   1
font3d          =   1
alignment       =   0
Begin VB.TextBox GPSIP
   Appearance      =   0  'Flat
   Height          =   285
   Left            =   1680
   TabIndex        =   38
   Text            =   "152.122.10.105"
   Top             =   2520
   Width           =   2865
End
Begin VB.TextBox GPSPort
   Appearance      =   0  'Flat
   Height          =   285
   Left            =   6180
   TabIndex        =   37
   Text            =   "15001"
   Top             =   2520
   Width           =   1515
End
Begin VB.TextBox DataSrvrPort
   Appearance      =   0  'Flat
   Height          =   285
   Left            =   6180
   TabIndex        =   34
   Text            =   "15000"
   Top             =   570
   Width           =   1515
End
Begin VB.TextBox DataSrvrIP
   Appearance      =   0  'Flat
   Height          =   285
   Left            =   1680
   TabIndex        =   33
   Text            =   "152.122.10.104"
   Top             =   570
   Width           =   2865
End
Begin VB.CommandButton DataSrvrConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   25
   Top             =   90
   Width           =   855
End
Begin VB.CommandButton InstSrvrConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   24
   Top             =   1050
   Width           =   855
End
Begin VB.CommandButton LightSrvrConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   23
   Top             =   2970
   Width           =   855
End
Begin VB.CommandButton GPSSrvrConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   22
   Top             =   2010
   Width           =   855
End
Begin VB.CommandButton PanelConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   21
   Top             =   3930
   Width           =   855
End
Begin VB.CommandButton WindSrvrConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   20
   Top             =   3450
   Width           =   855
End
Begin VB.CommandButton CtrlConDBOpen
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Open"
   Height          =   375
   Left            =   8160
   TabIndex        =   19
   Top             =   4410
   Width           =   855
End
Begin VB.TextBox CtrlConDB
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   Height          =   285
   Left            =   1680
   LinkItem        =   "WindSpeed"
   TabIndex        =   18
   Top             =   4440
   Width           =   6015
End
Begin VB.CommandButton BrowseDBFiles
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "..."
   Height          =   255
   Left            =   7800
   TabIndex        =   17
   Top             =   4440
   Width           =   255
End
Begin VB.CommandButton CtrlConConnect
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "Save"
   Height          =   375
   Left            =   8160
   TabIndex        =   16
   Top             =   1530
   Width           =   855
End
Begin VB.ComboBox DataSrvrPath
   Appearance      =   0  'Flat
   Height          =   315
   Left            =   1680
   TabIndex        =   15
```

```
      Text            =   "Combo1"
      Top             =   120
      Width           =   6375
   End
   Begin VB.ComboBox InstSrvrPath
      Appearance      =   0  'Flat
      Height          =   315
      Left            =   1680
      TabIndex        =   14
      Text            =   "Combo1"
      Top             =   1080
      Width           =   6375
   End
   Begin VB.ComboBox CtrlConPath
      Appearance      =   0  'Flat
      Height          =   315
      Left            =   1680
      TabIndex        =   13
      Text            =   "Combo1"
      Top             =   1560
      Width           =   6375
   End
   Begin VB.ComboBox GPSPath
      Appearance      =   0  'Flat
      Height          =   315
      Left            =   1680
      TabIndex        =   12
      Text            =   "Combo1"
      Top             =   2040
      Width           =   6375
   End
   Begin VB.ComboBox LightPath
      Appearance      =   0  'Flat
      Height          =   315
      Left            =   1680
      TabIndex        =   11
      Text            =   "Combo1"
      Top             =   3000
      Width           =   6375
   End
   Begin VB.ComboBox WindPath
      Appearance      =   0  'Flat
      Height          =   315
      Left            =   1680
      TabIndex        =   10
      Text            =   "Combo1"
      Top             =   3480
      Width           =   6375
   End
   Begin VB.ComboBox PanelPath
      Appearance      =   0  'Flat
      Height          =   315
      Left            =   1680
      TabIndex        =   9
      Text            =   "Combo1"
      Top             =   3960
      Width           =   6375
   End
   Begin VB.Label Label12
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "GPS IP"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   35
      Top             =   2550
      Width           =   630
   End
   Begin VB.Label Label11
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "GPS Port"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   4650
      TabIndex        =   36
      Top             =   2550
      Width           =   795
   End
   Begin VB.Label Label10
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Data Server Port"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   4650
      TabIndex        =   32
      Top             =   600
      Width           =   1440
   End
   Begin VB.Label Label9
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Data Server IP"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   31
      Top             =   600
      Width           =   1275
   End
   Begin VB.Label Label1
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Data Server"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   4
      Top             =   150
      Width           =   1035
   End
   Begin VB.Label Label3
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Instrument Server"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   5
      Top             =   1110
      Width           =   1515
   End
   Begin VB.Label Label4
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Light"
      ForeColor       =   &H80000008&
      Height          =   195
```

```
      Left            =   120
      TabIndex        =   6
      Top             =   3030
      Width           =   435
   End
   Begin VB.Label Label5
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "GPS"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   30
      Top             =   2070
      Width           =   390
   End
   Begin VB.Label Label6
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Panel"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   29
      Top             =   3990
      Width           =   495
   End
   Begin VB.Label Label7
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Wind"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   28
      Top             =   3510
      Width           =   450
   End
   Begin VB.Label Label2
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Database File"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   27
      Top             =   4470
      Width           =   1185
   End
   Begin VB.Label Label8
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000005&
      BackStyle       =   0  'Transparent
      Caption         =   "Control Console"
      ForeColor       =   &H80000008&
      Height          =   195
      Left            =   120
      TabIndex        =   26
      Top             =   1590
      Width           =   1350
   End
End
Begin Threed.SSPanel MainPanel
      Height          =   6375
      Left            =   0
      TabIndex        =   7
      Top             =   0
      Width           =   9375
      _version        =   65536
      _extentx        =   16536
      _extenty        =   11245
      _stockprops     =   15
      backcolor       =   -2147483633
      bevelwidth      =   2
      font3d          =   3
   End
   Begin MSComDlg.CommonDialog PrintSet
      Left            =   10080
      Top             =   6120
      _version        =   65536
      _extentx        =   847
      _extenty        =   847
      _stockprops     =   0
   End
   Begin MSComDlg.CommonDialog CtrlConDBDialog
      Left            =   9480
      Top             =   6120
      _version        =   65536
      _extentx        =   847
      _extenty        =   847
      _stockprops     =   0
   End
End
Attribute VB_Name = "SystemSettings"
Attribute VB_Creatable = False
Attribute VB_Exposed = False
Option Explicit

Dim CtrlConShareDB As Database
Dim ShareList As Snapshot

Private Sub BrowseDBFiles_Click()
  CtrlConDBDialog.DefaultExt = "MDB"
  CtrlConDBDialog.Filter = "Access Database|*.MDB"
  CtrlConDBDialog.DialogTitle = "Set Control
Console Database"
  CtrlConDBDialog.Flags = OFN_NOREADONLYRETURN Or
OFN_OVERWRITEPROMPT Or OFN_PATHMUSTEXIST
  CtrlConDBDialog.CancelError = True
  On Error Resume Next
  CtrlConDBDialog.Action = DLG_FILE_OPEN
  If Err = 0 Then
    CtrlConDB.Text = CtrlConDBDialog.FileName
  End If
  On Error GoTo 0
End Sub

Private Sub ClosePreferences_Click()
  SystemSettings.Hide
End Sub

Private Sub CtrlConConnect_Click()
  Dim i As Integer
  i = WriteProfileString("Frasca", "CtrlConPath",
CtrlConPath.Text)
End Sub

Private Sub CtrlConDBOpen_Click()
  Dim i As Integer
  i = WriteProfileString("Frasca", "CtrlConDB",
CtrlConDB.Text)
End Sub

Private Sub CtrlConPath_Change()
  CtrlConConnect.DEFAULT = True
```

```
End Sub

Private Sub CtrlConPath_Click()
  If Trim$(CtrlConPath.Text) = "[none]" Then
    CtrlConPath.Text = ""
  End If
End Sub

Private Sub DataSrvrConnect_Click()
  Dim i As Integer
  i = WriteProfileString("Frasca", "DataSrvrPath",
DataSrvrPath.Text)
  i = WriteProfileString("Frasca", "DataSrvrIP",
DataSrvrIP.Text)
  i = WriteProfileString("Frasca", "DataSrvrPort",
DataSrvrPort.Text)
End Sub

Private Sub DataSrvrPath_Change()
  DataSrvrConnect.DEFAULT = True
End Sub

Private Sub DataSrvrPath_Click()
  If Trim$(DataSrvrPath.Text) = "[none]" Then
    DataSrvrPath.Text = ""
  End If
End Sub

Private Sub Form_Load()
  Dim DataSrvrPathStr As String * 50
  Dim GPSPathStr As String * 50
  Dim LightPathStr As String * 50
  Dim WindPathStr As String * 50
  Dim PanelPathStr As String * 50
  Dim InstSrvrPathStr As String * 50
  Dim CtrlConPathStr As String * 50
  Dim CtrlConDBFile As String * 100
  Dim IPStr As String * 25
  Dim PortStr As String * 25
  Dim i As Integer

  SystemSettings.Move ((Screen.Width -
SystemSettings.Width) / 2), ((Screen.Height -
SystemSettings.Height) / 2), SystemSettings.Width,
SystemSettings.Height

  ' Load Paths

  i = GetProfileString("Frasca", "DataSrvrPath",
"DataSrvr|Main", DataSrvrPathStr, 50)
  SystemSettings.DataSrvrPath.Text =
DataSrvrPathStr
  i = GetProfileString("Frasca", "GPSPath",
"GPSSrvr|Main", GPSPathStr, 50)
  SystemSettings.GPSPath.Text = GPSPathStr
  i = GetProfileString("Frasca", "CtrlConPath",
"CtrlCon|Main", CtrlConPathStr, 50)
  SystemSettings.CtrlConPath.Text = CtrlConPathStr
  i = GetProfileString("Frasca", "LightPath",
"LightSrvr|Main", LightPathStr, 50)
  SystemSettings.LightPath.Text = LightPathStr
  i = GetProfileString("Frasca", "WindPath",
"WindSrvr|Main", WindPathStr, 50)
  SystemSettings.WindPath.Text = WindPathStr
  i = GetProfileString("Frasca", "PanelPath",
"PanlSrvr|Main", PanelPathStr, 50)
  SystemSettings.PanelPath.Text = PanelPathStr
  i = GetProfileString("Frasca", "InstSrvrPath",
"InstSrvr|Main", InstSrvrPathStr, 50)
  SystemSettings.InstSrvrPath.Text =
InstSrvrPathStr
  i = GetProfileString("Frasca", "CtrlConDB",
"ctrlcon.mdb", CtrlConDBFile, 100)
  SystemSettings.CtrlConDB.Text = CtrlConDBFile
  i = GetProfileString("Frasca", "DataSrvrIP",
"152.122.10.104", IPStr, 25)
  SystemSettings.DataSrvrIP.Text = IPStr
  i = GetProfileString("Frasca", "DataSrvrPort",
"15000", PortStr, 25)
  SystemSettings.DataSrvrPort.Text = PortStr
  i = GetProfileString("Frasca", "GPSIP",
"152.122.10.105", IPStr, 25)
  SystemSettings.GPSIP.Text = IPStr
  i = GetProfileString("Frasca", "GPSPort",
"15001", PortStr, 25)
  SystemSettings.GPSPort.Text = PortStr

