

TR 99175.30-02

**INTENT INFERENCE,
CONFIDENCE ASSESSMENT, AND
HAZARD PRIORITIZATION
STATUS REPORT**

Jimmy Krozel, Ph.D.,
Tysen Mueller, and Dave Schleicher

Prepared for:

NASA Ames Research Center
Moffett Field, CA 94035

Under NASA Contract NAS2-98001

March, 2000

Submitted by:



SEAGULL TECHNOLOGY, INC.
16400 Lark Ave., Suite 100
Los Gatos, CA 95032-2547

ACKNOWLEDGMENTS

This research was funded by NASA Langley Research Center under Research Task Order 30 (RTO 30) contract NAS2-98001 for NASA's Advanced Air Transportation Technologies (AATT) program. The authors appreciate the program support of Mark Ballin at NASA Langley Research Center and the additional guidance and support of Richard Barhydt, Sheila Conway, Monica Hughes, Mike Palmer, Dave Williams, and David Wing at NASA Langley Research Center.

ABOUT THE AUTHORS



Jimmy Krozel, Manager, Interdisciplinary Research Engineering (Ph.D., Aeronautical and Astronautical Engineering, Purdue University, 1992). Since joining Seagull Technology in Aug., 1995, Dr. Krozel has worked on aircraft conflict detection and resolution for Free Flight, the database management of aircraft, terrain, and navigation aid data, and aircraft terminal area routing around heavy weather. His research interests are in the areas of advanced visualization systems, air traffic control, Free Flight, route planning, terrain reasoning, autonomous vehicles, and intelligent path prediction. Dr. Krozel performed his M.S. thesis research under a fellowship at the NASA Ames Research Center, and performed his Ph.D. thesis as a Hughes Doctoral Fellow at the Hughes Research Laboratory. Dr. Krozel is a member of the American Institute of Aeronautics and Astronautics, American Association for Artificial Intelligence, Tau Beta Pi, Sigma Xi, and Sigma Gamma Rho. Dr. Krozel has a total of 17 technical publications. Dr. Krozel is on the editorial board of *Applied Intelligence*, Kluwer Academic Publishers.



K. Tysen Mueller, Senior Research Engineer (MA, Physics, California State University, Long Beach, 1971). Tysen Mueller has more than 30 years experience as a systems engineer in the areas of guidance, navigation, and computer simulation, including Kalman filtering. He has 10 years of experience as a product and program manager in GPS/Inertial (GPS/INS) and Differential GPS (DGPS) systems. This experience includes applications to the design, development, and evaluation of Automatic Vehicle Location Systems (AVLS) and Wide Area DGPS (WADGPS) networks. At Seagull, Mr. Mueller has developed GPS-based systems for precision farming, consisting of a GPS clock correction code, and Ionospheric delay correction code. He has provided system engineering consulting support to Hughes for the development of the FAA's Wide Area Augmentation System (WAAS) and for the Japanese MTSAT Augmentation System (MSAS). He is the co-author of a patent on WADGPS network technologies, and is a member of the Institute of Navigation (ION).



David Schleicher, Manager, Aviation Systems Modeling (MS, Aeronautics and Astronautics, Stanford University, 1989). David Schleicher specializes in the systems analysis of aircraft and air traffic management systems with an emphasis on economic and performance analysis. His analysis activities have included the investigation of safety, economic, and acoustic effects of new air traffic management procedures and the economic and performance impacts of advanced aircraft designs, especially high-speed rotorcraft. Mr. Schleicher is a member of the American Institute of Aeronautics and Astronautics, the Air Traffic Control Association, Phi Beta Kappa, Sigma Xi Research Society, and Tau Beta Pi. He was the recipient of a Master's Degree Scholarship (1988-89) for his studies at Stanford. Mr. Schleicher is also a private pilot.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. INTENT INFERENCE	3
2.1 Theory.....	6
2.2 A Tracking Filter to Facilitate Better Near Term Trajectory Estimates.....	20
2.2 Scenarios.....	20
3. CONFIDENCE ASSESSMENT	27
3.1 Input Data.....	28
3.2 Theory.....	36
3.3 Connection with Intent Inference and Confidence Level.....	50
3.4 Scenarios.....	61
4. HAZARD PRIORITIZATION	77
4.1 Theory.....	78
4.2 Scenarios.....	89
5. CONCLUSIONS AND RECOMMENDATIONS	91
5.1 Conclusions.....	91
5.2 Recommendations: Process to Complete Development of RTO 30 Modules.....	91
6. REFERENCES	95
7. APPENDICES	99

LIST OF ACRONYMS

AATT.....	Advanced Air Transportation Technologies
ADS-B.....	Automatic Dependent Surveillance – Broadcast
AIRMET.....	Airman’s Meteorological Information
ATC.....	Air Traffic Control
AOP.....	Autonomous Operations Planner
ATM.....	Air Traffic Management
CPA.....	Closest Point of Approach
CPDLC.....	Controller Pilot Data Link Communications
DAG TM.....	Distributed Air/Ground Traffic Management
dBZ.....	Decibel
DST.....	Decision Support Tool
ETA.....	Estimated Time of Arrival for an aircraft
FIS.....	Flight Information Service
FFSim.....	Free Flight Simulation Infrastructure
FMS.....	Flight Management System
ft.....	feet
FTE.....	Flight Technical Error
GPS.....	Global Positioning System
ID.....	Identification
IFR.....	Instrument Flight Rules
LTI.....	Long Term Intent
MASPS.....	Minimum Aviation System Performance Standards
METAR.....	Meteorological Aerodrome Report
NAS.....	National Airspace System
NASA.....	National Aeronautics and Space Administration
NEXRAD.....	Next Generation Radar
NMAC.....	Near Mid-Air Collision
nmi.....	Nautical mile
NTI.....	Near Term Intent
NWS.....	National Weather Service
PAZ.....	Protected Airspace Zone
PIREP.....	Pilot Report
rms.....	root mean square
RNP.....	Required Navigation Performance
RTA.....	Required Time of Arrival
RTCA.....	Radio Technical Commission for Aeronautics
RTO.....	Research Task Order
SA.....	Selective Availability (for GPS)
SID.....	Standard Instrument Departure
SIGMET.....	Significant Meteorological Information

LIST OF ACRONYMS

(Continued)

STAR.....	Standard Terminal Arrival Route
SUA.....	Special Use Airspace

TBD.....To Be Determined
TCP.....Trajectory Change Point
TIS.....Traffic Information Service
VFR.....Visual Flight Rules

1.0 INTRODUCTION

NASA's real-time Free Flight Simulation (FFSim) is designed to be a test bed for investigating Free Flight technical, operational, and policy concepts and their associated procedures and pilot and controller Decision Support Tools (DSTs). The Autonomous Operations Planner (AOP) and an advanced Flight Management System (FMS) are two major components that will be evaluated with FFSim. Three (among many) components of AOP are the intent inference, information confidence assessment, and hazard prioritization modules. These three modules and a couple of other supporting data management modules are developed in this Research Task Order (RTO) 30 effort. Figure 1.1 illustrates the high-level design of RTO 30 developed functions, and their interaction with other AOP and FFSim modules.

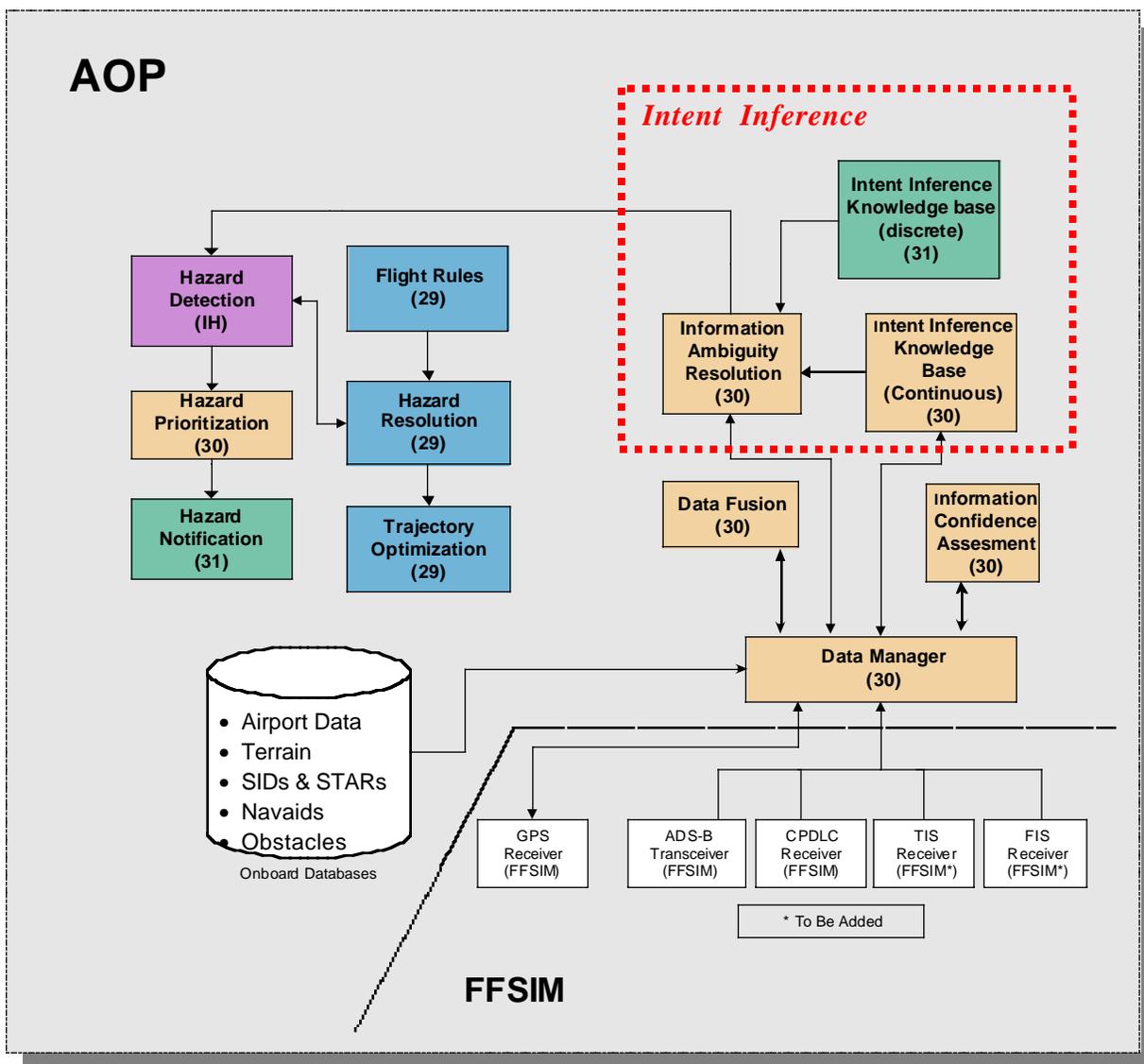


Figure 1.1. High-level AOP functions, RTO 30 developed functions, and their interaction with an Advanced FMS.

Intent inference is implemented to infer future paths and states of nearby aircraft; this type of intelligent information processing is a valuable input to conflict detection and resolution algorithms, situation awareness modules, information ambiguity resolution, and fundamental to each of these, trajectory estimation. Information confidence assessment is necessary for each AOP component to interpret the validity of the data from that component and not to over-react to data that is uncertain. Hazard prioritization is needed to rank order the important hazards (e.g., flying into terrain is worse than flying into weather) and to facilitate the process of notifying the aircraft crew about potential problems.

This report is organized as follows:

- Task 1) Intent Inference is covered in Chapter 2,
- Task 2) Confidence Assessment is covered in Chapter 3, and
- Task 3) Hazard Prioritization is covered in Chapter 4.

Chapter 5 presents our conclusions and recommendations. Chapter 6 presents references. Chapter 7 presents the following:

- Appendix A: Functional Design Requirements for RTO 30,
- Appendix B: Algorithms to convert grid weather data to triangulated data, and
- Appendix C: Derivations of CPA and Time-To-CPA for Hazard Detection.

2.0 INTENT INFERENCE

This research effort investigates the theory and implementation of an intent inferencing algorithm applicable to Free Flight [RTCA95]. As new Free Flight procedures remove jetway routing, positive control, and other constraints, an added emphasis will be placed on distributed control techniques. In the AATT program at NASA, such a system is being researched by the Distributed Air/Ground Traffic Management (DAG TM) Team [DAG99]. In Free Flight, Automatic Dependent Surveillance – Broadcast (ADS-B) or some other communications mechanism will be used to communicate state and intent data between aircraft in a distributed control system. In such a system, state and intent information (e.g., the position and speed for state information and for intent information, the next one or two waypoints or even the current flight plan) will be exchanged between aircraft. The Autonomous Operations Planner (AOP) being researched at NASA Langley demonstrates such a flight deck. There is a need for intent inference in the AOP system since in a Free Flight environment there is no guarantee that, even if flight plans or intent are broadcast to nearby aircraft, these plans will be followed. Even with acceptable Required Navigation Performance (RNP) levels, there still is the possibility that the crew of an aircraft might have changed their plans without inputs into the Flight Management System (FMS) or ADS-B broadcast, the equipment for ADS-B broadcast might fail, or deviations from a flight plan might be excessive due to strong winds, weather avoidance, or emergency diversions.

Figure 2.1 illustrates the intent inference problem. In this example, an aircraft has lateral deviations due to flight technical and navigation errors while flying to a way point, severe weather is just ahead, and there is a good reason why the aircraft might be (1) flying North of the storm (because the current heading is that way), (2) flying directly to the next way point and ignoring the weather (because all the past data indicates this), or (3) flying to the South of the storm and skipping a way point (because the airline has a policy to avoid storms and arrive on schedule). In general, the problem of intent inference is to determine: What is the aircraft intent?

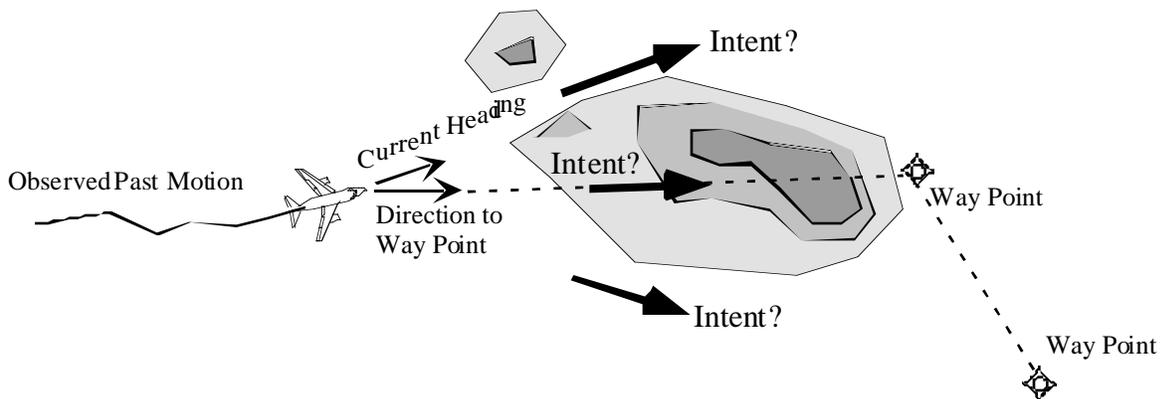


Figure 2.1. What is the aircraft intent?

In RTO 30 research, intent data is broadcast from one aircraft to another and the following cases information processing of intent information need to be performed:

1. The broadcast intent message is present, but needs to be verified true;
2. The broadcast intent message is present, is verified false (or erroneous), and an intent must be inferred from all available information;

3. The broadcast intent message is missing (due to data dropout or an unequipped aircraft), and the intent needs to be inferred from all available information.

In each of these three cases, the intent inference algorithm designed in this research effort will result in the intent of the aircraft and a level of confidence in the intent inference conclusion.

By matching observed state changes or pilot actions to a decision making model residing in an intent inference knowledge database, as we show in this Chapter, one can infer intent. For the aircraft situation in AOP, nearby aircraft motion (e.g., following the next one or two waypoints, a filed flight plan, Standard Instrument Departure (SID), etc) often follows a set of well defined and ordered actions. In the literature, Zhao, et al [ZHH98] specify several classifications of such pilot intent models, including: motive intent, objective intent, trajectory intent, and cost intent. These actions are modeled in the intent inference knowledge database using several engineering formats, and the knowledge base can be searched for an inferred intent with one of several correlation functions. We present an approach to solving this problem which combines a tracking filter (Kalman filter or a low pass filter) to estimate the state information and to predict short term trajectories, and intent models to predict long term paths.

Our approach to solving the intent inference problem has unique benefits and does not parallel any other approach in the literature. Other intent inference methodology include the following:

- Operator Plan Analysis Logic (OPAL) [G94, HG89, RGC88], where plan-goal graphs, as shown in Figure 2.2, are used to explain the causal relationship between the observed discrete actions and the goals or intent of the operator.

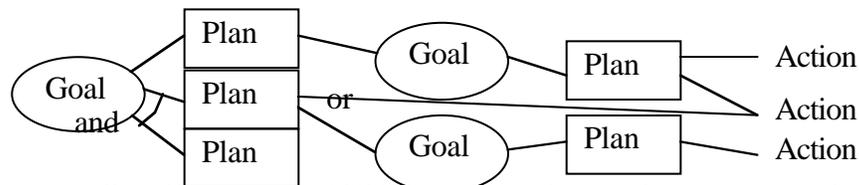


Figure 2.2. An Plan-Goal Graph (PGG) is an acyclic graph connecting plans and goals, including “and” and “or” connections (as shown).

- Operator Function Model expert system (OFMspert) [BM93,CMJ95,RJM88], where an expert system uses a blackboard architecture and operator function models, a heterarchic-hierarchical network of nodes that represent operator control functions, to replicate and infer the explanation of the actions of the human operator.
- Generalized plan recognition [KaA86, SSG78], where a deductive inference system performs intent inference by recognizing the plan the operator is implementing, based on observations of the operator, an action taxonomy (an exhaustive set of sequences of discrete admissible actions), and constraints.
- Event tracking using Soar [TR94, TR95, TR96], where the Soar knowledge based system is used to reason about the decisions made in the problem solving task of the operator, triggering rules that create problem space, goals, and subgoals that lead to an explanation of the situation.

These methods are primarily used for discrete intent inference problems. These discrete inference methods are not applicable to the continuous intent inference problem of RTO 30. As shown in

Table 2.1, continuous intent inference fills in the gap between trajectory prediction and discrete intent inference. Tracking filter-based prediction, while useful (as shown in Chapter 3) for short time periods ahead of an aircraft, does not address the intent inference problem when it comes to explaining factors like weather, special use airspace, turbulence, etc., which are factors not easily incorporated into the aircraft equations of motion modeled in the tracking filter. Using a tracking filter is most applicable when *coupled* with the intent inference method as we show in this report. In summary, the benefits of our approach to intent inference are:

- provides a method of verifying broadcast intent, if a broadcast intent exists,
- by exploiting a tracking filter, accounts for missing or delayed data collection,
- by exploiting a tracking filter for trajectory prediction, provides both near-term trajectory prediction and far-term flight path prediction, and
- provides a continuous output of the best estimate of the intent of the aircraft being tracked by the AOP system.

Table 2.1. Prediction theories and the time of horizon where the theory is applicable to a general problem domain.

PREDICTION THEORY	Tracking Filter Prediction	Continuous Intent Inference	Discrete Intent Inference
APPLICATION	Trajectory Estimation and Prediction	Path Prediction	Discrete Event Changes
TIME HORIZON	Seconds – Tens of Seconds	Tens of Seconds – Minutes	Asynchronous / Event Driven

The intent inference problem can affect the actions of the ownship as well as other nearby aircraft. For example, as shown in Figure 2.3, Yang and Kuchar [YK98] illustrates the difference between an aircraft that is informed about the intent of another aircraft compared to the aircraft that is not informed. In this example, the difference is between a correct alert and a false alert for separation assurance conflict detection. If the intent is not known, then a false alert occurs due to the fact that the ownship aircraft is likely to perform conflict resolution maneuvers when it is not necessary. If the intent is known, then there is no conflict alert and both aircraft continue on their current course.

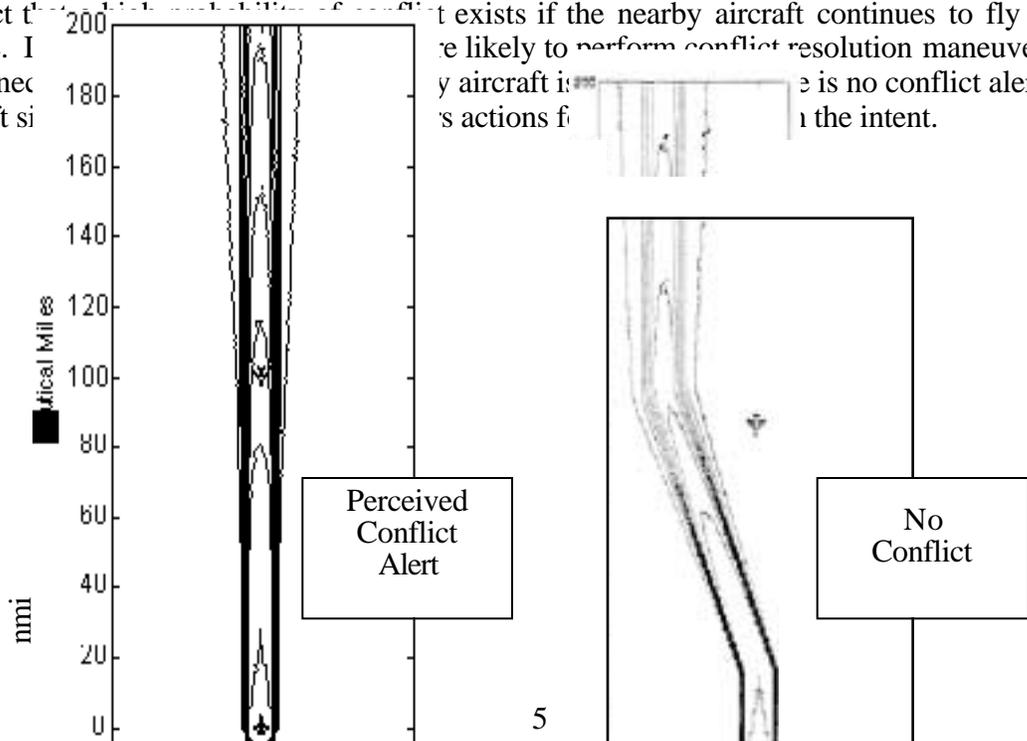


Figure 2.3. Conflict detection without knowledge of intent (left) and with knowledge of intent (right) based on the work of Yang and Kuchar [YK98] (printed with permission of Prof. J. Kuchar, M.I.T., Dept. of Aeronautics and Astronautics).

In the next sections of this report, we detail the theory behind our intent inference method, we provide some scenarios that can be used to verify our intent inference algorithms, and we present the process required to develop an intent inference module for the AOP system.

2.1 Theory

The intelligent inference algorithm is based on the theoretical foundation *Intelligent Path Prediction for Vehicular Travel* [K92, KA91, KA93, KA95]. Our solution will incorporate any broadcast or data linked intent messages from the nearby aircraft being tracked. Furthermore, when domain knowledge data are available (e.g., weather data, terrain data, SIDs, STARs, etc.), these data will be taken into consideration by the intent inference module. The intent inference module will be able to verify that a tracked aircraft is following the intent that is broadcast, and, if the aircraft is seemingly following some other intent, then the module will identify the most plausible intent for the aircraft being tracked. Finally, the intent inference module will also be able to use the most plausible intent to predict the future motion of the vehicle being tracked. Next, we provide the basis behind this theory.

2.1.1 Basis of Theory

The process of human flight control can be analyzed through two classical theories: control theory and psychology. From control theory, flying an aircraft incorporates: stability, control, guidance, and navigation. From psychology, reasons for making decisions while flying an aircraft can be explained by understanding human decision making processes.

Based on the analysis of Stengel [S93], a model for intelligent flight control can be postulated. This model includes sensing, regulation, and decision making, as shown in Figure 2.4. The key sensory inputs to the human are associated with seeing (ownship and other aircraft, runways, etc.), feeling (forces felt by the body), and hearing (sounds correlated with speed and inner ear effects from gravity). These senses are connected to the brain to influence decision making. Decision making governs regulation with neuromuscular responses driven by learned associations between stimuli and desirable actions. Furthermore, through the decision making process the pilot plans and sets goals that affect navigation and guidance command inputs. Such planning and decision making is dependent on the pilots knowledge base of both rules of flight and past experience.

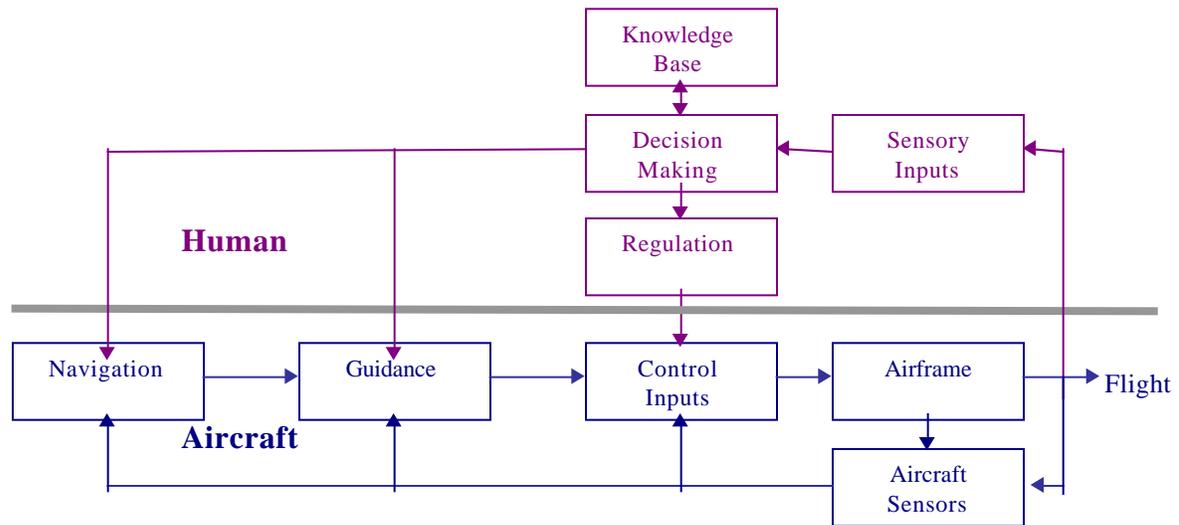


Figure 2.4. Process for human decision making while flying an aircraft.

For the aircraft model, navigation, guidance, and control functionality might be supplied by the pilot or by computer-based control systems (e.g., FMS). For the modestly equipped aircraft, a pilot will perform all guidance and navigation logic and will combine these with regulation logic to affect control inputs (minimally, the stick, throttle, and rudder). In the most advanced cockpits, navigation, guidance, and control systems are automated with potentially minimal input by the pilot. For instance, autopilots allow the pilot to command a constant heading angle or airspeed. The maneuvering precision will differ based on whether the tasks of navigation, guidance, and control logic are performed by a human pilot or by automatic control.

The human decision making model has a hierarchical structure. In the inner most loops, stability augmentation and reflexive control is performed with fast update rates, perhaps 1-10 Hz. In the outer loops, guidance and navigation changes may be made every minute to ten minutes. The central nervous system supports such a hierarchy for decision making by providing a structure that includes declarative actions, procedural actions, and reflexive actions – this is the basis of the intelligent flight control system proposed by Stengel [S93]. In the research of intent inference, we will next show that this hierarchical structure and separation between declarative actions from reflexive actions is also useful.

Next, consider the relationship between the human decision making process of flying an aircraft and the problem of predicting the intent and path of an aircraft being observed by an outside observer. These concepts are similar but different:

- **Intent Inference** is the process of determining *what* the tracked vehicle is most likely attempting to do.
- **Path Prediction** is the process of determining *how* the tracked vehicle will most likely accomplish what it is intending to do.

The path of an aircraft is directly a result of the control inputs determined by regulation and/or guidance decisions. The intent of the pilot is to use guidance and control inputs to follow a path plan or a navigation plan, e.g., a flight plan. In accordance with the aircraft flight control model set forth so far, intent inference is related to inferring the declarative and procedural decisions of the pilot, and path prediction is related to inferring the path that the pilot attains from regulatory and

reflexive control inputs. They are related through the hierarchical structure that was presented earlier.

Intent inference and path prediction are solved simultaneously in our approach. We will first discuss these individually, and then we will show how the two are combined. Intent inference may be considered to be at a higher level of abstraction than path prediction, since intent inference specifies *what* and path prediction specifies *how*. Predicting the intent of a vehicle can be abstracted into the process presented in Table 2.2. Likewise, predicting the motion of a vehicle can be abstracted into the process presented in Table 2.3.

Table 2.2. Process for intent inference.

Step	Process for Intent Inference
1	Build a knowledge database of plausible intent models for the vehicle being tracked and domain knowledge about the environment the vehicle is traveling through
2	Observe and record state information about the vehicle being tracked and nearby traffic,
3	Best fit the plausible intent models with the observed state information and determine the “correlation” of fit, and
4	Rank the intent models by “correlation” of fit to determine the most likely intent of the observed vehicle.

Table 2.3. Process for path prediction.

Step	Process for Path Prediction
1	Build a dynamics model for the vehicle being tracked, based on the inferred intent of the vehicle,
2	Observe and record state information about the vehicle being tracked,
3	Estimate states and/or parameters in the dynamics model whether observed directly or not, and
4	Project the dynamics model into the future in order to predict future motion.

Since the process for intent inference and the process for path prediction relate in a hierarchical nature, we combine these processes into a single algorithm that both predicts intent and predicts the future path of the aircraft being tracked, as shown in Table 2.4. Note that because of the hierarchical nature of decision making, the intent is likely to remain constant for a long time as inner loop commands change more frequently. If path prediction is pursued with very short look-ahead times, then the outer loop guidance and navigation commands can be assumed to be constant. However, as look-ahead times become longer, one should expect that the guidance and navigation commands may change, depending on where the pilot is in terms of following the outer loop decision logic. Thus, the model used for the path prediction must adapt to the context set forth by the outer loop decisions. From this discussion, the hierarchical structure explains why intent inference is necessary to do accurate path prediction, especially when the prediction time is long enough to include the declarative or procedural decisions of the outer loops.

Table 2.4. The combined process for intent inference and path prediction.

