Flight Segment Identification as a Basis for Pilot Advisory Systems

Dr. Wallace E. Kelly III*
Blue Rock Research and Development, Inc., Asheboro, North Carolina, 27502

Dr. John H. Painter†
Altair Corporation, College Station, Texas,

Flight Segment Identification (FSI) is the process of monitoring aircraft state variables and flight events to identify in real-time the phase of flight or operational procedure of an aircraft. It is a dynamic classification problem in which the state space is highly dimensional and the boundaries between the various flight phases are not crisply defined. Examples of flight segments include "enroute cruise," "holding," and "initial approach." We explain the role of Flight Segment Identification (FSI) in building pilot advisory systems, including a new distinction that we propose between state-based FSI and procedural FSI. State-based FSI uses the aircraft state variables to infer the flight operation *currently being executed* by the pilot. Procedural FSI tracks the flight operation that the pilot *should be executing* now – based on events and flight rules that are largely outside the control of the pilot. We present one approach to performing Flight Segment Identification based on fuzzy sets and how we applied this solution to the NASA High Volume Operations concept. Finally, we discuss our results and conclusions from recent flight tests of a Flight Segment Identifier.

Nomenclature

$m_A(x)$	=	the fuzzy degree of membership of data x in fuzzy set A.
l_i	=	a prototype point, to define a hypertrapezoidal fuzzy set i.
s	=	the crispness factor of a hypertrapezoidal partitioning.
$d(x, y)$	=	the Euclidean distance between data points x and y.
$r_{i\|j}(x)$	=	an intermediate distance value used for calculating hypertrapezoidal fuzzy membership.

I. Introduction

Avionics software with artificial intelligence could assist pilots in following flight procedures. There are several motivations for doing this. The first is that the amount of available information in future cockpits will continue to grow and the complexity of the avionics to manage that information will likewise increase. AI technologies can help monitor and prioritize the information flow in the cockpit. Secondly, "smarter" software in the cockpit would simplify the task of flight for newer and less experienced pilots. NASA and other industry leaders foresee a new class of "personal air vehicles," which brings aviation to more people. Making those vehicles easier to operate is an important research goal for the industry. The goal was recently described by NASA as "simplify the operation of small aircraft such that the specialized skills, knowledge, and associated training are reduced to levels comparable to operating an automobile or boat."[1] Another motivation, which is the primary focus of this paper, is that avionics with Pilot Advisor functionality can enable new flight procedures that make our National Airspace System operate more efficiently.

It is worth clarifying what is meant by "avionics software with artificial intelligence." The goal is to engineer onboard computer systems that assist the pilot much like an instructor pilot might monitor and advise a student pilot.

---

* President, Blue Rock Research, 4998 Old NC-49, Asheboro, NC, 27205-0117, Member, AIAA.
† President, Altair Corporation, P.O. Box 10046, College Station, TX, 77842-0046, Member, AIAA.

Completely replicating that kind of expertise would be quite a challenge. The industry is no where near that level of capability. But we are making progress.

Besides the in-flight benefits, there is also the potential for a training aid during simulation. A “verbose mode” of a Pilot Advisor can be used to pause a flight simulator when a student makes a significant mistake. Either a printed commentary may be projected on the display or synthetic voice may give the required training commentary. After digesting the commentary, the student may then un-pause the simulator and proceed with the flight. This training mode in a Pilot Advisor would be extremely useful in teaching such procedures as Instrument Approaches (ILS, GPS, etc.)

Providing in-flight pilot advising or aviation training software are worthy goals in themselves. But there is a problem looming in the National Airspace System that can also be addressed by “smarter” avionics. Many observers believe that the NAS is not prepared for the expected rise in air traffic over the next two decades. That rise is due to two factors – an increase in airline business and an increase in the number of smaller aircraft. Several companies are developing Very Light Jets and Personal Air Vehicles. These smaller vehicles are expected to place an additional strain on the NAS that can not be easily addressed with additional infrastructure. The solution may be to better utilize “community airports.”

Why are “community airports” important? As shown in Fig. 1, only 22 percent of the population of the United States lives within 30 minutes of a major hub airport, such as Chicago O'Hare International Airport or the Dallas/Fort Worth International Airport. Forty-one percent live within 30 minutes of a regional airport such as Bryan-College Station's Easterwood Airport. But 93 percent of the people in the United States live near a small, community airport. These community airports currently serve a relatively small number of general aviation pilots. While the major hubs are straining under the load of increasing air travel, community airports are underutilized. Community airports have the potential to become a huge national asset, if we can overcome the barriers to personal air transportation.

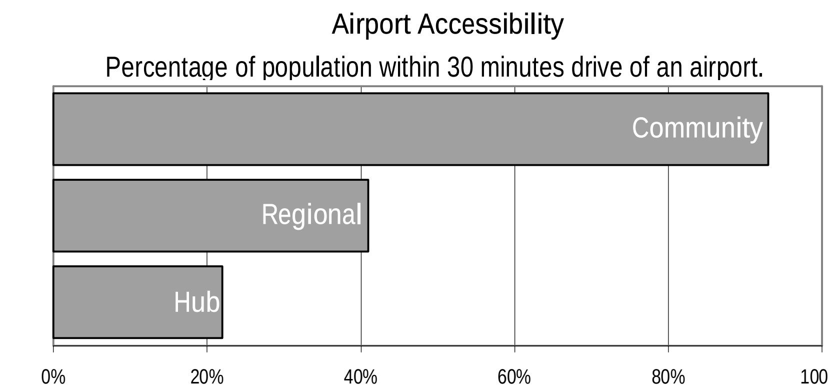


Figure 1. Airport Accessibility.

One of the barriers to personal air transportation is the lack of full air traffic control infrastructure at the community airports. The High Volume Operations (HVO) concept was developed by NASA researchers on the Small Aircraft Transportation System (SATS) research project to address this need. The concept is intended to open up consistent (i.e., all-weather) access to the large number of community airports across the country. But, rather than duplicating the infrastructure that exists at larger airports, HVO uses a combination of ground sequencing software, new flight procedures, and on-board pilot advisory software. One researcher describes HVO as follows:

> A concept for multiple operations during Instrument Meteorological Conditions (IMC) at non-tower, non-radar airports is described. The objective is to provide an automated service which will support separation assurance for aircraft operating in the airport airspace. This type of service will enable the use of a large number of airfields which currently have limited use in IMC. The service must be provided with minimal infrastructure and at low cost.[2]

Researchers expect the aviation industry to rely on “automated services” to grow the NAS. The “automated services” exist both on the ground and in the aircraft.

The primary focus of this paper is on using artificial intelligence to enable new types of flight procedures that improve the ability of National Airspace Systems to accommodate the expected growth in air traffic. For the authors, artificial intelligence is a general term that refers to embedding into systems the knowledge and logic to perform functions that are generally performed by a human today. For example, humans think about their flights in specific stages. That is, pilots maintain a mental model of their flight as a series of flight segments. It would be helpful, for reasons discussed below, if the onboard software similarly maintained a model of the segments of a flight and tracked the aircraft as it operated in and transitioned between those flight segments. We call that software process, Flight Segment Identification (FSI). FSI provides context for the avionics to provide pilot advisories, information and display management.

II. Pilot Advisory Systems

A. Automated Safety and Training Avionics (ASTRA)

The work described in this paper follows more than a decade of research at Texas A&M University on pilot advisory systems.[3-6] Texas A&M University has been maturing algorithms, software and displays to help those piloting small aircraft in all weather conditions. The result is ASTRA – Automated Safety and Training Avionics. In ASTRA, artificial intelligence is used to assist with decision making in the cockpit and to anticipate problems before they occur.