  On Error Resume Next
  Set CtrlConShareDB =
OpenDatabase(SystemSettings.CtrlConDB.Text)
  Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='Data Server'));")
  ShareList.MoveFirst
  If Err = 0 Then
    While Not ShareList.EOF
      DataSrvrPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
    Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='Instrument Server'));")
    ShareList.MoveFirst
    While Not ShareList.EOF
      InstSrvrPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
    Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='Control Console'));")
    ShareList.MoveFirst
    While Not ShareList.EOF
      CtrlConPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
    Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='GPS'));")
    ShareList.MoveFirst
    While Not ShareList.EOF
      GPSPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
```

```
    Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='Wind'));")
    ShareList.MoveFirst
    While Not ShareList.EOF
      WindPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
    Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='Light'));")
    ShareList.MoveFirst
    While Not ShareList.EOF
      LightPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
    Set ShareList =
CtrlConShareDB.CreateSnapshot("SELECT  DISTINCTROW
Computer.ComputerID, Share.ShareName FROM Computer,
Share, Computer INNER JOIN Share ON
Computer.ComputerID = Share.ComputerID WHERE
((Share.ShareType='Flat Panel'));")
    ShareList.MoveFirst
    While Not ShareList.EOF
      PanelPath.AddItem "\\" +
Trim$(ShareList("Computer.ComputerID")) + "\NDDE$|"
+ Trim$(ShareList("Share.ShareName"))
      ShareList.MoveNext
    Wend
    DataSrvrPath.AddItem "[none]"
    InstSrvrPath.AddItem "[none]"
    CtrlConPath.AddItem "[none]"
    GPSPath.AddItem "[none]"
    WindPath.AddItem "[none]"
    LightPath.AddItem "[none]"
    PanelPath.AddItem "[none]"
  End If
  On Error GoTo 0
End Sub

Private Sub GPSPath_Change()
  GPSSrvrConnect.DEFAULT = True
End Sub

Private Sub GPSPath_Click()
  If Trim$(GPSPath.Text) = "[none]" Then
    GPSPath.Text = ""
  End If
End Sub

Private Sub GPSSrvrConnect_Click()
  Dim i As Integer

  i = WriteProfileString("Frasca", "GPSPath",
GPSPath.Text)
  i = WriteProfileString("Frasca", "GPSIP",
GPSIP.Text)
  i = WriteProfileString("Frasca", "GPSPort",
GPSPort.Text)
End Sub

Private Sub InstSrvrConnect_Click()
  Dim i As Integer
  i = WriteProfileString("Frasca", "InstSrvrPath",
InstSrvrPath.Text)
End Sub

Private Sub InstSrvrPath_Change()
  InstSrvrConnect.DEFAULT = True
End Sub

Private Sub InstSrvrPath_Click()
  If Trim$(InstSrvrPath.Text) = "[none]" Then
    InstSrvrPath.Text = ""
  End If

End Sub

Private Sub LightPath_Change()
  LightSrvrConnect.DEFAULT = True
End Sub

Private Sub LightPath_Click()
  If Trim$(LightPath.Text) = "[none]" Then
    LightPath.Text = ""
  End If
End Sub

Private Sub LightSrvrConnect_Click()
  Dim i As Integer
  i = WriteProfileString("Frasca", "LightPath",
LightPath.Text)
End Sub

Private Sub PanelConnect_Click()
  Dim i As Integer
  i = WriteProfileString("Frasca", "PanelPath",
PanelPath.Text)
End Sub

Private Sub PanelPath_Change()
  PanelConnect.DEFAULT = True
End Sub

Private Sub PanelPath_Click()
  If Trim$(PanelPath.Text) = "[none]" Then
    PanelPath.Text = ""
  End If
End Sub

Private Sub PrinterSetup_Click()
  PrintSet.DialogTitle = "Printer Setup"
  PrintSet.Flags = PD_PRINTSETUP
  PrintSet.Action = 5
End Sub

Private Sub WindPath_Change()
  WindSrvrConnect.DEFAULT = True
End Sub

Private Sub WindPath_Click()
  If Trim$(WindPath.Text) = "[none]" Then
    WindPath.Text = ""
  End If
End Sub

Private Sub WindSrvrConnect_Click()
  Dim i As Integer

  i = WriteProfileString("Frasca", "WindPath",
WindPath.Text)
End Sub
```

```
Sub EyeMat (M() As Double)
' [ 1 0 0 0 ]
' [ 0 1 0 0 ]
' [ 0 0 1 0 ]
' [ 0 0 0 1 ]

  M(1, 1) = 1: M(1, 2) = 0: M(1,3) = 0: M(1, 4) = 0
  M(2, 1) = 0: M(2, 2) = 1: M(2,3) = 0: M(2, 4) = 0
  M(3, 1) = 0: M(3, 2) = 0: M(3,3) = 1: M(3, 4) = 0
  M(4, 1) = 0: M(4, 2) = 0: M(4,3) = 0: M(4, 4) = 1
End Sub

Sub PerspMat (M() As Double, d As Double)
  Dim invD As Double
  Dim i As Integer
'
' [ 1  0  0  0 ]
' [ 0  1  0  0 ]
' [ 0  0  0  0 ]
' [ 0  0 1/d 1 ]

  If d = 0 Then
    invD = 0
  Else
    invD = 1 / d
  End If

  For i = 1 To 4
    M(4, i) = M(3, i) * invD + M(4, i)
  Next i
  For i = 1 To 4
    M(3, i) = 0
  Next i
End Sub

Sub ProjMat (M() As Double, zp As Double, Q As
Double, dx As Double, dy As Double, dz As Double)
  Dim Qdz As Double
  Dim dxDz As Double
  Dim tempM() As Double
  Dim i As Integer
  Dim j As Integer

  ReDim tempM(4, 4)

  If (dz = 0) Then
    Q = 0
    dxDz = 0
    dyDz = 0
  Else
    dxDz = dx / dz
    dyDz = dy / dz
  End If
  If (Q = 0) Then
    Qdz = 0
  Else
    Qdz = 1 / (Q * dz)
  End If

  For i = 1 To 4
    tempM(1,i) = M(1,i)-M(3,i)*dxDz+M(4,i)*zp*dxDz
  Next i
  For i = 1 To 4
    tempM(2,i) = M(2,i)-M(3,i)*dyDz+M(4,i)*zp*dyDz
  Next i
  For i = 1 To 4
    tempM(3,i)=-M(3,i)*zp*Qdz+M(4,i)*zp*(zp*Qdz+1)
  Next i
  For i = 1 To 4
    tempM(4, i) = -M(3, i) * Qdz + M(4, i) * (zp *
Qdz + 1)
  Next i

  For i = 1 To 4
    For j = 1 To 4
      M(i, j) = tempM(i, j)
    Next j
  Next i
End Sub

Sub RotXMat (M() As Double, angle As Double)
  Dim cosA As Double
  Dim sinA As Double
  Dim tempM() As Double
  Dim i As Integer
  Dim j As Integer

  ReDim tempM(2, 4)

  cosA = Cos(angle): sinA = Sin(angle)
  For i = 1 To 4
    tempM(1, i) = cosA * M(2, i) - sinA * M(3, i)
  Next i
  For i = 1 To 4
    tempM(2, i) = sinA * M(2, i) + cosA * M(3, i)
  Next i

  For i = 1 To 2
    For j = 1 To 4
      M(i + 1, j) = tempM(i, j)
    Next j
  Next i
End Sub

Sub RotYMat (M() As Double, angle As Double)
  Dim cosA As Double
  Dim sinA As Double
  Dim tempM() As Double
  Dim i As Integer
  Dim j As Integer

  ReDim tempM(2, 4)

  cosA = Cos(angle): sinA = Sin(angle)
  For i = 1 To 4
    tempM(1, i) = cosA * M(1, i) + sinA * M(3, i)
  Next i
  For i = 1 To 4
    tempM(2, i) = -sinA * M(1, i) + cosA * M(3, i)
  Next i

  For i = 0 To 1
    For j = 1 To 4
      M(2 * i + 1, j) = tempM(i, j)
    Next j
  Next i
End Sub

Sub RotZMat (M() As Double, angle As Double)
  Dim cosA As Double
  Dim sinA As Double
  Dim tempM() As Double
  Dim i As Integer
  Dim j As Integer

  ReDim tempM(2, 4)
```

THREED.BAS

```
  cosA = Cos(angle): sinA = Sin(angle)
  For i = 1 To 4
    tempM(1, i) = cosA * M(1, i) - sinA * M(2, i)
  Next i
  For i = 1 To 4
    tempM(2, i) = sinA * M(1, i) + cosA * M(2, i)
  Next i

  For i = 1 To 2
    For j = 1 To 4
      M(i, j) = tempM(i, j)
    Next j
  Next i
End Sub

Sub ScaleMat (M() As Double, x As Double, y As
Double, z As Double)
  M(1, 1) = x * M(1, 1): M(1, 2) = x * M(1, 2):
M(1, 3) = x * M(1, 3): M(1, 4) = x * M(1, 4)
  M(2, 1) = y * M(2, 1): M(2, 2) = y * M(2, 2):
M(2, 3) = y * M(2, 3): M(2, 4) = y * M(2, 4)
  M(3, 1) = z * M(3, 1): M(3, 2) = z * M(3, 2):
M(3, 3) = z * M(3, 3): M(3, 4) = z * M(3, 4)
End Sub

Sub Transform3D (x() As Double, M() As Double)
  Dim tempX() As Double
  Dim XLB As Integer
  Dim XUB As Integer
  Dim i As Integer
  Dim j As Integer
  Dim k As Integer

  ReDim tempX(4)
  XLB = LBound(x, 1)
  XUB = UBound(x, 1)
  For i = XLB To XUB
    For j = 1 To 4
      tempX(j) = 0
      For k = 1 To 4
        tempX(j) = tempX(j) + M(j, k) * x(i, k)
      Next k
    Next j
    For j = 1 To 3
      x(i, j) = tempX(j) / tempX(4)
    Next j
  Next i
End Sub

Sub TransMat (M() As Double, x As Double, y As
Double, z As Double)
  M(1, 4) = M(1, 4) + x
  M(2, 4) = M(2, 4) + y
  M(3, 4) = M(3, 4) + z
End Sub
```

TRIGCODE.BAS

```
Option Explicit

Global DtoR As Double
Global Deg2Rad As Double
Global PIo2 As Double
Const ZERO = 1E-20

Function asin (ByVal X As Double) As Double
  asin = atn2(X, Sqr(1 - X * X))
End Function

Function atn2 (ByVal Y As Double, ByVal X As
Double) As Double
  Dim tempAtn As Double
  If (Abs(X) < ZERO) Then
    tempAtn = Sgn(Y) * PIo2
  Else
    tempAtn = Atn(Y / X)
  End If
  If (X < 0) Then
    If (Y < 0) Then
      tempAtn = tempAtn - 2 * PIo2
    Else
      tempAtn = tempAtn + 2 * PIo2
    End If
  End If
  atn2 = tempAtn
End Function

Sub initTrig ()
  PIo2 = Atn(1) * 2
  DtoR = PIo2 / 90
  Deg2Rad = DtoR
End Sub

Function radialFromWP (ByVal Y As Double, ByVal X
As Double) As Double
  Dim tempAtnR As Double
  tempAtnR = atn2(-X, -Y)
  If (tempAtnR < 0) Then tempAtnR = tempAtnR + 4 *
PIo2
  radialFromWP = tempAtnR
End Function

Function recipAng (Angle As Double) As Double
  If Angle < 2 * PIo2 Then
    recipAng = 2 * PIo2 + Angle
  Else
    recipAng = Angle - 2 * PIo2
  End If
End Function