Step	Process for Intent Inference and Path Prediction
1	Build a knowledge database of plausible intent models for the vehicle being tracked and domain knowledge about the environment the vehicle is traveling through
2	Observe and record state information about the vehicle being tracked and nearby traffic,
3	Build a dynamics model for the vehicle being tracked, based on the plausible intent of the vehicle,
4	Estimate states and/or parameters in the dynamics model whether observed directly or not, and
5	Combine the plausible intent models with the observed state information and determine the “correlation” of fit, and
6	Rank the intent models by “correlation” of fit to determine the most likely intent of the observed vehicle. → RETURN INTENT
7	Project the dynamics model into the future in order to predict future motion. → RETURN PREDICTED PATH

2.1.2 Intent Inference Knowledge Base and Domain Knowledge

The intent inference knowledge base consists of plausible cost criteria for travel and domain knowledge. The first step in building an intent inference module is to use knowledge engineering and operations analysis to build this knowledge base.

The theory of intent inference is based on identifying a particular intent model (Step 4) from a set of cost criteria that explain intent (Step 1). For this purpose, the aircraft flight domain has a rich set of cost criteria that will reside in the intent inference knowledge base, including but not limited to the cost criteria listed in Table 2.5. A set of cost criteria is established in a database suitable for use by the intent inference algorithm. This database holds all the associated constants and parameters needed to provide the direction that an aircraft should proceed given any arbitrary set of initial conditions. While some intent models are quite simple, for instance the model to hold heading, others are very complex, for instance the model to avoid weather.

Table 2.5. Plausible cost criteria for intent inference in the aircraft Free Flight domain.

Criteria	Intent Description	Required Data and Domain Knowledge Needed to Complete Intent Model
C1	Hold Heading	Velocity Data
C2	Hold Altitude	Altitude Data
C3	Hold Speed	Velocity Data or Airspeed Data
C4	Hold Course	Position and Velocity Data
C5	Hold Coordinated Turn	Velocity Data and Bank Angle Data or Turn Rate Data

C6	Go To Waypoint (fly over)	Position and Velocity Data and TCP or Flight Plan Data or Navigation Database
C7	Skip Waypoint; Fly to Next Waypoint (fly by)	Position and Velocity Data and TCP+1 or Flight Plan Data or Navigation Database
C8	Fly Direct To (waypoint, final approach fix, metering fix, or airport location)	Position and Velocity Data and TCP or Flight Plan Data or Airport, Metering Fix, or Final Approach Fix Data
C9	Return to Flight Plan from Heading Deviation	Position and Velocity Data and Flight Plan Data
C10	Return to Flight Plan from Speed Deviation	Position and Velocity Data and Flight Plan Data
C11	Return to Flight Plan from Altitude Deviation	Altitude and Velocity or Altitude Rate Data and Flight Plan Data
C12	Return to Flight Plan from Lateral Offset (hold radial)	Position and Velocity Data and Flight Plan Data
C13	Fly a list of Waypoints (Flight Plan, SID, or STAR)	Position and Velocity Data and SIDs and STARs Database or Flight Plan Data
C14	Avoid Special Use Airspace (SUA)	Position and Velocity Data and SUA Data (location and time)
C15	Avoid Turbulence	Position and Velocity Data and Turbulence Report Data (location and time)
C16	Avoid Hazardous Weather	Position and Velocity Data and Weather Precipitation Data
C17	Avoid Aircraft	Position and Velocity Data for Ownship and Position and Velocity Data for Intruder(s) (Traffic Data) and Alert Zone of Intruder Aircraft and Protected Airspace Zone of Intruder Aircraft
C18	Avoid Terrain	Position and Velocity Data and Terrain Database
C19	Avoid Hazard Region (General)	Position and Velocity Data and Hazard Region Data (location and time)
C20	Fly Holding Pattern	Position and Velocity Data and Holding Pattern Data
C21	Fly Wind Optimized Route (to Waypoint or Airport)	Position and Velocity Data and Wind Data or Waypoint Database or Airport Database or FMS Data (describing wind optimized route)
C22	Meet Required Time of Arrival RTA (at waypoint or Final Approach Fix)	Position and Velocity Data and FMS Data (describing RTA optimized route)

Next, as indicated in Table 2.5, the intent inference knowledge base requires domain knowledge. Domain knowledge is used to build the situation assessment model, or a map of the situation, which includes the factors that might influence the pilot's decision making. The domain knowledge database may include the information in Table 2.6.

Table 2.6. Data categories and domain knowledge databases needed for intent inference.

Data	Data Description	Database or Module
D1	Position (Latitude, Longitude, Altitude)	State Data Estimates from Data Fusion Module
D2	Velocity (North, East, Vertical)	State Data Estimates from Data Fusion Module
D3	Acceleration (North, East, Vertical)	State Data Estimates from Data Fusion Module
D4	Bank Angle	State Data Estimates Module
D5	Airspeed	State Data Estimates Module
D6	TCP (Waypoint)	ADS-B Module; Navigation Aid Database
D7	TCP+1 (Waypoint)	ADS-B Module; Navigation Aid Database
D8	Navigation Equipment Performance	ADS-B Module
D9	Flight Plan	Flight Plan Data from ADS-B Module ; CPDLC Module
D10	SIDS and STARS	Navigation Aid Database
D11	Airport Location	Airport Database
D12	Alternate Airport Location	Airport Database
D13	Final Approach Fix Location	Airport Database
D14	Metering Fix Location	Airport Database; Navigation Aid Database
D15	Conflict Detection Alert Status	CD&R Module
D16	Conflict Resolution Waypoints	CD&R Module
D17	Precipitation Hazard	FIS Module
D18	Precipitation Hazard Avoidance Route	Hazard Avoidance Module
D19	Special Use Airspace (SUA)	FIS Module
D20	Special Use Airspace Avoidance Route	Hazard Avoidance Module
D21	Turbulence	Turbulence Data from FIS PIREP
D22	Turbulence Airspace Avoidance Route	Hazard Avoidance Module
D23	Terrain Data	Terrain Database
D24	Terrain Avoidance Route	Hazard Avoidance Module
D25	Hazard Region (General)	Hazard Region Data
D26	Hazard Region Avoidance Route	Hazard Avoidance Module
D27	Wind Field Data	FIS Module
D28	FMS Generated Route	FMS Module

Note: All data is for nearby aircraft unless otherwise specified.

The domain knowledge comes from three sources in the AOP. These sources are:

1. Inputs via the data fusion module – this module takes inputs from ADS-B, TIS, FIS, and CPDLC and provides the intent inference module with necessary data and confidence levels for these data.
2. Other modules within AOP – these modules include the GPS receiver, the CD&R module, a hazard avoidance module, and FMS Module which provide data to the intent inference module for reasoning about intent.
3. Databases onboard AOP – these databases provide static data, including airport, navigation, SUA, and terrain elevation data.

The specifications for the input modules and required databases are identified in Table 2.7.

Table 2.7. Required data inputs and databases for intent inference.

Module	Description	Data Required
M1	Data Fusion Module	<p>4. Ownship State Estimate (FMS Data): Ownship Position, Velocity, Heading</p> <p>2. ADS-B Data Nearby Aircraft Position, Velocity, Heading TCP, TCP+1, Navigation Uncertainty Category – Position, Flight Plan Emergency/Priority Status, Alternate Airport, Time-to-Go to TCP, Turn Indication</p> <p>5. TIS Data (as an alternate source of traffic data) Nearby Aircraft Relative Position, Heading, Altitude Rate</p> <p>6. FIS Data Weather Map (dBZ Grid), Polygon Hazard Region Data, Turbulence PIREP, Wind Field Data</p> <p>7. CPDLC Data Vertical Request (level), Vertical Clearance (maintain level, climb to, descend to), Crossing Constraints (cross position at level, cross position at and maintain level at speed) Route Modification (proceed to, cleared route, fly heading), Speed Changes (maintain, maintain or greater, maintain or less), SUA Status, Mayday</p>
M2	CD&R Module	<p>1. Alert Zone Status (Alert or No Alert)</p> <p>2. Heading Conflict Resolution Solution Waypoints</p> <p>3. Altitude Conflict Resolution Solution Waypoints</p> <p>4. Speed Change Conflict Resolution Waypoints</p>
M3	FMS Module	<p>1. FMS Data describing flight path as (latitude, longitude) linked list that nearby aircraft would take if performing a wind optimized route to a specified waypoint</p> <p>2. FMS Data describing flight path (as (latitude, longitude) linked list) that nearby aircraft would take if performing a RTA optimized route to a specified waypoint</p>
M4	Hazard Avoidance Module	<p>Input data may be weather data, SUA, terrain, or any general hazard region; Outputs are:</p> <p>1. Waypoints describing avoidance route to TCP location</p> <p>2. Waypoints describing avoidance route to TCP+1 location</p> <p>3. Waypoints describing avoidance route to Metering Fix Location</p>
M5	Airport Database	Airport Location, Alternate Airport Location, Final Approach Fix Location
M6	Navigation Database	Nav aids Data (latitude, longitude, altitude)
M7	SUA Database	SUA Regions description, if not provided in the CPDLC message
M8	Terrain Database	Terrain Elevation Data
M9	Obstacle Database	Obstacle Data

2.1.3 Method of Path Correlation

The actions of a decision maker may be observed and analyzed both locally (instantaneously) and globally (over a time window). For instance, a local decision is a change in state in an instantaneous or very short time horizon, such as a single decision to hold course, turn, climb, descend, etc. A global decision is a series of local decisions which occur over a longer time horizon, perhaps a sequence of decisions to solve a navigation task to capture a way point, follow a STAR, avoid weather, etc. For instance, while manually flying a flight plan, if a pilot turns to

capture a way point after drifting away from the nominal flight plan, the local decision might be to turn left or right towards the way point. If one looks at the state variables at any one instant, it might be difficult to infer that the pilot is flying to a way point, as depicted in Figure 2.5. For this discussion, we assume that the heading direction and the course are the same directions. Mathematically, by defining a unit vector $\vec{\psi}$ in the direction of the heading ψ and a unit vector $\vec{\phi}$ in the direction ϕ to the way point, then the dot product:

$$\text{Local Correlation} = \vec{\psi} \cdot \vec{\phi} \quad (2.1)$$

will indicate (locally) if the aircraft is heading towards the way point (dot product of 1) or away from the way point (dot product -1). This dot product acts as a good measure of correlation (providing a number between -1 and 1) between the intent to fly directly to the way point and the current aircraft state as indicated by the current heading.

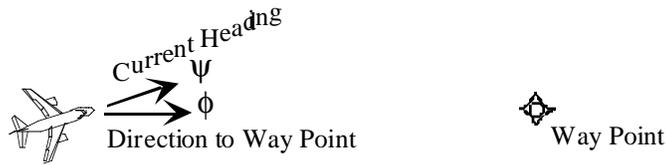


Figure 2.5. A pilot is not heading directly to a way point, but is generally heading that way.

Next, this local analysis of the intent of the aircraft can be extended to a global analysis by considering the correlation between a history of state variables with a series of decisions that would support a consistent intent. For the fly-to-next-way-point example, one would continuously integrate the local correlation measure (the dot product) over the observed flight path:

$$\text{Global Correlation} = \frac{1}{k} \int_{\text{flight path}} \vec{\psi} \cdot \vec{\phi} \, ds \quad (2.2)$$

where ds is a differential element along the flight path and k is a non-dimensionalization constant based on the cost of flight. In this research, a good value for k is simply the arc distance between the current location of the aircraft and some characteristic domain location (e.g., a waypoint, airport, or other applicable location):

$$k = \int_{\text{flight path}} ds \quad (2.3)$$

In the previous work of [K92], several global correlation functions were investigated, including fading window and moving window functions, as shown in Figure 2.6, and functions that accommodate lost data or data arriving at discontinuous update rates. Thus, this theory will address the requirement FDR*1.11 that the algorithm gracefully handle situations where the nearby aircraft data arrives intermittently or with occasional unreliable data points. The global correlation function applicable to a moving window of size l_w is described by:

* Functional Design Requirements (FDRs) are listed in Appendix A.

$$Global\ Correlation = \frac{1}{k} \int_{l_0}^l \bar{\psi} \cdot \bar{\phi} \, ds \quad (2.4)$$

where $l_0 = \max\{0, l - l_w\}$, and k is defined appropriately. In addition to a moving window, a fading memory moving window correlation function can also be used, as discussed in [K92].

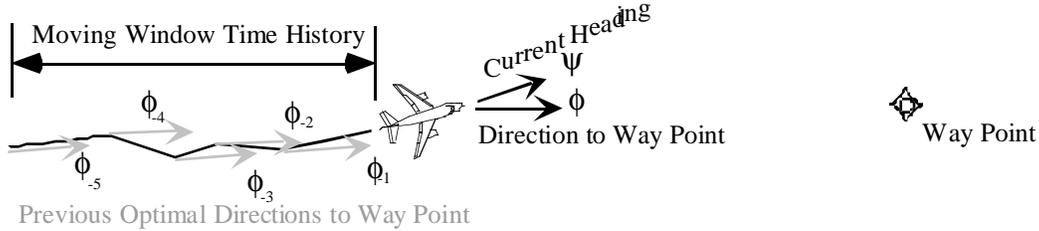


Figure 2.6. A global analysis investigates the previous headings and previous optimal directions to the way point in order to collect data to support or reject the conclusion that the aircraft is flying to the way point.

The global correlation measure can be used to validate the broadcast intent message. The global correlation measure has a range:

$$-1 \leq Global\ Correlation \leq 1 \quad (2.5)$$

If the broadcast intent is used to model the unit vectors $\bar{\phi}$, as explained in the next section, and if the correlation value is high and close to the value 1, then the broadcast intent can be considered valid. If the correlation value is zero or negative, then the broadcast intent is either invalid or is simply a poor model of intent. Engineering analysis and judgement must be exercised to determine when to decide if the global correlation value is high enough to consider an intent validated. Experimentation will be used to investigate the global correlation values (defined with a moving window) for typical circumstances of aircraft attempting to follow a constant intent but having normal guidance system and flight technical errors. After experimentation, a cutoff for identifying valid vs invalid broadcast intent messages will be determined. In any event, if the intent inference algorithm is capable of presenting an intent that better explains the motion of the aircraft in comparison to the intent that is broadcast by the aircraft, then this additional information should probably not be dismissed. More than one explanation of intent might apply at one time, for instance, flying a particular flight plan and flying to the next waypoint as described in the flight plan might both have very high global correlation values and are both quite valid intent models.

2.1.4 Decision Making Maps for Plausible Intents

A cost of travel or cost criterion C determines the optimal (or near-optimal, as discussed later in this section) decisions that would be made to solve vehicle navigation tasks given any initial condition. For instance, the minimum distance to a way point has a local cost $c=1$ where the total distance to the way point solves the mathematical problem:

$$Cost\ of\ Travel = C = \int_{flight\ path} 1 \, ds \quad (2.6)$$

However, distance is only one of many useful metrics for travel, so the cost of travel is generalized to:

$$\text{Cost of Travel} = C = \int_{\text{flight path}} c(x, y, z, t) ds \quad (2.7)$$

where the local cost function $c(x, y, z, t)$ may be a function of many domain variables (e.g., the presence of weather, turbulence, another aircraft, or Special Use Airspace, etc., as recorded in the intent inference knowledge base), which determines a positive value for c . In a specific intent model, there is only one value for c defined for every point in space (x, y, z) and for any time t . This is the most general form of the cost of travel C which forms the basis for the work performed in this RTO 30.

Mathematically, the motion of the aircraft over time determines the set of heading unit vectors $\vec{\psi}$ which point in the direction of the velocity of the vehicle. The gradient C determines the direction for the unit vector $\vec{\phi}$. A map of unit vectors $\vec{\phi}$ can be derived from the gradient C at multiple points as shown in Figures 2.7 and 2.8. In mechanization, the entire map is not needed, only the gradient information at the current and past locations of the aircraft are needed for the correlation function, eq. (2.4). In this RTO 30 work, we will specify how the direction for the unit vector $\vec{\psi}$ can be established within the intent inference module or how it can also be obtained by analyzing the results from a function call to a trajectory optimization module, such as that being built in RTO 29. Note that the “correlation” of fit (Step 4) is determined by correlating unit vectors $\vec{\phi}$, determined by the gradient C , with the observed state data. If future motion is to be predicted, the cost criterion C that best explains the motion of the aircraft can be used to generate the predicted future motion by exploiting the technique of dynamic programming. Note as stated in FDR1.12 and FDR1.13, there must be care taken for such dynamic programming to work under the computer size and speed limitations of modern avionics platforms. Thus, the dynamic programming search horizon will extend out to a limited range in front of the aircraft with a grid sizing that is limited by the avionics requirements.

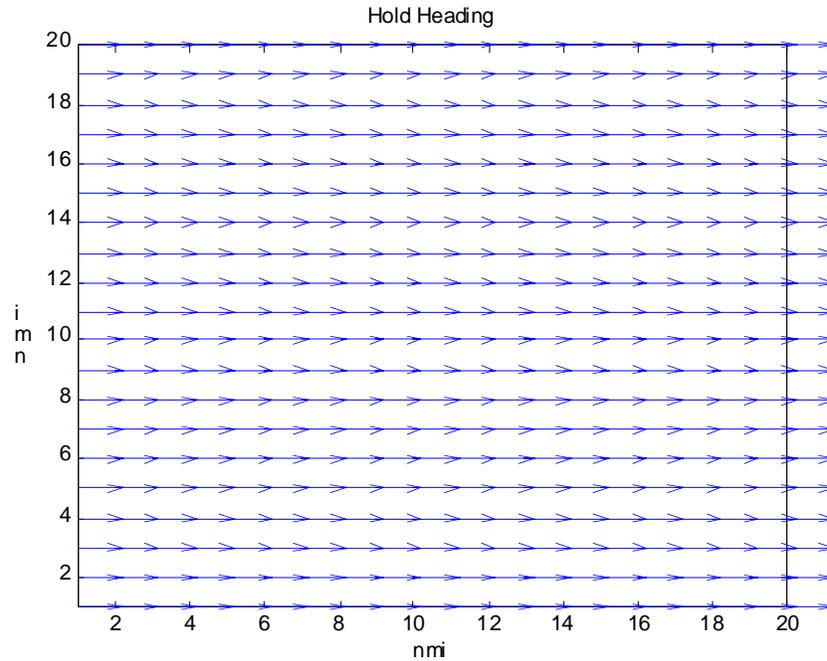


Figure 2.7. Example of the optimal decisions (headings) that should be used to fly a constant heading at any initial condition.

Hazardous airspace constraints may be modeled for Special Use Airspace (SUA may include restricted airspace, alert areas, and Military Operations Areas (MOAs)), Flow-Constrained Areas (FCAs), turbulence regions, or hazardous weather avoidance regions, such as SIGMETs. A *constraint region* C_{NOGO} is simply defined by a list of vertices that specify a polygon region:

$$C_{NOGO} = Polygon(c_1 c_2 \dots c_n) \quad (2.8)$$

where c_1, c_2, \dots, c_n locate the polygon vertices. Figure 2.9 illustrates the polygon constraint region

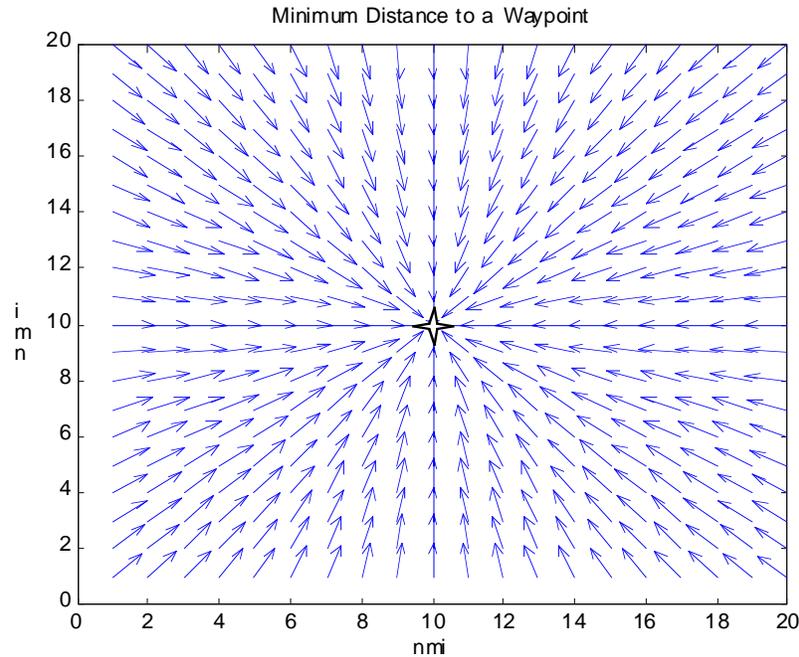


Figure 2.8. Example of the optimal decisions (headings) that should be used to fly to a waypoint using minimum distance to travel to the way point.

model for a SUA region north of LAS. SUA may be restricted at only specific times of day. Thus, SUA polygon constraint regions modeled have a list of vertices and a time period associated with them. If the constraint is inactive then the intent inference algorithm will simply ignore the SUA. With constraint regions as shown in Figure 2.10, the minimum distance route to a waypoint can be determined using the SUA as a constraint. In general, though, a hazard avoidance algorithm may have to be consulted to arrive at the set of waypoints that avoid the hazard. In such cases, for instance with weather avoidance routes, a set of weather avoidance routes is more useful than simply the optimal weather avoidance route, since the pilot may fly any of the weather avoidance routes. Figure 2.11 illustrates such a mapping of optimal directions to proceed in a weather avoidance problem.

In an iterative manner, intent models are coded and tested one at a time. The easiest intent models are addressed first, with more complex models later. The process pursues flying to a waypoint and holding altitude before addressing holding patterns, weather avoidance, and conflict detection and resolution intent models. In general, models are build individually and in some cases by building up the complexity of a previously built intent inference model.

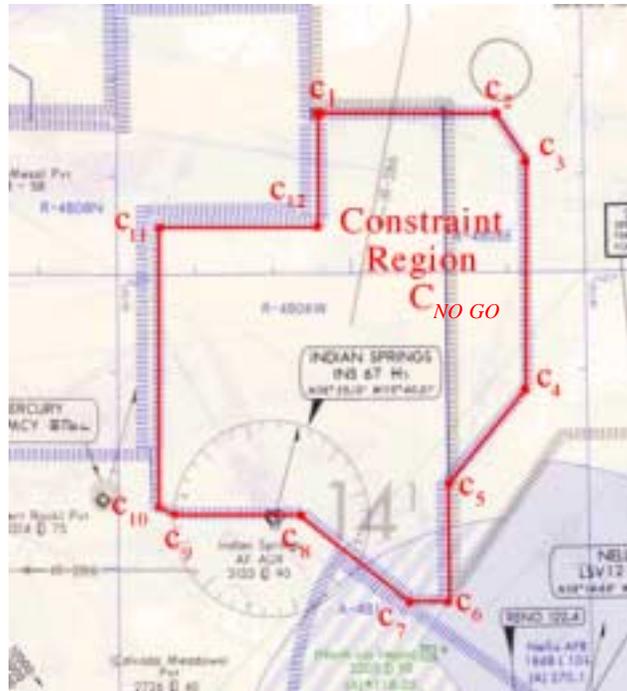


Figure 2.9. Modeling of a Special Use Airspace (SUA) region North of Las Vegas.

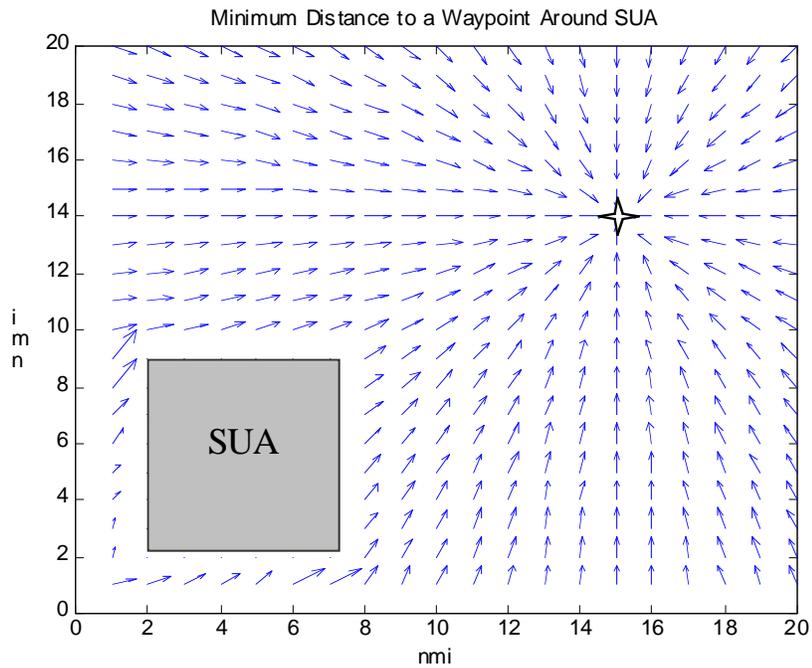


Figure 2.10. Example of the optimal decisions (headings) that should be used to arrive at the way point using minimum distance to travel to the way point subject to avoiding SUA airspace from any initial condition.

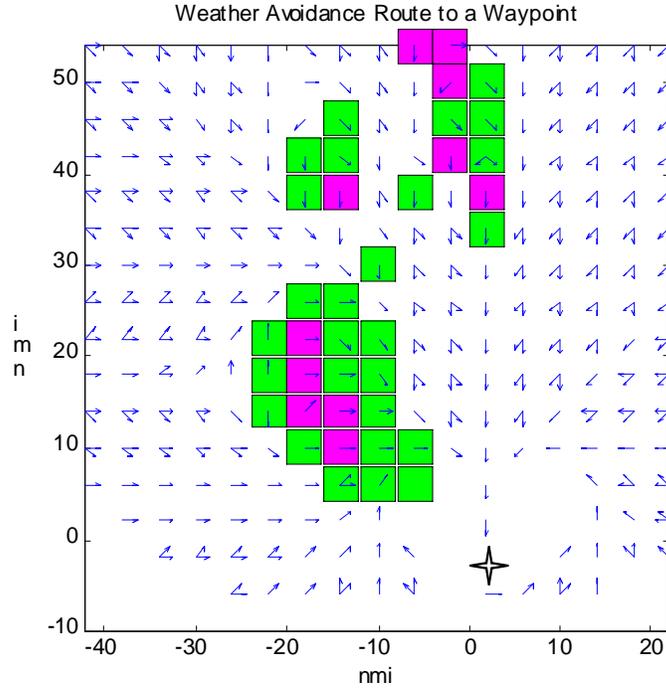


Figure 2.11. Example of the optimal decisions (headings) that should be used to arrive at the way point (with a South heading) using a weather avoidance route.

A unique feature of our method is that it is capable of considering multi-objective criteria and near-optimal criteria for travel. For example, sometimes a pilot may be avoiding weather while skipping a way point, or, a pilot might be avoiding SUA or bad weather while meeting a Required Time of Arrival (RTA). When applicable, multi-objective optimization criteria are captured in the theory by use of convex combinations of cost criteria (Step 3), modeled by:

$$C = \alpha_1 C_1 + \alpha_2 C_2 \quad (2.9)$$

where the candidate cost criterion C explains the pilot intent by combining two objectives C_1 and C_2 weighted by α_1 and α_2 . The proposed approach can best fit the parameters α_1 and α_2 to explain the past observed state history and then to model future intent. Also, the proposed approach can also be extended to model near-optimal decisions made by a pilot. As described in [KA95], a near optimal cost criteria \tilde{C} is within a small tolerance ε of being optimal:

$$\tilde{C} \leq (1 + \varepsilon)C \quad (2.10)$$

where C is the optimal. The parameter ε is determined by analyzing the state data of the nearby aircraft. For example, if an optimal decision policy C is to minimize the time to the next way point, then the near-optimal decision policy \tilde{C} might be to come within 10% (e.g., $\varepsilon=0.10$) of the optimum arrival time. Using multi-objective cost criteria and near-optimal cost criteria are two methods of expanding a simple basis set of cost criteria, so that if needed in the AOP application, a very rich set of cost criteria can always be available for explaining the intent of a pilot.

2.2. A Tracking Filter to Facilitate Better Near-Term Trajectory Estimations

The proposed solution to intent inference is coupled with a tracking filter in order to produce state estimates and to predict motion spanning near-term (e.g., seconds to minutes) to far-term (e.g., minutes to hours) prediction times. This tracking filter is a Kalman filter when the ADS-B data is available, and a Low Pass Filter (LPF) when only TIS-B data is available, as discussed in Chapter 3. A tracking filter by itself could not within reasonable accuracy predict intent beyond near-term time horizons, since the tracking filter in its general form is not capable of modeling a knowledge base as diverse as the one described for this application. However, the intent inference method described so far, while applicable to both near-term and far-term predictions, can be enhanced by a tracking filter for producing nearby aircraft state estimates, better (more accurate) near-term trajectory predictions, and a measure of uncertainty of these estimates.

The tracking filter provides a convenient sequential fading window filter. It provides the best estimate of the state variables needed for the correlation function, equation (2.4), of the intent inference method. Thus, it assists on the input side of the intent inference module by providing stable estimates of the state variables, even if, for instance, the state variables do not arrive at a continuous uninterrupted rate. For example, the tracking filter would be useful for estimating the inputs during lost transmissions of ADS-B or other input sources. Chapter 3 will go into fine detail as to how the tracking filter is used for estimating state variables and assigning confidence levels to them. While the tracking filter can assist the intent inference algorithm with stable inputs, conversely, the intent inference method proposed in this RTO can be used to adjust tracking filter parameters, such as the process noise. With these adjustments, a tracking filter can produce better near-term trajectory predictions for the nearby aircraft.

The same tracking filter that is used for confidence assessment can also be used for path prediction in the intent inference algorithm. Thus, there will be only one tracking filter central to both the confidence assessment module and the intent inference module. The confidence assessment module will use the tracking filter for producing the best estimate of the state variables, and the intent inference module will use the tracking filter to project forward in time for path predictions.

2.3 Scenarios

The engineering approach for testing the intent inference algorithm includes the use of several scenarios; these scenarios are reviewed next. First, we present a theoretical scenario to understand analytical variations in intent model correlation values with respect to some common variations in flight conditions. Then, several simulation based test scenarios are enumerated. Simulation-based test scenarios are designed to test the merits of several plausible intent models while other intent models should show no or little correlation with the data of the scenario. Real or synthetic data can be used in the test scenarios, based on the availability of data.

2.3.1. Theoretical Scenario: Investigation of Flying to a Waypoint

The geometry of this scenario is described in Figure 2.12, where an aircraft is attempting to fly along the x -axis from the origin to the waypoint located at position $x=a$. The location of the aircraft is described by (x,y) . In this analytical study, we consider flight paths that are modeled through analytical, mathematical expressions.