The ASTRA Program commenced in 1994 in the departments of Aerospace Engineering and Electrical Engineering at Texas A&M University. The principals were two faculty, one of whom had been an Air Force Navigator and the other an Air Force Test Pilot. Motivation was provided by the then current development of the Air Force's Pilot Associate, whose results were available to the principals. In the early days of the A&M research, ASTRA was known as the Poor Man's Pilot Associate.

As a result of the first round of NASA (Langley Research Center) funding from 1994 through 1998, a medium fidelity, fixed-base Flight Simulator (3 screens) was created, which allowed immediate "flight" evaluation of cockpit software. The physical simulator was created from the fuselage and cockpit of a surplus Air Force T-37 jet trainer. The instrument panel was gutted and two CRT projection screens were emplaced. Later upgrades replaced the CRTs with LED touch screens. Three overhead projectors onto three screens in front of the cockpit yielded a 150-degree field of view, providing a high-level of experienced reality to the pilot.

The first generation of the ASTRA software included a projected HUD, including ILS approach display, plus the usual instrument tapes. An innovation for that time was a "Virtual Runway," which was a runway projection, based on assumed GPS navigation precision. The runway was very useful for instrument flight. Cockpit displays included a moving map based on a Jeppesen aeronautical database. Software modules included the first-generation Flight Segment Identifier (FSI) and a rudimentary rule-based (CLIPS) Pilot Advisor (PA). The FSI was based on standard Fuzzy Logic. Later generations improved both the FSI and PA.

All the software in the Flight Simulator was developed by A&M graduate students, in pursuit of M.S. and Ph.D. degrees. The generations of software corresponded to the generations of students that passed through the ASTRA program. The second generation yielded software and display modules for weather avoidance and collision avoidance. The second generation also created a six-degree of freedom autopilot and flight management system for the light twin aircraft which supported the research. There have now been about four generations of students and of ASTRA software.

Today, A&M's ASTRA has progressed to support of interlinked multiple pilot stations, as well as the medium-fidelity original flight simulator, all operating in the same flight context. Figure 2 includes recent photos of the laboratory. The balance of this paper adds to the understanding of current ASTRA capabilities.

Figure 2. TAMU Engineering Flight Simulator.

B. Flight Segments

A significant innovation in the ASTRA architecture (see Fig. 3) is the specific acknowledgement of the problem of "flight segment identification." In this case, the term flight segment is used generically to mean any qualitative description of the current operating state that would be useful for the purposes of advising a pilot. That is, how

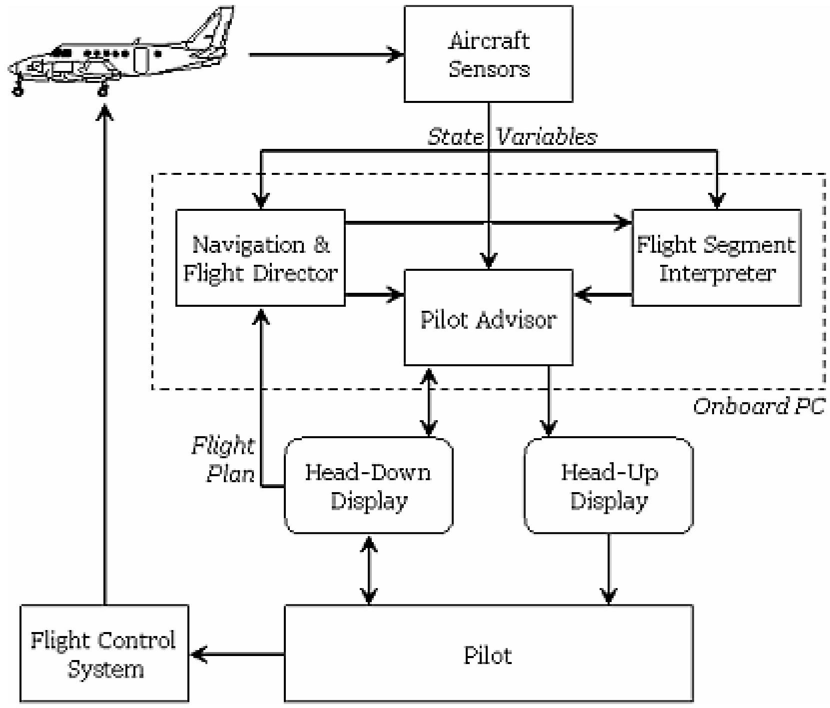


Figure 3. Automated Safety and Training Avionics Architecture (circa 1994).

might an expert pilot describe the current situation? (One can see why the term artificial intelligence is appropriate to this problem.)

The specific list of flight segments vary from application to application. For general pilot advising over the entire course of a flight the flight segments might be defined as follows: *taxi*, *takeoff*, *climb out*, *cruise*, *initial approach*, *final approach*, and *landing*. In a recent application of the ASTRA architecture to the High Volume Operations concept the flight segments were the specific steps in the HVO procedures: *vertical entry*, *lateral entry*, *holding high*, *holding low*, *base segment approach*, etc. For a homeland security application the flight segments might be: *enroute*, *diverting due to weather*, *on approach*, *off-course*, *unexpected maneuvering*, etc.

Similarly, the action that is taken based on the identified flight segment varies from application to application. For general pilot advising, the identified flight segment provides the basis for an expert system that generates warnings, cautions, and advisories on the pilot display. On NASA's Small Airport Transportation System program, Blue Rock Research used the inferred flight segment to automate a synthetic vision Highway In The Sky (HITS) display. Once the software "knew" what the pilot should be doing, it commanded the HITS to guide the pilot in performing that procedure. In the homeland security application, knowing that the pilot's actions do not match what the pilot should be doing could generate an alarm to security officials on the ground.

Regardless of the application, the flight segment identification problem is the same. Given the aircraft and environment variables that are available to the avionics, "*What is the pilot currently doing?*" and "*What should the pilot be doing?*" We answer these questions by classifying the current operating mode into discrete sets. Years of research at Texas A&M University has shown that being able to robustly answer these questions in real time is the linchpin for successfully engineering pilot advising software.

C. Example Application

Texas A&M University and Blue Rock Research recently partnered with the North Carolina and Upper Great Plains (NC&UGP) SATSLAB to modify, augment, and apply the ASTRA-developed technologies to the High Volume Operations (HVO) development and demonstration. The HVO concept is a good example of the kinds of new procedures that can be implemented in the NAS. The new procedures improve efficiency, relying largely on onboard automation like ASTRA.

As described in the introduction of this paper, the HVO concept was designed to increase the throughput of community airports, removing one of the technology barriers to on-demand air taxi services. The goal is to "enable simultaneous operations by multiple aircraft in non-radar airspace at and around small non-towered airports in near all-weather."[7] Researchers would like to accomplish this goal without duplicating the ground infrastructure that exists at today's larger airports. The HVO solution includes the following elements:

- *Self-Controlled Area (SCA)*
  The flight operations area defined around a community airport, in which HVO flight rules can apply.
- *Airport Management Module (AMM)*
  Ground-based software installed at HVO airports, which assigns entry type (either vertical or horizontal) and the landing sequence for the approach aircraft.
- *Conflict Detection and Alerting (CD&A)*
  An onboard separation assurance system which alerts the pilot when aircraft are projected to fly unacceptably close to each other; likely to be integrated into a cockpit display of traffic information.
- *Pilot Advising (PA)*
  Onboard software which provides information (generally in the form of textual, graphical, or auditory cues) that is helpful to the pilot performing HVO procedures.