Function sgnDiff (ByVal r1 As Double, ByVal r2 As
Double) As Double
  Dim tempDiff As Double
  If (r1 < 0) Then r1 = r1 + 4 * PIo2
  If (r2 < 0) Then r2 = r2 + 4 * PIo2
  tempDiff = r1 - r2
  If (Abs(tempDiff) > 180 * DtoR) Then
    tempDiff = tempDiff - Sgn(tempDiff) * 4 * PIo2
  End If
  sgnDiff = tempDiff
End Function

Function smallBisect (ByVal r1 As Double, ByVal r2
As Double) As Double
  Dim tempBisect As Double

  tempBisect = (r1 + r2) / 2
  If (Abs(r1 - r2) > 2 * PIo2) Then tempBisect =
recipAng(tempBisect)
  If (sgnDiff(r1, r2) > 0) Then tempBisect =
tempBisect - 4 * PIo2
  smallBisect = tempBisect
End Function
```

APPENDIX G: DATA ANALYSIS PROGRAMS

allrot.m
autocorr.m
clocksta.m
complim.m
desired.m
dispplot.m
dlmread.m
dowinds.m
extract.m
fpdata.m
fselect.m
import.m
limplots.m
lin2mat.m
lin2resp.m
linfit.m
loaddata.m
longstat.m
lookup.m
plotproc.m
process.m
qdata.m
quickt2.m
readcat.m
repstr.m
rotate.m
rotproc.m
saverot.m
savesumm.m
showcat.m
stderr.m
stepfit.m
stepplot.m
strcode.m
taeplot.m
wind.m
windproc.m
xteboxes.m
xtelimit.m
xteplot.m
xteslice.m
xtestats.m

allrot.m

```
% allrot
%  - make the rotate code redo its calculation

disp(['   Running allrot using file: ', filename])

rotflag = 1;
```

autocorr.m

```
function [R, avg] = autocorr(summary)
% AUTOCORR - Estimate of the Auto-Correlation

[m,n] = size(summary);

% Throw away all the flight plan data in the first
columns
data = summary(:,12:n);
avg  = sum(data)./m;

n = n - 11;
Ra = zeros(m,n*n);

for i = 1:m
   x = data(i,:);
   Rxx = x'*x;
   Ra(i,:) = Rxx(:)';
end
R = zeros(n,n);
if (m == 1)
   R(:) = Ra;
else
   R(:) = sum(Ra)./m;
end
```

clocksta.m

```
% clocksta
%  - processing code to collect Clock statistics

disp(['   Running clocksta using file: ',
filename])

intrv = diff(FlightData(:,DataSrvrTime));

meanI = mean(intrv);
varI = var(intrv);
ticks = round(intrv/55);
medianT = median(ticks);
minT = min(ticks);
maxT = max(ticks);
procsum = [procsum, medianT, meanI, varI, maxT,
minT];
```

complim.m

```
function complim(dist, xnum, xavg, xvar, xerr,
ynum, yavg, yvar, yerr)

xll = xavg - xerr;
xul = xavg + xerr;
yll = yavg - yerr;
yul = yavg + yerr;
polydist = [dist,fliplr(dist)]';
xpoly = [xll,fliplr(xul)]';
xind = find(finite(xpoly));

% First calculate a T test between the means
P = quickt2(xnum, xavg, xvar, ynum, yavg, yvar);
avgind = find(P < .05);

% Now compare the variances
P1 = fcdf(xvar./yvar,xnum-1,ynum-1);
P2 = fcdf(yvar./xvar,ynum-1,xnum-1);
% But only where the T-test fails.
varind = find((max(P1,P2) > 0.975) & ~(P < .05));

% and plot the result
hold on
set(gca,'Box','on');
patch(xpoly(xind),polydist(xind),'m');
plot(xll,dist,'m-',xul,dist,'m-',yll,dist,'y-
',yul,dist,'y-')
plot([yll;yul],[dist;dist],'y-')
plot(xll(avgind),dist(avgind),'cx',xul(avgind),dist
(avgind),'cx')
plot(yll(avgind),dist(avgind),'cx',yul(avgind),dist
(avgind),'cx')
plot(xll(varind),dist(varind),'go',xul(varind),dist
(varind),'go')
plot(yll(varind),dist(varind),'go',yul(varind),dist
(varind),'go')
grid
xlabel('XTE Limits [nm]')
ylabel('Distance to Go [nm]')
hold off
```

desired.m

```
function [dtX, dtY] = desired
% DESIRED
%  - returns x and y coordinates for the desired
track of experiment three

% Angle of the arc
ang = 41*pi/180;
alp = 7*(1/tan((pi - ang)/2) - 1/tan((pi-5/7)/2));
i = sqrt(-1);

% Initial segment - straight to begining of dme arc
tempP = exp(-i*ang)*([-1;0] - i*7) + (alp + i*18);
dtX = real(tempP);
dtY = imag(tempP);

% DME arc - circle segment of arcang radians
arcang = ang - asin((alp+2/pi)/7);
rads = [arcang/25:arcang/25:arcang]';
tempP = 7*exp(i*(-pi/2-ang+rads)) + (alp + i*18);
dtX = [dtX; real(tempP)];
dtY = [dtY; imag(tempP)];
```

desired.m

```
% Insert a NaN to separate the pieces
dtX = [dtX; NaN];
dtY = [dtY; NaN];

% Turn arc - circle segment of pi/2 radians
rads = [0:pi/40:pi/2]';
tempP = 2/pi*exp(i*(pi/2-rads)) + (-2/pi + i*(11-
2/pi));
dtX = [dtX; real(tempP)];
dtY = [dtY; imag(tempP)];

% Final segment - straight line to the origin
dtX = [dtX; 0];
dtY = [dtY; 0];
```

dispplot.m

```
% dispplot
%  - plots the tracks split into different figures
for each display
disp(['   Running dispplot using file: ',
filename])

if (PlanData(7) == 'T')
   subplot(2,2,1)
elseif (PlanData(7) == 'P')
   subplot(2,2,2)
elseif (PlanData(7) == 'V')
   subplot(2,2,3)
elseif (PlanData(7) == 'X')
   subplot(2,2,4)
end
% flip for crosswinds
if ((PlanData(1) == 1) | (PlanData(1) == 3))
   flip = -1;
elseif ((PlanData(1) == 2) | (PlanData(1) == 4))
   flip = 1;
end
%plot(TrackX*flip, TrackY)
plot(TrackX, TrackY)
if (~ishold)
%  axis('equal')
%  axis([-.5 .5 2 4])
   grid
   hold on
   if (PlanData(7) == 'T')
      title('T Display')
   elseif (PlanData(7) == 'P')
      title('P Display')
   elseif (PlanData(7) == 'V')
      title('V Display')
   elseif (PlanData(7) == 'X')
      title('X Display')
   end
end
```

dlmread.m

```
function m=dlmread(filename, dlm, r, c, rng)
%DLMREAD Read a ASCII delimited file into a matrix.
%  M = DLMREAD(FILENAME, DLM, R, C) reads data
from the ASCII delimited
%  format FILENAME, using the delimiter DLM.  The
data is read starting
%  at file offset row R and column C.
%  M = DLMREAD(FILENAME, DLM) reads data from the
ASCII delimited
%  format FILENAME, using the delimiter DLM.
Reading the entire matrix
%  is equivalent to R=C=0 since database indexing
begins at (0,0) in the
%  upper left corner.
%  A final optional argument, RNG can be used to
only import a range,
%  either indexed or named.
%
%  See also CSVREAD, CSVWRITE, WK1READ, WK1WRITE.

%     Brian M. Bourgault 10/22/93
%  Copyright (c) 1984-94 by The MathWorks, Inc.
%

   %
   % test for proper filename
   %
   if ~isstr(filename)
      error('delread: Filename must be a string
argument!');
   end
   all=0;
   %
   % check/set row,col offsets
   %
   if ~exist('r')
      r = 0;
   end
   if ~exist('c')
      c = 0;
   end

   %
   % delimiter defaults to Comma for CSV
   %
   if ~exist('dlm')
      dlm = ',';
   end

   %
   % get the upper-left and bottom-rignt cells
   % of the range to read into MATLAB
   %
   if exist('rng')
      if ~isstr(rng)
         ulc = rng(1:2);
         brc = rng(3:4);
      else
         x = str2rng(rng)
         ulc = x(1:2);
         brc = x(3:4);
      end
   else
      all = 1;
      rng = [ 0 0 ];
      ulc = [0 0];
      brc = [0 0];
   end

   %
   % open the file
   %
   fid = fopen(filename,'r');
   if fid == (-1)
      error(['dlmread: Could not open file
filename ']);
```

dlmread.m

```
   end

   %
   % Read delimited format
   %
   eol = 10;          % End Of Line char
   loc = [1 1];       % starting location of return
matrix
   line = fgets(fid); % get the 1st line, if
any...
   %
   % read till eof
   %
   while(line ~= [ -1 ])
      i = 1;
      j = 1;
      while(i <= length(line))
         %
         % read chars from line, parsing
delimiters & numbers
         %
         num = [];
         j = 1;
         while(line(i) ~= dlm & line(i) ~= eol)
            %
            % build number string from characters
on the line
            %
            num(j) = line(i);
            i = i + 1;    % overall line index
            j = j + 1;    % number string index
         end

         %
         % found a delimiter or <eol>
         %
            if(all | ((loc >= ulc) & (loc <= brc)))
               if( num ~= [])
                  num(j) = 0; % null terminate
string
                  temp = str2num(setstr(num));
                  if (temp == [])
                     temp = 0;
                  end
                  m(loc(2)+r, loc(1)+c) = temp;
               else
                  % no number found between
delimiters
                  m(loc(2)+r, loc(1)+c) = 0;
               end
            end

            if(line(i)   == dlm)
               % delimiter, set location to next row
and get next line
               loc(1) = loc(1) + 1;
               i = i + 1;
            else
               if(line(i) == eol)
                  % eol, set location to next row
and get next line
                  loc(2) = loc(2) + 1;
                  loc(1) = 1;
                  i = i + 1;
               end
            end
         end
         % get next line of file
         line = fgets(fid);
      end
      % close file
      fclose(fid);
```

dowinds.m

```
clear all
all = zeros(21,1);
for i = 3:21
   eval(['load phb',sprintf('%03d', i)])
   [hdg,spd,w] = wind(FlightData, TrueHeading,
GroundTrack, Airspeed, GroundSpeed);
   all(i) = hdg;
end
```

extract.m

```
function [yd] = extract(xd, tab, which)
% EXTRACT - pull out table slices keeping exactly
one per x point
%   which = 1 will grab the last of any repeats
%   which = -1 will grab the first of any repeats

% Lookup XTE values along the track
[y,x] = lookup(tab,xd(:));

% Add all the numbers we want to make sure are part
of the sequence
% also find the unique numbers in the sequence and
% the last (or first) one of each of the repeats.
% and the order that xd has been sorted into
if which > 0
   [xs,I] = sort([xd(:);x]);
   ys = [ones(size(xd(:),1),size(tab,2)-1)*NaN;y];
   fx = find(diff([xs;Inf]));
   xdi = find(I <= length(xd(:)));
   xdi = I(xdi);
elseif which < 0
   [xs,I] = sort([x;xd(:)]);
   ys = [y;ones(size(xd(:),1),size(tab,2)-1)*NaN];
   fx = find(diff([-Inf;xs]));
   xdi = find(I > length(x));
   xdi = I(xdi)-length(x);
end
ys = ys(I,:);

yd = zeros(size(xd(:),1),size(tab,2)-1);