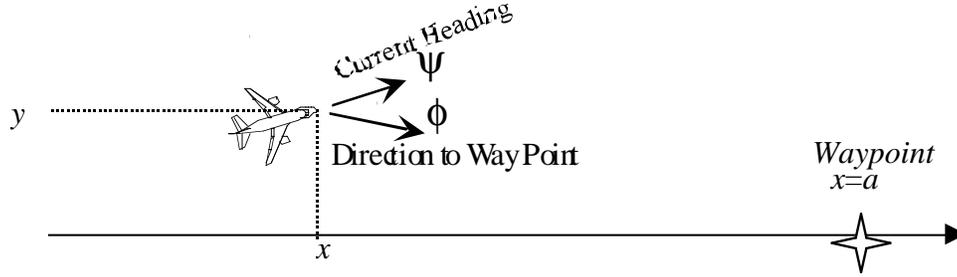


Figure 2.12. A pilot with aircraft located at (x,y) is heading to a waypoint located at $(a,0)$.

We assign the cost to travel to be a uniform unitary weight $c(x,y,t) = 1$, so that:

$$C = \int_{\text{flight path}} 1 ds, \quad (2.11)$$

where we expect the pilot is trying to fly a straight line to the waypoint, thus minimizing the cost to the waypoint.

Case I is the ideal decision making path for the perfect autopilot or pilot, where the aircraft flies perfectly along the x -axis to the waypoint. This example is the ideal Case I, described parametrically by the function:

$$\begin{aligned} x &= v_x t \\ y &= 0 \end{aligned} \quad (2.12)$$

where $v_x=1$ is the ground speed and t is time. To define the constant k for path correlation, we use the function $k=x$, which basically describes the path of the ideal path length as a function of time t . From this, we have a global correlation function:

$$\text{Global Correlation} = \frac{1}{k} \bar{\psi} \bar{\phi} ds = \frac{1}{x} 1 ds = 1 \quad (2.13)$$

where $\bar{\psi} \bar{\phi} = 1$ for all time t since the aircraft flies in the direction of the optimal path for all time t . This Case I provides the baseline for other flight paths to be compared.

Case II and III consider a pilot who drifts off of the x -axis in a motion that is described by the analytic function:

$$\begin{aligned} x &= v_x t \\ y &= A \sin\left(\frac{n\pi t}{a}\right) \end{aligned} \quad (2.14)$$

where $v_x=1$ is the ground speed, t is time parameter, n is a variable that describes the sinusoidal drift with amplitude A of a fictitious flight path described by (x,y) , and a describes the location of the waypoint as shown in Figure 2.12. For the flight path of Case II, let $n=1$, and let A vary, and for the flight path of Case III let n vary and let A be fixed. Use $k=x$ as defined from the ideal flight path, so that:

$$\text{Global Correlation} = \frac{1}{x} \bar{\psi} \bar{\phi} dt \quad (2.15)$$

where $\bar{\psi}$ (not normalized here) is defined by the heading direction (the derivative of the function in equation (2.14)), that is,

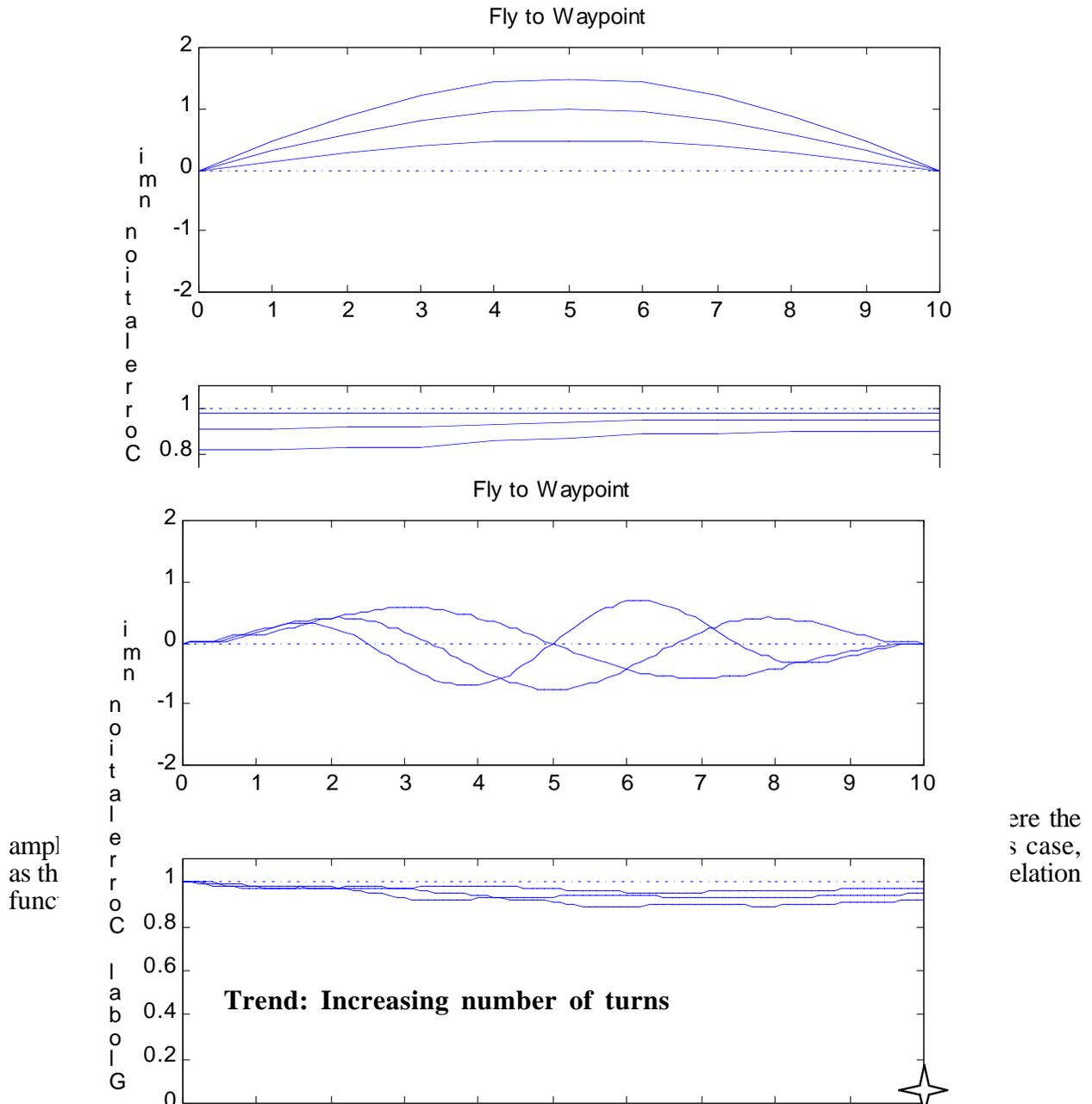
$$\bar{\psi} = 1 - \frac{n\pi A}{a} \cos\left(\frac{n\pi t}{a}\right), \quad (2.16)$$

and $\bar{\phi}$ is described by the direction from the point (x,y) to the point $(a,0)$, that is,

$$\bar{\phi} = \frac{a-x}{\sqrt{(a-x)^2 + y^2}} - \frac{y}{\sqrt{(a-x)^2 + y^2}}. \quad (2.17)$$

Figure 2.13 illustrates the comparison between the baseline Case I and Case II where the amplitude deviation away from the optimal route increases. In this case, as the deviation away from the baseline increases, the correlation function decreases.

Figure 2.13. If a pilot is heading to a waypoint located at $(a,0)$ and the flight path



local deviation from the baseline (as shown in the top plot) causes a decrease in the correlation value (as shown in the bottom plot).

In general, as shown in [K92], local path variations around a baseline path cause a decrease in the correlation function. As would be expected, all of the resultant correlation values are relatively close to yet less than the baseline correlation value. In the next set of test scenarios, the scenarios are set up to compare competing global intent models. In a similar way to what has been shown here for the “fly to waypoint” example, local variations in aircraft motion will act to reduce correlation values, but in general, all correlation values will be reduced in a similar manner – an observation based on experience with this theory.

2.3.2. Test Scenarios

The following scenarios are designed to test competing models of intent. In Scenarios 1-5 the following information is the same for each case:

- Tracking of a single nearby aircraft by the ownship,
- Neighboring aircraft is broadcasting position, velocity, and intent including next two waypoints with no lost data in the transmission of intent,
- No separation conflicts or separation alerts with respect to the ownship, and
- No weather or turbulence present.

Later, in Scenarios 6-12, further conditions are varied.

Scenario 1

Key Feature:

- Nearby aircraft flies waypoints within RNP limits

Expected Outcome:

- High ranking by correlation function for: Broadcast intent verified, Flying to Waypoint, Holding Altitude
- Other intent models should have low values

Scenario 2

Features:

- Aircraft breaks RNP limits early and flies directly to TCP+1

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Skipping Waypoint (to TCP+1), Holding Altitude
- Other intent models should have low values

Scenario 3

Features:

- Aircraft breaks RNP limits and flies to a Navaid unspecified in ADS-B but in the future waypoints contained in the flight plan

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Flying to Waypoint (all waypoints in flight plan tested and best fit returned), Holding Altitude
- Other intent models should have low values

Scenario 4

Features:

- Aircraft breaks RNP limits and flies to a Navaid unspecified in ADS-B message nor flight plan

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Flying to Waypoint (all Navaid within 50 miles are tested for best fit), Holding Altitude
- Other intent models should have low values

Scenario 5

Features:

- Aircraft outside RNP limits yet flying towards flight plan

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Capturing Flight Plan (from heading deviation), Holding Altitude
- Other intent models should have low values

In scenarios 6-10, the following conditions are held constant:

- Tracking of a single nearby aircraft by the ownship
- No separation conflicts or separation alerts with respect to the ownship, and
- No weather or turbulence present.

Scenario 6

Features:

- Tracked aircraft is within ADS-B range limit but no intent information (TCP or TCP+1) is within the ADS-B message

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Fly to Waypoint (all NavAids within 50 miles are tested for best fit), Holding Altitude
- Other intent models should have low values

Scenario 7

Features:

- Tracked aircraft is within ADS-B range limit but entire ADS-B message is missing or intermittent

Expected Outcome:

- Kalman filter should provide best estimate of nearby aircraft location and velocity and provide these data to intent inference algorithm
- Medium ranking (due to uncertainties in estimation) by correlation function for: Broadcast intent not verified, Fly to Waypoint (all NavAids within 50 miles are tested for best fit), Holding Altitude
- Other intent models should have low values

Scenario 8

Features:

- Tracked aircraft is broadcasting in ADS-B message a flight plan or (TCP and TCP+1) that violate a recently issued SUA activation

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Avoid SUA, Fly to Waypoint (corner location of SUA), Holding Altitude
- Other intent models should have low values

In scenarios 9-12, the conditions are fully specified on a Scenario by Scenario basis.

Scenario 9

Features:

- Ownship tracking nearby aircraft
- Neighboring aircraft is broadcasting position, velocity, and intent including next two waypoints with no lost data in the transmission of intent
- Separation conflict between ownship and nearby aircraft determined by ownship CD&R subsystem using state information about ownship and nearby aircraft
- Nearby aircraft broadcasts a conflict resolution set of waypoints; Nearby aircraft changes state vectors to indicate resolution activity
- No weather or turbulence present

Expected Outcome:

- High ranking by correlation function for: Broadcast intent verified (based on CD&R solution and TCP and TCP+1), Avoiding Collision (CD&R algorithm provides alternative frontside, backside, topside, and bottomside maneuver waypoints – best match found from this set), Holding Altitude (if CD&R maneuver is horizontal maneuver) or else Holding Heading (if CD&R maneuver is vertical maneuver)
- Other intent models may have high values given the broadcast of new TCP and TCP+1, but the Avoid Collision intent inference has precedence over all other intents since the conflict alert condition is on; broadcast TCP and TCP+1 that solve CD&R problem confirm valid solution.

Scenario 10

Features:

- Ownship tracking two nearby aircraft
- Both neighboring aircraft are broadcasting position, velocity, and intent including next two waypoints with no lost data in the transmission of intent
- Separation conflict between the two nearby aircraft (not with ownship) determined by ownship CD&R subsystem using state information about the two nearby aircraft
- Neither nearby aircraft broadcasts a conflict resolution set of waypoints; Neither aircraft changes state vectors to indicate resolution activity
- No weather or turbulence present

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified (due to estimated conflict alert condition in violation of acceptable intent), Avoiding Collision (CD&R algorithm provides alternative frontside, backside, topside, and bottomside maneuver waypoints)
- Other intent models may have high values given the perceived OK status of these two aircraft, but the Avoid Collision intent inference has precedence over all other intents

Scenario 11

Features:

- Ownship tracking nearby aircraft
- Neighboring aircraft is broadcasting position, velocity, and intent including next two waypoints with no lost data in the transmission of intent
- Nearby aircraft does not follow RNP for TCP and deviates to avoid bad weather

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Avoid Hazardous Weather (Hazard avoidance algorithm provides alternative frontside, backside, topside, and bottomside maneuver waypoints around hazard),
- Skip Waypoint may have high correlation value if weather avoidance path provides opportunity to fly directly to TCP+1
- Other intent models should have low values

Scenario 12

Features:

- Ownship tracking nearby aircraft
- Neighboring aircraft is broadcasting position, velocity, and intent including next two waypoints and flight plan with no lost data in the transmission of intent
- Nearby aircraft does not follow RNP for TCP and deviates to avoid bad weather

Expected Outcome:

- High ranking by correlation function for: Broadcast intent not verified, Avoid Hazardous Weather (Hazard avoidance algorithm provides alternative frontside, backside, topside, and bottomside maneuver waypoints around hazard)
- Skip Waypoint may have high correlation value if weather avoidance path provides opportunity to fly directly to TCP+1
- Fly to Waypoint (all waypoints in flight plan tested) may have high correlation value if weather avoidance puts aircraft in a position to skip TCP and TCP+1 and go to a waypoint further in the flight plan
- Other intent models should have low values

3.0 CONFIDENCE ASSESSMENT

The AOP system needs a confidence assessment module to establish the accuracy of traffic management information for other AOP modules. For instance, the intent inference module exploits the confidence assessment information to prepare a best estimate of the state data and to address the issue of missing or delayed data.

The relationship of the confidence assessment task to the intent inference and hazard assessment tasks is illustrated in Figure 3.1. This figure also shows the primary four data sources to be exploited for this study: ADS-B, TIS, FIS, and CPDLC data. Note: Regional weather covers the region that both aircraft will occupy during the total time interval of conflict detection and resolution. Local weather covers the region that both aircraft will occupy from the current time to the predicted time of conflict.

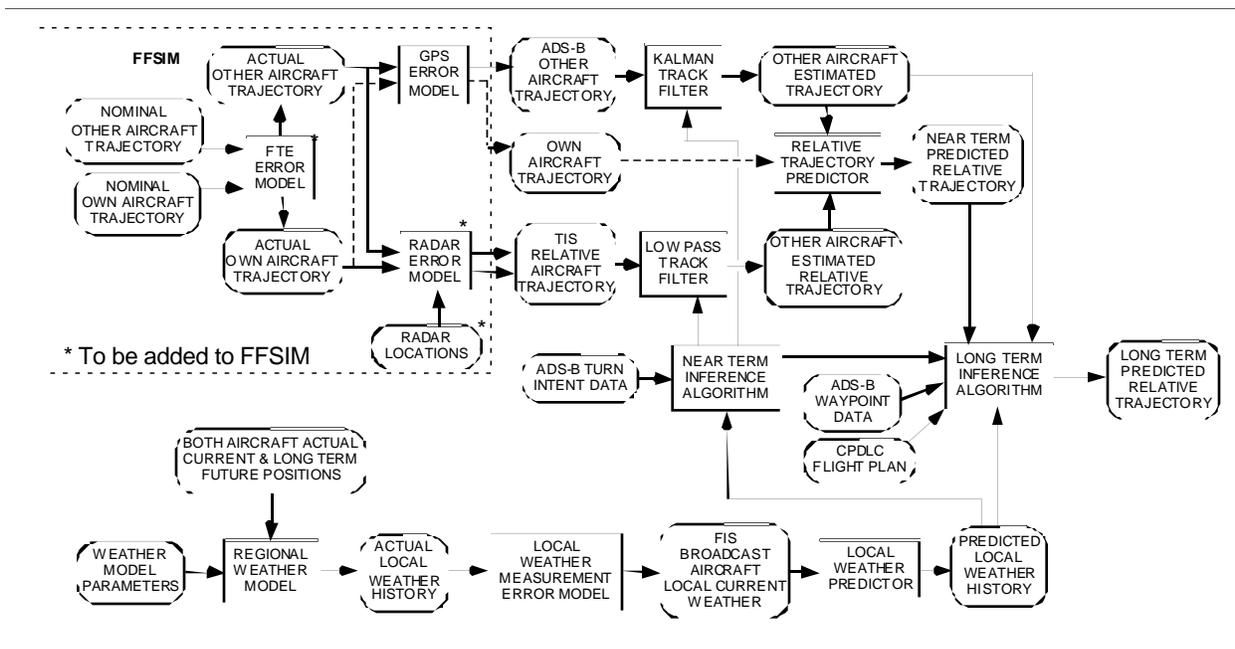


Figure 3.1 Other aircraft continuous intent inference data flow diagram.

The other aircraft trajectory will not only include the nominal flight path history but also the small perturbations about the nominal due to pilot or autopilot steering errors, or Flight Technical Errors (FTE). In addition, the trajectory information will be further corrupted by the sensors that are used to measure the trajectory parameters. Under ADS-B, the sensor is the GPS receiver onboard the other aircraft while under TIS, the primary sensor will be the surveillance radar. The reason that these sensor and FTE errors have to be considered is that they corrupt the predicted nearby aircraft trajectory. These errors are expected to be modeled under the RTO-21 FFSIM effort.

The upper part of Figure 3.1 illustrates the ADS-B processing while the middle part illustrates the TIS processing. For either path, a trajectory for both aircraft would have to be provided. In the case of the ADS-B path, both the threat and own trajectory are corrupted by GPS sensor and FTE errors. The threat trajectory is processed through the Kalman track filter that accepts as inputs the ADS-B threat trajectory and any near-term intent, such as aircraft turns or decelerations. The current estimated threat trajectory is combined with the own current trajectory to establish a current relative trajectory. This current relative trajectory is then propagated forward for

several tens of seconds or up to a few minutes using a relative trajectory predictor. This near-term predicted relative threat trajectory is then fed to the long-term inference algorithm.

The TIS path is somewhat similar to the ADS-B path except that the TIS data is provided in the form of relative trajectory data. Since the TIS data is of much lower accuracy and available at a much lower data frequency (12 seconds vs 1 second for ADS-B), the intent is to use it only when the ADS-B data is not available (e.g., the other aircraft does not have an ADS-B transponder or the transponder is not operating).

Due to the lower quality and lower frequency of this TIS data, a different Kalman track filter is proposed. This track filter is a Low Pass Filter (LPF) that provides a current relative trajectory using the relative TIS trajectory and any near-term intent data. This current relative trajectory is also fed to the relative trajectory predictor algorithm where it is propagated forward to provide a near-term intent assessment. This near-term intent assessment is then fed to the long-term inference algorithm.

A variety of qualitative data is expected to eventually be available over the FIS data link, which is still undergoing definition by the relevant RTCA committee. The focus under this effort will be the broadcast of weather cells to pilots in the area. The lower path in Figure 3.1 illustrates, in a preliminary form the generation of the actual weather in the region of interest and how this data is corrupted by imperfect measurements and data frequency. In the latter case, a near-term weather prediction has to be generated by the Airline Operations Center (AOC) or onboard the aircraft. Since this prediction will not be perfect, this will lead to uncertainties in the prediction. As shown in Figure 3.1, both the actual and the estimated weather models will be generated under the RTO 30 study.

The long-term inference algorithm relies on the near-term relative threat trajectories, the near-term intent, such as may be provided in the ADS-B message, and the flight plans that might be provided by CDPLC data. In addition, the potential hazards to the current aircraft flight path, such as provided by the FIS weather data and by the hazard assessment tasks, will be used as input. From these inputs, the long-term intent of the other aircraft will be inferred.

3.1 Input Data

3.1.1 ADS-B Message Data

The ADS-B state vector, shown in Figure 3.1, is defined in Table 3.1. The principal Kalman filter inputs are the time of applicability, latitude, longitude, geometric altitude, North velocity, East velocity, and geometric vertical rate. The position and velocity measurement uncertainty is provided by elements 5 and 10, respectively, in Table 3.1. The position error categories are further defined in Table 3.2 while the velocity uncertainty categories are defined in Table 3.3.

Table 3.1. The ADS-B state vector report definition [RTCA232, RTCA242, SK99a].

Element	Contents
1	Participant Address
2	Latitude
3	Longitude
4	Geometric Altitude
5	Navigation Uncertainty Category NUC_p – Position
6	Geometric Position Valid (Horizontal/Vertical)
7	*North Velocity
8	*East Velocity
9	*Geometric Vertical Rate
10	Navigation Uncertainty Category NUC_r – Velocity
11	Barometric Altitude (Pressure Altitude)
12	* Barometric Altitude Rate
13	* Air Speed (True/IAS)
14	* Ground Speed, Ground Track (True/Mag Heading)
15	* Turn Indication
16	Time of Applicability
17	Report Mode (Acquisition, Track, Default)

* An indication that no data is available should be provided if appropriate

The navigation position uncertainty categories are defined in Table 3.2 while the velocity uncertainties are defined in Table 3.3 with accuracy specified at the 95% (2σ) confidence level.

Table 3.2. Navigation Position Uncertainty Category, NUC_p (95% Confidence) [RTCA232, RTCA242, SK99a].

Category	Uncertainty	Category	Uncertainty
0	No Integrity	5	< 0.5 nmi
1	< 20 nmi (RNP-10)	6	< 0.2 nmi
2	< 10 nmi (RNP-5)	7	< 0.1 nmi
3	< 2 nmi (RNP-1)	8	TBD
4	< 1 nmi (RNP-0.5)	9	TBD

Table 3.3. Navigation Velocity Uncertainty Category, NUC_v (95% Confidence) [RTCA232, RTCA242, SK99a].

Category	Horizontal Velocity Uncertainty	Vertical Velocity Uncertainty
0	Unknown	Unknown
1	< 20 m/s	< 50 fps
2	< 3 m/s	< 15 fps
3	< 1 m/s	< 5 fps
4	< 0.3 m/s	< 1.5 fps

The turn indication information that describes near-term intent is provided by elements 15 and 16 of Table 3.1. For long-term intent, as described by the next waypoint, or Trajectory Change Point (TCP), latitude, longitude, altitude, time of applicability and time to reach it is obtained from the ADS-B Mode-Status Report as summarized in Table 3.4. The following TCP

latitude, longitude, altitude, and the time to reach it are obtained from the ADS-B TCP+1 On-Condition Report, as summarized in Table 3.5.

Table 3.4. The ADS-B Mode-Status Report definition [RTCA232, RTCA242, SK99a].

Element	Contents
1	*Participant Address
2	*Call Sign
3	*Participant Category
4	*Surveillance Support Code
5	*Emergency/Priority Status
6	*Class Codes
7	TCP Latitude
8	TCP Longitude
9	TCP Altitude (Baro Alt/FL)
10	Time-To-Go TTG
11	Operational Mode Specific Data
12	Flight Mode Specific Data
13	Time of Applicability

** Elements 1-6 comprise a Partial Mode-Status Report.*

Table 3.5. The ADS-B TCP+1 On-Condition Report definition [RTCA232, RTCA242, SK99a].

Element	Contents
1	Participant Address
2	TCP+1 Latitude
3	TCP+1 Longitude
4	TCP+1 Altitude (Baro/FL)
5	TCP+1 Time-To-Go TTG
6	Time of Applicability

3.1.2 TIS-B Data

The Traffic Information Service – Broadcast (TIS-B) provides a much more limited data set of other aircraft relative to own aircraft. It is based primarily on ground-based radar measurements of aircraft position that are not as accurate as the GPS-based ADS-B aircraft state data. TIS-B is intended for those aircraft that are not equipped with ADS-B transponders or for those aircraft that are equipped with ADS-B but cannot ‘see’ a non-equipped aircraft. The latter represents the mixed equipage case. The TIS traffic information data block is summarized in Table 3.6.

Table 3.6. The TIS-B traffic information block [RTCA239, SK99b].

Element	Content
1	Traffic Bearing (deg)
2	Traffic Range (nmi)
3	Relative Altitude (ft)
4	Relative Altitude Rate
5	Traffic Heading
6	Traffic Status

Due to the use of minimum bit encoding, TIS-B message elements are quantized as summarized in Tables 3.7 through 3.12.

Table 3.7. The TIS-B traffic bearing encoding [RTCA239, SK99b].

Encoding	Range of Values
0	$0^{\circ} \leq B < 6^{\circ}$
1	$6^{\circ} \leq B < 12^{\circ}$
2	$12^{\circ} \leq B < 18^{\circ}$
:	:
59	$354^{\circ} \leq B < 360^{\circ}$

Table 3.8. The TIS-B traffic range encoding [RTCA239, SK99b].

Encoding	Range of Values (nmi)	Encoding	Range of Values (nmi)
0	$0 \leq R \leq 1/8$	8	$1\ 7/8 < R \leq 2\ 1/4$
1	$1/8 < R \leq 3/8$	9	$2\ 1/4 < R \leq 2\ 3/4$
2	$3/8 < R \leq 5/8$	10	$2\ 3/4 < R \leq 3\ 1/2$
3	$5/8 < R \leq 7/8$	11	$3\ 1/2 < R \leq 4\ 1/2$
4	$7/8 < R \leq 1\ 1/8$	12	$4\ 1/2 < R \leq 5\ 1/2$
5	$1\ 1/8 < R \leq 1\ 3/8$	13	$5\ 1/2 < R \leq 6\ 1/2$
6	$1\ 3/8 < R \leq 1\ 5/8$	14	$6\ 1/2 < R \leq 7$
7	$1\ 5/8 < R \leq 1\ 7/8$	15	$7 < R$

Table 3.9. The TIS-B traffic relative altitude encoding [RTCA239, SK99b].

Encoding	Relative Altitude (ft)	Encoding	Relative Altitude (ft)
0	$0 \leq \Delta h \leq 100$	16	No Altitude Reported
1	$100 \leq \Delta h < 200$	17	$-100 < \Delta h < 0$
2	$200 \leq \Delta h < 300$	18	$-200 < \Delta h \leq -100$
:	:	:	:
9	$900 \leq \Delta h < 1000$	26	$-1000 < \Delta h \leq -900$
10	$1000 \leq \Delta h < 1500$	27	$-1500 < \Delta h \leq -1000$
11	$1500 \leq \Delta h < 2000$	28	$-2000 < \Delta h \leq -1500$
:	:	:	:
14	$3000 \leq \Delta h < 3500$	30	$-3000 < \Delta h \leq -2500$
15	$3500 < \Delta h$	31	$\Delta h \leq -3000$

Table 3.10. The TIS-B traffic heading encoding [RTCA239, SK99b].

Encoding	Range of Values
0	$0^{\circ} \leq B < 45^{\circ}$
1	$45^{\circ} \leq B < 90^{\circ}$
2	$90^{\circ} \leq B < 135^{\circ}$
:	:
7	$315^{\circ} \leq B < 360^{\circ}$

Table 3.11. The TIS-B traffic altitude rate encoding [RTCA239, SK99b].

Encoding	Range of Values (ft/min)
----------	--------------------------

0	unused
1	$500 < v_h$, Climbing
2	$-500 > v_h$, Descending
3	$-500 \leq v_h \leq 500$, Level

Table 3.12. The TIS-B traffic status encoding [SK99b].

Encodin g	Range of Values (ft/min)
0	Proximity Alert
1	Traffic Alert

TIS-B data have a few shortcomings. Absent in Table 3.6 are the time of applicability and the relative velocity. As seen in Table 3.8, the data provided by TIS-B is very short range when you consider that the minimum en route separation requirement is 5 nmi. Finally, the TIS-B data message specifications do not provide any indication of the accuracy of the data.

3.1.3 FIS Data

Flight Information Service (FIS) data provides weather and other flight advisory information to pilots for the purpose of enhancing situation awareness and strategic decision making. The exact FIS data that will be broadcast are still not fully defined by the RTCA. A list of products that have been *proposed* by RTCA-169 is summarized in Table 3.13. The cruise products that will probably most directly affect the long-term intent of another aircraft is the weather data. Precipitation Maps, provided by the Graphical Weather Service, and the Hazardous Weather Advisories, are the principal data sources of interest for AOP Long-Term Intent algorithms.

The data contents of the FIS-B messages are transmitted with a lossless compression mechanism. For the AOP, there is little benefit in simulating the compression and de-compression processes. We assume that the compression mechanism does not have to be simulated. The raw data without compression can be transmitted to the AOP, and if needed, corrupt or missing data can be included without the need to simulate compression.

Based on the latest draft versions of the FIS MASPS, the contents of the FIS messages are designed to be composed of:

- METARs and PIREPs, with a maximum of 120 minutes in age,
- Weather forecasts within a maximum lifespan as indicated in the message,
- Mosaic Weather Products with a maximum of 75 minutes lifespan,
- Graphical Maps for METARs and CATMETs with a maximum of 75 minutes lifespan, and
- NEXRAD Weather Radar Mosaic Precipitation Maps with a maximum of 10 minutes of lifespan.

As shown in Table 3.14, the FIS products include text, graphic, and gridded types. The only gridded format data are GRIB format forecasts for winds, temperature, relative humidity, vertical velocity, and other weather parameters. The graphic type format is used more often and is very much different from the gridded format, in that the graphic type format simply describes the colors of a display image while the gridded format describes the raw data. The size and range of colors in the graphic format depends on the compression mechanism chosen for the TIS product installed in an aircraft. Six different compression mechanisms are available, as shown in Table 3.15. For instance, one graphic format allows up to 65,536 pixels of colors or data codes ranging from 0 to

255, and another graphic format only allows colors or data codes ranging from 0 to 15. The soon-to-be-published FIS MASPS describes the details.

Table 3.13. Operational usage of initial FIS Request/Response products [RTCA232].

FIS Product	Operational/Regulatory (Approach/Departure)	Hazard Avoidance (Approach/Departure)	Hazard Avoidance (Cruise)	Optimization (Cruise)	Situational Awareness (Cruise)
Precipitation Map (Graphical Weather Service)			X		X
Terminal Weather Information for Pilots (TWIP)		X			X
Automated Terminal Information Service (ATIS)	X				X
Hazardous Weather Advisory			X		X
Surface Observation					X
Terminal Forecast					X
Winds/Temperature				X	
Pilot Reports (PIREPs)			X		X
Notice to Airmen (NOTAMs)					X
Runway Visual Range (RVR)	(Local)				(Remote)

Table 3.14. Expected FIS-B Products (from FIS MASPS [RTCA195]).