Figure 4 is a diagram of a typical SCA. It consists of two initial approach fixes (IAF), each with two holding altitudes, an intermediate fix (IF), and the final approach fix (FAF). The NASA-defined HVO procedures define the steps that a pilot goes through to transition from outside the SCA, to one of the IAFs, and finally onto approach to the runway. If a missed approach is executed, the pilot flies to the AMM-assigned Missed Approach Holding Fix (MAHF), which is one of the two IAFs. Examples of the HVO procedures include using an onboard system and datalink to request entry from the AMM, holding at a higher altitude if another aircraft is holding at the lower altitude of the assigned IAF, monitoring the lead aircraft to know when one can initiate an approach, etc.

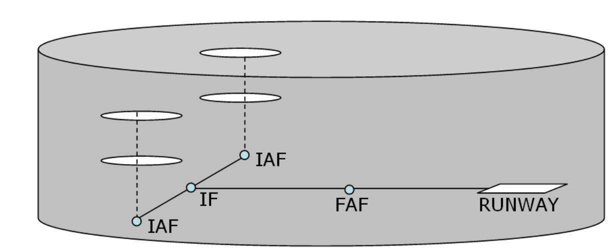


Figure 4. Anatomy of a typical SCA.
Includes initial approach fix (IAF), two holding altitudes, intermediate fix (IF), and final approach fix (FAF).

The authors developed the state diagram in Fig. 5 which includes fourteen flight segments describing the various stages of HVO operations. The altitudes (3,000 and 2,000 feet) are merely representative altitudes. The conditions for transitioning from one stage to the next are not shown in the diagram but are detailed in Refs. 7 and 8. While not a formal requirement in the NASA specifications, it is reasonable to organize the HVO PA logic around this diagram. Our PA system could then cue the pilot that it is now time to descend from 3,000 feet to 2,000 feet. Or, when the PA detects that the pilot is performing a missed approach, it could highlight the AMM-assigned MAHF on the moving map. Therefore, our PA software must have a model of what the pilot should be doing next (e.g., descending from 3,000 feet to 2,000 feet) and what the pilot is actually doing now (e.g., executing a missed approach). *Flight segment identification is the real-time process of evaluating models for flight procedures and pilot actions.*

In addition to performing the prescribed HVO procedures, pilots are also responsible for monitoring separation between aircraft in the SCA. The HVO procedures are designed such that if everything goes as planned, the aircraft will maintain separation. However, there is always the possibility for errors – someone may initiate their approach too soon, for example. In this case, onboard CD&A logic is expected to notify the pilot of the possible loss of separation. The NASA-defined CD&A algorithms include the concept of procedural conformance.[9] In the HVO concept, each aircraft is required to have software that monitors its aircraft's state to determine if it is conforming to the HVO procedures. The HVO requirements state that each aircraft broadcast a "conformance bit" in an extended ADS-B message. *The inclusion of ownship conformance monitoring places a considerable design requirement on the HVO avionics – a design requirement that can be met by flight segment identification.*

In Part III, we describe the details of our approach to flight segment identification, using HVO procedures as an example application.

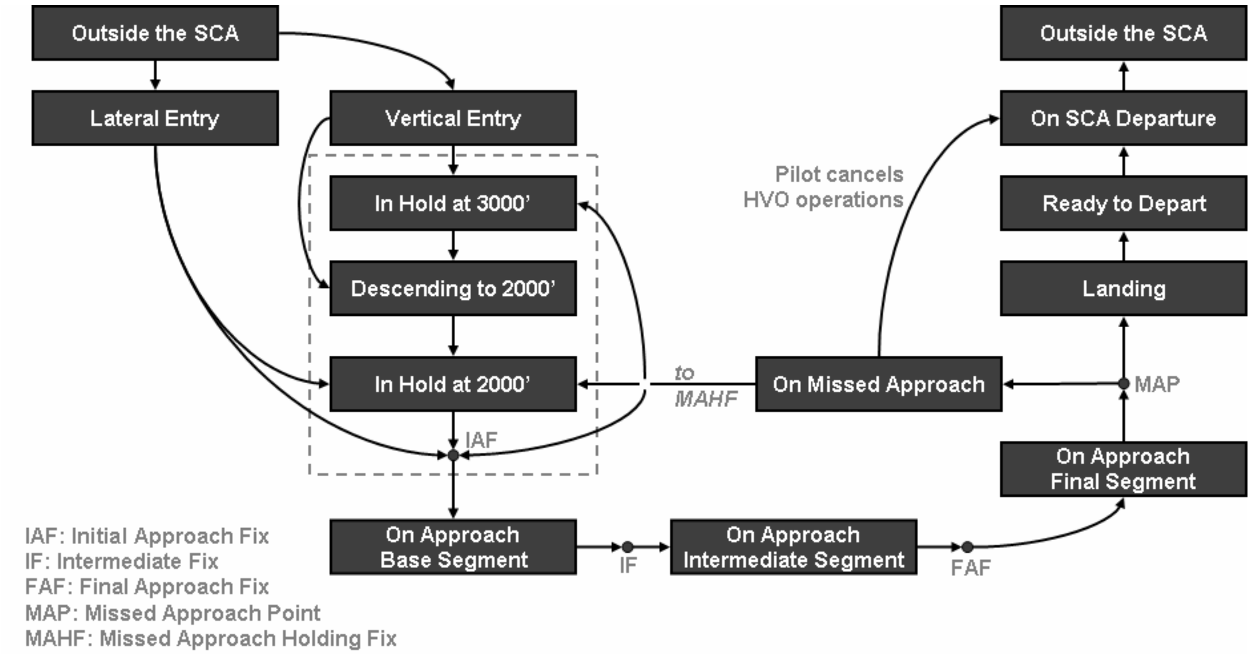


Figure 5. Aircraft's state diagram for High Volume Operations.

III. Flight Segment Identification

Flight Segment Identification (FSI) is the process of monitoring aircraft state variables and flight events to identify in real-time the phase of flight or operational procedure in which an aircraft is operating. FSI can be used to identify the phase of flight that the pilot *is* flying. FSI can also be used to identify the phase of flight that the pilot *should be* flying. One reason to implement FSI in avionics is to support the generation of pilot advisory messages, like those shown in Fig. 6. The messages may be textual, graphical, auditory, or even haptic. Regardless, pilot advisory systems need the context that FSI can provide.

A. State-based FSI

If avionics has a requirement for identifying the flight segment, how might that be implemented? One approach is to include a button for the pilot to press or a knob for the pilot to turn to set the current flight segment. This approach to flight segment identification has the obvious drawbacks of increasing pilot workload and being error-prone. A preferred implementation would track the flight segments automatically without requiring pilot intervention.