% Finally extract the ones we want and put them
back in the
% original order.
yd(xdi,:) = ys(fx,:);

if (size(xd,1) < size(xd,2))
   yd = yd';
end
```

fpdata.m

```
% Flight Plan Information
%  Types: 1 = Left Step  ( 1), Right Wind, Left
Turning DME
%       2 = Left Step  ( 1), Left Wind,  Straight
%       3 = Right Step (-1), Left Wind,  Right
Turning DME
%       4 = Right Step (-1), Right Wind, Straight
%     Type Plate Wind  RWY  Step  MAPAlt    Disp
FPInfo = real([
        1  'a'    81   36    1  500     'X';
        2  'a'   351   36    1  500     'X';
        3  'a'   351   36   -1  500     'T';
        4  'a'    81   36   -1  500     'V';
        1  'b'   167  122    1  900     'V';
        2  'b'    77  122    1  900     'V';
        3  'b'    77  122   -1  900     'X';
        4  'b'   167  122   -1  900     'P';
        1  'c'   242  197    1  700     'T';
        2  'c'   152  197    1  700     'T';
        3  'c'   152  197   -1  700     'P';
        4  'c'   242  197   -1  700     'X';
        1  'd'   325  280    1  300     'P';
        2  'd'   235  280    1  300     'P';
        3  'd'   235  280   -1  300     'V';
        4  'd'   325  280   -1  300     'T';
        1  'p'   209  164    1  600     'X';
        2  'p'   119  164    1  600     'T';
        3  'p'   119  164   -1  600     'P';
        4  'p'   209  164   -1  600     'V']);

% Experiment Matrix
%  Indices into above matrix for each subject type
%  Practise runs are 16 + abs(RunNum)
ExpMatrix = [     1  14   3   5  ;
             15   7  13  16  ;
              8   9   6   2  ;
             10   4  12  11  ;
             11  10   4  12  ;
              2   8   9   6  ;
             16  15   7  13  ;
              5   1  14   3  ;
              4  12  11  10  ;
              9   6   2   8  ;
              7  13  16  15  ;
             14   3   5   1  ;
              3   5   1  14  ;
             13  16  15   7  ;
              6   2   8   9  ;
             12  11  10   4  ;
             18  18  18  18  ;
             20  20  20  20  ;
             19  19  19  19  ;
             17  17  17  17  ];
```

fselect.m

```
function flist = fselect(pilot, selection)
% FSELECT
%  flist = fselect(pilot, selection)
%     pilot - initials of pilot, eg. 'sar'
%           - only works for a single pilot at a
time
%     selection - boolean expression
%
%  returns the list of file numbers for which the
boolean expression returns true
%  useful expression variables are
%     FPType = 1, 2, 3, 4;
%     Plate = 'a', 'b', 'c', 'd', 'p';
%     Disp = 'X', 'V', 'T', 'P';
%     LStep = True for Left Step, False for Right
Step;
%     SWind = True for Same as Step Wind, False
for Opposite to Step Wind
%     Arc = True for Arc approaches, False for
Straight approaches
%     Data = True for Data runs, False for
Practise runs
%  eg. '~LStep & (Disp == ''X'')'

% Load the general flight plan information
fpdata

% Save the current directory to come back to after
processing each subject
OrigDir = pwd;

fail = 0;
flist = [];
eval(['cd ',pilot], 'fail = 1;')
if (fail)
   disp(['Unable to change to subject directory:
',pilot])
   return
end

eval('catalog', 'fail = 1;')
% Return to the original directory
eval(['cd(''',OrigDir,''')'])
if (fail)
   disp(['Unable to load catalog from directory:
',pwd])
   return
end

fpindx = Files(:,2);                  % Get the run
number for each file
pracs = find(fpindx < 0);             % Find the
practise runs
fpindx(pracs) = -fpindx(pracs) + 16;     % Convert
the practise run numbers to indices
fpindx = ExpMatrix(fpindx,SubjectType); % Look the
runs up in the experiment matrix

% Finally, grab the data from the Flight Plan
information
FPType = FPInfo(fpindx, 1);
Plate = FPInfo(fpindx, 2);
SWind = (FPType == 2) | (FPType == 4);
LStep = (FPType == 1) | (FPType == 2);
Arc = ~SWind;
Disp = FPInfo(fpindx, 7);
dindx = find(Files(:,3) > 0);
Disp(dindx) = Files(dindx,3);
Data = (Plate ~= 'p');

eval(['flist = ', selection, ';'], 'fail = 1;');
if (fail)
   disp(['Unable to evaluate selection criterion:
', selection])
   flist = [];
   return
```

```
end

flist = mat2str(Files(find(flist),1)');      %
Return the file numbers corresponding to the
selection
flist = flist(2:(length(flist)-1));
flindx = find(flist == ' ');
flist(flindx) = ones(size(flindx)).*real(',');
flist = setstr(flist);
```

import.m

```
function [] = import(SubjID, RunNo)
%IMPORT Load a Frasca datafile.
%  IMPORT(SUBJID, RUNNO) reads data from the
FRASCA data file
%  specified by the Subject ID and Run Number
given as arguments.
%  (The filename is created by concatenating the
Subject ID and Run Number
%  thus import('SAR', 1) loads the files
SAR001.CFG and SAR001.DAT)

%
% test for proper subject ID
%
if ~isstr(SubjID)
   error('loaddata: SUBJID must be a string
argument!');
end
sp = abs(' ');
dq = abs('"');

%
% Load the configuration file
%
filename = sprintf('%s%03d.cfg', SubjID, RunNo);
ind = find(filename == ' ');  % Remove spaces from
the filename
filename(ind) = [];
fid = fopen(filename,'r');
if (fid == -1)
   error(['loaddata: Could not open the
configuration file ',filename]);
end

% Discard the first line
tline = fgetl(fid);

% Read the Experiment, Profile and Flight Plan
names, discard Subject ID and Run Number
tline = fgetl(fid);
ind = find(tline == '"');
tline(ind) = abs('''')*ones(size(ind));
ind = find(tline == ',');
tline = [tline(1:(ind(2)-1)),'; FlightPlan =
',tline((ind(2)+1):(ind(3)-1)),';'];
tline = [tline(1:(ind(1)-1)),'; Profile =
',tline((ind(1)+1):length(tline))];
tline = ['Experiment = ',tline];
eval(tline);

% Read in the next set of quoted text and save it
as the comment
tline = fread(fid, 1);
nextch = '';
Comment = [];
while (nextch ~= dq)
   Comment = [Comment,nextch];
   nextch = fread(fid,1);
end
tline = fgetl(fid);

% Discard the number of labels as recorded in the
config file
tline = fgetl(fid);
%nlabels = fscanf(fid, '%d');

% Read all the labels and process them
tline = fgetl(fid);
ind = find((tline == ' ')|(tline == '"'));  %
Remove spaces and double quotes
% ind =
[ind,find((tline=='a')|(tline=='e')|(tline=='i')|(t
line=='o')|(tline=='u'))];    % Remove vowels
tline(ind) = [];
ind = find(tline == ',');
varlens = [ind,(length(tline)+1)]-[0,ind]-1;
maxlabellen = max(varlens);
sreqd = maxlabellen - varlens;
nlabels = length(sreqd);
tline =
[tline,sp*ones(1,sreqd(nlabels)),sprintf('=%4d;',nl
abels)];
for k = (nlabels-1):-1:1
   tline = [tline(1:(ind(k)-
1)),sp*ones(1,sreqd(k)),sprintf('=%4d;',k),tline((i
nd(k)+1):length(tline))];
end
eval(tline);
labels = zeros(maxlabellen+6,nlabels);
labels(:)=tline;
labels = setstr(labels');

% Start reading the waypoint information
tline = fgetl(fid);
k = 0;
WPTable = [];
while (tline ~= -1)
   % Count the number of waypoints
   k = k + 1;

   % Replace any null strings with ""
   tline = strrep(tline, '#NULL#','""');

   % Convert " to '
   ind = find(tline == '"');
   tline(ind) = abs('''')*ones(size(ind));

   % Make sure there are 6 characters in each of
the first two fields
   sreqd = 6-ind(2)+ind(1)+1;
   if (sreqd > 0)
      tline = [tline(1:(ind(2)-
1)),sp*ones(1,sreqd),tline(ind(2):length(tline))];
      ind(3:4) = ind(3:4)+sreqd;
   end
   sreqd = 6-ind(4)+ind(3)+1;
   if (sreqd > 0)
      tline = [tline(1:(ind(4)-
1)),sp*ones(1,sreqd),tline(ind(4):length(tline))];
   end

   % Parse the matrix in text form
   eval(['WPTable=[WPTable; ',tline,'];']);

   % define temporary identifiers for the waypoint
names and types
```

```
    eval(['WPTable(k,1:6),'= k;']);
    if (find(WPTable(k,7:12) ~= sp))
       eval([WPTable(k,7:12),'= k;']);
    end
    tline = fgetl(fid);
end
fclose(fid);

%
% Open the data file
%
filename = strrep(filename, '.cfg', '.dat');
fid = fopen(filename,'r');
if (fid == -1)
    error(['loaddata: Could not open the data file
',filename]);
end

%
% Read the file until eof
%
tline = fgetl(fid); % get the 1st text line, if
any...
FlightData = [];
while (tline(1) ~= -1)
    % remove unwanted characters
    ind = find((tline<=sp)|(tline==dq));
    tline(ind) = [];

    % replace commas with spaces
    ind = find(tline == ',');
    tline(ind) = sp*ones(size(ind));

    % put in zeroes for the blank columns
    tline = repstr(tline, '  ', ' 0 ');

    % Parse the matrix in text form
    eval(['FlightData=[FlightData; ',tline,'];']);

    % get next text line of file
    tline = fgetl(fid);
end

% close file
fclose(fid);

% Clear out the waypoint and waypoint type
variables
%for k = 1:size(WPTable,1)
%   eval(['clear ',WPTable(k,1:12)]);
%end

% Clear out the temporary variables
clear sreqd varlens k tline fid ind sp dq nextch
ans

% Save the data into a matlab file
filename = strrep(filename, '.dat', '.mat');
eval(['save ',filename]);
```

limplots.m

```
function limplots(summ)
% LIMPLOTS - create figures showing 95% XTE limits
xdispind = find(summ(:,11) == 'X');
tdispind = find(summ(:,11) == 'T');
vdispind = find(summ(:,11) == 'V');
pdispind = find(summ(:,11) == 'P');

figure(1)
clf

dist = 18:-.2:0;
[m,n] = size(summ);

subplot(1,4,1)
[xnum,xavg,xvar,xerr,xtth,xout] =
xtelimit(summ(xdispind,12:n),dist);
xll = xavg - xerr;
xul = xavg + xerr;
sigindx = find(xtth);
plot(xll,dist,'y-',xavg,dist,'g-',xul,dist,'y-
',xavg(sigindx),dist(sigindx),'rx',...
    xout',dist'*ones(1,size(xout,1)),'r+')
grid
xlabel('XTE Limits [nm]')
ylabel('Distance to Go [nm]')
title('XTE Only')

subplot(1,4,2)
[tnum,tavg,tvar,terr,ttth,tout] =
xtelimit(summ(tdispind,12:n),dist);
tll = tavg - terr;
tul = tavg + terr;
sigindx = find(ttth);
plot(tll,dist,'y-',tavg,dist,'g-',tul,dist,'y-
',tavg(sigindx),dist(sigindx),'rx',...
    tout',dist'*ones(1,size(tout,1)),'r+')
grid
xlabel('XTE Limits [nm]')
ylabel('Distance to Go [nm]')
title('Triangle')

subplot(1,4,3)
[vnum,vavg,vvar,verr,vtth,vout] =
xtelimit(summ(vdispind,12:n),dist);
vll = vavg - verr;
vul = vavg + verr;
sigindx = find(vtth);
plot(vll,dist,'y-',vavg,dist,'g-',vul,dist,'y-
',vavg(sigindx),dist(sigindx),'rx',...
    vout',dist'*ones(1,size(vout,1)),'r+')
grid
xlabel('XTE Limits [nm]')
ylabel('Distance to Go [nm]')
title('Vector')

subplot(1,4,4)
[pnum,pavg,pvar,perr,ptth,pout] =
xtelimit(summ(pdispind,12:n),dist);
pll = pavg - perr;
pul = pavg + perr;
sigindx = find(ptth);
plot(pll,dist,'y-',pavg,dist,'g-',pul,dist,'y-
',pavg(sigindx),dist(sigindx),'rx',...
    pout',dist'*ones(1,size(pout,1)),'r+')
grid
xlabel('XTE Limits [nm]')
ylabel('Distance to Go [nm]')
title('Predictor')