Field	Product	Type
0	METAR and SPECI	Text
1	TAF and Amended TAF	Text
2	SIGMET	Text
3	Convective SIGMET	Text
4	AIRMET	Text
5	PIREP	Text
6	AWW	Text
7	Winds and Temperatures Aloft	Text
8-50	Reserved	
51	National NEXRAD, Type 0 - Low dynamic range	Graphic
52	National NEXRAD, Type 1 - 8 level	Graphic
53	National NEXRAD, Type 2 - 8 level	Graphic
54	National NEXRAD, Type 3 - 16 level	Graphic
55	Regional NEXRAD, Type 0 - Low dynamic range	Graphic
56	Regional NEXRAD, Type 1 - 8 level	Graphic
57	Regional NEXRAD, Type 2 - 8 level	Graphic
58	Regional NEXRAD, Type 3 - 16 level	Graphic

Field	Product	Type
59	Individual NEXRAD, Type 0 - low dynamic range	Graphic
60	Individual NEXRAD, Type 1 - 8 level	Graphic
61	Individual NEXRAD, Type 2 - 8 level	Graphic
62	Individual NEXRAD, Type 3 - 16 level	Graphic
63-80	Reserved	
81	Radar echo tops graphic, scheme 1: 16-level	Graphic
82	Radar echo tops graphic, scheme 2: 8-level	Graphic
83	Storm tops and velocity	Graphic
84-100	Reserved	
101	Lightning strike type 1 (pixel level)	Graphic
102	Lightning strike type 2 (grid element level)	Graphic
103	Lightning strike type 3 (raster encoding scheme)	Graphic
104-150	Reserved	
151	Point phenomena, vector format	Graphic
152-200	Reserved	
201	Surface conditions/winter precipitation graphic	Graphic
202	Surface weather systems	Graphic
203-250	Reserved	
251	National METAR Graphic	Graphic
252	CATMET format	Graphic
253	Regional METAR Graphic	Graphic
254	AIRMET, SIGMET: Bitmap encoding	Graphic
255	AIRMET, SIGMET: Vector representation, closed curves	Graphic
256	AIRMET, SIGMET: Vector encoding scheme	Graphic
257-300	Reserved	
301	Gridded Weather Forecast Products	Gridded

Because of the lack of gridded format precipitation map data in the FIS messages, AOP modules will have to use graphic weather data interpreted in gridded form. NEXRAD precipitation products in the FIS messages are in graphic form, however, they can be converted from radar levels to intensity values based on the relationships defined in Tables 3.15 through 3.17. While precise radar reflectivity values (in dBZ) cannot be determined, the levels transmitted in the graphic image can be replaced with the average value within the range to specify the radar reflectivity for a given level (Level 0-7 or Level 0-15, depending on NEXRAD Type).

Table 3.15. The 8-Level NEXRAD Type 1 encoding for the FIS message [RTCA195].

Code	Radar Level	Intensity (dBZ)
0	Level 0	dBZ <5, No data
1	Level 1	5 ≤ dBZ < 18
2	Level 2	18 ≤ dBZ < 30
3	Level 3	30 ≤ dBZ < 41
4	Level 4	41 ≤ dBZ < 46
5	Level 5	46 ≤ dBZ < 50
6	Level 6	50 ≤ dBZ < 57
7	Level 7	57 ≤ dBZ

Table 3.16. The 8-Level NEXRAD Type 2 encoding for the FIS message [RTCA195].

Code	Radar Level	Intensity (dBZ)
0	Level 0	dBZ < 20, No data
1	Level 1	20 ≤ dBZ < 25
2	Level 2	25 ≤ dBZ < 30
3	Level 3	30 ≤ dBZ < 35
4	Level 4	35 ≤ dBZ < 40
5	Level 5	40 ≤ dBZ < 50
6	Level 6	50 ≤ dBZ < 60
7	Level 7	60 ≤ dBZ

Table 3.17. The 16-Level NEXRAD Type 3 encoding for the FIS message [RTCA195].

Code	Radar Level	Intensity (dBZ)
0	Level 0	dBZ < 5, No data
1	Level 1	5 ≤ dBZ < 10
2	Level 2	10 ≤ dBZ < 15
3	Level 3	15 ≤ dBZ < 20
4	Level 4	20 ≤ dBZ < 25
5	Level 5	25 ≤ dBZ < 30
6	Level 6	30 ≤ dBZ < 35
7	Level 7	35 ≤ dBZ < 40
8	Level 8	40 ≤ dBZ < 45
9	Level 9	45 ≤ dBZ < 50
10	Level 10	50 ≤ dBZ < 55
11	Level 11	55 ≤ dBZ < 60
12	Level 12	60 ≤ dBZ < 65
13	Level 13	65 ≤ dBZ < 70
14	Level 14	70 ≤ dBZ < 75
15	Level 15	75 ≤ dBZ

The FIS message also allows for a vector format. The vector format allows for a series of connected line segments or a polygon region to be identified using connected line segments described with a series of vertices. This data field allows the transmission of weather fronts (as a series of connected line segments), icing regions (lines segments connected to form a polygon), turbulence regions (lines segments connected to form a polygon), etc. For the AOP, it is assumed that polygon data transmitted in graphical format can also be transmitted through this vector format. Thus, instead of having to deduce polygon regions from graphical format, the FIS vector format will be used to define polygon regions in FIS data.

The AIRMET/SIGMET vector encoding scheme is an example of a message that exploits the vector format. Table 3.18 illustrates the data in an AIRMET/SIGMET field. Note that the type of 3D region that is encoded is a special type of polyhedron; that is, a single list of vertices describes the same polygon for the top as well as the bottom of the polyhedron. All sides of the polyhedron are vertical, connecting the top vertices to the bottom vertices.

All FIS data are time stamped. Time stamps include day of month (optional), time in hours, time in minutes, and time in seconds (optional).

Table 3.18. The AIRMET/SIGMET vector encoding definition for FIS [RTCA195].

Data Item	Resolution	Definition
-----------	------------	------------

Hazard Type		For AIRMET <ul style="list-style-type: none"> • S (Sierra) IFR & Mt. Obscuration • T (Tango) Turb & Strong Sfc Wind • Z (Zulu) Icing & Freezing Lvl For SIGMET <ul style="list-style-type: none"> • Icing • Turbulence • Thunderstorms
Likelihood (optional)		Low; Moderate; High
Severity		Trace; Light ; Moderate; Severe
Altitude 1	9 bits	Top of hazard layer in hundreds of feet
Altitude 2	9 bits	Bottom of hazard layer in hundreds of feet
Movement Direction	6 bits	Direction in tens of degrees
Movement Speed	8 bits	Speed in kts
Number of vertices	5 bits	Number of vertices
1 st Latitude	14 bits	Latitude coordinate for 1 st vertex of watch box polygon in hundredths of a degree; MSB = 1 for South Latitude; 0 – 9,000
1 st Longitude	15 bits	Longitude coordinate for 1 st vertex of watch box polygon in hundredths of a degree; MSB = 1 for East Longitude; 0 – 18,000
Next Latitude	14 bits	Latitude coordinate for next vertex of watch box polygon
Next Longitude	15 bits	Longitude coordinate for next vertex of watch box polygon
etc	“	“
Last Latitude	14 bits	Latitude coordinate for last vertex of watch box polygon; user’s software to connect last vertex to 1 st vertex (no need to transmit the last with this assumption)
Last Longitude	15 bits	Longitude coordinate for last vertex of watch box polygon; user’s software to connect last vertex to 1 st vertex (no need to transmit the last with this assumption)

3.2 Theory

The confidence assessment approach that has been selected is centered on using a Kalman Filtering with ADS-B data. With the TIS-B data, a Low Pass Filter (LPF) is used. The ADS-B message provides both current measurement of the other aircraft state and the nominal uncertainty.

The motivation for using a Kalman filter is to get an estimate of the current state of nearby aircraft with higher accuracy. Better predictions of the future position of nearby aircraft are produced by integrating the current velocity and acceleration, if available. While this is also the motivation for using the LPF with the TIS-B data, the TIS-B data does not provide any higher derivatives of the current relative position of the two aircraft nor does it provide any accuracy for the TIS-B position data. Without the higher derivatives, the predicted relative position of the two aircraft will have very low accuracy. Hence, the LPF provides an estimate of the relative velocity of the other aircraft as well as the uncertainty in both the position and velocity.

The following sections present the equations for the LPF. This filter estimates the relative ground speed and heading two aircraft using sequential relative North and East position measurements obtained from the TIS-B message. The principal source of the TIS-B data is the ground-based Secondary Surveillance Radar (SSR). The goal is to obtain a best estimate of the relative ground speed and heading such that when these estimates are integrated, the estimated relative position history is obtained with minimum error.

Also presented is a Kalman filter based on ADS-B data. This filter takes aircraft position and velocity data for nearby aircraft and estimates the position, velocity, and acceleration of these nearby aircraft. Since this filter also provides dynamic estimates of the uncertainty in these, estimates, more accurate estimates of the accuracy of the filter data are obtained. Since the LPF is a suboptimal version of the Kalman filter, the Kalman filter is presented first.

3.2.1 General Kalman Filter Theory

The Kalman filter is based on combining the predicted (modeled) and measured values of a state vector of interest in an optimal (minimum variance) way. Hence, the first step is to define the state equation that describes how the actual state vector changes with time. Next, a measurement equation must be defined that establishes the relationship between the measurement vector and the state vector. Since the Kalman filter is based on linear state and measurement equations, non-linear state and measurement equations must first be linearized.

If the actual state and measurement equations are given by:

$$\underline{\dot{x}} = \underline{f(x)} + \underline{n} \quad (3.1)$$

$$\underline{z} = \underline{h(x)} + \underline{m} \quad (3.2)$$

where, \underline{x} , \underline{z} are the state and measurement vectors, \underline{f} , \underline{h} are the nonlinear state and measurement function, and \underline{n} , \underline{m} are state and measurement Gaussian white noise vectors. Since the actual state and measurement vectors are unknown, the state estimate and measurement equations are given by:

$$\underline{\dot{\hat{x}}} = \underline{f(\hat{x})} \quad (3.3)$$

$$\underline{\hat{z}} = \underline{h(\hat{x})} \quad (3.4)$$

where $\underline{\hat{x}}$, $\underline{\hat{z}}$ are the state and measurement vector estimates.

Assume that the relationship between the estimate and the actual state can be expressed as follows:

$$\underline{\hat{x}} = \underline{x} + \underline{\delta x} \quad (3.5)$$

Differentiating (3.5) results in:

$$\underline{\dot{\hat{x}}} = \underline{\dot{x}} + \underline{\delta \dot{x}} \quad (3.6)$$

Substituting equations (3.5) and (3.6) into equation (3.3) gives:

$$\underline{\dot{x}} + \underline{\delta \dot{x}} = \underline{f(x + \delta x)} \quad (3.7)$$

and taking a first order Taylor series expansion of the term on the right-hand side of (3.7):

$$\underline{f(x + \delta x)} = \underline{f(x)} + \frac{\partial \underline{f(x)}}{\partial \underline{x}} \underline{\delta x} + \dots \quad (3.8)$$

Substituting (3.1) and (3.8) into (3.7): $\underline{\delta \dot{x}} \cong [F(x)] \underline{\delta x} - \underline{n}$ (3.9)

where, $[F(x)] = \frac{\partial \underline{f(x)}}{\partial \underline{x}}$ (3.10)

While equation (3.9) provides a linear state equation for the estimation error, it requires that the unknown actual state and process noise vectors. A further approximation that is made to

equation (3.9) is to use the estimated state to evaluate the state matrix and to ignore the process noise vector:

$$\underline{\delta \dot{x}} \dots [F(\underline{\hat{x}})] \underline{\delta x} . \quad (3.11)$$

With the estimated state obtained as a solution to equation (3.3) and the estimation error which is a solution to equation (3.11), the actual state is obtained using equation (3.5):

$$\underline{x} \cong \underline{\hat{x}} - \underline{\delta \hat{x}} . \quad (3.12)$$

Now the non-linear measurement equation (3.4) can be linearized in a similar fashion. Substitute equation (3.5) into (3.4):

$$\underline{\hat{z}} = \underline{h}(\underline{x} + \underline{\delta x}) \quad (3.13)$$

and expanding the right-hand side in a first-order Taylor series expansion results in:

$$\underline{h}(\underline{x} + \underline{\delta x}) = \underline{h}(\underline{x}) + \frac{\partial \underline{h}(\underline{x})}{\partial \underline{x}} \underline{\delta x} + \dots \quad (3.14)$$

Substituting equation (3.2) and (3.14) into equation (3.13) gives:

$$\underline{\delta z} \dots (\underline{\hat{z}} - \underline{z}) = [H(\underline{x})] \underline{\delta x} - \underline{m} \quad (3.15)$$

where,

$$[H(\underline{x})] \dots \frac{\partial \underline{h}(\underline{x})}{\partial \underline{x}} . \quad (3.16)$$

Since the actual state and the measurement noise is unknown, the following approximations can be made to equation (3.15):

$$\underline{\delta \hat{z}} \dots [H(\underline{\hat{x}})] \underline{\delta \hat{x}} . \quad (3.17)$$

A discrete form of equation (3.11) is obtained as follows:

$$\underline{\delta \hat{x}}_k^{(-)} = [\Phi]_{k,k-1} \underline{\delta \hat{x}}_{k-1}^{(+)} \quad (3.18)$$

where,

$$[\Phi]_{k,k-1} = \prod_{n=0}^{k-1} \frac{(t_k - t_{k-1})}{n!} [F]_{k-1} .$$

(3.19)

Also, from equation (3.3), the discrete estimated state prediction equation is:

$$\underline{\hat{x}}_k^{(-)} \cong \underline{\hat{x}}_{k-1}^{(+)} + (t_k - t_{k-1}) f(\underline{\hat{x}})_{k-1} \quad (3.20)$$

and,

$$\underline{\hat{x}}_k^{(+)} = \underline{\hat{x}}_k^{(-)} + \underline{\delta \hat{x}}_k^{(+)} . \quad (3.21)$$

Then the Kalman filter update is obtained as follows [Ge74]:

$$\underline{\delta \hat{x}}_k^{(+)} = \underline{\delta \hat{x}}_k^{(-)} + [K]_k \{ \underline{\delta z}_k - [H] \underline{\delta \hat{x}}_k^{(-)} \} \quad (3.22)$$

where,

$$[K]_k \dots [P]_k^{(-)} [H]^T \{ [H] [P]_k^{(-)} [H]^T + [R] \}^{-1} \quad (3.23)$$

$$[P]_k^{(-)} = [\Phi]_{k,k-1} [P]_{k-1}^{(+)} [\Phi]_{k,k-1}' + [Q]_{k-1} \quad (3.24)$$

$$[P]_k^{(+)} = \{[I] - [K]_k [H]\} [P]_k^{(-)} \{[I] - [K]_k [H]\}^T + [K]_k [R] [K]_k^T \quad (3.25)$$

or,
(3.26)

$$[P]_k^{(+)} = \{[I] - [K]_k [H]\} [P]_k^{(-)}$$

and,

$$[Q]_k = \sum_{n=1}^{\infty} \frac{(t_k - t_{k-1})^n}{n!} [J]_n \quad (3.27)$$

$$[J]_n = [F]_{k-1} [J]_{n-1} + ([F]_{k-1} [J]_{n-1})^T \quad \text{with, } [J] = [N] \quad (3.28)$$

where, $[K]$ is the Kalman gain matrix, $[P]$ is the estimation error covariance matrix, $[N]$ and $[Q]$ are continuous and discrete process noise matrices, and $[R]$ is the measurement noise matrix. Equation (3.25) is the general version of (3.26). While the latter can be used for any filter gain matrix, whether Kalman or not, the former is only valid for the Kalman filter gain matrix.

The key to confidence assessment is the estimation error covariance matrix. This matrix provides the Kalman filter a confidence estimate of how good the current state has been estimated. Hence, the key confidence assessment equations (3.23), (3.24) and (3.26) require the Kalman filter to be used.

The measurement matrix for the ADS-B takes on a simple diagonal form while a more complicated form is required for the TIS radar data:

$$[R]_{ADS-B} = \begin{bmatrix} \sigma_x^2, & 0, & 0, & 0 \\ 0, & \sigma_y^2, & 0, & 0 \\ 0, & 0, & \sigma_v^2, & 0 \\ 0, & 0, & 0, & \sigma_\psi^2 \end{bmatrix} \quad (3.29)$$

and,

$$[R]_{TIS} = [T]_k [r] [T]_k^T \quad (3.30)$$

$$\begin{bmatrix} \sigma_r^2, & 0, & 0 \\ 0, & \sigma_\theta^2, & 0, & 0 \end{bmatrix}$$

with,

$$[r] = \begin{bmatrix} 0, & 0, & -2\sigma_r^2 \sqrt{\Delta t^2} \\ 0, & 0, & \sqrt{\Delta t^2} \end{bmatrix}$$

$$\begin{bmatrix} 0, & 0, & -2\sigma_\theta^2 \sqrt{\Delta t^2} \\ 0, & 0, & \sqrt{\Delta t^2} \end{bmatrix}$$

(3.31)

$$\begin{bmatrix} \cos\theta, & -r\sin\theta, & 0, & 0 \\ \sin\theta, & r\cos\theta, & 0, & 0 \end{bmatrix}$$

and,

$$[T]_k = \begin{bmatrix} \frac{r\dot{e}^2}{\sqrt{G_{G,k}}} & 0, & \frac{\dot{r}}{\sqrt{G_{G,k}}} & \frac{r^2\dot{e}}{\sqrt{G_{G,k}}} \\ \frac{r\dot{e}}{\sqrt{G_{G,k}}} & 1, & -\frac{r\dot{e}}{\sqrt{G_{G,k}}} & \frac{r\dot{r}}{\sqrt{G_{G,k}}} \end{bmatrix}$$

(3.32)

where, r is the range and $\dot{\epsilon}$ the azimuth of the aircraft from the radar, \dot{r} and $\dot{\epsilon}$ are the range and azimuth rates, and V_G is the speed of the aircraft.

Since the TIS-B measurement noise matrix requires knowledge of the radar location as well as higher derivatives of the range and azimuth measurements, a default solution is to use diagonal matrix such as used for the ADS-B measurements with the statistics that are typical for a radar.

3.2.2 ADS-B Kalman Filter

The number of possible state and measurement combinations that can be used for a Kalman filter based on ADS-B message data is summarized in Table 3.14. The one selected as the baseline design for the ADS-B message data is Option 4.

Table 3.14. ADS-B Kalman filter state and measurement options.

Measurement	Option			
	1	4	5	10
North position, x_m	√	√	√	√
East position, y_m	√	√	√	√
Ground speed, $v_{G,m}$		√	√	√
Ground track, $\psi_{G,m}$		√	√	√
Ground acceleration $a_{G,m}$ (derived from $v_{G,m}$)				√
Ground track rate, $\dot{\psi}_{G,m}$ (derived from $\psi_{G,m}$)				√
States				
North position, x	√	√	√	√
East position, y	√	√	√	√
Ground speed, v_G	√	√	√	√
Ground track, ψ_G	√	√	√	√
Ground acceleration a_G			√	√
Ground track rate, $\dot{\psi}_G$			√	√

Instead of North and East position, the ADS-B message provides the latitude and longitude of the other aircraft. Since working in a local level coordinate system is advantageous, the following calculations are performed:

$$x_M = (\ddot{o} - \ddot{o}_R) R_P \cos(\ddot{\epsilon}_{G,R}) \quad (3.33)$$

and,

$$y_M = (\ddot{\epsilon}_G - \ddot{\epsilon}_{G,R}) R_M \quad (3.34)$$

with,

$$R_M = \frac{a \cos(\lambda_{G,R})}{\sqrt{1 - \epsilon^2 (\sin(\lambda_{G,R}))^2}} \quad (3.35)$$

$$R_P = \frac{a}{\sqrt{1 - \varepsilon^2 (\sin(\lambda_{G,R}))^2}} \quad (3.36)$$

where, x_M, y_M are the measured North and East local level position, a, ε are Earth equatorial radius and eccentricity, R_M, R_P are the meridian and prime radius of earth curvature, $\lambda_G, \lambda_{G,R}$ are the geodetic latitude of other and own (reference) aircraft, and ϕ, ϕ_R are the longitude of other and own (reference) aircraft.

With ADS-B data, the aircraft velocity is provided in orthogonal coordinates rather than polar coordinates. The conversion (pre-processing) required to obtain the polar coordinate velocity components are:

$$v_{G,Mk} = \sqrt{v_{GMx,k}^2 + v_{GMy,k}^2} \quad (3.37)$$

and,

$$\psi_{GM,k} = \arctan \frac{-v_{GMy,k}}{v_{GMx,k}} \quad (3.38)$$

where v_{GM} is the measured ground speed and ψ_{GM} is the measured ground track.

In the design of Kalman filters, there is a preference to select coordinate frames that uncouple the dynamics between different axes. Hence, the ground speed of an aircraft is less likely to be coupled to the track angle of the aircraft, compared to the coupling between the North and East velocity components.

Since the ADS-B ground speed and heading are based on relatively precise GPS data, the ground speed-derived ground acceleration and the track-derived track rate, that are considered for Option 4, are computed as follows:

$$a_{GM,k} \cong \frac{v_{GM,k} - v_{GM,k-1}}{(t_k - t_{k-1})} \quad (3.39)$$

and,

$$\psi_{GM,k} \cong \frac{\psi_{GM,k} - \psi_{GM,k-1}}{(t_k - t_{k-1})} \quad (3.40)$$

where a_{GM} is the measured ground acceleration and $\dot{\psi}_{GM}$ is the measured ground track rate.

The specific form of the Kalman filter for Option 4 consists of the following equations. The predicted state error is:

$$\begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \\ \delta\hat{a}_G \\ \delta\hat{\dot{\psi}}_G \end{matrix} \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \end{matrix} \begin{matrix} (-) \\ \\ \\ \\ \\ \downarrow \\ \downarrow \end{matrix} = [\hat{\Phi}]_{k,k-1} \begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \\ \delta\hat{a}_G \\ \delta\hat{\dot{\psi}}_G \end{matrix} \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \end{matrix} \begin{matrix} (+) \\ \\ \\ \\ \\ \downarrow \\ \downarrow \end{matrix} \quad (3.41)$$

The discrete predicted estimated state is:

$$\begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(-)} \Big|_k \equiv \begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(+)} \Big|_{k-1} + \Delta t \begin{matrix} -\hat{v}_G \cos \hat{\psi}_G \\ \hat{v}_G \sin \hat{\psi}_G \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \Big|_{k-1} \quad (3.42)$$

The Kalman filter update of the estimated state is:

$$\begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(+)} \Big|_k = \begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(-)} \Big|_k + [K]_k \begin{matrix} -x_M \\ y_M \\ v_{G,M} \\ \psi_{G,M} \\ a_{G,M} \\ \dot{\psi}_{G,M} \end{matrix} \Big|_k - [H] \begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(-)} \Big|_k ? \quad (3.43)$$

The measurement matrix is:

$$[H] = [I] \quad (3.44)$$

The derived measurements are:

$$v_{G,M,k} = \sqrt{v_{M,x,k}^2 + v_{M,y,k}^2} \quad (3.45)$$

$$\psi_{G,M,k} = \arctan \frac{v_{M,y,k}}{v_{M,x,k}} \quad (3.46)$$

$$a_{GM,k} \equiv \frac{v_{GM,k} - v_{GM,k-1}}{(t_k - t_{k-1})} \quad (3.47)$$

and,

$$\dot{\psi}_{Gm,k} \equiv \frac{\psi_{GM,k} - \psi_{GM,k-1}}{(t_k - t_{k-1})} \quad (3.48)$$

The Kalman filter updated estimate of state is then:

$$\begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(+)} \Big|_k = \begin{matrix} \hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{matrix}^{(-)} \Big|_k + \begin{matrix} \delta \hat{x} \\ \delta \hat{y} \\ \delta \hat{v}_G \\ \delta \hat{\psi}_G \\ \delta \hat{a}_G \\ \delta \hat{\dot{\psi}}_G \end{matrix} \Big|_k \quad (3.49)$$

The Kalman filter gain matrix is obtained as follows:

$$[K]_k = [P]_k^{(-)} [H]^T \{ [H] [P]_k^{(-)} [H]^T + [M] [R] [M]^T \}^{-1} . \quad (3.50)$$

The predicted (propagated) estimation error covariance matrix is:

$$[P]_k^{(-)} = [\Phi]_{k,k-1} [P]_{k-1}^{(+)} [\Phi]_{k,k-1}^T + [Q]_{k-1} . \quad (3.51)$$

The Kalman filter updated estimation error covariance matrix is:

$$[P]_k^{(+)} = \{ [I] - [K]_k [H] \} [P]_k^{(-)} . \quad (3.52)$$

The discrete process noise matrix is:

$$[Q]_k = \int_{n=1}^{\infty} \frac{(t_k - t_{k-1})^n}{n!} [J]_n \quad (3.53)$$

$$[J]_n = [F]_{k-1} [J]_{n-1} + ([F]_{k-1})^n [J]_{n-1} \quad \text{with, } [J] = [N] . \quad (3.54)$$

The continuous process noise covariance matrix is:

$$[N] = \begin{bmatrix} \sigma_{Nx}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{Ny}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{Nv}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{N\theta}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{Na}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{N\dot{\theta}}^2 \end{bmatrix} . \quad (3.55)$$

A convenient way to select the σ variables in equation (3.55) is to compute them using the initial σ and the corresponding reciprocal time constants [G74]. This assures that each first-order Gauss Markov process is bounded. Here are the appropriate equations:

$$\sigma_{Nx} = \sigma_{x,1} \sqrt{\beta_x} \quad (3.56)$$

$$\sigma_{Ny} = \sigma_{y,1} \sqrt{\beta_y} \quad (3.57)$$

$$\sigma_{Nv} = \sigma_{v,1} \sqrt{\beta_v} \quad (3.58)$$

$$\sigma_{N\psi} = \sigma_{\psi,1} \sqrt{\beta_\psi} \quad (3.59)$$

$$\sigma_{Na} = \sigma_{a,1} \sqrt{\beta_a} \quad (3.60)$$

$$\sigma_{N\dot{\theta}} = \sigma_{\dot{\theta},1} \sqrt{\beta_{\dot{\theta}}} . \quad (3.61)$$

and,

The transition matrix is:

$$[\hat{\Phi}]_{k,k-1} = \int_{n=0}^{\infty} \frac{(t_k - t_{k-1})^n}{n!} [\hat{F}]_{GPS} \Big|_{t-1} . \quad (3.62)$$

The state matrix is:

$$\begin{bmatrix} \hat{F}_{GPS} \end{bmatrix}_{k-1} \dots = \begin{bmatrix} -\hat{a}_x, & 0, & \cos \hat{\psi}_G, & -\hat{v}_G \sin \hat{\psi}_G, & 0, & 0 \\ 0, & -\hat{a}_y, & \sin \hat{\psi}_G, & \hat{v}_G \cos \hat{\psi}_G, & 0, & 0 \\ 0, & 0, & -\hat{a}_v, & 0, & 1, & 0 \\ 0, & 0, & 0, & -\hat{a}_\theta, & 0, & 1 \\ 0, & 0, & 0, & 0, & -\hat{a}_a, & 0 \\ 0, & 0, & 0, & 0, & 0, & -\hat{a}_\theta \end{bmatrix}_{k-1} \quad (3.63)$$

The measurement noise covariance matrix is expressed in North-East coordinate position, velocity, and acceleration coordinates is:

$$[R] = \begin{bmatrix} \sigma_{Mx}^2, & 0, & 0, & 0, & 0, & 0 \\ 0, & \sigma_{My}^2, & 0, & 0, & 0, & 0 \\ 0, & 0, & \sigma_{Mvx}^2, & 0, & 0, & 0 \\ 0, & 0, & 0, & \sigma_{Mvy}^2, & 0, & 0 \\ 0, & 0, & 0, & 0, & \sigma_{Max}^2, & 0 \\ 0, & 0, & 0, & 0, & 0, & \sigma_{May}^2 \end{bmatrix} \quad (3.64)$$

The acceleration σ variables in equation (3.64) are derived from the velocity σ variables as follows:

$$\sigma_{Max}^2 \cong 2 \frac{\sigma_{Mvx}^2}{\Delta t} \quad (3.65)$$

$$\sigma_{May}^2 \cong 2 \frac{\sigma_{Mvy}^2}{\Delta t} \quad (3.66)$$

Since the velocity and acceleration measurements that are used are expressed in terms of ground speed, ground track, ground acceleration, and ground track rate, a conversion matrix is required:

$$[M] = \begin{bmatrix} 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0 \\ 0, & 0, & \frac{v_{x,k}}{v_{G,k}}, & \frac{v_{y,k}}{v_{G,k}}, & 0, & 0 \\ 0, & 0, & -\frac{v_{y,k}}{v_{G,k}^2}, & \frac{v_{x,k}}{v_{G,k}^2}, & 0, & 0 \\ 0, & 0, & 0, & 0, & \frac{v_{x,k}}{v_{G,k}}, & \frac{v_{y,k}}{v_{G,k}} \\ 0, & 0, & 0, & 0, & -\frac{v_{y,k}}{v_{G,k}^2}, & \frac{v_{x,k}}{v_{G,k}^2} \end{bmatrix} \quad (3.67)$$

The initial conditions are:

$$\begin{array}{ccc}
\begin{array}{c} \hat{x}^{(+)} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{array} & = & \begin{array}{c} \hat{x}^{(-)} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \\ \hat{a}_G \\ \hat{\dot{\psi}}_G \end{array} = \begin{array}{c} x_M \\ y_M \\ v_{G,M} \\ \psi_{G,M} \\ a_{G,M} \\ \dot{\psi}_{G,M} \end{array}
\end{array} \quad (3.68)$$

$$\begin{array}{ccc}
\begin{array}{c} \delta\hat{x}^{(+)} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \\ \delta\hat{a}_G \\ \delta\hat{\dot{\psi}}_G \end{array} & = & \begin{array}{c} \delta\hat{x}^{(-)} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \\ \delta\hat{a}_G \\ \delta\hat{\dot{\psi}}_G \end{array} = \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}
\end{array} \quad (3.69)$$

$$[P]_t^{(+)} = [P]_t^{(-)} = \begin{array}{cccccc} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_a^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 \end{array} \quad (3.70)$$

where $[K]$ is the Kalman gain matrix, $[P]$ is the estimation error covariance matrix, $[N]$ and $[Q]$ are the continuous and discrete process noise matrices, and $[R]$ is the measurement noise matrix.

3.2.3 FIS-B Filter

The number of possible state and measurement combinations that can be selected for a Kalman filter TIS-B message data is summarized in Table 3.15. In addition, a Low-Pass Filter (LPF) approach is considered for TIS data. The baseline TIS-B filter selected is the LPF based in part on the poor accuracy of the TIS-B data and on the variable, but unknown, measurement noise statistics. The measurement noise statistics cannot be modeled accurately, since the location of the radar that produced these measurements is unknown. Hence, a Kalman filter cannot be optimized to work well under all TIS-B measurement conditions. Therefore, a simpler more stable LPF is the preferred choice.

Table 3.15. TIS-B Kalman filter and Low Pass Filter (LPF) state and measurement options.