If an FSI module should not rely on pilot input, then what is the basis for deciding what flight segment the pilot is currently flying? The decision is made based on the aircraft state variables – position relative to the flight plan, altitude, airspeed, vertical speed, etc. We have termed this decision *state-based flight segment interpretation* (S-FSI). In contrast to the procedural FSI described in the next section, the state-based FSI relies primarily on state data over which the pilot largely has control. It determines what the pilot is doing with the aircraft, without

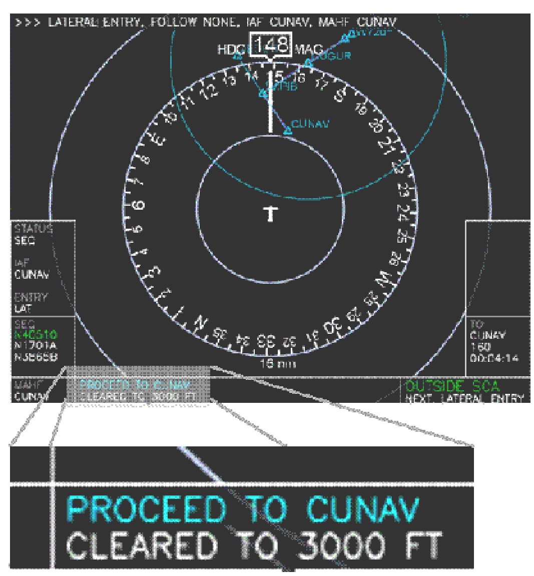


Figure 6. Example of pilot advisories during SATS HVO operations.

asking him.

The specific goal of the state-based FSI is determined by the system designers to support the application at hand. Depending on the application, the S-FSI process answers questions like, “What is the aircraft doing right now?” or “What phase of flight is the aircraft in?” or “What are the inferred intentions of the pilot?”

The SATS HVO concept is a good example of an application that benefits from S-FSI. One of the fields in the HVO extended ADS-B message is based in the pilot’s intent. Specifically, the “next waypoint type” field is expected to be “IAF” while the aircraft remains in a holding position. When the pilot intends to leave the hold and initiate the approach, the HVO software is required to begin broadcasting “MAHF” in the “next waypoint type” field. This functionality could have been implemented by adding a button the pilot presses upon begin the approach. In contrast, our software used the concept of flight segment identification to detect the pilot’s transitioning from “In Hold at 2,000’” to “On Approach Base Segment.” (Reference Fig. 5.)

B. Procedural FSI

There is another FSI decision (in addition to the state-based decision) that has proven useful. “What flight segment *should* the pilot be flying?” Knowing the answer to this question is useful, particularly if we compare it with the result of the S-FSI result. But how could one answer this question in software? We can not expect the pilot to input that information – particularly since we want to provide preemptive guidance. We can not rely on the aircraft state since this reflects what the pilot *is* doing, rather than what the pilot *should be* doing. Consequently, the FSI logic that makes the decision based (as much as possible) on variables and events that are outside the direct control of the pilot. In other words, the logic for this decision should not rely on the pilot. It should be based on the variables and events that define the flight procedures. We use the term *procedural flight segment identification* (P-FSI).

As with the state-based FSI, the specific goal is determined by the system designers to match the application. The P-FSI process answers questions like, “What flight procedure should the pilot be executing now?” or “For which phase of flight should the aircraft be configured?” The procedural FSI is different from the state-based FSI. It is based on data or events which are largely outside of the control of the pilot. For example, in the HVO application, the procedures specify when the pilot can enter the SCA, when the pilot can descend from the upper holding pattern, and when the pilot can begin the approach. The conditions include the messages received from the Airport Management Module (AMM), the position of the *other* aircraft in the pattern, etc. The logic for P-FSI encodes those rules to make a decision about what the pilot should be doing now. While the S-FSI module may detect that the pilot is “In Hold at 3,000,” the P-FSI module may indicate that there is no reason that the pilot should not be “Descending to 2,000.” (Reference Fig. 5.)

C. Flight Segment Display

The question arises, “In addition to the pilot advisory messages, should the identified flight segments be shown directly to the pilot?” On one hand, displaying the FSI results gives pilots the context for the flight – particularly for new procedures like HVO. It may also be helpful in interpreting the pilot advisory messages. For example, if a message is displayed that says “Proceed to CUNAV”, the pilot might understand better the motivation for proceeding to CUNAV, if there was also an indication that the P-FSI is recommending transitioning from “Holding at 2,000” to “On Approach Base Segment.”

The NC&UGP research system included the display of the FSI result – a feature of the display that we termed *Procedure Guidance*. Performing experiments of FSI display concepts was outside the scope of project. The topic deserves more research. At this time, our implementation displays both the *procedural* (i.e., “should be in”) and *state* (i.e., “is operating in”) FSI results in the lower, right-hand corner of the traffic display as shown in Figure 7. The specific details of the display depend on whether the *procedural* and *state*-based results match or at least follow a reasonable progression. In this first research implementation, we displayed the State-

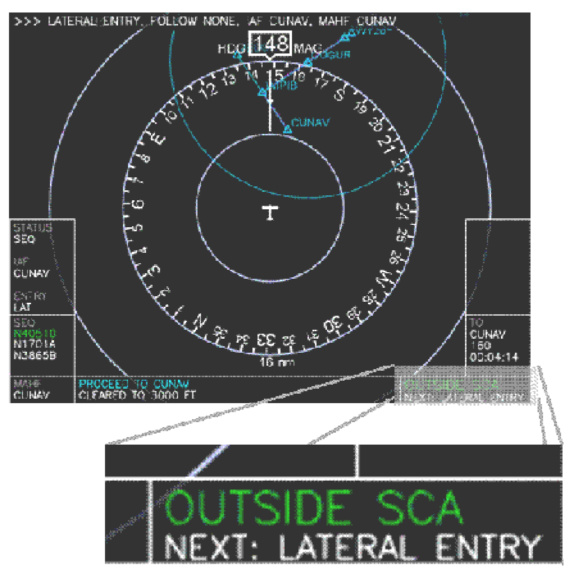


Figure 7. Example of flight segment display during SATS HVO operations.

based Flight Segment Identification (S-FSI) result in large letters as shown in Fig. 8a. If the Procedural Flight Segment Identification (P-FSI) result did not match the S-FSI exactly, we displayed it in smaller letters with the prefix “NEXT:” as shown in Fig. 8b and 8c.

Consider the following three cases and their corresponding examples in Fig. 8.

Case A: Both the procedural (P-FSI) and state-based (S-FSI) results agree. That is, the pilot seems to be performing the procedure that is required at the moment. In this case, the procedural guidance displays the FSI result in green, as shown in Fig. 8a.

Case B: The P-FSI and S-FSI do not match, but the P-FSI (i.e., should be in) reasonably follows the S-FSI (i.e., is operating in) result. For example, the pilot is still holding at the higher SCA holding altitude, but there is no reason that the pilot can not begin the descent to the lower holding altitude. In this case, the S-FSI would be “Holding High.” The P-FSI would be “Descending.” For these situations, we display the procedure guidance as shown in Fig. 8b. It says to the pilot, “You are currently holding high, but you should begin descending to the lower altitude.” The pilot advisory system may display a message to that effect explicitly.

Case C: The P-FSI and S-FSI do not match and the pilot seems to be performing a procedure that the pilot should not be performing. Figure 8c, for example, is displayed if the pilot seems to be performing a vertical entry, but should remain outside the SCA. Notice that the top line, which in all three cases is the State-based FSI result, changes to a yellow color.

The design of the Procedure Guidance display based on S-FSI and P-FSI results deserves more research. In fact, it is still an open question about whether or not the FSI results should be displayed at all.

(a)

(b)

(c)

Figure 8. Examples of procedure guidance on a pilot display.
(a) P-FSI and S-FSI results are identical. Pilot is performing expected procedure.
(b) P-FSI result is a flight segment that nominally follows the S-FSI result. The pilot can transition into the next flight segment.
(c) P-FSI result and S-FSI result do not match or follow the expected sequence. The pilot seems to be transitioning into an unexpected flight segment.