% Comparison plots
figure(2)
clf

subplot(1,3,1)
complim(dist, xnum, xavg, xvar, xerr, tnum, tavg,
tvar, terr);
```

```
title('X vs T')

subplot(1,3,2)
complim(dist, xnum, xavg, xvar, xerr, vnum, vavg,
vvar, verr);
title('X vs V')

subplot(1,3,3)
complim(dist, xnum, xavg, xvar, xerr, pnum, pavg,
pvar, perr);
title('X vs P')

figure(3)
clf
subplot(1,3,1)
complim(dist, tnum, tavg, tvar, terr, vnum, vavg,
vvar, verr);
title('T vs V')

subplot(1,3,2)
complim(dist, tnum, tavg, tvar, terr, pnum, pavg,
pvar, perr);
title('T vs P')

subplot(1,3,3)
complim(dist, vnum, vavg, vvar, verr, pnum, pavg,
pvar, perr);
title('V vs P')
```

lin2mat.m

```
function [A,B,C,D] = lin2mat(lam)
% lin2mat
%  - function to return the matrices of a 2 state system
% lam should be a two vector [Real Part, Imag Part, dx0]
% - it defines the eigenvalues for a second order system
% - if Imag Part < 0 then the eigenvalues will be complex: lam(1) +/- sqrt(-1)*lam(2)
% - if Imag Part > 0 then the eigenvalues will be real: lam(1) +/- lam(2)

A = [0, 1; abs(lam(2))*lam(2) - lam(1)*lam(1), 2*lam(1)];
B = [0;0];        % No inputs
C = [1,0];    % Output only the first state
D = 0;
```

lin2resp.m

```
function ysim = lin2resp(lam, t)
% lin2resp
%  - function to minimise 2 state system
% lam(3) gives the initial value for the velocity
% t is a vector of points at which the outputs
should be compared
[A,B,C,D] = lin2mat(lam);
x0 = [.25,lam(3)];

ysim = initial(A,B,C,D,x0,t);
```

linfit.m

```
function f = linfit(lam, t, y, sysresp)
% linfit
%  - function to minimise linear system
eval(['ysim = ',sysresp, '(lam, t);']);

f = (y - ysim).*exp(-t);
```

loaddata.m

```
function [wp,m] = loaddata(subjid, runno)
%LOADDATA Load a Frasca datafile.
%  LOADDATA(SUBJID, RUNNO) reads data from the
FRASCA data file
%  specified by the subject ID and Run Number
given as arguments.
%  (The filename is created by concatenating the
Subject ID and Run Number
%  thus loaddata('SAR', 1) loads the files
SAR001.CFG and SAR001.DAT.)

%
% test for proper subject ID
%
if ~isstr(subjid)
   error('loaddata: SUBJID must be a string
argument!');
end
sp = abs(' ');
dq = abs('"');
%
% Load the configuration file
%
filename = sprintf('%s%03d.cfg', subjid, runno);
ind = find(filename == ' ');  % Remove spaces from
the filename
filename(ind) = [];
fid = fopen(filename,'r');
if (fid == -1)
   error(['loaddata: Could not open the
configuration file ',filename]);
end

% Discard the first line
tline = fgetl(fid);

% Read the Experiment, Profile and Flight Plan
names
tline = fgetl(fid);
```

```
% Read in the next set of quoted text and save it
as the comment
tline = fread(fid, 1);
nextch = '';
Comment = [];
while (nextch ~= dq)
   Comment = [Comment,nextch];
   nextch = fread(fid,1);
end
tline = fgetl(fid);

% Discard the number of labels as recorded in the
config file
tline = fgetl(fid);
%nlabels = fscanf(fid, '%d');

% Read all the labels and process them
tline = fgetl(fid);
ind = find((tline == ' ')|(tline == '"')); %
Remove spaces and double quotes
% ind =
[ind,find((tline=='a')|(tline=='e')|(tline=='i')|(t
line=='o')|(tline=='u'))];    % Remove vowels
tline(ind) = [];
ind = find(tline == ',');
varlens = [ind,(length(tline)+1)]-[0,ind]-1;
maxlabellen = max(varlens);
sreqd = maxlabellen - varlens;
nlabels = length(sreqd);
tline =
[tline,sp*ones(1,sreqd(nlabels)),sprintf('=%4d;',nl
abels)];
for i = (nlabels-1):-1:1
   tline = [tline(1:(ind(i)-
1)),sp*ones(1,sreqd(i)),sprintf('=%4d;',i),tline((i
nd(i)+1):length(tline))];
end
eval(tline);
labels = zeros(maxlabellen+6,nlabels);
labels(:)=tline;
labels = setstr(labels');

% Start reading the waypoint information
tline = fgetl(fid);
k = 0;
WPTable = [];
while (tline ~= -1)
   % Count the number of waypoints
   k = k + 1;

   % Replace any null strings with ""
   tline = strrep(tline, '#NULL#','""');

   % Convert " to '
   ind = find(tline == '"');
   tline(ind) = abs('''')*ones(size(ind));

   % Make sure there are 6 characters in each of
the first two fields
   sreqd = 6-ind(2)+ind(1)+1;
   if (sreqd > 0)
      tline = [tline(1:(ind(2)-
1)),sp*ones(1,sreqd),tline(ind(2):length(tline))];
      ind(3:4) = ind(3:4)+sreqd;
   end
   sreqd = 6-ind(4)+ind(3)+1;
   if (sreqd > 0)
      tline = [tline(1:(ind(4)-
1)),sp*ones(1,sreqd),tline(ind(4):length(tline))];
   end

   % Parse the matrix in text form
   eval(['WPTable=[WPTable; ',tline,'];']);

   % define temporary identifiers for the waypoint
names and types
   eval([WPTable(k,1:6),'= k;']);
   if (find(WPTable(k,7:12) ~= sp))
      eval([WPTable(k,7:12),'= k;']);
   end
   tline = fgetl(fid);
end
fclose(fid);

%
% Open the data file
%
filename = strrep(filename, '.cfg', '.dat');
fid = fopen(filename,'r');
if (fid == -1)
   error(['loaddata: Could not open the data file
',filename]);
end

%
% Read the file until eof
%
tline = fgetl(fid); % get the 1st text line, if
any...
FlightData = [];
while (tline(1) ~= -1)
   % remove unwanted characters
   ind = find((tline<=sp)|(tline==dq));
   tline(ind) = [];

   % replace commas with spaces
   ind = find(tline == ',');
   tline(ind) = sp*ones(size(ind));

   % put in zeroes for the blank columns
   tline = repstr(tline, '  ', ' 0 ');

   % Parse the matrix in text form
   eval(['FlightData=[FlightData; ',tline,'];']);

   % get next text line of file
   tline = fgetl(fid);
end

% close file
fclose(fid);

% Clear out the temporary variables
clear sreqd varlens k tline fid ind sp dq nextch

% Save the data into a matlab file
filename = strrep(filename, '.dat', '.mat');
eval(['save ',filename]);
```

longstat.m

```
function [outstats] = longstat(yd)
% LONGSTAT - calculates along track statistics

finind = find(finite(yd));
yd = yd(finind);

n = length(yd);

outavg = sum(yd)/n;
outvar = sum((yd-outavg).^2)/(n-1);
outrms = sqrt(sum(yd.^2)/n);
outmad = sum(abs(yd-outavg))/n;

outstats = [outavg, outvar, outrms, outmad];
```

lookup.m

```
function [y,x] = lookup(tab,x0)
%LOOKUP   Table look-up.
%  [Y,X] = lookup(TAB,X0) returns a table of linearly interpolated rows from
%  table TAB, looking up X0 in the first column of TAB.
%     Y -- returns the interpolated table
%     X - returns the x values associated with each line of the table.
%       - if the first column of table is not monotonic then this will
%       - contain repeated values from X0.
%       - if any values in X0 are outside of the range of the first column of
%       - the table then they will be excluded.
%
%  See also INTERP1, INTERP2, TABLE1, TABLE2.


if (nargin ~= 2), error('Wrong number of input arguments.'), end

[m,n] = size(tab);
x0 = x0(:);       % Make sure x0 is a column
k0 = max(size(x0));

ncmp = sparse(0,0);
for k = 1:k0
%  ncmp = [ncmp, diff(x0(k) > tab(:,1))];
   ncmp = [ncmp, diff([x0(k) >= tab(1,1);x0(k) > tab(:,1)])];
end
[ki, kj] = find(ncmp);
x = x0(kj);
tab = [tab(1,1) - 1, tab(1,2:n); tab];
xp = (x - tab(ki,1))./(tab(ki+1,1) - tab(ki,1));

if (n*length(ki) < 10000)
   y = tab(ki,2:n) + (tab(ki+1,2:n) - tab(ki,2:n)).*(xp*ones(1,n-1));
else
   y = zeros(length(ki),n-1);
   for k = 1:length(ki)
      y(k,:) = tab(ki(k),2:n) + (tab(ki(k)+1,2:n) - tab(ki(k),2:n))*xp(k);
   end
end
```

plotproc.m

```
% plotproc
%  - plots all the tracks in a single figure

disp(['   Running plotproc using file: ', filename])

if (~ishold)
   figure
end
plot(TrackX, TrackY)
if (~ishold)
   axis('equal')
   hold on
end
```

```
function [summary] = process(SubjDirStr, FileListStr, FileProc)
% PROCESS - [summary] = process(SubjDirStr, FileListStr, FileProc)
%  SubjDirStr - directory to change to which contains the data to be processed
%        - this directory needs a file 'catalog.m' which lists the files
%        - separate multiple subjects with semi-colons
%        - eg. 'phb;dkj;cjf'
% FileListStr - list of files to process
%        - numbers refer to the file identifiers
%        - use a null vector to indicate all files
%        - use Inf to indicate all non-practise runs
%        - use -Inf to indicate all practise runs
%        - need one list for each subject, separate by semi-colons
%        - eg. 'Inf;;2,4:8,3'
% FileProcStr - processing procedure(s) to execute
%        - eg. 'windproc' will execute the wind processing code
%-            'windproc; plotproc' will execute both the wind and plotting codes
%   summary(:,1) = Subject Number
%   summary(:,2) = File Index
%   summary(:,3) = File Identifier
%   summary(:,4) = Run Number
%   summary(:,5) = Flight Plan type
%   summary(:,6) = Approach Plate
%   summary(:,7) = Wind Direction in Compass degrees
%   summary(:,8) = Runway Heading
%   summary(:,9) = Step Direction (+1 = Left Step; -1 = Right Step)
%   summary(:,10) = MAP Altitude
%   summary(:,11) = Display Used
%   summary(:,12:->) = data from the processing routines that were run