Measurement	Kalman Filter Options								LP F
	1	2	3	4	5	6a	6b	7	8
Relative north position, x_M	√	√	√					√	√

Relative east position, y_M	√	√	√					√	√
Relative raw ground speed, v_{GM}		√	√	√	√	√		√	√
Relative raw ground track angle ψ_{GM}		√	√	√	√		√	√	√
Relative raw ground acceleration, a_{GM}								√	
Relative raw ground track angle, Ψ_{GM}								√	
States									
Relative north position, x	√	√	√					√	√
Relative east position, y	√	√	√					√	√
Relative ground speed, v_G	√	√	√	√	√	√		√	√
Relative ground track angle ψ_G	√	√	√	√	√		√	√	√
Relative ground acceleration, a_G			√		√	√		√	
Relative ground track angle, Ψ_G			√		√		√	√	

The TIS-B message provides relative range and bearing to the other aircraft, as presented in Table 3.6. For convenience, these two variables can be converted into local level orthogonal coordinates as follows:

$$x_{M,k} = r_{M,k} \cos \theta_{M,k} \quad (3.71)$$

$$y_{M,k} = r_{M,k} \sin \theta_{M,k} \quad (3.72)$$

where $x_{M,k}$ and $y_{M,k}$ are the relative North and East position of other aircraft from own aircraft at time t_k and $r_{M,k}$ and $\theta_{M,k}$ are the relative range and bearing of other aircraft from own aircraft at time t_k .

In Table 3.15, raw ground speed and heading (or track angle) measurements are considered for Options 2 through 8, and raw ground accelerations and heading rate for Option 7. Raw measurements are based on numerical derivatives of the position measurements:

$$v_{GM,k} \cong \frac{\sqrt{(x_{M,k} - x_{M,k-1})^2 + (y_{M,k} - y_{M,k-1})^2}}{(t_k - t_{k-1})} \quad (3.73)$$

$$\psi_{GM,k} \cong \arctan \frac{-y_{M,k} - y_{M,k-1}}{x_{M,k} - x_{M,k-1}} \quad (3.74)$$

$$a_{GM,k} \cong \frac{v_{GM,k} - v_{GM,k-1}}{(t_k - t_{k-1})} \quad (3.75)$$

$$\dot{\psi}_{GM,k} \cong \frac{\psi_{GM,k} - \psi_{GM,k-1}}{(t_k - t_{k-1})} \quad (3.76)$$

Instead of a Kalman filter, another filter that is commonly used is the LPF. The LPF is popular due to its simplicity and stability. One of the drawbacks to using a Kalman filter is that the process noise, measurement noise, and any time constants must be selected correctly; if not, the Kalman filter will be sub-optimal and it may exhibit undesirable (unstable or biased) behavior.

The basic LPF for an estimate of a variable u based on measurements z is:

$$\hat{u}_k = (1 - \alpha)\hat{u}_{k-1} + \alpha z_k \quad (3.77)$$

where \hat{u}_k is the estimate of u , z_k is the measurement, and α is the filter gain ($0 < \alpha < 1$). If this equation is rewritten as follows:

$$\hat{u}_k = \hat{u}_{k-1} + \alpha(z_k - \hat{u}_{k-1}) \quad (3.78)$$

then equation (3.78) is very similar to the Kalman filter state update equation (3.22) with a transition and measurement matrix of unity. The updated LPF state error estimate is obtained as follows:

$$\begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{matrix} \begin{matrix} (+) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} = \begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{matrix} \begin{matrix} (-) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} + [G] \begin{matrix} -x_M \\ y_M \\ v_{G,M} \\ \psi_{G,M} \end{matrix} \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} - [H] \begin{matrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{matrix} \begin{matrix} (-) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} ? \quad (3.79)$$

The LPF updated estimate of state is then:

$$\begin{matrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{matrix} \begin{matrix} (+) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} = \begin{matrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{matrix} \begin{matrix} (-) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} + \begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{matrix} \begin{matrix} (+) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} . \quad (3.80)$$

The predicted state estimate is:

$$\begin{matrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{matrix} \begin{matrix} (-) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} \cong \begin{matrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{matrix} \begin{matrix} (+) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k-1 \end{matrix} + \Delta t \begin{matrix} -\hat{v}_G \cos\hat{\psi}_G \\ \hat{v}_G \sin\hat{\psi}_G \\ 0 \\ 0 \end{matrix} \begin{matrix} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k-1 \end{matrix} . \quad (3.81)$$

The predicted state error estimate is:

$$\begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{matrix} \begin{matrix} (-) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k \end{matrix} = [\hat{\Phi}]_{k,k-1} \begin{matrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{matrix} \begin{matrix} (+) \\ \downarrow \\ \downarrow \\ \downarrow \\ \leftarrow k-1 \end{matrix} . \quad (3.82)$$

The state transition matrix is:

$$[\hat{\Phi}]_{k,k-1} = \sum_{n=0}^{\infty} \frac{(t_k - t_{k-1})^n}{n!} [\hat{F}]_{k-1} . \quad (3.83)$$

The state matrix is:

$$\begin{bmatrix} \hat{F} \end{bmatrix}_{k-1} \dots = \begin{bmatrix} -\hat{a}_x, 0, \cos \hat{\psi}_G, -\hat{v}_G \sin \hat{\psi}_G \\ 0, -\hat{a}_y, \sin \hat{\psi}_G, \hat{v}_G \cos \hat{\psi}_G \\ 0, 0, -\hat{a}_v, 0 \\ 0, 0, 0, -\hat{a}_\psi \end{bmatrix}_{k-1} \quad (3.84)$$

The updated estimation error covariance matrix is:

$$[P]_k^{(+)} = \{[I] - [G][H]\}[P]_k^{(-)} \{[I] - [G][H]\} + [G][R][G]^T \quad (3.85)$$

The predicted estimation error covariance matrix is:

$$[P]_k^{(-)} = [\Phi]_{k,k-1} [P]_{k-1}^{(+)} [\Phi]_{k,k-1}^T + [Q]_{k-1} \quad (3.86)$$

The discrete process noise matrix is obtained as follows:

$$[Q]_k = \sum_{n=1}^{\infty} \frac{(t_k - t_{k-1})^n}{n!} [J]_n \quad (3.87)$$

$$[J]_n = [F]_{k-1} [J]_{n-1} + ([F]_{k-1} [J]_{n-1})^T \quad \text{with, } [J] = [N] \quad (3.88)$$

The continuous process noise covariance matrix is:

$$[N] = \begin{bmatrix} \sigma_{Nx}^2, 0, 0, 0 \\ 0, \sigma_{Ny}^2, 0, 0 \\ 0, 0, \sigma_{Nv}^2, 0 \\ 0, 0, 0, \sigma_{N\psi}^2 \end{bmatrix} \quad (3.89)$$

A convenient way to select the σ variables in equation (3.89) is to compute them using the initial σ variables and the corresponding reciprocal time constants [G74].

$$\sigma_{Nx} = \sigma_{x,1} \sqrt{\beta_x} \quad (3.90)$$

$$\sigma_{Ny} = \sigma_{y,1} \sqrt{\beta_y} \quad (3.91)$$

$$\sigma_{Nv} = \sigma_{v,1} \sqrt{\beta_v} \quad (3.92)$$

and

$$\sigma_{N\psi} = \sigma_{\psi,1} \sqrt{\beta_\psi} \quad (3.93)$$

The LPF gain matrix is:

$$[G] = \begin{bmatrix} \alpha_x, 0, 0, 0 \\ 0, \alpha_y, 0, 0 \\ 0, 0, \alpha_v, 0 \\ 0, 0, 0, \alpha_\psi \end{bmatrix} \quad (3.94)$$

The measurement matrix is:

$$[H] = [I] \quad (3.95)$$

The measurement noise matrix is:

$$[R] = \begin{bmatrix} \sigma_{Mx}^2, & 0, & 0, & 0 \\ 0, & \sigma_{My}^2, & 0, & 0 \\ 0, & 0, & \sigma_{Mv}^2, & 0 \\ 0, & 0, & 0, & \sigma_{M\psi}^2 \end{bmatrix} \quad (3.96)$$

where,

$$\sigma_{Mv}^2 = \frac{2(\sigma_{Mx}^2 + \sigma_{My}^2)}{(\Delta t)^2} \quad (3.97)$$

and,

$$\sigma_{M\psi}^2 = \frac{2[(y_{M,k} - y_{M,k-1})^2 \sigma_{Mx}^2 + (x_{M,k} - x_{M,k-1})^2 \sigma_{My}^2]}{[(y_{M,k} - y_{M,k-1})^2 + (x_{M,k} - x_{M,k-1})^2]} \quad (3.98)$$

The initial conditions are:

$$\begin{bmatrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{bmatrix}^{(+)} = \begin{bmatrix} -\hat{x} \\ \hat{y} \\ \hat{v}_G \\ \hat{\psi}_G \end{bmatrix}^{(-)} = \begin{bmatrix} -x_M \\ y_M \\ v_{G,M} \\ \psi_{G,M} \end{bmatrix} \quad (3.99)$$

where,

$$v_{GM,1} \equiv \frac{\sqrt{(x_{M,2} - x_{M,1})^2 + (y_{M,2} - y_{M,1})^2}}{(t_2 - t_1)} \quad (3.100)$$

and,

$$\psi_{GM,1} \equiv \arctan \frac{y_{M,2} - y_{M,1}}{x_{M,2} - x_{M,1}} \quad (3.101)$$

$$\begin{bmatrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{bmatrix}^{(+)} = \begin{bmatrix} -\delta\hat{x} \\ \delta\hat{y} \\ \delta\hat{v}_G \\ \delta\hat{\psi}_G \end{bmatrix}^{(-)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.102)$$

$$[P]_i^{(+)} = [P]_i^{(-)} = \begin{bmatrix} \sigma_x^2, & 0, & 0, & 0 \\ 0, & \sigma_x^2, & 0, & 0 \\ 0, & 0, & \sigma_v^2, & 0 \\ 0, & 0, & 0, & \sigma_\psi^2 \end{bmatrix} \quad (3.103)$$

Hence, equations (3.85) and (3.86) provide the confidence estimates for the states estimated with the LPF.

3.3 Connection with Intent Inference and Confidence Level

In Figure 3.1 was shown that the Kalman filter would provide a near-term estimate of the flight path of the other aircraft trajectory. It would also provide an estimate of the confidence of that

estimate using its continuously updated estimation error covariance matrix. In this figure was also shown an input from the near-term intent logic into this filter. This real-time adjustment of the Kalman filter will be explored in this section.

In that same figure, the output from the Kalman filter is fed into the long-term inference logic. This logic then predicts the long-term threat trajectory. While the terms near-term and long-term are subjective, they can be quantified by determining how far the TIS data can be used to predict the threat trajectory with an acceptable level of confidence. This can similarly be quantified for the ADS-B data predicted threat trajectories. Where the limit of the near-term prediction is reached for each type of threat trajectory, the long-term prediction has to take over. The confidence level to which the long-term threat trajectories can be predicted will also be explored in this section.

3.3.1 Near-Term Intent Confidence

Under Near-Term Intent (NTI), the Kalman filter determines the predicted trajectory of the other aircraft by propagating the current other aircraft state vector forward in time. The basic assumption used is that the current state vector, based on measurements performed over the recent past, is a good indicator of the future state vector of the other aircraft. If NTI information is available, such as that the aircraft has started a turn or is accelerating or decelerating, then this NTI information can be used to modify the performance of the Kalman filter.

Returning back to the general Kalman filter equations of Section 3.2.1, if the state and measurement equations for the threat trajectory are given by:

$$\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}}) \quad (3.104)$$

$$\underline{\hat{\mathbf{z}}} = h(\underline{\hat{\mathbf{x}}}) \quad (3.105)$$

then, the current estimated state of the threat trajectory is given by:

$$\underline{\hat{\mathbf{x}}}_k^{(+)} = \underline{\hat{\mathbf{x}}}_k^{(-)} + \underline{\delta\hat{\mathbf{x}}}_k^{(+)} \quad (3.106)$$

$$\underline{\delta\hat{\mathbf{x}}}_k^{(+)} = \underline{\delta\hat{\mathbf{x}}}_k^{(-)} + [K]_k \underline{\mathbf{z}}_k - [H] \underline{\hat{\mathbf{x}}}_k^{(-)} \quad (3.107)$$

where,

$$[K]_k = [P]_k^{(-)} [H] \{ [H] [P]_k^{(-)} [H]^T + [M] [R] [M]^T \}^{-1} \quad (3.108)$$

and

(3.109)

$$[P]_k^{(+)} = \{ [I] - [K]_k [H] \} [P]_k^{(-)}$$

The corresponding LPF current estimated state of the other trajectory is given by:

$$\underline{\hat{\mathbf{x}}}_k^{(+)} = \underline{\hat{\mathbf{x}}}_k^{(-)} + \underline{\delta\hat{\mathbf{x}}}_k^{(+)} \quad (3.110)$$

and,

$$\underline{\delta\hat{\mathbf{x}}}_k^{(+)} = \underline{\delta\hat{\mathbf{x}}}_k^{(-)} + [G]_k \underline{\mathbf{z}}_k - [H] \underline{\hat{\mathbf{x}}}_k^{(-)} \quad (3.111)$$

with,

$$[P]_k^{(+)} = \{ [I] - [G]_k [H] \} [P]_k^{(-)} \{ [I] - [G]_k [H] \}^T + [G]_k [R] [G]_k^T \quad (3.112)$$

Between measurements, the current estimated state is propagated forward to the next measurement time for use by either filter. If the current measurement time and the current time are not the same as illustrated in Figure 3.2, the last estimated state is propagated forward to the current time. Finally, if the trajectory is needed at a future time, the last estimated state is propagated forward to that future time.

For all three scenarios, the same equations are used, differing only in the propagation time interval:

$$\hat{\underline{x}}_k^{(-)} \cong \hat{\underline{x}}_{k-1}^{(+)} + \Delta t ? f(\hat{\underline{x}})_{k-1} \quad (3.113)$$

$$\delta \hat{\underline{x}}_k^{(-)} \cong [\Phi]_{k,k-1} \hat{\underline{x}}_{k-1}^{(+)} \quad (3.114)$$

$$[P]_k^{(-)} = [\hat{\Phi}]_{k,k-1} [P]_{k-1}^{(+)} [\hat{\Phi}]_{k,k-1} + [Q]_{k-1} \quad (3.115)$$

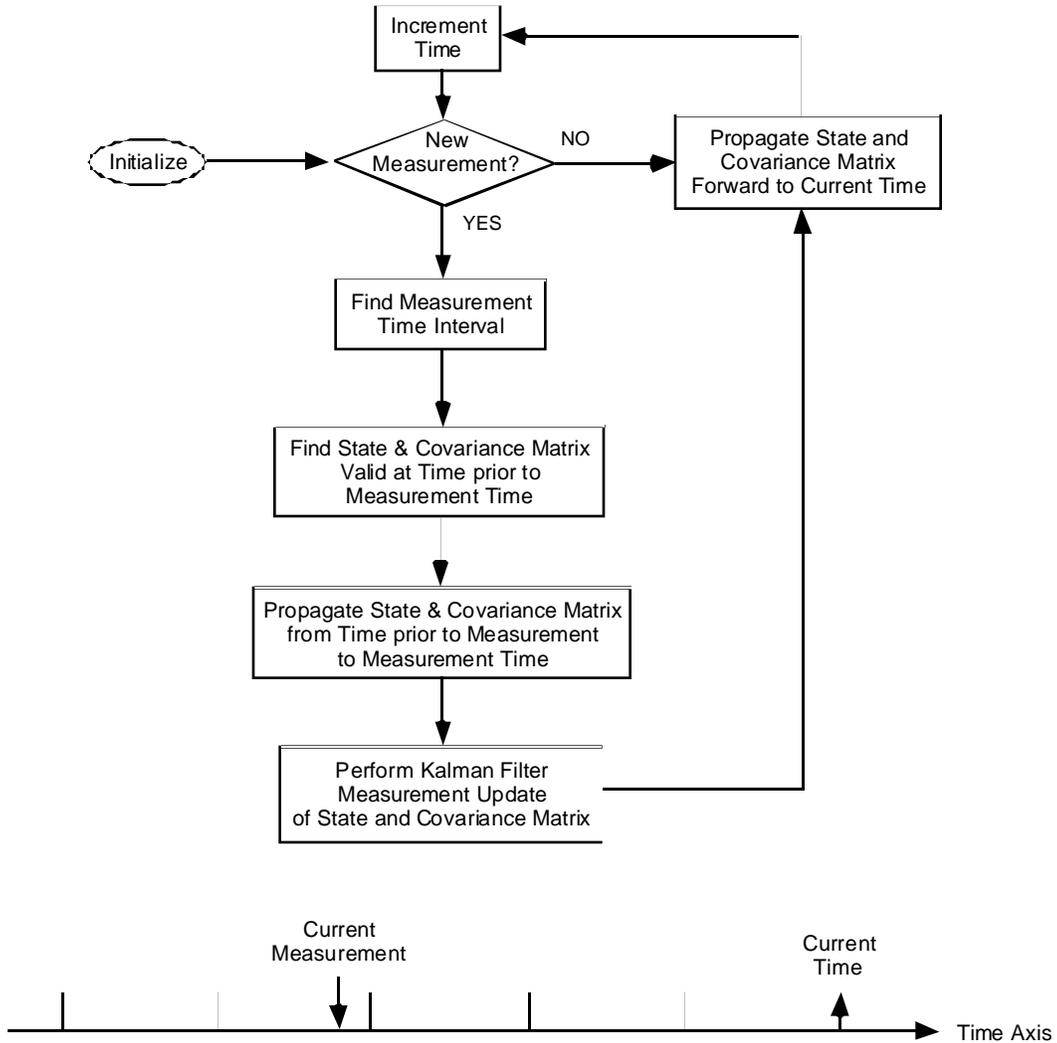


Figure 3.2. Kalman filter with latent measurement.

The measurement noise covariance matrix, $[R]$, represents the uncertainty in the measurements. The process noise covariance matrix, $[Q]$, in turn, describes the uncertainty in the future state and covariance matrix. Hence, if there is NTI information available that the other aircraft is decelerating or has started a turn, the process noise covariance matrix can be scaled to reflect this information. This will modify the estimation error covariance matrix and the Kalman filter gain matrix as follows:

$$[Q_{intent}] = [K_{intent} I Q I K_{intent}^T] \quad (3.116)$$

$$[K_{intent}] = \begin{bmatrix} k_x & 0 & 0 & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 & 0 & 0 \\ 0 & 0 & k_v & 0 & 0 & 0 \\ 0 & 0 & 0 & k_\psi & 0 & 0 \\ 0 & 0 & 0 & 0 & k_a & 0 \\ 0 & 0 & 0 & 0 & 0 & k_\delta \end{bmatrix} \quad (3.117)$$

with $k_x, k_y, k_v, k_a > 1$ if the nearby aircraft trajectory is decelerating (accelerating), $k_x, k_y, k_\psi, k_\delta > 1$ if it is turning, and $k_x, k_y, k_v, k_\psi, k_a, k_\delta = 1$ if it is maintaining its course. Equation (3.117) assumes that a 6-state Kalman filter is implemented, however, if a 4-state LPF is implemented, then the intent scaling matrix is a 4x4 matrix with the last two scale factors eliminated.

Since the estimation error covariance matrix is assumed to have a Gaussian probability density function, the diagonal elements that are the error variances can be used to derive the confidence limits of the estimated or predicted threat trajectory:

$$(\underline{\hat{x}} - n\sigma) \leq \underline{x} < (\underline{\hat{x}} + n\sigma) \quad (3.118)$$

with,
$$\sigma_i = \sqrt{P_{i,i}} \quad (3.119)$$

where σ_i is the estimation error standard deviation, and n=1 is the 68.3% confidence level, n=2 is the 95.4% confidence level, and n=3 is the 99.7% confidence level. Hence, equation (3.119) states that when n = 2, for example, the actual (but unknown) state of the other aircraft is located within the limits shown with a 95% confidence level (probability).

3.3.2 Long-Term Intent Confidence

Under the Long-Term Intent (LTI), the focus is more on whether the current estimated other aircraft state vector is close to an LTI trajectory for this aircraft or whether it will be close to this trajectory in the future. To quantify this approach, let us focus on the LTI of the other aircraft, as indicated by its last and current waypoint. Alternately, if the current course bearing of the other aircraft and the next waypoint is known, these can be used instead. Note that the use of these two waypoints can be generalized to handle cases where a weather front with storm cells or a hazardous (no fly) region is identified in front of the other aircraft. In that case, the current position of the aircraft or its last waypoint can be taken as one of the waypoints and a second waypoint can be selected such that the other aircraft will avoid the undesirable flight region as specified by the AOP intent inference module and/or the hazard detection and resolution module.

Two indicators locate an aircraft on a nominal LTI trajectory, as illustrated in Figure 3.3. First, the current position of the aircraft is expected to be close to the nominal LTI trajectory. Second, the ground track angle of the other aircraft is expected to be pointed toward the current waypoint of the nominal LTI trajectory. Figure 3.3 illustrates a situation where the last, current, and next waypoint are known. Hence there are at least two possible nominal LTI trajectories. The first LTI trajectory is defined by the last waypoint (n-1) and current waypoint (n). The second LTI trajectory is defined by the last (n-1) and next waypoint (n+1). The latter LTI trajectory reflects a situation where the other aircraft may have chosen or been directed to bypass the next waypoint.

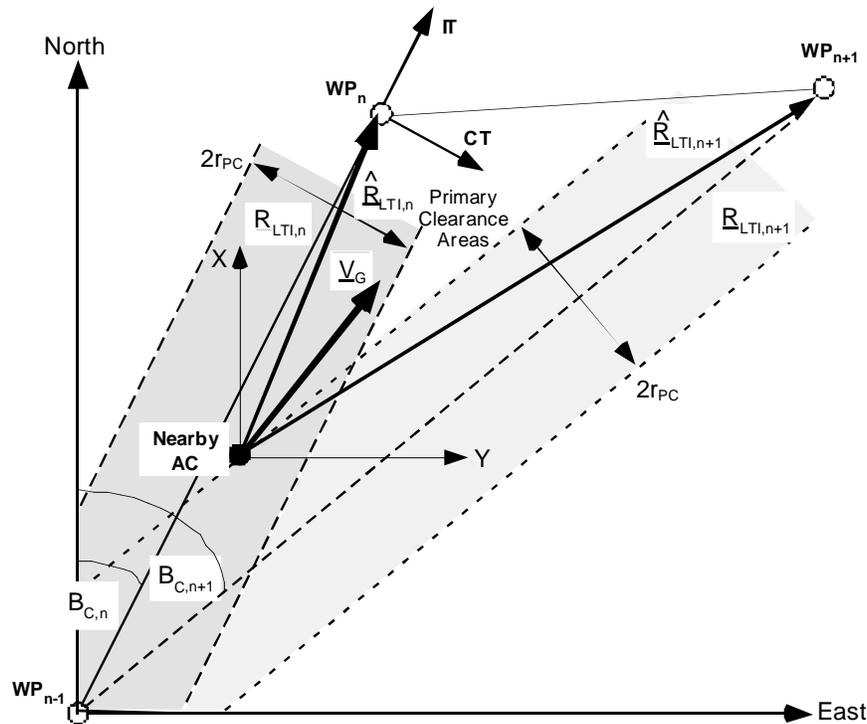


Figure 3.3. Alternate LTI trajectories.

Shown in Figure 3.3 is a band about the nominal LTI trajectory called the Primary Clearance Region (PCA). Current jetways and air routes have primary and secondary clearance regions where it is safe to fly without encountering any terrain or other hazard. For Free Flight, other criteria may be used to define these regions, such as the Required Navigation Performance (RNP) which describes the minimum lateral wander that an aircraft can make from its route.

Figure 3.4 illustrates the key parameters used to determine the probability that the other aircraft is following the nominal LTI trajectory. Figure 3.4 shows the position of the nearby aircraft in an in-track-crosstrack coordinate system, as defined by the last and current waypoint and the current position of the aircraft. The current ground speed vector has a track angle, Ψ_G , that can vary between Ψ_{min} and Ψ_{max} and still allow the nearby aircraft to pass within $\pm r_{PCA}$ of the current waypoint.

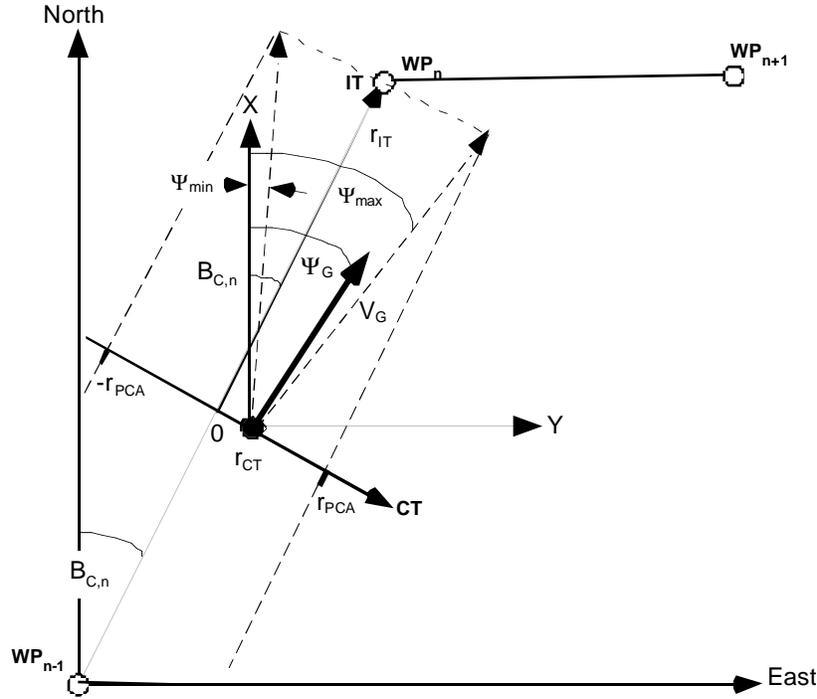


Figure 3.4. Nearby aircraft ground speed geometry relative to LTI trajectory.

To derive the probability that the nearby aircraft is on its nominal LTI trajectory, we start with the observed LTI trajectory. This trajectory is defined by the current position of the other aircraft and its current waypoint. The observed LTI trajectory in intrack-crosstrack coordinate position is

$$\begin{bmatrix} \hat{r}_{IT} \\ \hat{r}_{CT} \end{bmatrix} \sqrt{\downarrow} = \begin{bmatrix} \cos B_C & \sin B_C \\ -\sin B_C & \cos B_C \end{bmatrix} \begin{bmatrix} x_{WP} - \hat{x} \\ y_{WP} - \hat{y} \end{bmatrix} \sqrt{\downarrow} \quad (3.120)$$

where,

$$\begin{bmatrix} \cos B_C & \sin B_C \\ -\sin B_C & \cos B_C \end{bmatrix} \quad (3.121)$$

The relationship between the North-East and intrack-crosstrack coordinate system is illustrated in Figure 3.5. This figure illustrates the course bearing angle B_C , the true bearing angle B_T , the true track angle Ψ_G , and the ground speed V_G . This figure also illustrates the nominal LTI trajectory, represented by \underline{R}_{LTI} , and the observed LTI trajectory, represented by \underline{R}_{LTI} .

To determine the confidence level that the other aircraft is moving toward the current waypoint, it is convenient to focus on the track angle of the other aircraft. The first step is to rotate the 4x4 estimation error covariance matrix into lateral coordinates:

$$[P_{ITCT}] = [T_{LAT}] [P] [T_{LAT}] \quad (3.122)$$

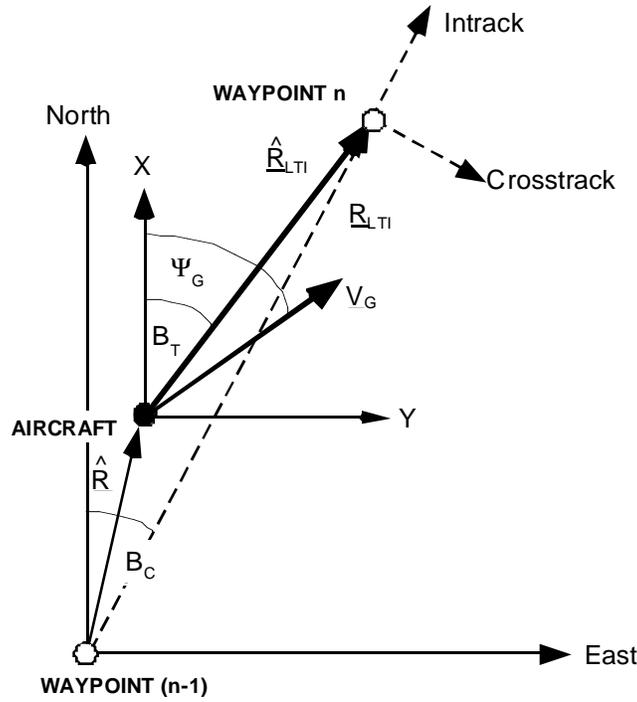


Figure 3.5. The local coordinate systems.

where,

$$[P_{ITCT}] = \begin{bmatrix} [P_{IT,CT}] & [C_{IT,CT;V_G,\Psi_G}] \\ [C_{IT,CT;V_G,\Psi_G}] & [P_{V_G,\Psi_G}] \end{bmatrix} \quad (3.123)$$

$$[P] = \begin{bmatrix} [P_{xy}] & [C_{xy,\Psi_V}] \\ [C_{xy,\Psi_V}] & [P_{\Psi_V}] \end{bmatrix} \quad (3.124)$$

$$[T_{LAT}] \dots \begin{bmatrix} [T_{B_C}] & [0] \\ [0] & [I] \end{bmatrix} \quad (3.125)$$

where $[P]$ is the Kalman filter or LPF estimation error covariance matrix and $[P_{ITCT}]$ is the Kalman filter estimation error covariance matrix for the intrack-crosstrack position and for the ground speed and track angle.