D. Path Guidance

In the SATS HVO application, the NC&UGP research system has S-FSI and P-FSI implemented for the NASA-defined procedures. The FSI results are inputs to the Pilot Advisor, providing a context for evaluating the expert system rules. The system also includes Procedure Guidance (i.e., the display of FSI results as described in the previous section.) There was one additional feature added late in the project – a “wouldn’t-it-be-cool-if-we-could-do-this” feature.

The P-FSI had been successfully implemented and flight tested in the NC&UGP research system. The system includes Highway-In-The-Sky (HITS) guidance based on Nav3D Corporation’s synthetic vision software as shown in Fig. 9. But, the P-FSI result was not being used to drive the HITS. While the P-FSI “knew” that the pilot should descend from 3,000 to 2,000, the HITS would continue to hold the pilot at 3,000’ until the pilot manually dialed down the altitude setting for the HITS holding pattern. The team decided to try connecting the P-FSI output to the HITS. The result was striking.

Again, performing formal experiments on the value of automating the HITS guidance was outside the scope of our SATS project. This is an area that deserves more research. But the subjective and anecdotal evidence from the flight tests suggests that this could be the most important application for FSI technologies. With the P-FSI commanding the HITS display, the HVO procedures could be flown by “following the magenta, wire-frame highway.”

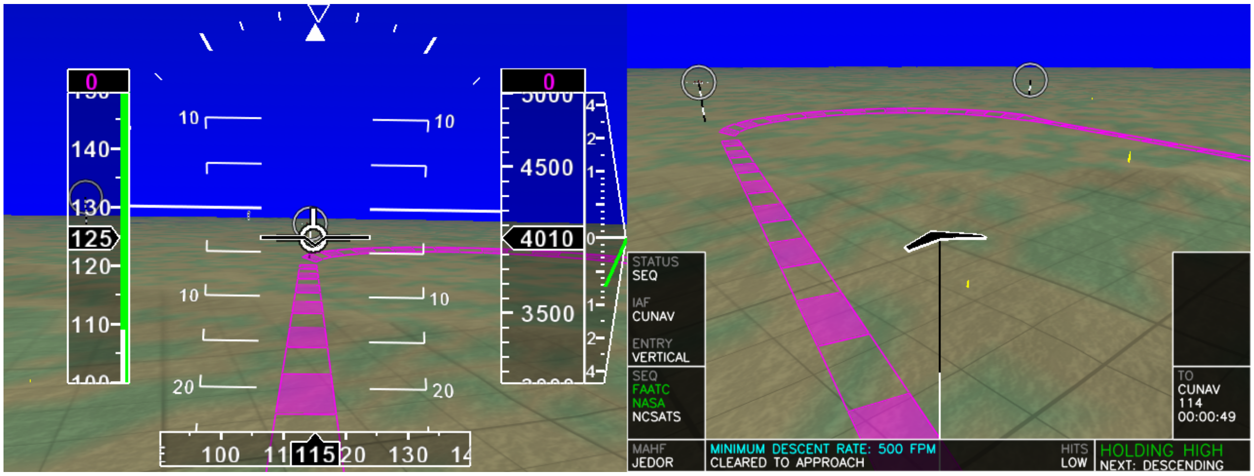


Figure 9. Screenshot from NC&UGP displays, including Highway-In-The-Sky.

IV. Fuzzy Sets and Flight Segment Identification

Our approach to flight segment identification has been based on fuzzy sets. Fuzzy sets are useful for describing sets in which the boundaries between the sets can not be crisply defined. The concept was introduced by Lotfi Zadeh in 1965.L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 - 353, June 1965[10] According to Zadeh, the stated purpose of fuzzy sets is to deal with "classes" that have no "sharply defined criteria of class membership." A fuzzy set is completely defined by its fuzzy membership function, $m(x)$, which gives the degree of membership of an element, x, in a fuzzy set. The classic example is that of the set of tall people. The height of a person will indicate whether or not a person is tall, but the boundary between *tall people* and *short people* can not be drawn at some exact height. Fuzzy sets allow the construction of system models when the sets that comprise the model are not clearly defined.

Why use fuzzy sets as a basis for flight segment identification? After all, the flight segment decision is a crisp decision. The control of the HITS display requires a specific selection of the next flight segment. While the flight segment decision must be a crisp all-or-nothing decision, the flight segments themselves do overlap in the aircraft state space and fuzzy sets are an excellent model of the ambiguity in defining some flight segments. However, there is another motivation for using fuzzy models of the flight segments. The degree of memberships in the fuzzy flight segments can be interpreted as a measure of certainty and used to derive confidences of the flight segment decisions. These certainties and confidence factors allow for filtering the mode decision, a feature we have found valuable in practice.

A. One Dimensional Fuzzy Sets

The membership function indicates the degree of membership of a crisp value in a fuzzy set. A variety of basic shapes can be used to design the fuzzy membership functions. The trapezoidal fuzzy membership function is the most popular. Trapezoids are easily specified and calculated. By far, fuzzy sets are usually defined in one domain at a time. That is, the fuzzy sets are one-dimensional, based on a single variable. Multiple domains are combined using rules, which have fuzzy sets as their antecedents.

The first implementation of flight segment identification logic in ASTRA was performed using a rule-base of one-dimensional fuzzy sets.[11] The rule base is shown in Fig. 10. Each row lists the one-dimensional fuzzy sets for a state-variable. Each column corresponds to a flight segment. The aircraft is operating in a given flight segment to the degree that the current values for power, angle of attack, roll, etc. match the one-dimensional membership function in the flight segment's column.