% Quick check on the parameters
goodparam = 1;
if (nargin ~= 3)
   disp('Process requires three parameters')
   goodparam = 0;
end
if (~isstr(SubjDirStr))
   disp('The first argument should be a string')
   goodparam = 0;
end
if (~isstr(FileListStr))
   disp('The second argument should be a string')
   goodparam = 0;
end
if (~isstr(FileProc))
   disp('The third argument should be a string')
   goodparam = 0;
end
% Separate the Subject Directory List and File Lists for each subject to be processed
sdi = [0,find(SubjDirStr == ';'),length(SubjDirStr)+1];
fli = [0,find(FileListStr == ';'),length(FileListStr)+1];

if (length(fli) ~= length(sdi))
   disp('Number of subjects and number of file lists don''t match')
   goodparam = 0;
end
if ~goodparam
   help process
   error(' ')
end

% Load the general flight plan information
fpdata

% Save the current directory to come back to after processing each subject
OrigDir = pwd;

summary = [];
for ksubj = 1:(length(sdi)-1)
```

```
% Grab the subject directory from the full list
SubjDir = SubjDirStr((sdi(ksubj)+1):(sdi(ksubj+1)-1));

% Grab the file list from the full list
fail = 0;
eval(['FileList = [',FileListStr((fli(ksubj)+1):(fli(ksubj+1)-1)),'];'],'fail = 1;');
if (fail)    % Default to process all of the files
   FileList = [];
end

% Go to the subject directory and
% Load the catalog of files in this directory
fail = 0;
eval(['cd(''',SubjDir,''')'], 'fail = 1;')
if (fail)
   disp(['Unable to change to subject directory: ',SubjDir])
else
   fail = 0;
   eval('catalog', 'fail = 1;')
   if (fail)
      disp(['Unable to load catalog from directory: ',pwd])
   else
      FileList = FileList(:);
      if (isempty(FileList))
         % Set default FileList to be all the files
         FileList = Files(:,1);
      elseif (any(isinf(FileList)))
         fs = [0,find(isinf(FileList))];
         pind = find(Files(:,2) < 0);
         npind = find(Files(:,2) > 0);
         % Replace each Inf with all but the practise runs
         % and each -Inf with only practise runs
         NewList = [];
         for f = 2:length(fs)
            NewList = [NewList;FileList((fs(f-1)+1):(fs(f)-1))];
            if (FileList(fs(f)) < 0)
               NewList = [NewList; Files(pind,1)];
            else
               NewList = [NewList; Files(npind,1)];
            end
         end
         FileList = [NewList; FileList((fs(f)+1):length(FileList))];
         clear NewList npind pind fs f
      end

      nfiles = length(FileList);
      for ind = 1:nfiles
         fileIND = find(Files(:,1) == FileList(ind));
         fname = [SubjIDENT, sprintf('%03d', FileList(ind))];
         if (isempty(fileIND))
            disp(['File not catalogued: ',fname])
         else
            disp(['Processing: ',fname])
            fileIND = fileIND(length(fileIND));  % Use the last one in the list
            fail = 0;
            eval(['load ',fname],'fail = 1;')
            if (fail)
               disp(['Unable to load data, file skipped: ',fname])
            else
               procsum = [];
               if (~exist('PlanData'))
                  corfp = Files(fileIND, 2);                  % Grab the run number
                  if (corfp < 0), corfp = 16 + abs(corfp); end  % Convert practise runs
                  corfp = ExpMatrix(corfp, SubjectType);        % Grab the correct flight plan number
                  PlanData = FPInfo(corfp,:);                 % Pull the data for this flight plan
                  if (Files(fileIND, 3) ~= 0)
                     PlanData(7) = Files(fileIND, 3);         % Replace the display type
                  end
               end
               fail = 0;
               eval(FileProc, 'fail = 1;')
               if (fail)
```

```
                        disp(['Unable to evaluate processing code: ', FileProc])
                        eval(['cd(''',OrigDir,''')'])
                        error(' ')
                    end
                    % Clear away all of the variables from the current file
                    WPTable = [WPTable(:,1:12),32*ones(size(WPTable,1),1)]';
                    WPTable = WPTable(:)';
                    eval(['clear ',WPTable]);
                    labels = [labels(:,1:maxlabellen),32*ones(nlabels,1)]';
                    labels = labels(:)';
                    eval(['clear ',labels]);
                    clear Comment Experiment FlightData FlightPlan PlanX PlanY
                    clear Profile RunNo SubjID TrackX TrackY WPTable filename labels
                    clear maxlabellen nlabels
                    subjectNo = strcode(SubjIDENT);
                    summary = [summary; [subjectNo, fileIND, Files(fileIND,1:2), PlanData, procsum]];
                    clear PlanData
                end
            end
        end
    end
  end
  eval(['cd(''',OrigDir,''')'])
end
```

qdata.m

```
% qdata.m
%   processing code to collect question data from
the catalogs
disp(['   Running qdata using file: ', filename])

procsum = [procsum, QuestionData];
```

quickt2.m

```
function P = quickt2(xnum, xavg, xvar, ynum, yavg,
yvar)
dfx = xnum - 1;
dfy = ynum - 1;
dfe = dfx + dfy;
msx = dfx .* xvar;
msy = dfy .* yvar;
difference = xavg - yavg;
pooleds = sqrt((msx + msy) .* (1./xnum + 1./ynum)
./ dfe);
ratio = difference ./ pooleds;
significance  = tcdf(ratio, dfe);
P = 2 * min(significance,1 - significance);
```

readcat.m

```
% readcat
%  - collects an english version of the file
information
%  - for post process display by showcat

disp(['   Running readcat using file: ', filename])
dispchar = setstr(PlanData(7));
if (PlanData(7) == 'T')
   display = ['Triangle '];
elseif (PlanData(7) == 'P')
   display = ['Predictor'];
elseif (PlanData(7) == 'V')
   display = ['T Vector '];
elseif (PlanData(7) == 'X')
   display = ['XTE Only '];
elseif (PlanData(7) == 'E')
   display = ['Elec HSI '];
end

platechar = setstr(PlanData(2));
if (PlanData(2) == 'p')
   plate = ['Practice  '];
elseif (PlanData(2) == 'a')
   plate = ['Marathon  '];
elseif (PlanData(2) == 'b')
   plate = ['Tavernier '];
elseif (PlanData(2) == 'c')
   plate = ['Fedhaven  '];
elseif (PlanData(2) == 'd')
   plate = ['Ochopee   '];
end

typechar = int2str(PlanData(1));
if (PlanData(1) == 1)
   type = ['LO, Arc '];
elseif (PlanData(1) == 2)
   type = ['LS, Str '];
elseif (PlanData(1) == 3)
   type = ['RO, Arc '];
elseif (PlanData(1) == 4)
   type = ['RS, Str '];
end

readsum = [filename, ': ', typechar, platechar, '
', dispchar, ' |', type, plate, display];

% Add a header so that showcat can find this
information
procsum = [procsum, '@!#rc', length(readsum),
readsum];
```

repstr.m

```
function s = repstr(s1,s2,s3)
%STRREP String search and replace utility.
%  S = REPSTR(S1,S2,S3) replaces all occurrences
of S2 in S1 with S3
%  including occurrences created by including s3
in s1.
%  Example:
%       s1='This is a good example';
%       strrep(s1,'good','great') returns 'This is
a great example'
%       strrep(s1,'bad','great')  returns 'This is
a good example'
%       strrep(s1,'','great')     returns 'This is
a good example'
%  See also FINDSTR.

%  Rick Spada  11-23-92
%  Copyright (c) 1984-94 by The MathWorks, Inc.

% Make sure we don't get any recursion
if (findstr(s3,s2) ~= [])
   error('repstr: search string contained in
replace string')
end

% Take care of trivial cases:
%   s2 > s1
%   s1 is empty
%   s2 is empty
s = s1;
if ((length(s2)>length(s1)) | isempty(s2) |
isempty(s1))
  return;
end;

% Find all occurrences of s2 in s
s2len = length(s2);
s2pos=findstr(s1,s2);
while (s2pos ~= [])
   % Replace only the first occurrence
   s = [s(1:(s2pos(1)-
1)),s3,s((s2pos(1)+s2len):length(s))];

   % Find all occurrences of s2 in s
   s2pos = findstr(s,s2);
end
```

rotate.m

```
function [x, y] = rotate(oldLat, oldLong, oldHdg,
refLat, refLong, flip)
% ROTATE - [x, y] = rotate(oldLat, oldLong, oldHdg,
refLat, refLong, flip)

longcoef = cos(refLat*pi/180);
scaleM = [0 60;
   -60*longcoef 0];
old = [oldLat(:)-refLat, oldLong(:)-
refLong]*scaleM;

oldHdg = (oldHdg+180)*pi/180;
rotM = [cos(oldHdg), -sin(oldHdg);
   sin(oldHdg), cos(oldHdg)];
new = old*rotM';

x = flip*new(:,1);
y = new(:,2);
```

rotproc.m

```
% rotproc
%  - processing code to rotate the plots and save
the result

% if rotflag exists then do the rotation even if it
has been done before
if (~exist('rotflag'))
   if (exist('TrackX'))
      % Calculation has already been completed and
saved
      % No need to do it again
      return
   end
end

disp(['   Running rotproc using file: ', filename])
saverot(filename, PlanData)
```

saverot.m

```
function [] = saverot(filename, PlanData)
% SAVEROT
%  - reloads the data file into clean stack space,
%    calculates the rotated track data and saves
%    everything back to the file

eval(['load ',filename])

% Clear any old data that may be lying around
TrackX = [];
TrackY = [];
TrackXTE = [];
TrackDist = [];
PlanX = [];
PlanY = [];

% Rotate the flight plan co-ordinates
[PlanX, PlanY] =
rotate(WPTable(:,13),WPTable(:,14),PlanData(4),...

   WPTable(1,13),WPTable(1,14),PlanData(5));
% Rotate the ground track co-ordinates
[TrackX, TrackY] =
rotate(FlightData(:,Latitude),FlightData(:,Longitud
e),PlanData(4),...