The cross-track position standard deviation, track angle standard deviation, and cross-track position-track angle correlation coefficient can be obtained from equations (3.123), as follows:

$$\sigma_{r_{CT}} = \sqrt{\rho_{r_{CT},r_{CT}}} \quad (3.126)$$

$$\sigma_{\Psi_G} = \sqrt{\rho_{\Psi_G,\Psi_G}} \quad (3.127)$$

and,

$$\rho_{r_{CT},\Psi_G} = \frac{-C_{r_{CT},\Psi_G}}{\sigma_{r_{CT}} \sigma_{\Psi_G}} \quad (3.128)$$

Then with equations (3.126) – (3.128), the total probability that the other aircraft is following the nominal LTI trajectory is obtained as follows:

$$p_{LTI,Total} = \frac{1}{2\pi\sigma_{r_{CT}}\sigma_{\psi_G}} \int_{-r_{PCA}}^{r_{PCA}} \int_{\psi_{min}}^{\psi_{max}} \frac{1}{2\sqrt{1-\rho_{r_{CT},\psi_G}^2}} \frac{1}{\sigma_{r_{CT}}^2} \frac{1}{\sigma_{r_{CT}}\sigma_{\psi_G}} \frac{1}{\sigma_{\psi_G}^2} e^{-\frac{(r-\hat{r}_{CT})^2}{2\sigma_{r_{CT}}^2} - \frac{(\psi-\hat{\psi}_G)^2}{2\sigma_{\psi_G}^2} - \frac{2\rho_{r_{CT},\psi_G}(r-\hat{r}_{CT})(\psi-\hat{\psi}_G)}{2\sigma_{r_{CT}}\sigma_{\psi_G}}} dr d\psi e \quad (3.129)$$

The integration limit, r_{PCA} , is the crosstrack radius of the primary clearance area around the LTI trajectory, as illustrated in Figure 3.4. The crosstrack position integral of equation (3.129) is illustrated in Figure 3.5.

The ground track integration limits, are computed as follows:

$$\psi_{min}(r_{CT}, r_{IT}) = B_C - \arctan \frac{-r_{PCA} + r_{CT}}{r_{IT}} \quad (3.130)$$

and,

$$\psi_{max}(r_{CT}, r_{IT}) = B_C + \arctan \frac{-r_{PCA} - r_{CT}}{r_{IT}} \quad (3.131)$$

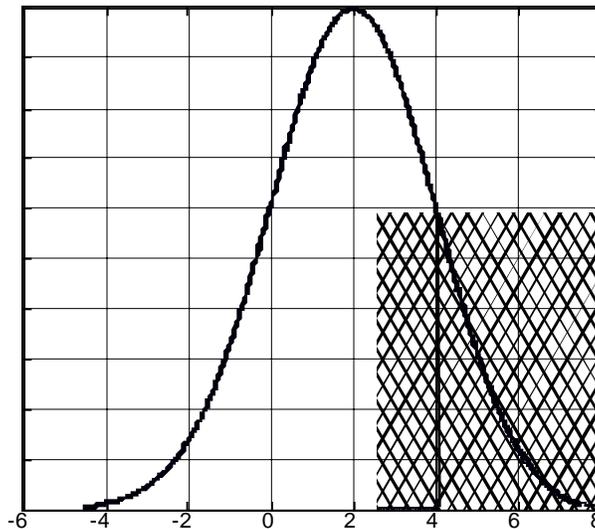


Figure 3.6. Probability that nearby aircraft is on LTI trajectory based on crosstrack position.

Equation (3.129) is difficult equation to solve since it is a coupled double integral but also because the track angle integration limits involve the crosstrack position, which is one of the integration variables. A reasonable simplification is to set the track angle integration limits equal to the estimated crosstrack position. This assumes that the standard deviation of the crosstrack position is small relative to the primary clearance area radius, r_{PCA} :

$$\psi_{\min}(\hat{r}_{IT}, \hat{r}_{CT}) = B_C - \arctan \frac{-r_{PCA} + \hat{r}_{CT}}{\hat{r}_{IT}} \quad (3.132)$$

and,

$$\psi_{\max}(\hat{r}_{IT}, \hat{r}_{CT}) = B_C + \arctan \frac{-r_{PCA} - \hat{r}_{CT}}{\hat{r}_{IT}} \quad (3.133)$$

Then equation (3.129) is evaluated using the limits defined by equations (3.132) and (3.133). A further simplification in the equation (3.129) is achieved since the correlation coefficient in equation (3.129) is negligible (close to zero). With equations (3.132) – (3.133), the double integral in equation (3.129) takes the form:

$$\rho_{LTI, Total} \cong \frac{1}{2\pi\sigma_{r_{CT}}\sigma_{\psi_G}} \int_{-r_{PCA}}^{r_{PCA}} e^{-\frac{(r-\hat{r}_{CT})^2}{2\sigma_{r_{CT}}^2}} dr \int_{\psi_{\min}}^{\psi_{\max}} e^{-\frac{(\psi-\hat{\psi}_G)^2}{\sigma_{\psi_G}^2}} d\psi \quad (3.134)$$

With equation (3.134), only the product of two separate integrals need to be evaluated. Expressed in terms of the error function $erf()$, equation (3.134) can be rewritten as follows:

$$\rho_{LTI, Total} = 0.25 \operatorname{erf} \frac{-r_{PC} + \hat{r}_D}{\sigma_{r_D} \sqrt{2}} + \operatorname{erf} \frac{-r_{PC} - \hat{r}_D}{\sigma_{r_D} \sqrt{2}} \operatorname{erf} \frac{\psi_{\max} - \hat{\psi}_G}{\sigma_{\psi_D} \sqrt{2}} - \operatorname{erf} \frac{\psi_{\min} - \hat{\psi}_G}{\sigma_{\psi_D} \sqrt{2}} \quad (3.135)$$

where,

$$\operatorname{erf}(z) \dots \frac{2}{\sqrt{\pi}} \int_0^z e^{-u^2} du \quad (3.136)$$

or,

$$\operatorname{erf}(z) = \frac{2z}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n}}{n! (2n+1)} \quad (3.137)$$

3.3.3 Confidence Assessment for other Data

The confidence assessment module assesses the confidence level of weather and other data. These data may be of the type of point data, region data (e.g, polygon regions), or grid data. Confidence assessment of these data assigns a number in the interval [0,1] to the data to indicate low (0) to high (1) level of confidence. Confidence assessment is performed independent of the data type and data format. Two methods of confidence assessment are designed for this purpose, discrete and fading memory. A discussion of these two confidence assessment methods follows.

3.3.3.1 Discrete Confidence Assessment

With discrete confidence assessment, the data confidence is assessed to be one of a limited number of discrete values. The simplest form of discrete confidence assessment is binary, where the confidence is either 0% (0) or 100% (1). A particular data type X is assigned a confidence assessment C within the set:

$$C(X) - \{0, 1\} \quad (3.138)$$

where the assignment may depend on any heuristic or domain knowledge. Yet another discrete confidence assessment may assign 100% (1), 90% (0.9), 50% (0.5), 10% (0.1), or 0% (0) confidence. Thus, in general, discrete confidence assessment assigns a discrete set of certainty values:

$$C(X) - \{C_0, C_1, C_2, \dots, C_n\} \quad (3.139)$$

based on a set of heuristic factors associated with the data. Discrete confidence assessment methods are used in AOP as the simplest form of confidence assessment or as a default condition

when more detailed or more accurate methods of confidence assessment are not available.

Consider an example of assigning confidence assessment for a turbulence SIGMET from the FIS message. Based on the FIS data item for likelihood (Low, Moderate, High), a direct correspondence can be established to form the confidence assessment:

$$C(X) = \{C_0 = 0 \text{ (no SIGMET)}, \\ C_1 = 0.5 \text{ (Low Likelihood)}, \\ C_2 = 0.7 \text{ (Moderate Likelihood)}, \\ C_3 = 0.9 \text{ (High Likelihood)}\}. \quad (3.140)$$

The confidence assessment level for discrete confidence assessment can remain fixed over the duration of valid time of the data (until a new data replaces old data, or the data expires as indicated by the data lifespan) or it may change with time, as with the fading memory confidence assessment method described next.

3.3.3.2 Fading Memory Confidence Assessment

With fading memory confidence assessment, each data type has a time constant τ associated with data collection, and as the time t passes, the confidence of the information decays accordingly with the associated time constant. The initial confidence assessment level can start at 1 or some other value dependent on a heuristic (as described above in Section 3.3.3.1). In general, if a particular data type $X(x,y)$ is collected at time T_0 , then we denote the data and time stamp by $X(x,y,T_0)$ and assign the confidence assessment C for the data X at the time t by an exponentially decaying function, as shown in Figure 3.7.

The time-decay confidence assessment model can apply to any data type or any data format. Thus, this time-decay confidence assessment model is useful as a default, provided a time decay constant τ is available.

3.3.3.3 Combining Confidence Assessment Levels in Data Fusion

There are several methods available to combine confidence assessment levels in data fusion. Among the methods are: Bayesian Theory, membership for Fuzzy Set Theory, possibility distributions, certainty factors in MYCIN-like systems, and mass, belief, and plausibility functions from Dempster-Shafer Evidence Theory. A review and comparison of these methods is performed by Bloch [B196].

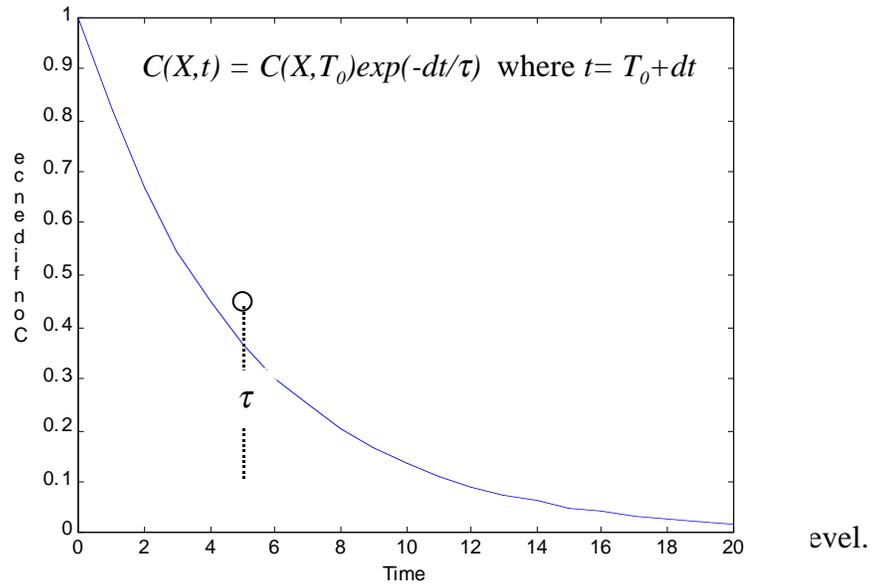


Figure 3.7. Ec

The MYCIN-like method for assigning certainty factors is implemented in RTO 30 to combine confidence assessment levels in the data fusion process. Fusion based on certainty factors [B196] is motivated by the early expert systems research. Here, certainty factors are defined on the interval $[-1,1]$ where positive indicates that the fused information supports the Assertion A and negative indicates that the fused information denies the Assertion A . [Note: a simple transformation converts the interval $[0,1]$ to $[-1,1]$ between the confidence assessment levels and the MYCIN certainty factor levels.] When information x and y are combined:

$$a) \ x + y - xy \quad \text{if } x \neq 0 \text{ and } y \neq 0 \text{ (} x \text{ and } y \text{ confirm } A) \quad (3.141)$$

$$b) \ x + y + xy \quad \text{if } x \leq 0 \text{ and } y \leq 0 \text{ (} x \text{ and } y \text{ deny } A) \quad (3.142)$$

$$c) \ x + y \quad \text{if } x \leq 0 \text{ and } y \neq 0, \text{ or } x \neq 0 \text{ and } y \leq 0 \text{ (} x \text{ and } y \text{ show mixed support for } A) \quad (3.143)$$

This classical method of combining certainty factors for data may be necessary for confidence assessment of combined data sources such as weather, multiple hazards, or airspace conflicts. With weather data in particular, the FIS message allows for individual pixels within a precipitation map graphic to indicate no data or lost data, for which the confidence level is assigned to be 0. Another weather map, for instance, the onboard weather map from the onboard weather radar, may have a reading for that pixel location, and the certainty factor method of equation (3.180) is applicable.

The MYCIN like operator for confidence assessment is a context independent variable behavior type operator [B196]. The operator does not depend on the data type: point, region, or grid. The operator is variable behavior because the outcome depends on the input conditions for x and y (confirm, deny, or mixed support). This confidence assessment has the properties:

- when x and y are positive, the operator confirms the event more than either of the individual confidence levels (as in Figure 3.8),
- when x and y are negative, the operator rejects the event more than either of the individual confidence levels, and
- in a mixed compromise when x and y are mixed signs, the sign of the result depends on the stronger confidence level.

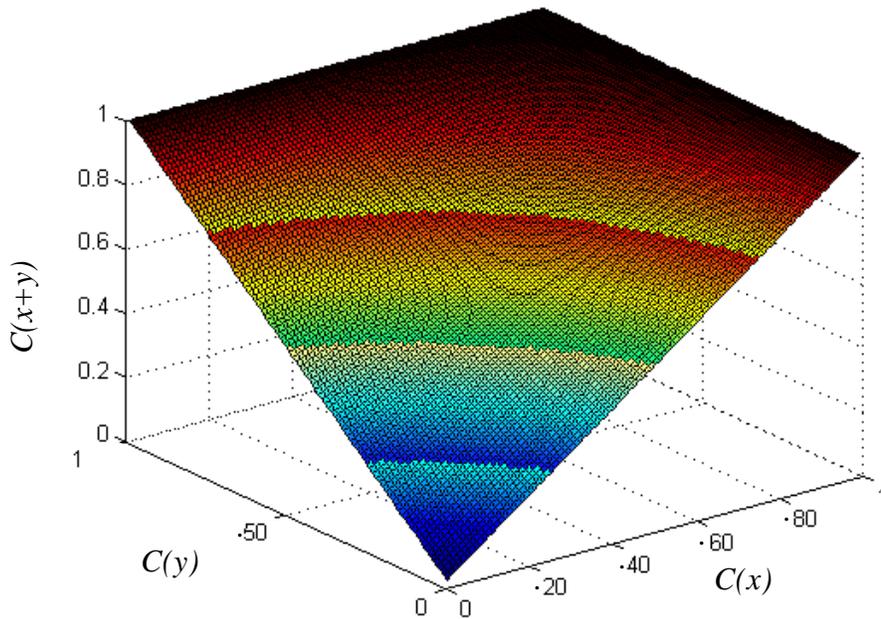


Figure 3.8. The MYCIN confidence assessment operator confirms an event more than either individual confidence levels that combine in data fusion.

3.4 Scenarios

The two filters presented in Section 3.2 are investigated further in this section using two simulations with several aircraft trajectories. For the ADS-B trajectory, a Monte Carlo simulation is used which includes aircraft lateral FTE and GPS navigation errors. Initially, the nearby aircraft trajectory is only being investigated, not the relative trajectory. Since the aircraft trajectory is based on a simulation, the true (GPS error-free) trajectory is also available. Actual radar track data for two aircraft are used to represent the TIS-B data. The radar error-free trajectory is obtained by using a moving window least squares filter on the radar position data to determine the true ground speed, heading, ground acceleration, and heading rate.

3.2.1 ADS-B Scenario with Kalman Filter

A Monte Carlo simulated GPS/INS aircraft trajectory is used as the basis for the ADS-B analysis, as illustrated in Figure 3.9. The statistics and constants that were used in the simulation equations are summarized in Table 3.16. Note that all the units in Table 3.16 were standardized within the simulation to distance in feet, speed in feet/sec, acceleration in feet/sec², angles in radians, angular rates in radians/sec, and time in seconds.

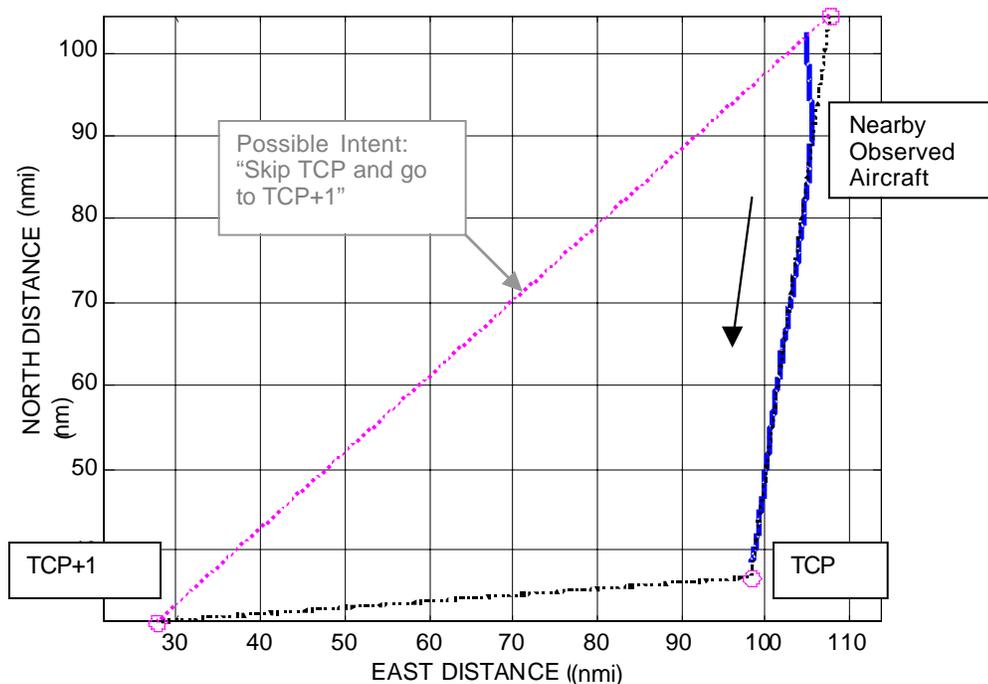


Figure 3.9. Plan view of aircraft path, ADS-B intent TCP and TCP+1, and alternate intent.

The trajectory variables are illustrated in Figure 3.9. The Kalman filter state estimates are presented in Figures 3.10 through 3.16. The NTI predicted position estimates are presented in Figure 3.17. The LTI estimated position and track angle statistics are presented in Figure 3.18. Finally, Figure 3.19 presents the probability that the other aircraft is on the nominal LTI trajectory.

In Figures 3.10 - 3.16, the relative error between the Kalman filter estimate and truth is presented. This error is the actual estimation error of the Kalman filter. The uncertainty bound around this estimation error is the 95% confidence limit that the unknown actual history is found within $\pm 2\sigma$ of the estimate. Hence, if this confidence estimate is computed correctly, one would expect most (19 out of 20) of the actual estimation error values to lie within this confidence interval.

Figure 3.17 illustrates the NTI aircraft position prediction capability of the Kalman filter. These predicted positions were obtained by propagating the current Kalman filter estimate and estimation error covariance matrix, without further measurements, 60 seconds in the future. The predicted positions were then compared to the actual positions to obtain the prediction error. The error bound surrounding the prediction error is the 95% confidence interval about the prediction estimate. This confidence interval is based on the standard deviation of the prediction as obtained from the predicted estimation error covariance matrix. As can be seen, the 95% confidence levels are reasonable, particularly for the East position prediction.

The discrete increases in the confidence limits shown in these Figures reflects the use of NTI data by the Kalman filter that the aircraft is turning or accelerating. Since no NTI data was available from the ADS-B message or the NTI inference algorithm, the use of NTI data was simulated. Specifically whenever the other aircraft absolute track rate or acceleration exceeded a specified value, the scaling on the process noise matrix of the Kalman filter was changed. In Figure 3.18, the bands shown around the mean cross-track position and mean track angle represent the respective integration limits. A primary clearance area radius of 1 nmi was used instead of the current jetway radius of 4 nmi.

Table 3.16. The parameters used in the ADS-B Kalman filter simulation.

Parameter	Units	Symbol	Value
Initial North Position Sigma	nmi	σ_x	0.2
Initial East Position Sigma	nmi	σ_y	0.2
Initial Ground Speed Sigma	kts	σ_{vg}	0.1
Initial Track Sigma	deg	$\sigma_{\psi g}$	5
Initial Ground Acceleration Sigma	kts/sec	σ_{Ag}	0.1
Initial Track Rate Sigma	deg/sec	$\sigma_{\dot{\psi} g}$	0.2
GPS North Position Measurement Noise Sigma	m	σ_{MX}	0.05
GPS East Position Measurement Noise Sigma	m	σ_{MY}	0.05
GPS North Velocity Measurement Noise Sigma	m/s	σ_{MVx}	0.5
GPS East Position Measurement Noise Sigma	m/s	σ_{MVy}	0.5
North Position Reciprocal Time Constant	1/sec	β_x	0.33
East Position Reciprocal Time Constant	1/sec	β_y	0.33
Ground Speed Reciprocal Time Constant	1/sec	β_{vg}	0.56
Ground Track Reciprocal Time Constant	1/sec	$\beta_{\psi g}$	1
Ground Acceleration Reciprocal Time Constant	1/sec	$\beta_{\dot{v} g}$	0.56
Ground Track Rate Reciprocal Time Constant	1/sec	$\beta_{\dot{\psi} g}$	0.56
Primary Clearance Area Radius	nmi	r_{PCA}	1
Computation and Measurement Time Interval	sec	Δt	1
NTI North Position Process Noise Scale Factor		k_x	1 or 2.5
NTI East Position Process Noise Scale Factor		k_y	1 or 2.5
NTI Ground Speed Process Noise Scale Factor		k_v	1 or 2.5
NTI Ground Track Process Noise Scale Factor		k_w	1 or 5
NTI Ground Acceleration Process Noise Scale Factor		k_a	1 or 2.5
NTI Ground Track Rate Process Noise Scale Factor		k_w	1 or 2.5
NTI Prediction Time Interval	sec	Δt_{PRED}	60

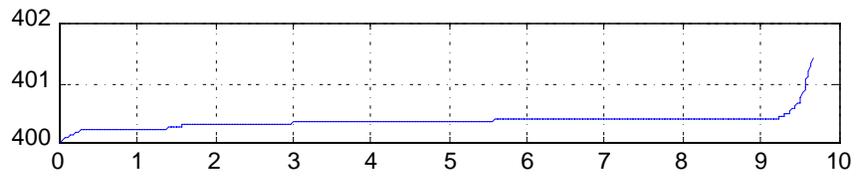
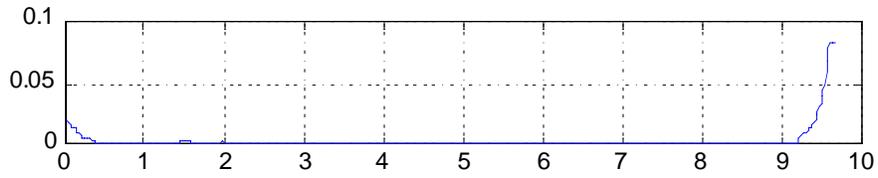
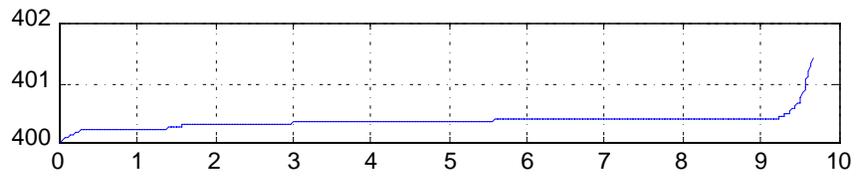


Figure 3.10. The ADS-B aircraft trajectory truth data.



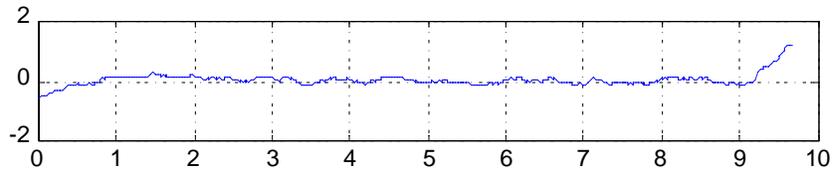
rms = 0.004 kts/sec

Figure 3.11. The Kalman filter ground acceleration estimate.



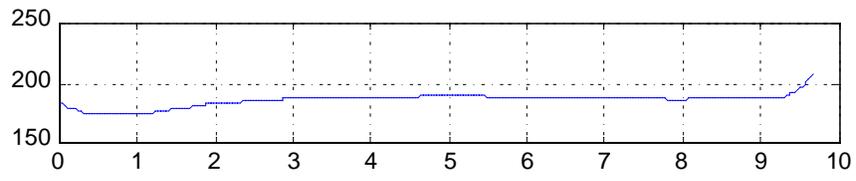
rms = .025 kts

Figure 3.12. The Kalman filter ground speed estimate.



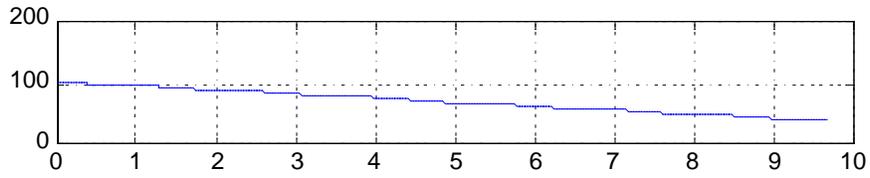
rms = 0.025 deg/sec

Figure 3.13. The Kalman filter track rate estimate.



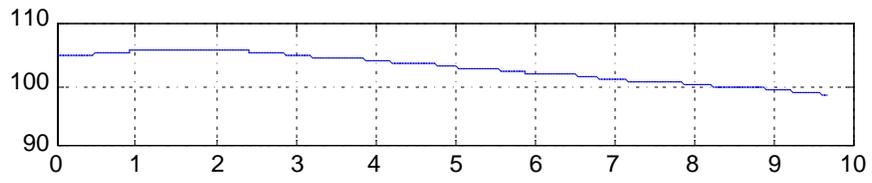
rms = 0.11 deg

Figure 3.14. The Kalman filter track angle estimate.



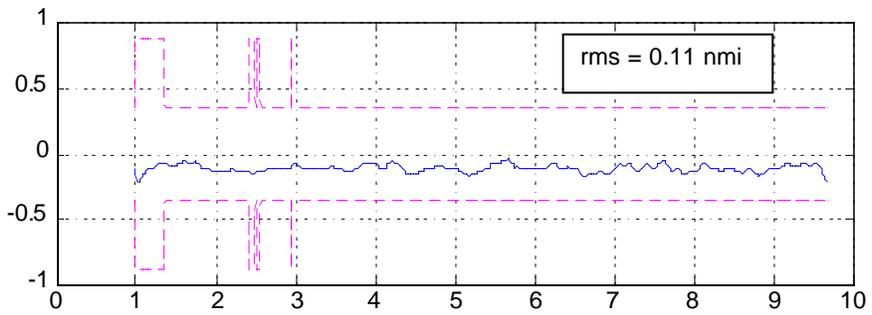
rms = 0.002 nmi

Figure 3.15. The Kalman filter North position estimate.



rms = 0.012 nmi

Figure 3.16. The Kalman filter East position estimate.



rms = 0.11 nmi

rms = 0.30 nmi

Figure 3.17. The Kalman filter 60 second NTI position prediction errors.

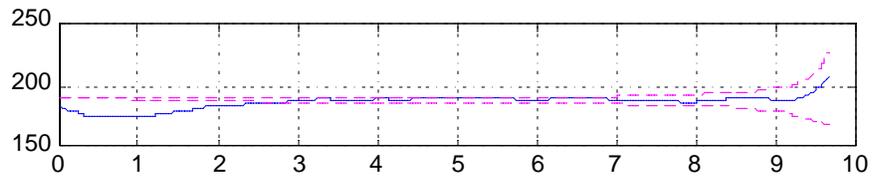


Figure 3.18. The Kalman filter crosstrack position and track angle LTI statistics.

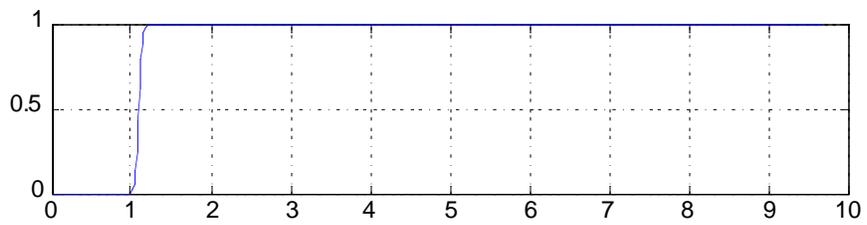


Figure 3.19. The Kalman filter LTI probabilities.

3.2.2 TIS-B Scenario with LPF

The radar track data for the two aircraft that were used to evaluate the LPF with TIS-B data is illustrated in Figure 3.20. The own aircraft is moving from the upper left to the South while the other aircraft is moving from the left to the East. Figure 3.21 shows the relative trajectory for the same two trajectories of Figure 3.20. The raw and truth variables for the relative trajectory are presented in Figures 3.22 and 3.23. As can be seen in Figure 3.22, the two aircraft will nominally pass within about 10 nmi of each other.

The simulation was based on the LPF with TIS-B data and the parameters shown in Table 3.17. All units were standardized to distance in feet, speed in feet/sec, acceleration in feet/sec², angles in radians, angular rates in radians/sec, and time in seconds.

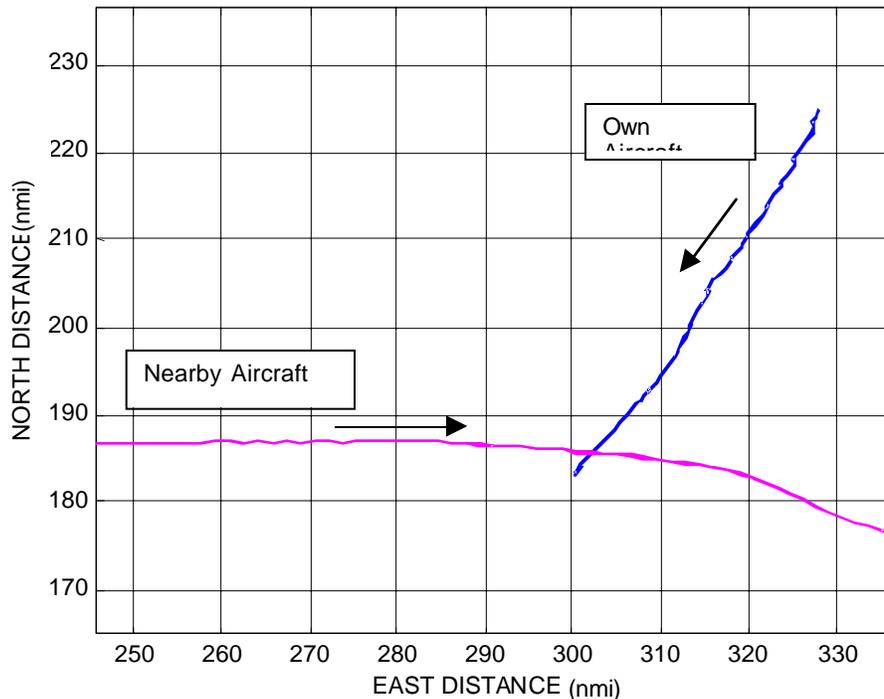


Figure 3.20. Flight paths for two aircraft in close proximity.