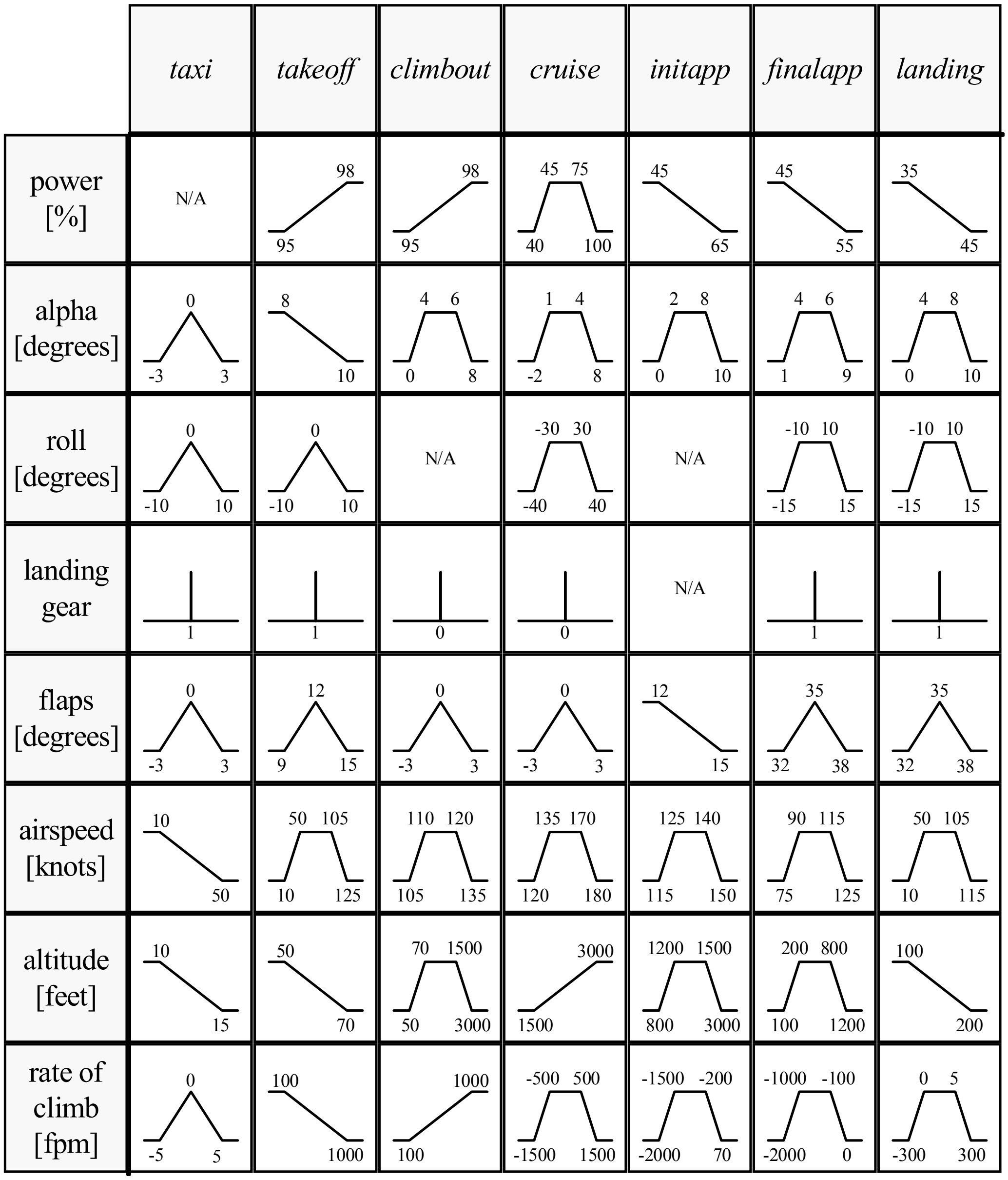


Figure 10. Rules base of one-dimensional fuzzy sets for flight segment identification. (circa 1995)

The one-dimensional FSI provided a glimpse of the usefulness of an avionics system which maintains a qualitative assessment of the current flight procedure. However, it also revealed the challenges of the flight segment identification problem. Tuning the fuzzy set definitions and rule base is a time-consuming, trial-and-error process. And, more importantly, one-dimensional fuzzy sets have a fundamental shortcoming – they do not model correlation between variables in defining the flight segments.

While one-dimensional fuzzy reasoning is still largely the state-of-the-art for fuzzy systems, correlation between input variables of a fuzzy system can lead to complications. By "correlation" is meant the condition that a fuzzy set describing a system state is represented by an irregular, smoothly connected region in a multivariable state space. The "footprint" of such a mode on the *x-y* plane could look something like the solid ellipse in Fig. 11. One-dimensional membership functions cannot by themselves represent such a relationship. The current practice approximates a smooth representation by composition of two or more single-variable regions. Such a composition is shown in dashed lines in Fig. 11.

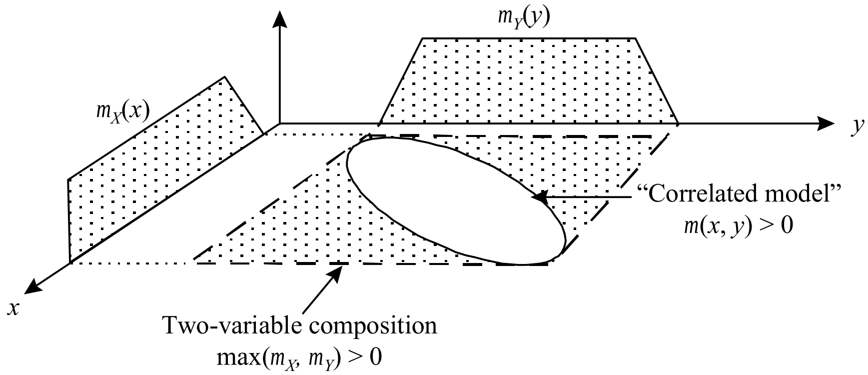


Figure 11. Footprint of fuzzy set when the input variables are correlated.

The use of fuzzy inference for flight mode interpretation has revealed that this standard fuzzy logic approach is insufficient for application in complex systems. To address some fundamental shortcomings in the current state-of-the-art, the authors developed the *hypertrapezoidal fuzzy membership function* (HFMF).

B. Bayesian Isomorphism

Before introducing a method for specifying multi-dimensional fuzzy sets, it is useful to consider a somewhat theoretical question of the relationship between fuzzy set theory and Bayesian decision theory. There is an isomorphic relationship between the two. That relationship has been shown in Refs. **Error! Reference source not found.** and 13. The parallel between the two approaches can be developed by assuming certain constraints on the design of the fuzzy sets and on the logical connectives used to operate on fuzzy sets.

The first design constraint is that the fuzzy sets must conform to Eq. 1. Equation 1 states that for all points x in the state space, the degrees of membership $m_i(x)$ in all the fuzzy sets sum to exactly one. This also implies that there is full coverage of the state space.

$$\sum_i m_i(x) = 1 \quad \forall x \tag{1}$$

The other requirement is that the fuzzy logic connectives, used to perform logical operations on fuzzy sets, must be based on *multiplication* (for fuzzy AND) and *addition* (for fuzzy OR). These are in contrast to the more widely used *min* function (for fuzzy AND) and *max* function (for fuzzy OR). The former was first proposed by Bellman and Zadeh[12] as the "soft connectives" in addition to the "hard connectives" of *min* and *max*. The soft connectives have the advantage of being mathematically consistent with Bayesian decision theory. This is important because it is the basis for the next section on designing multi-dimensional fuzzy sets.

C. Hypertrapezoidal Fuzzy Sets

To overcome the shortcomings of the one-dimensional state-of-the-art in fuzzy set theory, we explored the options for multidimensional fuzzy sets. Those options included fine-grained rule-base composition of multidimensional relationships, conditional fuzzy membership functions, and multi-dimensional Gaussian functions. For various reasons all these options are still inadequate for the engineering problem of flight segment identification.[13]

An important consideration in the development of N-dimensional membership functions is that they be specified with only a few parameters. The standard method for defining one-dimensional trapezoidal membership functions is with four points – *a*, *b*, *c*, and *d*, as shown in Fig. 12. This method, however, is impractical for defining membership functions on multiple dimensions. The extension of the trapezoidal membership function into a two-dimensional space would require at least eight points, as shown in Fig. 13.

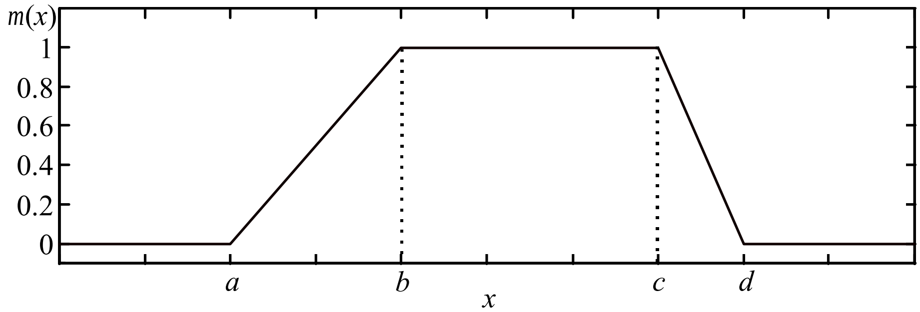


Figure 12. Defining a one-dimensional trapezoidal membership function.