   WPTable(1,13),WPTable(1,14),PlanData(5));

% Reset so that the MAP is the origin
TrackX = TrackX - PlanX(MAP);
TrackY = TrackY - PlanY(MAP);
PlanX = PlanX - PlanX(MAP);
```

saverot.m

```
PlanY = PlanY - PlanY(MAP);

% For the DME runs, calculate new XTE and Distance to go numbers
if ((PlanData(1) == 1) | (PlanData(1) == 3))
   i = sqrt(-1);
   ang = 41*pi/180;
   alp = 7*(1/tan((pi - ang)/2) - 1/tan((pi-5/7)/2));
   zone1 = TrackY > (tan(pi/2 - ang)*(TrackX+alp) + 18);
   zone2 = TrackY > (TrackX + 11);
   zone3 = TrackX < (-2/pi);
   zone4 = TrackY > (11 - 2/pi);
   init = find(zone1);
   dme = find(~zone1 & zone2 & zone3);
   turn = find(~zone3 & zone4);
   final = find(~zone2 & ~zone4);
   %lenDME = 7*ang - (alp + 2/pi);        % Length at infinity after transformation
   lenDME = 7*ang - 7*asin((alp + 2/pi)/7);    % Correct length

   % In the initial segment rotate about the start of the dme arc
   tempP = TrackX(init) + i*TrackY(init);
   tempP = tempP + i*7*exp(-i*ang) - (alp + i*18);
   tempP = tempP * exp(-i*(pi/2 - ang));
   TrackDist = imag(tempP) + 17;              % + lenDME + 1 + (11 - 2/pi);
   TrackXTE = real(tempP);

   % Insert a Nan to separate the pieces
   %TrackDist = [TrackDist; 17];
   %TrackXTE = [TrackXTE; NaN];

   % In the dme segment calculate polar coordinates about the center of the arc
   % and subtract a piece of arc at the end to account for the early change to the turn
   tempP = TrackX(dme) + i*TrackY(dme);
   tempP = tempP - (alp + i*18);
   %TrackDist(dme+1) = -(ang + pi/2 + angle(tempP)) * 7 + 17;    % + 1 + (11 - 2/pi);
   %TrackXTE(dme+1) = abs(tempP);
   %%TrackDist(dme+1) = TrackDist(dme+1) - TrackXTE(dme+1).*asin((alp+2/pi)./TrackXTE(dme+1));
   %TrackXTE(dme+1) = 7 - TrackXTE(dme+1);
   TrackDist(dme) = -(ang + pi/2 + angle(tempP)) * 7 + 17;    % + 1 + (11 - 2/pi);
   TrackXTE(dme) = abs(tempP);
   TrackXTE(dme) = 7 - TrackXTE(dme);

   % Insert a Nan to separate the pieces
   TrackDist = [TrackDist; 12];
   TrackXTE = [TrackXTE; NaN];

   % In the turn segment calculate polar coordinates about the center of the turn
   tempP = TrackX(turn) + i*TrackY(turn);
   tempP = tempP - ((-2/pi) + i*(11 - 2/pi));
   TrackDist = [TrackDist; angle(tempP) * 2/pi + 11];      % + (11 - 2/pi);
   TrackXTE = [TrackXTE; abs(tempP) - 2/pi];

   % Insert a Nan to separate the pieces
   TrackDist = [TrackDist; 11];
   TrackXTE =  [TrackXTE; NaN];

   % In the final zone no transformations need to be made
   TrackDist = [TrackDist; TrackY(final)];
   TrackXTE = [TrackXTE; TrackX(final)];

   % Clear all of the temporary variables
   clear zone1 zone2 zone3 zone4 init dme turn final
   clear alp ang lenDME tempP i
else
   TrackDist = TrackY;
   TrackXTE = TrackX;
end

% Save everything
eval(['save ', filename])
```

savesumm.m

```
function [] = savesumm(summary)
%SAVESUMM
%  savesumm(summary) - saves the matrix to a new file in space delimited ascii format

% Remove any catalog information collected by readcat
% First find the readcat mark in the string
mark = findstr(summary(1,:), '@!#rc');

if (~isempty(mark))
% readcat mark found
   start = mark;
   len = summary(1,start+5);
   vect = [];
   for i = 1:length(start)
      vect = [vect,start(i):(start(i)+len(i)+5)];
   end
   summary(:,vect) = [];
end

% Find a new filename
found = 0;
indx = 0;
while ((~found) & (indx <= 999))
   fname = sprintf('summ%03d.dat', indx);
   if (exist(fname) ~= 2), found = 1; end
   indx = indx + 1;
end

% Open the file
fid = fopen(fname,'wt');

[m,n] = size(summary);
%  summary(:,1) = Subject Number
%  summary(:,2) = File Index
%  summary(:,3) = File Identifier
%  summary(:,4) = Run Number
%  summary(:,5) = Flight Plan type
%  summary(:,6) = Approach Plate
%  summary(:,7) = Wind Direction in Compass degrees
%  summary(:,8) = Runway Heading
%  summary(:,9) = Step Direction (+1 = Left Step; -1 = Right Step)
%  summary(:,10) = MAP Altitude
%  summary(:,11) = Display Used
%  summary(:,12:->) = data from the processing routines that were run
```

savesumm.m

```
% Convert the string codes in the first column back
into Subject ID's
SubjID = strcode(summary(:,1));

% Print out a list of labels and create the format
string for printing
% each line of the matrix.
fprintf(fid, 'SubjID SubjNo SeqNo RunNo AppType
AppPlate Disp DispNo Wind Step Turn');
formstr = abs(['%c']')*ones(size(SubjID(1,:)));
formstr = setstr(formstr(:)');
formstr = [formstr, ' %d %d %d %d %c %c %d %d %d
%d'];
for k = 1:(n-11)
   fprintf(fid, ' var%03d', k);
   formstr = [formstr, ' %f'];
end
fprintf(fid, '\n');
formstr = [formstr, '\n'];

% Convert display codes to numbers
displays = ['X'; 'T'; 'V'; 'P'; 'E'];
[sd,I] = sort(displays);
dispno = lookup([sd,I],summary(:,11));

% Convert Subject ID's to numbers
dir
subjects;     % First load the list of all subjects
[ss,I] = sort(strcode(SubjectList(:,1:SubjIDlen)));
subjno = lookup([ss,I],summary(:,1));

% Recode the first few columns to be more useful in
Sysstat
outsum = [SubjID, subjno, summary(:,[2,4:6,11]),
dispno, -((abs(summary(:,5)-2.5)*2)-2),...
      summary(:,9), -rem(summary(:,5)-2,2),
summary(:,[12:n])];

% Print out the matrix
disp(['Saving summary information to file: ',
fname]);
fprintf(fid, formstr, outsum');
fclose(fid);
```

showcat.m

```
function [] = savesum(summary)
% showcat
%  - displays the readable catalog collected by
readcat

% First find the readcat mark in the string
mark = findstr(summary(1,:), '@!#rc');
if (isempty(mark))
% readcat mark not found
   return;
end

start = mark(1)+6;
len = summary(1,start-1);
disp(summary(:,start:(start+len-1)))
```

stderr.m

```
function y = stderr(t,x)
% STDERR - create mean and standard error bar plots
%  y = stderr(D)
%     outputs the mean and standard error values

% Calculate mean and standard deviation
[m,n] = size(x);
avg = sum(x)/m;
if (m == 1) + (n == 1)
   m = max(m,n);
   sdev = norm(x-avg);
else
   sdev = zeros(size(avg));
   for i=1:n
      sdev(i) = norm(x(:,i)-avg(i));
   end
end
if m == 1
   sdev = 0;
else
   sdev = sdev / sqrt(m-1);
end
serr = sdev/sqrt(m);

y = zeros(3,n);
y(1,:) = avg;
y(2,:) = avg-serr;
y(3,:) = avg+serr;
y(4,:) = sdev;

% Draw the plot
errorbar(t,avg,serr,serr,'o')
```

stepfit.m

```
% stepfit
%  - processing code for fitting a model to the
step
disp(['  Running stepfit using file: ', filename])

% Lookup XTE values at 0.02nm intervals along the
track
% after the step
xd = [3.7:-0.05:0]';
y = extract(xd, [TrackDist, TrackXTE], 1);

t = 3.7 - xd;
y(1) = .25;      % Make sure the first point has
correct step XTE
%w = ((abs(PlanData(1)-2.5)*2)-2);  % Calculate
wind direction from flight plan type
w = (y(2) - y(1))/0.05;
%lam = leastsq('linfit',[-1,-1,-2,-2,w],[1,1e-6,1e-
6],[],t,y,'lin4resp'); % 1/(4*sqrt(2)-1)
%lam = leastsq('linfit',[-1,-2,-2,w],[1,1e-6,1e-
6],[],t,y,'lin3resp'); % 1/(4*sqrt(2)-1)
lam = leastsq('linfit',[-1,-1,w],[0,1e-6,1e-
6],[],t,y,'lin2resp'); % 1/(4*sqrt(2)-1)

% Calculate the model response
ysim = lin2resp(lam, t);

% Collect some statistics from the model response
% 1) Max Value
```

taeplot.m

```
% taeplot
%  - plots TAE versus distance to end
disp(['   Running taeplot using file: ', filename])

if (PlanData(7) == 'T')
   subplot(1,4,1)
elseif (PlanData(7) == 'P')
   subplot(1,4,2)
elseif (PlanData(7) == 'V')
   subplot(1,4,3)
elseif (PlanData(7) == 'E')
   subplot(1,4,4)
end
[n,m] = size(FlightData);
% flip for crosswinds
if ((PlanData(1) == 1) | (PlanData(1) == 4))
   flip = 1;
elseif ((PlanData(1) == 2) | (PlanData(1) == 3))
   flip = -1;
end

plot(FlightData(2:n,TAE)*flip,
FlightData(2:n,DistToEnd))
if (~ishold)
%  axis('equal')
   axis([-.8 .8 5 22])
   grid
   hold on
   if (PlanData(7) == 'T')
      title('T Display')
   elseif (PlanData(7) == 'P')
      title('P Display')
   elseif (PlanData(7) == 'V')
      title('V Display')
   elseif (PlanData(7) == 'E')
      title('E Display')
   end
end
```

wind.m

```
function [dirn, speed, W] = wind(FlightData,
TrueHeading, GroundTrack, Airspeed, GroundSpeed)

hdg = FlightData(:,TrueHeading)*pi/180;
trk = FlightData(:,GroundTrack)*pi/180;
aspd = FlightData(:,Airspeed);
gspd = FlightData(:,GroundSpeed);

tvel = aspd.*exp(sqrt(-1).*hdg);
gvel = gspd.*exp(sqrt(-1).*trk);

W = gvel(2:(length(gvel))) - tvel(1:(length(tvel)-
1));

wspd = abs(W);
whdg = angle(W)*180/pi;
wsd = std(wspd);
wav = mean(wspd);
ind = find(abs(wspd-wav)>(2*wsd));
W(ind) = [];
whdg(ind) = [];
wspd(ind) = [];

speed = mean(wspd);
dirn = mean(whdg)+180;
if dirn > 360, dirn = dirn - 360; end
if dirn < 0, dirn = dirn + 360; end
```

windproc.m

```
% windproc
%  process files for wind information

disp(['   Running windproc using file: ',
filename])

% do the calculations
[hdg,spd] = wind(FlightData, TrueHeading,
GroundTrack, Airspeed, GroundSpeed);

% Save the data and clean up
if (~exist('procsum'))
   procsum = [hdg, spd];
else
   procsum = [procsum, hdg, spd];
end
clear corfp corhdg hdg spd
```

xteboxes.m

```
function xteboxes(x, dist)
% XTEBOXES - draws box plots at each XTE slice
%  x = columns recording sets of xte values for
different distance slices
%  dist = vector of distances, ie. 18:-.2:0
%  requires one value in dist for each column in x

[xrow,xcol] = size(x);

if min(xrow,xcol) > 1
   [m n] = size(x);
   xx = x(:,1);
   yy = xx;
else
   n = 1;
   xx = x;
   yy = x;
end

%lb = 1:n;
lb = dist;

notch = 1;
sym = 'r+';
vert = 0;
whis = 1.5;

step = min(abs(diff(dist)));
xlims = [min(lb)-0.5*step, max(lb)+0.5*step];

k = find(~isnan(x));
ymin = min(min(x(k)));
```

stepfit.m

```
maxval = max(ysim);
minval = min(ysim);
if abs(minval)>maxval
   maxval = minval;
end
% 2) Distance to First Zero Crossing (Intercept
Distance)
zerodist = lookup([ysim, t], 0);
zerodist = [zerodist; 3.7];
zerodist = zerodist(1);
% 3) Intercept Angle
intang = extract([.15;.14], [ysim, t], -1);
intang = atan((.15-.14)/diff(intang))*180/pi;