Figures 3.23 through 3.27 illustrate how accurately the LPF estimates the relative current ground speed, track angle, North position, and East position of the two aircraft. In Figures 3.24 through 3.27, are presented the actual LPF estimation errors, shown in the panels labeled relative. These errors are bounded by 95% confidence limits that are obtained by using the ± 2 sigma limits. Figure 3.28 then shows the prediction error when the current Kalman filter and LPF position state estimate is propagated forward into the future by 1 minute, without any additional measurements, and this estimate is compared to the true position. Finally, Figure 3.29 presents the fixed LPF gain values and compares these to the dynamically set Kalman filter gain values.

The discrete increases in the sigma values reflect the use of NTI information, indicating that the aircraft is turning or accelerating. Since no NTI inference data was available, the simulation triggered the use of NTI data in the LPF whenever the absolute value of the heading rate or the acceleration exceeded a specified value. The excessive triggering of the NTI data reflects the noisiness of the data used to implement this simple NTI logic.

Table 3.17. The parameters used for the TIS-B Low Pass Filter (LPF) simulation.

Parameter	Units	Symbol	Value
Initial North Position Sigma	nmi	σ_x	0.16
Initial East Position Sigma	nmi	σ_y	0.16
Initial Ground Speed Sigma	kts	σ_{vg}	20
Initial Track Sigma	deg	$\sigma_{v\theta}$	7.5
Radar North Position Measurement Noise Sigma	nmi	σ_{MX}	0.2
Radar East Position Measurement Noise Sigma	nmi	σ_{MY}	0.2
North Position Reciprocal Time Constant	1/sec	β_x	0.083
East Position Reciprocal Time Constant	1/sec	β_y	0.083
Ground Speed Reciprocal Time Constant	1/sec	β_{vg}	0.083
Ground Track Reciprocal Time Constant	1/sec	$\beta_{v\theta}$	0.083
North Position LPF Gain		α_x	0.25
East Position LPF Gain		α_y	0.25
Ground Speed LPF Gain		α_{vg}	0.10
Ground Track LPF Gain		$\alpha_{v\theta}$	0.10
Computation and Measurement Time Interval	sec	Δt	12
NTI North Position Process Noise Scale Factor		k_x	1 or 4
NTI East Position Process Noise Scale Factor		k_y	1 or 4
NTI Ground Speed Process Noise Scale Factor		k_v	1 or 4
NTI Ground Track Process Noise Scale Factor		$k_{v\theta}$	1 or 4
NTI Prediction Time Interval	sec	Δt_{PRED}	60

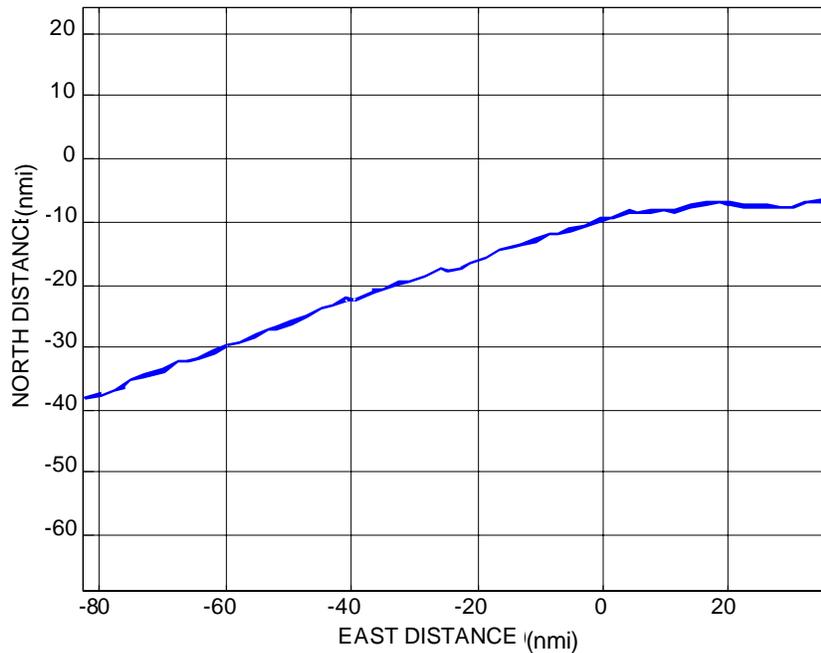


Figure 3.21. The relative path for the two aircraft.

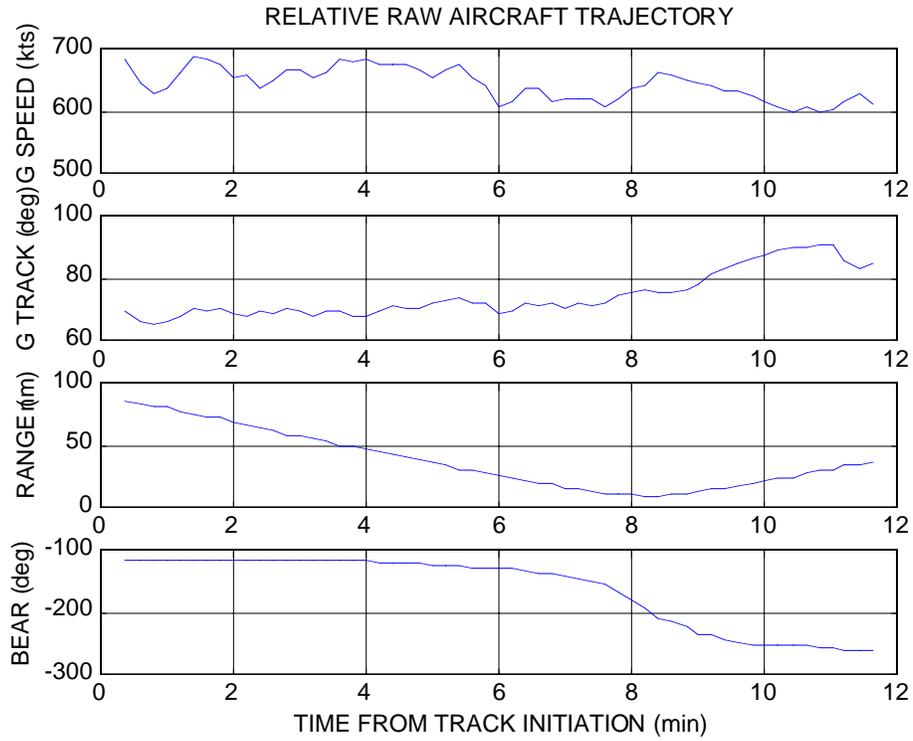


Figure 3.22. The relative raw trajectory variables.

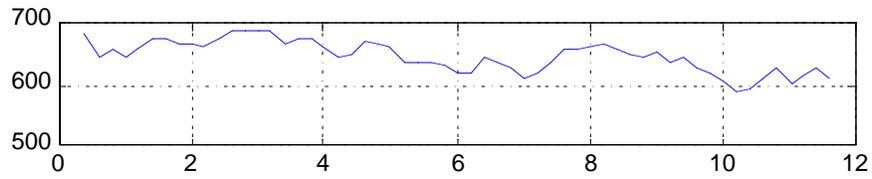
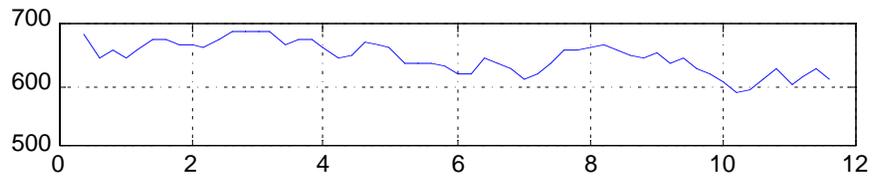
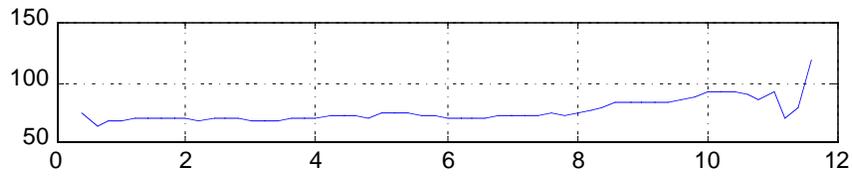


Figure 3.23. The relative trajectory variables.



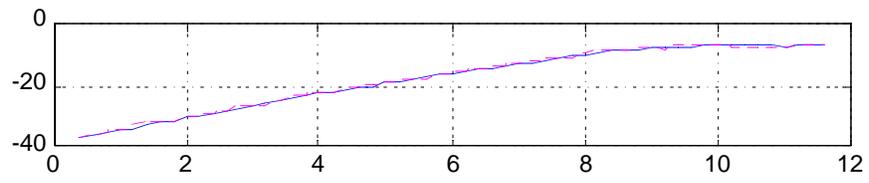
rms = 20.2 kts

Figure 3.24. The LPF ground speed statistics.



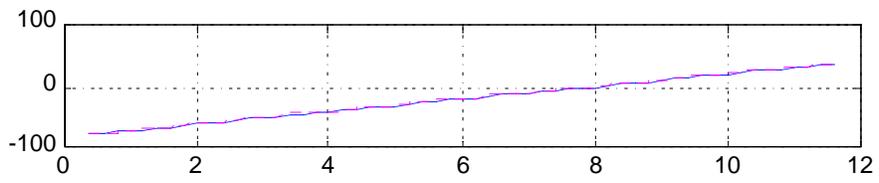
rms = 5.6
deg

Figure 3.25. The LPF ground track statistics.



rms = 0.25 nmi

Figure 3.26. The LPF North position statistics.



rms = 0.19 nmi

Figure 3.27. The LPF East position statistics.

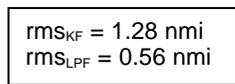
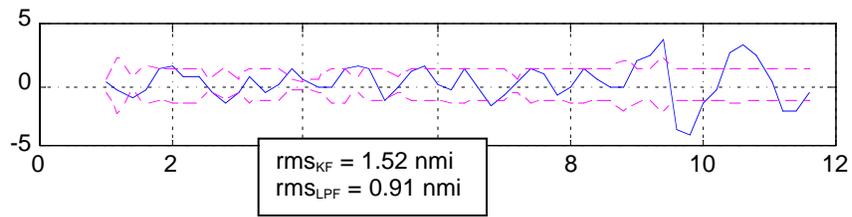


Figure 3.28. The LPF and Kalman filter 60 second position prediction statistics.

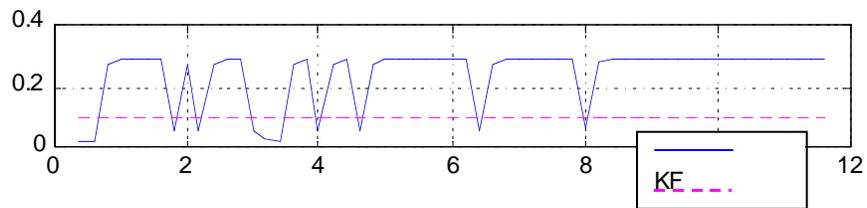


Figure 3.29. The LPF and Kalman filter gain values.

3.4.4 Lessons Learned from Scenario Investigations

In these scenarios, a number of lessons were learned. A number of different Kalman filters were explored to estimate the state of the other aircraft based on ADS-B data, and the 6-state Kalman filter with 6-measurement states provided the best performance. The focus was initially on the horizontal dynamics of the other aircraft. The 6 states and measurements consisted of North and East position, ground speed and track angle, and ground acceleration and track angle rate. Since the ADS-B data does not include measurements of the ground acceleration and track angle rate, these were obtained by numerically differentiating the ground speed and track angle measurements.

For the TIS-B data, a number of Kalman filters and a LPF were investigated for determining the estimated relative state of both aircraft. A 4-state LPF with 4-measurement states provided the best performance for estimating the horizontal relative states of both aircraft. The performance of the LPF was found to be better overall than the Kalman filter since the LPF is less sensitive to inaccuracies in the knowledge of the measurement and process noise statistics than the Kalman filter. The four states and measurements are the North and East relative position and the relative ground speed and track angle. Since the TIS-B does not provide measurements of the relative ground speed and track angle, these were computed by numerically differentiating the position measurement data.

NTI confidence was based on propagating the current best state estimate, and the corresponding estimation error covariance matrix, forward into the near future (e.g.: 30-60 sec). This assumes that the current state estimate, based primarily on measurements of the recent past trajectory, is a good indicator of the near-term future.

The current state estimate has a confidence level that is defined by the standard deviation of the estimation error. This estimation error is obtained from the dynamically computed estimation error covariance matrix. When the current state is propagated forward under NTI without any additional measurements, the estimation error covariance matrix increases through the addition of the process noise matrix. In other words, the process noise matrix provides an estimate of the future uncertainty of the state estimate.

When NTI information is available such as that the other aircraft is turning or accelerating, this information can be used to scale the process noise matrix to reflect the increased uncertainty of the future state of the aircraft. Such NTI information would come directly from the ADS-B message, while NTI acceleration information is not broadcast.

The LTI confidence estimate of the other aircraft can be derived by focusing on the nominal and current LTI trajectory of the other aircraft. By comparing the nominal to the current observed LTI of the other aircraft, the probability (confidence level) that the other aircraft is following its nominal LTI trajectory can be derived. This process is computed in the Information Ambiguity Resolution module, which verifies the LTI. If the LTI cannot be verified or does not exist, then the Intent Inference module will estimate the LTI, as discussed in Chapter 2. The nominal LTI trajectory is defined by the last and current waypoint of the other aircraft as provided by the AOP Intent Inference module, ADS-B, or CPDLC message.

To derive the LTI confidence estimate and verify intent, the focus was on whether the other aircraft position currently is within the airway or jet way of the nominal LTI trajectory. This route is generally defined by a primary and secondary clearance regions centered about piecewise linear segments between waypoints. Under Free Flight, the RNP is used to define the boundaries of the route. In addition to requiring the other aircraft to have its position within the nominal LTI trajectory route, another requirement incorporates the current ground speed velocity vector. Hence, for the aircraft to be following its nominal LTI trajectory, its track angle must also be directed near the current waypoint.

To evaluate the performance of the Kalman filter and LPF, as well as the NTI and LTI confidence assessment algorithm, ADS-B and TIS-B scenarios were selected. The ADS-B scenario consisted of an aircraft using GPS navigation and with FTE flying between two waypoints. The

TIS-B scenario was obtained from two aircraft radar track histories as measured by a TRACON radar. The two aircraft trajectories were artificially arranged to introduce a near-conflict. Performance of the Kalman filter with the ADS-B scenario resulted in a current North and East position root-mean-square (rms) error of 0.002 and 0.012 nmi, respectively. Hence, the North and East position estimate has a 95% confidence level of +/- 0.004 and 0.024 nmi, respectively. When these estimates are propagated 60 seconds into the future under NTI conditions, the rms position errors increased to 0.11 and 0.30 nmi, respectively. LTI confidence estimates were also computed for each point along the trajectory. The performance of the LPF with the TIS-B scenario resulted in a North and East position rms error of 0.25 and 0.19 nmi, respectively. When this state estimate was propagated 60 seconds into the future, the North and East position rms errors increased to 0.91 and 0.56 nmi, respectively.

4.0 HAZARD PRIORITIZATION

In its flight through the NAS, an aircraft may encounter many possible hazards to its safe passage from the flight's origin to its destination. A hazard prioritization function in AOP is needed to determine the order hazards should be dealt with by the aircraft's flight crew (see Figure 4.1). This function is used to enhance flight crew situational awareness and maximize the safety and efficiency of hazard avoidance maneuvers.

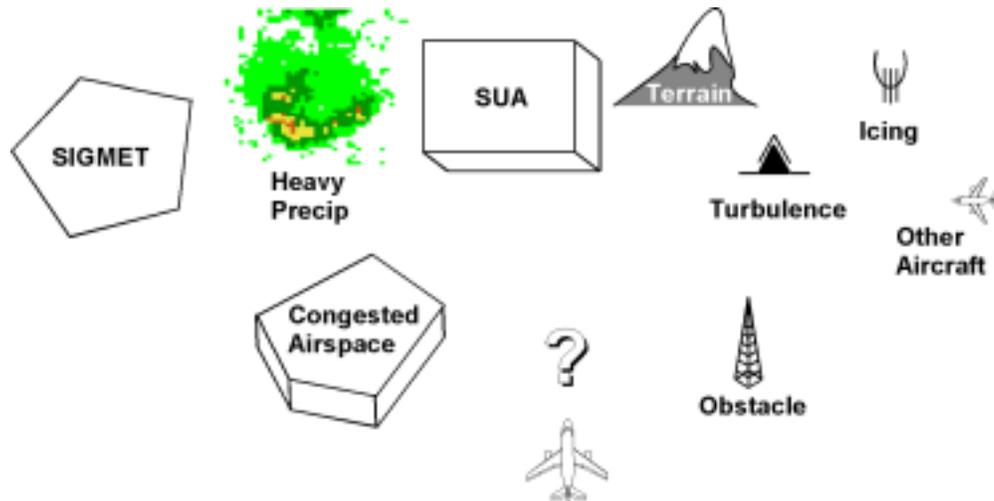


Figure 4.1. In what order should the flight crew deal with the known hazards?

In the AOP design, the architecture of the Hazard Prioritization model is directly supported by the Hazard Detection model and directly supports the Hazard Notification model, as shown in Figure 4.2.



Figure 4.2. Hazard Prioritization model connectivity within the AOP.

The nominal data flow supporting the hazard prioritization function is shown in Figure 4.3. Data from ADS-B, TIS, FIS, and CPDLC messages, along with onboard database information are converted by the Hazard Detection model into a hazard list that is fed into the Hazard Prioritization model. The Hazard Prioritization model incorporates a series of user preferences into a series of filter and sorting algorithms that produce the eventual prioritized threat list as output that is passed onto the Hazard Notification model.

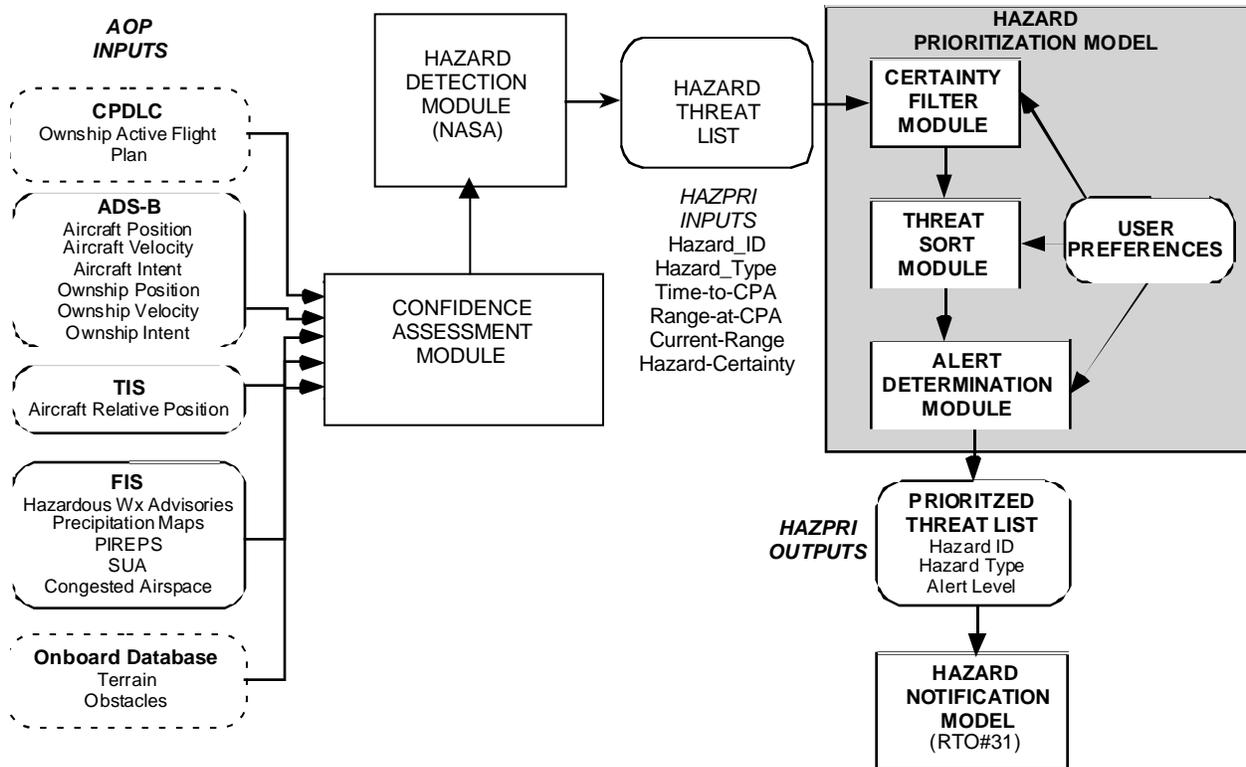


Figure 4.3. Hazard prioritization data flow diagram.

4.1 Theory

The hazard prioritization method is based on an approach centered around a series of filter and sorting algorithms based on key hazard priority factors shown in Table 4.1. In the next sections, the hazard prioritization inputs and outputs and method are described.

Table 4.1. The key hazard priority factors used in the Hazard Prioritization model.

Factor	Description
Hazard Type	Types include aircraft, convective weather, turbulence, Special Use Airspace, congested airspace, terrain
Time-to-Closest Point of Approach (CPA)	The predicted time before the aircraft will be at its closest point of approach for the given hazard
Range-at-CPA	The predicted minimum horizontal and vertical ranges between the aircraft and a given hazard
Current Range	The current horizontal and vertical range between the aircraft and a given hazard
Hazard Certainty	The probability that the given conflict (or a more severe one) will occur between the aircraft and the given hazard
User Preferences	A set of hazard prioritization preferences that includes the relative priority of hazard types, alert level thresholds, and hazard certainty filter criteria.

4.1.1 Hazard Prioritization Inputs and Outputs

The necessary inputs for the Hazard Prioritization model consists of a hazard list provided by the Hazard Detection model, or otherwise obtained from other AOP models, and a set of hazard prioritization user preferences. The output from the hazard prioritization process is a prioritized hazard list that is sent to the Hazard Notification model. These inputs and outputs are now described in more detail.

4.1.1.1 Hazard List Inputs

The hazard list is comprised of key information about all flight hazards known to the AOP. For each flight hazard, the hazard list includes:

- a unique hazard identification name,
- the hazard type,
- the time-to-CPA,
- the horizontal and vertical ranges at CPA,
- the current horizontal and vertical ranges from the hazard, and
- a probabilistic hazard certainty.

Possible hazard types to be handled by the Hazard Prioritization model include those shown in Table 4.2.

Table 4.2. Possible hazard types considered for the Hazard Prioritization model.

Hazard Type	Data Source	Description
Air Traffic	ADS-B, TIS	Aircraft (fixed-wing, rotary-wing), balloons, etc.
SIGMETs (incl. Convective SIGMETs)	FIS	Predicted areas of hazardous weather phenomena including severe/extreme turbulence, severe icing, duststorms, sandstorms, volcanic ash, tornadoes, severe thunderstorms
Turbulence	FIS/PIREP	Pilot Report of air turbulence; such turbulence ranges in severity from impacting ride quality to safety.
Special Use Airspace	FIS	Volumes of airspace that are off-limits to commercial air traffic for given periods of time; typically used by military aircraft
Heavy Precipitation	FIS	Areas of current or forecasted heavy precipitation that have associated dangerous convective activity
Congested Airspace	FIS	Volumes of airspace and associated time periods for which the forecasted or current air traffic demand is greater than capacity
Icing	FIS/PIREP	Pilot Report of icing
Wind Shear	FIS/PIREP	Pilot Report of wind shear
Terrain	Onboard Terrain Database	Static natural contour of the Earth's surface
Obstacles	Onboard Obstacle Database	Static man-made structures close to the ground

For RTO 30, hazard prioritization algorithms are designed to handle the prioritization of all such hazard types, assuming the hazard list is available to the Hazard Prioritization model. If this assumption does not hold true, the Hazard Prioritization model will have to generate the hazard list itself. In this case, algorithms to generate the required hazard information for the hazard types in Table 4.2 will be generated in the order shown, while adhering to contract budget and time constraints.

4.1.1.2 User Preference Inputs

In addition to the hazard list, the Hazard Prioritization model uses a set of user-defined inputs in order to tailor the hazard prioritization method to support specific AOP research objectives. The user preferences include:

- different certainty thresholds per hazard type,
- the number of desired threat levels,
- the threat level per hazard type,
- the number of desired alert levels, and
- alert level criteria based on combinations of time-to-CPA, range-at-CPA, and current range.

More detailed specification of these user preference inputs for each hazard prioritization algorithm are found in Section 4.1.2.

4.1.1.3 Prioritized Hazard List Outputs

Output by the Hazard Prioritization model, the prioritized hazard list is comprised of an ordered list of flight hazards that are relevant for flight crew consideration. This ordered list is in decreasing order of consideration priority and includes:

- a unique hazard identification name,
- the hazard type, and
- the hazard alert level.

For RTO 30, it is assumed that this prioritized hazard list is what is needed by the Hazard Notification model. If this assumption does not hold true, the Hazard Prioritization model will be modified to provide the additional necessary information, while adhering to contract budget and time constraints.

4.1.2 Hazard Prioritization Method

The hazard prioritization methodology implemented in the Hazard Prioritization model is based on modules that filter the hazards based on hazard certainty, sort the hazards based on user-assigned hazard threat levels, and then sort the hazards based on assigned alert levels based on user-specified separation time and distance criteria. Each of these hazard prioritization algorithms is now discussed.

4.1.2.1 Hazard Certainty Filter

The first component of the hazard prioritization process is the filtering of hazards based on threat certainty. For any potential hazard, there will be some level of uncertainty that a conflict between the hazard and ownship will occur. This uncertainty arises due to a large number of factors:

- age of the data (e.g., old versus new PIREP),
- information source (e.g., TIS vs ADS-B),
- uncertainties in the current positions, velocities (as expressed for aircraft in ADS-B Navigational Uncertainty Categories (see [RTCA242, SK99a]), and future intent of both the hazard (e.g., SUA schedule) and ownship (e.g., flight plan),
- dynamics of the atmosphere (e.g., wind speed uncertainty), and
- dynamic nature of the type of hazard (e.g., convective activity may grow or shrink).

In the hazard prioritization method, a scheme is incorporated that takes conflict uncertainties into account. Previous researchers have developed various methods for aircraft-to-aircraft conflict analysis [KMH96, PE97, YK97]. For all hazards identified by the AOP, we assume that the Hazard Prioritization model will receive a probabilistic value in the range [0,1] of conflict certainty

for each hazard from the Hazard Detection model. Note that this certainty level is not equivalent to that being generated by the Information Confidence Assessment model. The certainty level being generated by the Information Confidence Assessment model is the certainty that a given aircraft is at a particular position at the current time. This is not equal to what the Hazard Prioritization model needs, which is the certainty that a given hazard *will conflict with ownship at a future time*. If the assumption of the availability of probabilistic conflict certainties for each hazard from the Hazard Detection model is not valid, then the implementation of the hazard certainty filter will need to be revised.

The hazard certainty filter works as follows. If the incoming hazard certainty for a given hazard is below the user-defined threshold for a corresponding hazard type, the hazard is removed from the hazard list. Otherwise, the hazard information is retained. A set of incoming hazards, H , exists with unique hazards, H_{ii} , which are the i th instance of unique hazard type t , that have the respective hazard certainties of C_{ii} . Also, the set H has n unique hazards and m unique hazard types. The hazard certainties are within the range:

$$0 \leq C_{ii} \leq 1, \quad (4.1)$$

where C_{ii} represents the probabilistic certainty of a future conflict with H_{ii} . The extreme cases exist when $C_{ii} = 0$, signifying that a future conflict with H_{ii} has no support, and $C_{ii} = 1$, signifying that conflict with H_{ii} currently exists or is certain. For example, if three hazards exist such that three aircraft with call signs AAL100, AAL200, and AAL300 with corresponding hazard certainties of 0.5, 0.6, and 0.7, then:

- H is the set $\{H_{11}, H_{12}, H_{13}\} = \{AAL100, AAL200, AAL300\}$
- $n = 3$ unique hazards, $m = 1$ unique hazard type, and $t = 1$ for aircraft hazard type, and
- $C_{11} = 0.5$; $C_{12} = 0.6$; and $C_{13} = 0.7$.

For every hazard type t , the user defines a user-preferred hazard certainty threshold, T_t , which represents a lower limit on the acceptable hazard certainty. Therefore, all T_t represent a probabilistic certainty of a future conflict with the hazard H_{ii} such that:

$$0 \leq T_t \leq 1, \quad (4.2)$$

where T_t represents a lower certainty threshold for conflicts with hazards H_{ii} . Special cases occur when $T_t = 0$, signifying that the user wants to be aware of all hazards with finite conflict probabilities, and $T_t = 1$, signifying that the user does not want to know any hazard information. In absence of user input, the default value of T_t is 0.5, or whatever RTCA or other recommendation is standard in the industry.

The filtering algorithm works by comparing the input hazard certainty, C_{ii} , with the corresponding hazard certainty threshold, T_t . Then, all hazards H_{ii} for which $C_{ii} \leq T_t$ are removed from H (see Figure 4.4) and the resulting H is passed on to the hazard threat level sorting algorithm. For example, in the three aircraft example above, if the user defines $T_1 = 0.6$, then AAL100 (with $C_{11} = 0.5$) and AAL200 (with $C_{12} = 0.6$) are removed from the set of hazards.

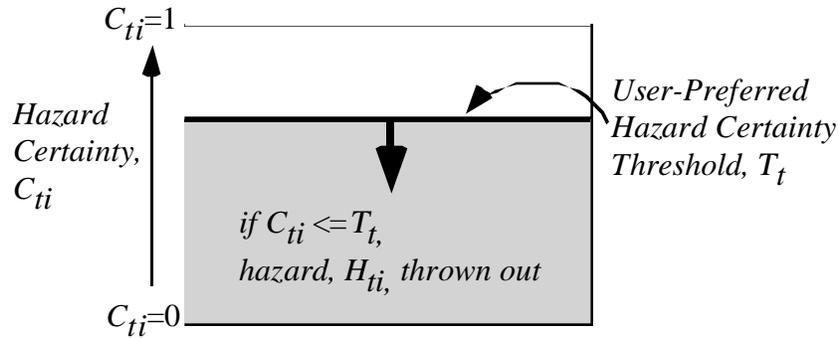


Figure 4.4. Hazards with low certainty are filtered from the set of hazards.

4.1.2.2 Hazard Sorting by Threat Level

The second component of the hazard prioritization process is the sorting of hazards based on user-assigned threat levels. Previous investigations into airborne hazard prioritizations such as [H98] and [FHK98] suggest that safety is the pre-eminant factor in prioritizing hazards. Such a safety criteria leads one to prioritize certain hazard types for which a conflict could be catastrophic (e.g., terrain or other aircraft) above those for which safety would not be impacted (e.g., minor turbulence or SUA). However, [H98] and [FHK98] have also shown that other factors and the relative importance to individual users come into play in the hazard prioritization process. For example, the factors of maintaining ride quality or following individual company policies have user-dependent levels of importance. Therefore, in order to provide NASA researchers the capability to investigate different hazard prioritization schemes, a scheme to sort hazards based on user-defined threat levels per hazard type is implemented.