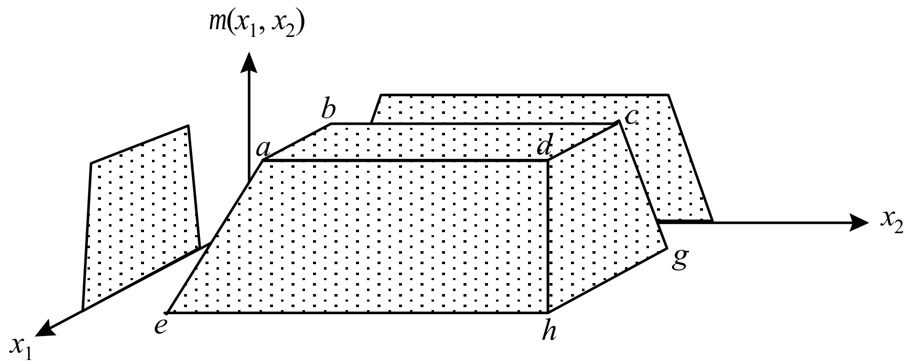


Figure 13. Defining a two-dimensional trapezoidal membership function.

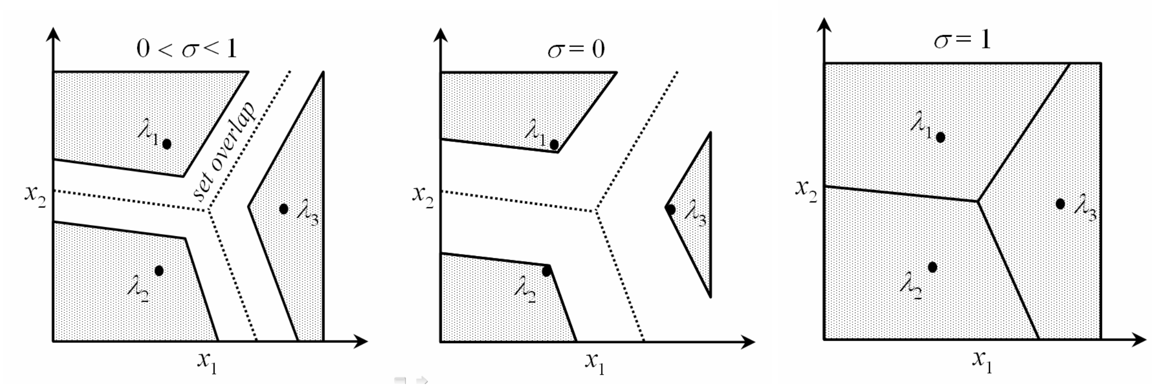
Another important consideration is that the multidimensional fuzzy sets should enforce the requirement of Eq. 1 and use the alternate fuzzy logic connectives that are isomorphic with Bayesian probabilistic reasoning mentioned in the previous section. Membership functions defined in such a manner are referred to as a *fuzzy partitioning*. Fuzzy membership functions based on Gaussian probability density functions can easily be extended to *N* dimensions. However, they do not exhibit the desirable property of Eq. 1. Trapezoidal membership functions, on the other hand, can be defined with the design constraint of Eq. 1.

Based on the requirements outlined above, we developed a new mechanism for specifying and calculating multidimensional fuzzy membership functions.[13-15] Termed *hypertrapezoidal fuzzy membership functions* (HFMF), this new development is a major advancement in the practical application of fuzzy logic to engineering problems.

As an alternative to trying to define all the corners of N-dimensional fuzzy sets, consider the use of a single point in the state space as the defining parameter of an N-dimensional fuzzy set. Each fuzzy set in a fuzzy partitioning would then have an associated N-dimensional vector which is a typical value for that set. We chose to call such an N-dimensional vector the *prototype point*. The prototype point, l_i, for a fuzzy set, S_i, with a membership function, $m_i(x)$, satisfies the following equations.

$$\begin{aligned} m_i(l_i) &= 1 \\ m_j(l_i) &= 0 \quad j \neq i \end{aligned} \tag{2}$$

Figure 14a shows a simple example of a fuzzy partitioning in two dimensions using three prototype points to define three fuzzy sets leaving some area of overlap between the sets.



(a) (b) (c)

Figure 14. Prototype points defining a fuzzy partitioning.

A measured value, x, which is an N-dimensional point in the state space of a fuzzy partitioning, has a degree of membership in a fuzzy set based on its Euclidean distance from the prototype point for that set. For example, if $x = l_1$, then $m_1(x) = 1$, $m_2(x) = 0$, and $m_3(x) = 0$. As another example, if x is equidistant from all three prototype points, then $m_1(x) = 0.333$, $m_2(x) = 0.333$, and $m_3(x) = 0.333$. This is the basis of hypertrapezoidal fuzzy membership functions and has proven to be quite useful in inferring operational flight segments of an aircraft.

One additional parameter is needed for defining an N-dimensional fuzzy partitioning. The *crispness factor* determines how much overlap exists between the sets of two adjacent prototype points. We chose to define the range of the crispness factor, s, to be [0, 1]. For $s = 1$, no overlap exists between the sets, and the partitioning reduces to a minimum distance classifier. Figures 14b and 14c show the resulting partitions of the above example for the two extremes $s = 0$ and $s = 1$.

Given a sensor measurement, x, the HFMFs can now be calculated using standard trigonometry. First, a distance measure, $r_{i|j}$, is calculated for each pair of prototype points, as shown Eq. 3. Here, $d(x,y)$ is the Euclidean distance between x and y.

$$r_{i|j}(x) = \frac{d^2(x, l_i) - d^2(x, l_j)}{d^2(l_i, l_j)} \tag{3}$$

Then the pair-wise membership functions are calculated for each pair of prototype points, as shown in Eq. 4. Here, $\vec{v}_{ji}$ is a vector from l_j to l_i, $\vec{v}_{jx}$ is a vector from l_j to x, and $\vec{v}_{ji} \cdot \vec{v}_{jx}$ is the dot product of the two vectors.

$$m_{i|j}(x) = \begin{cases} 0; & r_{i|j}(x) \geq 1 - s \\ 1; & r_{i|j}(x) \leq s - 1 \\ \dfrac{\vec{v}_{ji} \cdot \vec{v}_{jx} - \frac{s}{2} \cdot d^2(l_j, l_i)}{(1-s) \cdot d^2(l_j, l_i)}; & \text{otherwise} \end{cases} \tag{4}$$

Finally, the degree of membership, $m_i(x)$, of measured input, x, can be determined based on product inference as shown in Eq. 5. Here M is the number of fuzzy sets in the partition.

$$m_i(x) = \frac{\prod_{j=1 \neq i}^{M} m_{i|j}(x)}{\sum_{k=1}^{M} \left(\prod_{j=1 \neq k}^{M} m_{k|j}(x) \right)} \tag{5}$$

Notice that Eqs. 3, 4, and 5 are general for N dimensions, including N=1. These three equations allow for the design of N-dimensional membership functions using only N+1 parameters. Additionally, the desirable property of Eq. 1 is enforced.

Figure 15 shows an example of the fuzzy sets that were defined for light twin aircraft using hypertrapezoidal fuzzy membership functions. In Fig. 15 all but two variables (rate of climb and indicated airspeed) are fixed so that a 3D plot could be drawn using just those two variables. As can be seen, the relationships between variables in fuzzy sets can be defined more richly than they could be defined with one-dimensional sets. Correlation is modeled and with the small number of parameters, the technique scales to many dimensions.