% 4) Max deviation after first zero crossing
if zerodist == 3.7
   maxremn = 0;
else
   distind = [3.7:-0.05:zerodist];
   remnant = lookup([t,ysim],distind);
   maxremn = max(remnant);
   minremn = min(remnant);
   if abs(minremn) > maxremn
      maxremn = minremn;
   end
end
% 5) Natural frequencies and damping factors
eigv = [lam(1)+sqrt(sign(lam(2)))*lam(2); lam(1) -
sqrt(sign(lam(2)))*lam(2)];
[Wn, Z] = damp(eigv);

% 6) Maximum intercept angle before zero crossing
distind = [0:0.01:zerodist];
remnant = lookup([t,ysim],distind);
[md, ii] = min(diff(remnant));
maxang = atan(-md/.01)*180/pi;
anglocn = 3.7 - (ii-1)*.01;

procsum = [procsum, lam, maxval, zerodist, intang,
maxremn, Wn', Z', maxang, anglocn];
```

stepplot.m

```
% stepplot
%  - processing code for plotting steps
%  - runs stepfit first to obtain the least square
model parameters

% Run stepfit first
stepfit

disp(['   Running stepplot using file: ',
filename])

% Plot the results
plotno = Files(fileIND, 2);
if (plotno > 8)
   figure(2)
   plotno = plotno - 8;
else
   figure(1)
end
subplot(2, 4, plotno)
plot(t,y,t,ysim)
title(filename)
xlabel(['lam = ', mat2str(lam,3)])
grid
```

strcode.m

```
function out = strcode(in, N)
% STRCODE - converts from/to a character string
to/from a number
%  out = strcode(in);
%  If the input argument is a string then the
output will be a number
%  that is a numeric coding of that string
%  If the input argument is a number then the
output will be the string
%  that is represented by the number
%  If N is supplied then the output string will be
padded to be at least
%  N characters wide
%  (the maximum returnable string length is 11
characters)

if isstr(in)
   [m,n] = size(in);
   spcind = find(in == abs(' '));
   in(spcind) = ones(size(spcind))*(abs('a')-1);
   out = (abs(lower(in))-abs('a')+1)*(27.^[0:(n-
1)]');
else
   temp = in;
   out = [];
   while (max(temp) > 0)
      out = [out, rem(temp, 27)];
      temp = fix(temp/27);
   end
   spcind = find(out == 0);
   out = out + abs('a') - 1;
   out(spcind) = ones(size(spcind))*abs(' ');
   out = setstr(out);
   if nargin > 1
      [m,n] = size(out);
      if n < N
         out = [out, setstr(ones(m,N-n)*abs('
'))];
      end
   end
end
```

```
      title('T Display')
   elseif (PlanData(7) == 'P')
      title('P Display')
   elseif (PlanData(7) == 'V')
      title('V Display')
   elseif (PlanData(7) == 'X')
      title('X Display')
   elseif (PlanData(7) == 'E')
      title('E Display')
   end
end
```

xteslice.m

```
% xteslice
%  - processing code to record XTE slices

disp(['   Running xteslice using file: ',
filename])

% Lookup XTE values at 0.2nm intervals along the
track
xd = [18:-0.2:0];
yd = extract(xd, [TrackDist, TrackXTE], 1);

procsum = [procsum, yd];
```

xtestats.m

```
% xtestats
%  - processing code to record along track
statistics from XTE slices

disp(['   Running xtestats using file: ',
filename])

% Lookup XTE values at 0.05nm intervals along the
track
% Roll attitude for the entire run
xd = [18:-.05:0];
finind = find(finite(TrackXTE));
yd = extract(xd, [TrackDist(finind),
FlightData(:,Roll), FlightData(:,Pitch)], 1);
rollstats = longstat(yd(1,:));
pitchstats = longstat(yd(2,:));

% During the arc
xd = [16:-0.05:13];
yd = extract(xd, [TrackDist, TrackXTE], 1);
arcstats = longstat(yd);

% Starting 4 miles before the FAF (just after the
turn)
xd = [10:-0.05:8];
yd = extract(xd, [TrackDist, TrackXTE], 1);
turnstats = longstat(yd);

% during the sensitivity change
xd = [8:-0.05:6];
yd = extract(xd, [TrackDist, TrackXTE], 1);
sensstats = longstat(yd);

% Starting at the FAF and continuing for 2 miles
xd = [6:-0.05:4];
yd = extract(xd, [TrackDist, TrackXTE], 1);
fafstats = longstat(yd);

% Starting 2 miles before the MAP
xd = [2:-0.05:0];
yd = extract(xd, [TrackDist, TrackXTE], 1);
mapstats = longstat(yd);


procsum = [procsum, rollstats, pitchstats,
turnstats, sensstats, fafstats, mapstats,
arcstats];
```

```
ymax = max(max(x(k)));
dy = (ymax-ymin)/20;
ylims = [(ymin-dy) (ymax+dy)];

lf = (max(xlims) - min(xlims)) *
min(0.15,0.5*step/(max(lb)-min(lb)));

% Scale axis for vertical or horizontal boxes.
set(gca,'NextPlot','add','Box','on');
axis([ylims xlims]);
%set(gca,'YTick',lb);
set(gca,'XLabel',text(0,0,'XTE [nm]'));
set(gca,'YLabel',text(0,0,'Distance to Go [nm]'));

if n==1
   vec = find(~isnan(yy));
   if ~isempty(vec)
      boxutil(yy(vec),notch,lb,lf,sym,vert,whis);
   end
else
   for i=1:n
      z = x(:,i);
      vec = find(~isnan(z));
      if ~isempty(vec)
   boxutil(z(vec),notch,lb(i),lf,sym,vert,whis);
      end
   end
end
set(gca,'NextPlot','replace');
```

xtelimit.m

```
function [xnum,xavg,xvar,xerr,xtth,xout] =
xtelimit(x,dist)
% XTELIMIT - draws 95% limits based on XTE data

[m,n] = size(x);
p = 1 - 0.025;
np = norminv(p);
% for each slice
xnum = zeros(1,n);
xavg = zeros(1,n);
xvar = zeros(1,n);
xout = ones(size(x))*NaN;
xout(1,:) = zeros(1,n);
for indx = 1:n
   k = find(finite(x(:,indx)));
   xnum(indx) = length(k);
   if (xnum(indx) > 1)
      xavg(indx) = sum(x(k,indx))/xnum(indx);
      cent = x(k,indx) - xavg(indx);
      xvar(indx) = sum(cent.*cent)/(xnum(indx)-1);
      testk = k;
      testavg = xavg(indx);
      done = 0;
      while (~done)
% Find the point that is the furthest distance
from the current mean
         [y, testout] = max(abs(cent));
% Eliminate that point and check if it is an
outlier
         outk = testk(testout);
         testk(testout) = [];
         testnum = length(testk);
% Even if the current one is an outlier,
% if there are fewer than 10 points left then
stop after this one
         if testnum <= 10, done = 1; end
         testavg = sum(x(testk,indx))/testnum;
         cent = x(testk,indx) - testavg;
         testvar = sum(cent.*cent)/(testnum-1);
% If it was an outlier then record it and go back
to check the next point
% otherwise, stop processing.
         if (abs(x(testout,indx) -
testavg)>norminv(.99995)*sqrt(testvar))
            xout(1,indx) = xout(1,indx) + 1;
            xout(xout(1,indx)+1,indx) =
x(outk,indx);
            xnum(indx) = testnum;
            xavg(indx) = testavg;
            xvar(indx) = testvar;
         else
            done = 1;
         end
      end
   else
      xavg(indx) = NaN;
      xvar(indx) = NaN;
   end
end
tval = xavg./sqrt(xvar./xnum);
sig = tcdf(tval,xnum-1);
sig = 2* min(sig,1-sig);
xtth = (sig <= 0.05);
xerr = np*sqrt(xvar.*chi2inv(p,xnum-1)./(xnum-1));

% Trim xout to remove unneccessary NaN's
p = max(xout(1,:));
xout = xout(2:(p+1),:);
```

xteplot.m

```
% xteplot
% - plots XTE versus distance to end
disp(['   Running xteplot using file: ', filename])

if (PlanData(7) == 'T')
   subplot(1,4,1)
elseif (PlanData(7) == 'P')
   subplot(1,4,2)
elseif (PlanData(7) == 'V')
   subplot(1,4,3)
elseif (PlanData(7) == 'X')
   subplot(1,4,4)
elseif (PlanData(7) == 'E')
   subplot(1,4,4)
end
%[n,m] = size(FlightData);
% flip for crosswinds
if ((PlanData(1) == 2) | (PlanData(1) == 4))
   flip = 1;
elseif ((PlanData(1) == 1) | (PlanData(1) == 3))
   flip = -1;
end

plot(TrackXTE*flip, TrackDist)
if (~ishold)
%  axis('equal')
   axis([-.8 .8 0 18])
   grid
   hold on
   if (PlanData(7) == 'T')
```

REFERENCES

[Bennett95] Bennett, S., *personal communication*, July 1995.

[Clement67] Clement, W.F., Best, J.J., Stapleford, R.L., *Analysis of Display-Pilot-Vehicle Dynamics for ILS Approach of a large Jet Aircraft*, Systems Technology, Inc., TM163-C, May 1967.

[Etkin80] Etkin, B., The turbulent wind and its effect on flight. *AIAA Aircraft Systems Meeting*, Anaheim, CA, AIAA-80-1836, 1980.

[FAA95] Federal Aviation Administration, *Technical Standard Order C129a: Airborne Supplemental Navigation Equipment Using the GPS*, FAA, Washington, DC, 1995.

[Gallaher77] Gallaher, P.D., Hunt, R.A. and Williges, R.C., A regression approach to generate aircraft predictive information. *Human Factors*, vol. 19, pp549-566, 1977

[Grunwald85] Grunwald, A.J., Predictor Laws for Pictorial Flight Displays, *Journal of Guidance*, vol. 8, pp545-552, Sept-Oct 1985.

[Jansen81] Jansen, C.J., Non-Gaussian Atmospheric Turbulence Model for Flight Simulator Research, *AIAA Journal of Aircraft*, vol19: 374-379, 1981.

[Kelley60] Kelley, C.R., *Developing and testing the effectiveness of the predictor instrument*, Santa Monica, CA, Dunlap and Associates, Inc., Technical report 252-60-1, March, 1960.

[Kelley68] Kelley, C.R., *Manual and Automatic Control*. New York, Wiley and Sons. 1968

[McRuer65] McRuer, D., Graham, D., Krendel, E., Reisener Jr., W., *Human Pilot Dynamics in Compensatory Systems: Theory, Models and Experiments with Controlled Element and Forcing Function variations*. AFFDL-TR-65-15, July 1965.

[McRuer67] McRuer, D., Jex, H., Clement, W., Graham, D., *A Systems analysis theory for displays in manual control*, Systems Technology, Inc., TR 163-1, October 1967.

[Moesdijk78] van de Moesdijk, G.A.J., *Non-Gaussian structure of the simulated turbulent environment in piloted flight simulation*, Delft University of Technology, Memorandum M-304, 1978.

[Oman95] Oman, C.M., Huntley, M.S., Rasmussen, S.A.M., *The use of Analog Track Angle Error Display for Improving Simulated GPS Approach Performance*, DOT-VNTSC-FAA-95-29, 1995.

[RTCA91] RTCA Special Committee 15, *Minimum Operational Performance Standards For Airborne Supplemental Navigation Equipment Using Global Positioning System*, DO-208, Washington, DC, 1991.

[Wells89] Wells, D., ed., *Guide to GPS positioning*, Fredericton, NB, Canada, Canadian GPS Associates. 1989.

[Wickens92] Wickens, C.D., *Engineering Psychology and Human Performance*, Illinois, HarperCollins, 1992.