Using HFMF models of flight segment, we were able to accurately infer flight segments directly from aircraft state variables. Figure 16 compares the inferred flight segments to the flight segment that the pilot stated he was in during the data collection process.

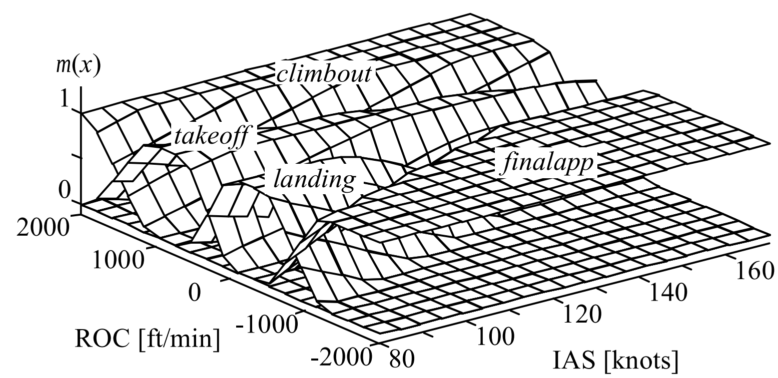


Figure 15. Plots of four fuzzy sets for an altitude of 500 feet above ground.

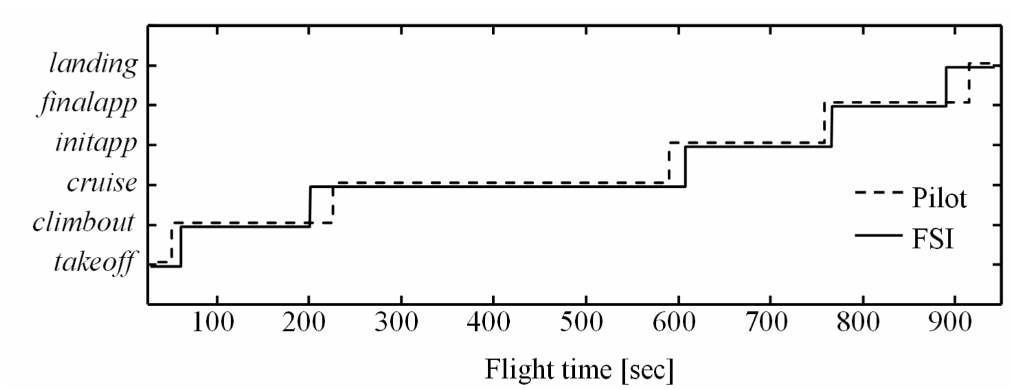


Figure 16. Experimental results for Hypertrapezoidal Flight Segment Identifier.

V. Conclusion and Future Work

Over a decade of research has shown the value of explicitly modeling flight segments and evaluating those models in real-time. Flight segment identification (FSI) is an enabler for context-aware pilot advising, procedure guidance, and automated Highways-In-The-Sky. We explain the difference between State-based FSI, which identifies the flight segment that the pilot is currently flying, and Procedural FSI, which identifies the flight segment that the pilot should be flying. Pilot advisories and intent-based conflict detection were both successfully implemented and flight tested on the NASA SATS project. Our SATS demonstration showed that flight segment identification enables “smarter” avionics which can support new, efficient flight procedures for the NAS.

There are several areas that deserve additional research and development. The “automated path guidance” feature, in which the P-FSI result commands the HITS deserves more attention. Our subjective observation is that this feature will become a “must-have” for synthetic vision highways. The “procedure guidance” display should also be studied some more. Does such a display improve a pilot’s execution of new procedures? Finally, there is the need to automate the design of FSI models for specific aircraft and procedures. Currently, building the models is a time-consuming, trial-and-error process. Commercialization will require a training algorithm that adapts the basic models to specific aircraft models and new procedures.

Acknowledgments

The authors would like to thank the NASA SATS project leadership (Sally Johnson, Sally Viken, and Brian Baxley) for the opportunity to apply flight segment identification technology to the High Volume Operations concept. We are also grateful to Dr. John Valasek and Jie Rong of Texas A&M University for their fantastic support in transitioning ASTRA to the HVO application. The NC&UGP SATSLab leadership at RTI International (Lou Williams, Michael Heck, Dennis Wilt, Jim Henion, and Randall Davis) deserve special recognition for managing the diverse team and making the flight tests such a success. Keith Alter and Andrew Barrows of Nav3D Corporation went “beyond the call of duty” in helping us integrate the FSI with their synthetic vision libraries. Finally, it is hard to imagine how everything would have fallen together without the engineering excellence of Paul Snow. Thank you to all.

References

[1] Anonymous, “Personal Air Transportation Technologies for Flight Demonstration”, Small Business Innovation Research & Small Business Technology Transfer Program Solicitation, 2005.

[2] V. A. Carreño, Concept for Multiple Operations at Non-Tower Non-Radar Airports During Instrument Meteorological Conditions, Proceedings of the 22nd Digital Aviation Systems Conference, DASC 2003, Indianapolis, Indiana, 12-16 October 2003.

[3] J. H. Painter, “Knowledge-based control,” in Proceedings of the 1992 IEEE Symposium on Computer Aided Control System Design, Napa, CA, March 1992, pp. 138-147.

[4] T. H. Nguyen and D. T. Ward, “A neural-network based inference engine for a general aviation pilot advisor,” Paper #97-0221 of the AIAA 35th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1997.

[5] J. Painter, W. Kelly, J. Trang, K. Lee, P. Branham, J. Crump, D. Ward, K. Krishnamurthy, D. Woo, W. Alcorn, and R. Yu, “Decision Support for the General Aviation Pilot”, Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, October 12-15, 1997, pp. 88-93.

[6] J. Rong and J. Valasek, “Onboard Pilot Decision Aid For High Volume Operations In Self-Controlled Airspace”, Proceeding of the 23rd International Digital Avionics Systems Conference, 2004.

[7] C. Adams, M. Consiglio, K. Jones, D. Williams, T. Abbott, “Development of the SATS HVO Operational Concept: Nominal Operations,” NASA Langley Research Center, June 29, 2003.

[8] G. Dowek, C. A. Muñoz, and V. A. Carreño, “Abstract Model of the SATS Concept of Operations: Initial Results and Recommendations,” NASA/TM-2004-213006, March 2004.

[9] C. Adams, V. Carreno, M. Consiglio, C. Munoz, D. Williams, “SATS HVO Conflict Detection and Alerting Functional Requirements,” NASA Langley Reseach Center, December 4, 2003.

[10] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338 - 353, June 1965.

[11] V. Q. Harral, “Genetic algorithms for tuning fuzzy membership functions in flight control software,” M.S. Thesis, Department of Electrical Engineering, Texas A&M University, College Station, TX, December 1995.

[12] R. E. Bellman and L. A. Zadeh, “Decision-making in a fuzzy environment,” *Management Science*, vol. 17, no. 4, pp. 141-164, December 1970.

[13] W. Kelly, Dimensionality in Fuzzy Systems, Ph.D. Dissertation, Texas A&M University, July 1997.

[14] W. Kelly and J. Painter, “Soft Computing in the General Aviation Cockpit”, The First On-line Workshop on Soft Computing, On the Internet, served by Nagoya University, Japan, August 19-30, 1996, pp. 151-156.

[15] W. Kelly and J. Painter, “Hypertrapezoidal Fuzzy Membership Functions”, 5th IEEE International Conference on Fuzzy Systems, New Orleans, September 8, 1996, pp. 1279-1284.