This scheme involves the specification of a user-defined threat level for each hazard type and a total number of distinct threat levels. Associated with the set of incoming hazards H^i is a user-defined threat level L^t defined for each hazard type t . The user also defines the total number of distinct threat levels P which are ordered in decreasing levels of threat from 1 to P .

The hazard sorting procedure is accomplished using an insertion sort algorithm [Kn98]. As a simple, stable, and in-place sorting algorithm, this algorithm achieves an ordered sort in $O(p^2)$ operations and works quickly when P is relatively small (i.e., $P \leq 12$) [http://members.xoom.com/thomasn/s_man.htm]. If P becomes very large, other sorting algorithms such as a quicksort should be considered.

The insertion sort algorithm works as follows:

1. Designate H' as the set of n hazards; $H' = \{H'_j\}$ with $j = 1, 2, \dots, n$.
2. Choose $j = 2$ and let $H'' = H'$, the eventual output set of hazards.
3. If $j = n+1$, then go to step 8. Otherwise, take H'_j out of H'' .
4. Starting at the k th hazard in H'' , identified as H''_k , where the threat level is greater than the threat level of the chosen hazard, move all hazards inclusively between H''_k and H''_{j-1} one position down (i.e., $H''_{l+1} = H''_l$ for $l = k, k+1, \dots$ to $j-1$ and $H''_k = \{0\}$).
5. Set $l = k + 1$.
6. If the threat level for H'_j is less than the threat level for H''_l , then set $H''_{l-1} = H'_j$, set $j = j + 1$ and go to step 3.
7. Set $l = l + 1$ and go to step 6.
8. Stop. H'' is the output, sorted hazard list.

A simple example of a hazard insertion sort with $n = 3$ hazards is shown in Figure 4.5.



Figure 4.5. Hazards are sorted based on threat level with an insertion sort.

Default values for the user-specified threat levels are as shown in Table 4.3. Note that the default threat levels are designated as four levels in decreasing level of severity from 1 to 4.

Table 4.3. Default Hazard Threat Levels.

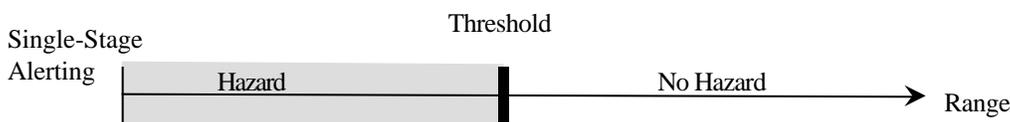
Hazard Type	Threat Level*	Reasoning
Air Traffic	1	Encountering other air traffic would be catastrophic.
SIGMETs (incl. Convective SIGMETs)	2	Encountering SIGMETs would be very hazardous to the safety of flight.
Turbulence	3	Encountering turbulence would affect ride quality which is assumed to be important.
Special Use Airspace	4	Encountering SUA would not be hazardous to the safety of flight, but would impact the legality of the flight.
Heavy Precipitation	2	Encountering areas of heavy precipitation that have associated dangerous convective activity would be very hazardous to the safety of flight
Congested Airspace	4	Encountering congested airspace would not be hazardous to the safety of flight, but could impact the flight's schedule adherence.
Wind Shear	2	Encountering wind shear would be very hazardous to the safety of flight.
Icing	2	Encountering icing conditions would be very hazardous to the safety of flight
Terrain	1	Encountering terrain would be catastrophic.
Obstacles	1	Encountering obstacles would be catastrophic

*Note: 1= most threatening, 4= least threatening.

4.1.2.3 Hazard Sorting by Alert Level

The third and final component of the hazard prioritization process is the sorting of the hazards based on an assigned alert level. Alert levels being considered by ATM researchers typically consist of some combination of geometric and temporal criteria such as the time-to-CPA, horizontal and vertical ranges-at-CPA, and current horizontal and vertical separation distances [KP97, KY97], and, in some instances, number of conflict-free aircraft maneuvering options [YK97]. The work of Kuchar [KH95, Ku96] presents a unified methodology to airborne alerting logic. Figure 4.6 illustrates the logic for a single-stage and two-stage alerting logic. For conflict detection between aircraft, the conflict can occur at the Alert Zone in a single-stage alert, or a caution can be established before the Alert Zone conflict. Multi-stage alerts, as designed by the RTCA SC186 Working Group 1 ADS-B based conflict detection and resolution operational concept [RTCASC186 Ops Concept], allows warnings to alert as the severity of the situation gets worse, as shown in Figure 4.7. In Figure 4.7, low, medium, high, and critical alerts trigger at closer ranges:

1. Critical Alert – triggered by a prediction of a near-mid air collision (i.e., CPA distance within 500 feet horizontally and 100 feet vertically) in one minute,
2. High Level Alert – triggered by the violation of the Protected Airspace Zone (PAZ) (i.e., for the terminal area, current separation distance is within 3 nmi horizontally and 1000 feet vertically),
3. Medium Level Alert – triggered by a prediction of a PAZ conflict within 2 minutes, and
4. Low Level Alert – triggered by a prediction of a PAZ conflict within 5 minutes.



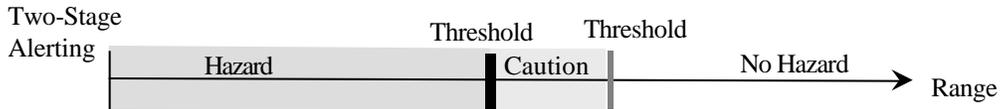


Figure 4.6. Single-stage alerting logic and two-stage alerting logic.

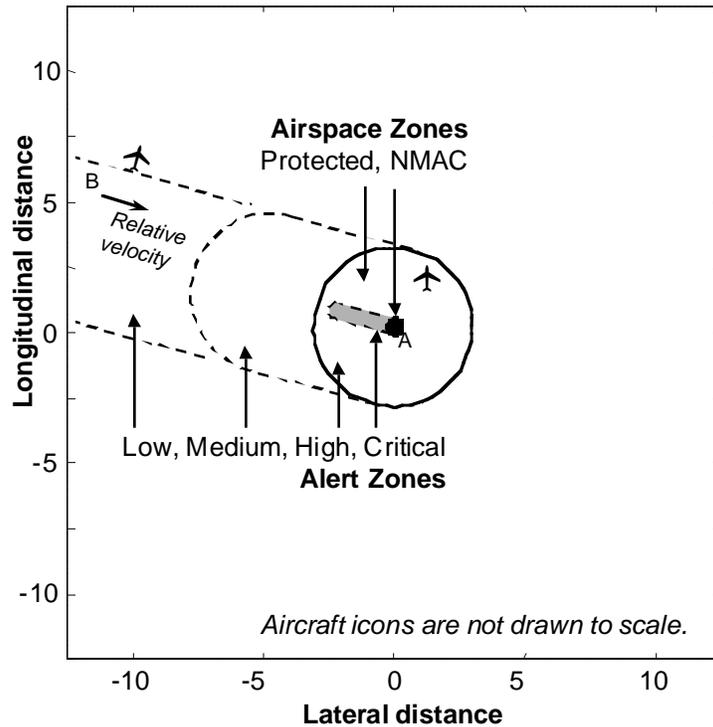


Figure 4.7. Multi-stage Protected Airspace Zone alerting logic allowing for low, medium, high, and critical alerts.

Key factors for determining alert levels include the time-to-CPA and hazard ranges. These two key factors are now discussed.

Time-to-CPA influences hazard prioritization by how long the flight crew has to respond to a hazard. The alerting logic based in the Traffic Alert and Collision Avoidance System (TCAS) was developed on a time-to-CPA criteria known as “ τ ”. The classical τ is defined by the aircraft-to-aircraft range divided by range rate, assuming that both aircraft are modeled as point hazards. A more precise determination of time-to-CPA which requires the broadcast of aircraft velocity information is:

$$\tau = -\frac{r \cdot c}{\dot{c} \cdot c}, \quad (4.1)$$

where r is the vector locating a target aircraft with respect to the ownship, and c is the relative motion vector. This time-to-CPA equation was developed in [KP97] and is compatible with hazard prioritization of ADS-B equipped aircraft threats such as those that can be currently configured in the NASA FFSim architecture. Other time-to-CPAs can be generated for line, region, or volumetric hazards based on the ownship velocity vector, hazard motion data, and hazard point, line, region, or volume data as shown in Appendix C. For longer-term time-to-CPA calculations, aircraft intent

information may need to be incorporated into velocity-based schemes. This intent information may take the form of flight plan waypoints or Trajectory Change Points (TCPs) as defined in the ADS-B message [RTCA242], and different time-to-CPA algorithms will need to be developed in either case.

Hazard ranges are important in the prioritization of hazards due to their impact on the hazard threat level. In the case of traffic hazards, hazard ranges are often classified as being within either a “critical” Near Mid-Air Collision (NMAC) zone, a Protected Airspace Zone (PAZ), or an Alert Zone (AZ) (see Figure 4.7). Prepared for the RTCA SC-186 committee Working Group 1, [K99] provides the basis for establishing different hazard threat levels that affect hazard prioritization for conflict detection and resolution. An example calculation of aircraft-to-aircraft miss distance, the norm of \bar{r}_f , based on velocity vector information would be:

$$\bar{r}_f = \hat{c} \left\langle \bar{r} \leftarrow \hat{x} \right\rangle. \quad (4.2)$$

Developed in our previous AATT research [KP97], this miss distance equation, uses the same vector assumptions as equation (4.1), but also assumes that \hat{c} is the unit vector in the direction of the relative motion vector \bar{c} .

Calculation of miss distance for area (e.g., SIGMET) or volumetric hazards (e.g., SUA) is based on the miss distance from the area or volume vertices. Figure 4.8 illustrates the CPA for a aircraft flying near SUA. Miss distance from a point hazard such as a PIREP may need to factor in the uncertainty in the hazard region based on typical size and characteristics of the weather phenomena and the age of the reported data.

Hazard alert levels are typically based in part upon either the current or forecasted minimum range (or “miss distance”) between ownship and a given hazard. Also, the alert level can be a function of ownship location (e.g., the PAZ horizontal limit may be defined as being 5 nmi in en route airspace and 3 nmi in terminal airspace) as well as phase of flight (e.g., the PAZ vertical limit may be defined as being greater during climbs and descents).

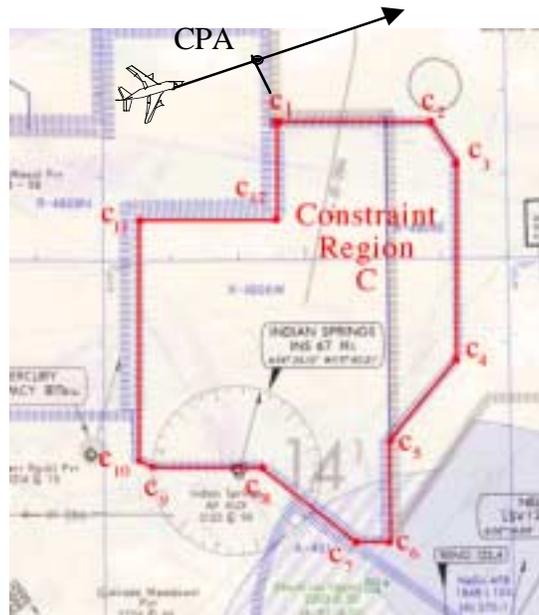


Figure 4.8. Modeling of aircraft miss distance to a SUA region North of Las Vegas.

In the current hazard prioritization implementation, the necessary time-to-CPA and hazard range information are assumed to be supplied by the Hazard Detection model. If this is not the case, then additional appropriate algorithms will have to be designed and coded. Because of the high risk of this assumption that is being made, a preliminary investigation into useful time-to-CPA and range determination algorithms was performed and is included at the end of this report as Appendix C.

Given the budget and time constraints inherent in the development of the hazard prioritization method and the desire for maximum NASA researcher flexibility, a hazard prioritization alert sorting algorithm was chosen based on maximum user flexibility and spatial-temporal alert level criteria. This final sorting algorithm is based on a combination of an alert level assignment algorithm and a final alert level sort algorithm.

The alert level assignment algorithm is based on a series of user-specified alert level criteria (see Table 4.4). First, the user must specify the number of desired alert levels. Then, for each alert level, the user must specify a desired combination of alert thresholds based on desired alert level criteria.

These alert thresholds can be of two types: forecast or nowcast. The forecast alerts are based on the expected future relative states of ownship and the hazard at the CPA. Forecast alerts involve some user-specified combination of time-to-CPA, horizontal range-at-CPA, and vertical range-at-CPA threshold values. Nowcast alerts are based on the current relative state of ownship and the hazard. Since a temporal threshold has no meaning in this context, the nowcast alert is based only on current horizontal and vertical ranges.

Table 4.4. User-specifiable alert level criteria in the Hazard Prioritization model

Criteria	Symbol	Units	Comments
Number of Alert Levels	n_a	-	The number of alert levels
Alert Level	A^l		The l th alert level; between 1 and n_a ; most critical = 1, least critical = n_a .
Alert Type	a_i	-	$a_i = 0 =$ nowcast alert; $a_i = 1 =$ forecast alert; Nowcast alerts are based on current horizontal and vertical ranges; Forecast alerts are based on a combination of time-to-CPA, and horizontal and vertical range-at-CPA.
Current Horizontal Range Threshold	x_T^l	nmi	Horizontal range-based threshold for the l th alert level based on current state; Typically used in high-level, nowcast alerts and in combination with a current vertical range threshold
Current Vertical Range Threshold	z_T^l	ft	Vertical range-based threshold for the l th alert level based on current state; Typically used in high-level, nowcast alerts and in combination with a current horizontal range threshold
Horizontal Range-at-CPA Threshold	ξ_T^l	nmi	Horizontal range-based threshold for the l th alert level at predicted CPA; Typically used in lower-level, forecast alerts and in combination with a time-to-CPA threshold
Vertical Range-at-CPA Threshold	ζ_T^l	ft	Vertical range-based threshold for the l th alert level at predicted CPA; Typically used in lower-level, forecast alerts and in combination with a time-to-CPA threshold
Time-to-CPA Threshold	τ_T^l	min	Time-based threshold for the l th alert level; Typically used in lower-level, forecast alerts and in combination with range-at-CPA thresholds

In all alert threshold cases, an alert of the appropriate level is triggered when the value of the hazard list data is below the appropriate threshold, and a hazard is assigned the most critical, alert level of all valid alerts. For each hazard H''_j , such that H'' is the set of n hazards; $H'' = \{H''_j\}$, $j = 1, 2, \dots, n$, with existing values of current horizontal range, x^j , current vertical range, z^j , horizontal range-at-CPA, ξ^j , vertical range-at-CPA, ζ^j , time-to-CPA, τ^j , and n_a number of user-defined alert levels, its corresponding alert level, A_j is determined by:

$$A_j = \min \{ \text{all valid } A^i \}, \quad (4.3)$$

for $i = 1, 2, \dots, n_a$. In the case of nowcast alerts, $a_i = 0$, a valid A^i is defined if and only if:

$$x^j \leq x_T^i \text{ or } z^j \leq z_T^i. \quad (4.4)$$

In the case of forecast alerts, $a_i = 1$, a valid A^i is defined if and only if:

$$\xi^j \leq \xi_T^i \text{ or } \zeta^j \leq \zeta_T^i \text{ and } \tau^j \leq \tau_T^i. \quad (4.5)$$

Default values for the user-specified alert level threshold information are shown in Table 4.5. Note that the default number of alert levels n_a is designated as 3 (analogous to the traffic alerting levels from [RTCA SC-186 CD&R Ops Concept]) in decreasing levels of severity from 1 to 3.

Table 4.5. Default Alert Level Thresholds.

Alert Level A^i	Alert Type a_i	x_T^i * (nmi)	z_T^i * (ft)	ξ_T^i * (nmi)	ζ_T^i * (ft)	τ_T^i (min)	Comments
1	0	5	1000	-	-	-	High Level Alert– will exist when the separation distance between the hazard and flight is less than the designated miss distance
2	1	-	-	5	1000	2 min	Medium Level Alert– will exist when the flight is predicted to miss the hazard by less than the designated miss distance in less than 2 minutes
3	1	-	-	5	1000	5 min	Low Level Alert – will exist when the flight is predicted to miss the hazard by less than the designated miss distance in less than 5 minutes.

*Note: Range-based thresholds may in the future be dependent on airspace and/or hazard type.

Once the alert levels are designated, the hazard list, previously sorted on threat types, is sorted based on alert level. This final algorithm in the alert level sorting module is a sorting of *all* of the hazards based on the assigned alert levels. Note that this scheme is different than the hazard prioritization performed by TCAS. With TCAS, threat prioritization occurs for the highest RA alert levels, but in the case of the lower TA alert levels, all TA threats are of equal priority [RTCA/DO-185].

The final sorting algorithm is based on a double insertion sort that works in the same manner as the one in the threat sort module (Figure 4.5), but sorting on alert level as opposed to threat level. The first insertion sorts based on the discrete alert level assigned to the hazard.

After the first insertion sort, sorting is performed for each set of hazards at a specified alert level. This second insertion sort is based on slant distance to the hazard, in the case of $a_i=0$ alerts, or time-to-CPA, in the case of $a_i=1$ alerts. This sorting algorithm works on the H'' output list from the threat sort module and generates a new, prioritized hazard output list H''' . The list H''' consists of a prioritized list of hazard IDs, hazard types, and alert levels for use by the Hazard Notification model and any other AOP model that needs the information.

4.1.3 Hazard Prioritization Updates

In order to perform its function, the hazard prioritization methodology needs to communicate inputs and outputs with the AOP Hazard Detection and Hazard Prioritization models. Once every second, the hazard prioritization methodology implemented in the Hazard Prioritization model will look for an updated input hazard list from the AOP Hazard Detection model. The identification of a new input hazard list then drives the hazard prioritization process to generate a new, prioritized hazard list, which, upon completion, is then output to the Hazard Notification model.

4.2 Scenarios

The engineering approach for testing the hazard prioritization algorithm includes the use of several simulation-based test scenarios; these scenarios are reviewed next. These scenarios are designed to test the validity of the coded hazard prioritization algorithms. Synthetic data are planned for use in the test scenarios, but real data can be integrated in the test scenarios, based on the availability of data.

4.2.1 Test Scenarios

The following scenarios are designed to test the hazard prioritization algorithm. In Scenarios 1-3, the hazards all have a certainty level of 1; in Scenario 4-5, hazard certainty levels range between 0 and 1.

Scenario 1

Key Feature:

- A series of aircraft in head-on conflicts at different look-ahead times.

Expected Outcome:

- Aircraft with less time-to-CPA will have higher alert levels.

Scenario 2

Key Feature:

- A series of aircraft with the same heading and distance from ownship, but at different radial positions from ownship.

Expected Outcome:

- Aircraft with lower distance-at-CPA and time-to-CPA will have lower alert levels.

Scenario 3

Key Feature:

- A static set of co-located aircraft, SIGMET, turbulence, SUA, heavy precipitation, congested airspace, wind shear, and icing at a fixed distance from ownship

Expected Outcome:

- All hazards should have the same alert level, but be in order of the user preferred threat level.

Scenario 4

Key Feature:

- A static set of co-located aircraft, SIGMET, turbulence, SUA, heavy precipitation, congested airspace, wind shear, and icing at a fixed distance from ownship with varying certainty levels between 0 and 1.

Expected Outcome:

- All hazards with certainty levels below the user-preferred certainty threshold should not show up in the prioritized hazard list.

Scenario 5

Key Features:

- A dynamic set of aircraft, SIGMET, turbulence, SUA, heavy precipitation, congested airspace, wind shear, and icing hazards at varying distances from ownship with varying certainty levels between 0 and 1.
- Different user preference criteria (incl. one where turbulence is considered a high threat level hazard and one where it has a low threat level)

Expected Outcome:

- The prioritized hazard list will vary based on the different user preference criteria (esp., turbulence will vary in hazard priority).

5.0 CONCLUSIONS AND RECOMMENDATIONS

In this document, we specify the status of the engineering work towards the intent inference, confidence assessment, and hazard prioritization modules for AOP.

5.1 Conclusions

The intent inference approach is to derive several (twenty or more) intent models that describe intelligent behaviors of a pilot. With these models, the motion of an aircraft being tracked by the ownship can be analyzed to infer intent. In the case that the aircraft is broadcasting intent, the Information Ambiguity Resolution module will verify that the aircraft is actually following the broadcast intent. In the event that the intent is missing from the broadcast or if there is no broadcast of intent, then the Intent Inference module infers the intent and predicts the future flight path. When prediction is requested within tens of seconds, the prediction is based on a tracking filter (a Kalman filter if ADS-B data are used, or a low pass filter for TIS-B data), and when the prediction spans larger time horizons, then the prediction is performed based on the tracking filter that is biased by the path predicted by intent inference models (Long Term Intent).

The confidence assessment approach is based on using a tracking filter to prepare the best estimate of state data and to address the issue of missing or delayed data. The key to confidence assessment is in the estimation error covariance matrix, which provides a confidence estimate of how good the current state has been estimated. From our analysis using TIS-B data, we found that the TIS-B data has such low accuracy that it should only be used when the ADS-B data is not available. Also, due the low accuracy of TIS-B data as well as no time tags for these data, the TIS-B data should not be fused with the ADS-B data when both are available.

The hazard prioritization approach is based on a set of flexible algorithmic filters and sorting that maximizes the ability for NASA researchers to test different alerting schemes. Hazard types, times to point of closest approach, miss distances from hazards, hazard certainties, and user preferences for threat levels and alert levels are considered in the prioritization. Our approach handles various hazards types: points, line segments, polygons, and polyhedron volumes.

5.2 Recommendations: Process to Complete Development of RTO 30 Modules

The process to complete the development of the RTO 30 modules includes the following high level objectives. Here we assume that the AOP receives further development and refinement while the RTO 30 algorithms and module designs await a restart.

- Update Proposed Engineering Approach. The intent inference algorithm will need to have the Proposed Engineering Approach re-evaluated to account for changes that take place to the AOP architecture in the time period when the RTO 30 task is dormant. Input systems, database content, new modules, scenarios, or modifications to existing AOP modules may occur and these changes are likely to change our Proposed Engineering Approach.
- Update Software Design. Requirements will be revisited with respect to any changes in the AOP architecture. Then the analysis step and development of use cases will be performed again if there were any changes in requirements. The object models will be updated to reflect any changes in the design and or engineering approach. A new Software Design Document will be created to describe the design of the RTO 30 modules that are compatible with the current state of the AOP-FFSim system.
- Year 2000 Problem Changes. The Year 2000 Compliance for the software will not be an issue since the restart will occur after January, 2000 (or after all Year 2000 effects are over). The Functional Design Requirement for Year 2000 Compliant code will be dropped.

- Update RTCA SC-195 Committee Specifications for FIS. The FIS specifications are soon to be released and all FIS data requirements have been based on the latest available draft copies of RTCA MASPS. The Proposed Engineering Approach and Software Design Document will have to be modified to correspond with the final release from the RTCA.
- Procure Standardized Airport and Navigation DAFIF Database. The RTO 30 intent inference and hazard prioritization systems needs a standardized set of data that is kept onboard in a database management system. More specifically, airport, runway, navigation aids, SIDs, STARs, SUA, and other data are needed as specified in our intent inference onboard database requirements. These data are provided in an industry standard database provided by the government (NIMA) in their Digital Aeronautical Flight Information File (DAFIF) database product. This data can be acquired (free of charge to NASA or at a rate of roughly \$130 per month for commercial industry customers) with the proper paper work provided by NASA. It will accelerate our progress if NASA were to procure these data (free of charge) and have them available when RTO 30 restarts.
- Intent Inference Module Iterative Design. The intent inference module development is planned to proceed with its iterative software design cycle. At first, the software will be designed to Verify Intent; this algorithm will either verify the intent or it will indicate that the intent cannot be verified. Next, the software for Fly to Waypoint will be designed and tested with the TCP waypoint. This algorithm will then be modified to perform Fly to Waypoint using TCP+1 or any waypoint in the flight plan. Last, the Fly to Waypoint intent model will be modified to search for any valid waypoint within 50 nmi of the ownship. Iterations are used to incrementally add intent models to the intent inference algorithm. After Fly to Waypoint capabilities are added, other software will be developed, for instance, the altitude hold intent model, heading hold, and maintain speed models. The final intent models and most complex intent models are those that include calls (communications or message passing) with other AOP modules, for instance, hazard avoidance, CD&R, or FMS. At first, the hazard avoidance intent models will be developed for turbulence, SUA, and weather avoidance intent considerations. Then, the CD&R module in AOP will be called to formulate the options for intent models that verify that a conflict resolution is being performed by a nearby aircraft in conflict with the ownship or by two nearby aircraft in conflict with each other. Finally, the FMS module will be called to identify intent models that include wind optimized routes or RTAs. The approach is iterative; the intent inference algorithm capabilities are incrementally built up through iterations in engineering specifications, software design, software coding, and verification and validation using test scenarios.
- Intent Inference Module Ability to Access Information from AOP CD&R Module. The intent inference module uses other AOP modules, for instance the CD&R Module, to compute solutions to problems that are posed in terms of data from nearby aircraft. For instance, the solutions to frontside, backside, topside, and bottomside maneuvers for CD&R are computed in the CD&R Module. However, these CD&R solutions are usually computed for determining only the best conflict resolution path for the ownship and perhaps for the nearby aircraft as well for coordination purposes. Not only the optimal solution, but all feasible solutions are useful to the intent inference algorithm when considering plausible CD&R solution waypoints to predict the intent of the other nearby aircraft. In CD&R cases, the intent inference algorithm considers several intents for the nearby aircraft:
 - (1) the nearby aircraft either receives the CD&R waypoints or computes its own CD&R waypoints that match the ownship solution and proceeds to use the CD&R solution and broadcasts the TCP and TCP+1 that correctly implement the CD&R solution, (this is the nominal and expected case).

(2) the nearby aircraft flies its current TCP and TCP+1 and does not update these with the CD&R solution because it did not receive the new CD&R solution waypoints from the ownship, (this is the unequipped or non-cooperative case for CD&R).

(3) the nearby aircraft computes its own CD&R waypoints that differ from the ownship solution and the nearby aircraft proceeds to use the CD&R solution (e.g., a vertical solution) that does not match the ownship solution (e.g., a horizontal solution), that is, they fail to coordinate actions, (this is the case of mixed or mis-communicated CD&R).

These cases indicate that there is a need for the intent inference module to use the CD&R module, perhaps with function calls, to ask questions like: what are the frontside, backside, topside, and bottomside solutions to the cooperative and non-cooperative CD&R problem between ownship and this nearby aircraft? Not only the optimal solution is needed, but multiple solutions are needed so that all plausible actions of the nearby aircraft can be explained from the CD&R options available. Furthermore, the intent inference algorithm may be tracking two aircraft that are in conflict with each other but are not in conflict with the ownship. In such cases, one could not expect the CD&R algorithms onboard ownship to necessarily produce the CD&R results that these two aircraft might implement. At best, the ownship can determine through its own CD&R module that there is indeed a conflict between these two nearby aircraft and infer that conflict resolution is taking place or soon to take place; the ADS-B broadcast may or may not confirm this intent, however, this will only effect the level of confidence and not the intent inferred.

- Intent Inference Module Ability to Access Information from AOP Hazard Resolution Module. The intent inference module uses other AOP modules, for instance the Hazard Resolution Module, to compute solutions to problems that are posed in terms of data from nearby aircraft. For instance, the solutions to hazard resolution maneuvers for avoiding turbulence, SUA, terrain, and weather are typically computed in the Hazard Resolution Module for the ownship. Yet, the intent inference algorithm uses the hazard avoidance solution waypoints to predict the intent of the other nearby aircraft. This type of query to an ownship hazard resolution algorithm is very non-conventional. The ownship hazard avoidance algorithms typically only compute hazard avoidance solutions for the ownship, but the intent inference algorithm needs the ability to perform “what if” queries to the ownship hazard avoidance module so that the intent inference algorithm can get the waypoints that are valid hazard avoidance solutions for the nearby aircraft (not the ownship) and test them as plausible intents of nearby aircraft.
- Intent Inference Module Ability to Access Information from AOP FMS Module. The intent inference module uses other AOP modules, for instance the FMS Module, to compute solutions to problems that are posed in terms of data from nearby aircraft. In particular, wind optimized routes and RTA trajectories that are normally computed by the FMS Module for the ownship need to be computed with respect to the parameters and initial conditions of nearby aircraft. This is so the intent inference algorithm can compare the wind optimized routes or RTA routes with the state data of the nearby aircraft in order to infer the intent of the nearby aircraft as “following wind optimized route” or “following RTA trajectory”. This type of query to an ownship FMS is very non-conventional and perhaps infeasible. The ownship FMS algorithms typically only compute trajectory solutions for the ownship, but what the intent inference algorithm needs is the ability to perform “what if” type queries to the FMS module so that the intent inference algorithm can get the waypoints that are valid FMS optimized solutions and test them as plausible intents of these nearby aircraft. If this type of call statement to the FMS is infeasible, then the intent inference module will have to omit the intent model for “following wind optimized route” or “following RTA trajectory”.
- Confidence Assessment Module Development. At this time, the software design for confidence assessment is finished, however, the design needs to be re-assessed when the RTO 30 project restarts to address changes in the AOP architecture and input modules.

- Confidence Assessment Module Verification and Validation. There is a need for a large set of track data for multiple types of aircraft performing many different types of scenarios. For best verification results, both GPS and ground radar track data should be obtained, however, either types of data are useful to our testing of the algorithms. Data sets procured from NASA flight tests or sets available from NASA data archives will serve this purpose well. In the interim period before RTO 30 work is restarted, the gathering of track data for testing and validation should be performed.
- Hazard Prioritization Module Integration with other AOP Modules. The hazard prioritization module is expected to receive hazard data from the Hazard Detection model and output prioritized hazard data to the Hazard Notification module, and, perhaps, other AOP modules. A key next step for technical risk mitigation is to define the interfaces between the hazard prioritization software and the rest of the AOP to ensure that the correct data is being passed to and from the module. Critical information to be specified for the interfaces include the data input/output from the hazard prioritization module including the time-to-CPA and range-at-CPA data for each identified hazard, and the dynamics of how the modules communicate with one another. If the correct data is not available from the other AOP modules, the hazard prioritization software will have to include the design and implementation of new algorithms. To mitigate this potential problem, a preliminary investigation into useful CPA and time-to-CPA algorithms is included in Appendix C.
- Hazard Prioritization Module Iterative Design. After the hazard prioritization software design is approved by NASA, the hazard prioritization module will be implemented in software with an iterative design, implementation, and verification cycle. This software implementation will be driven by the specified set of multiple scenarios. Scenarios with increasing levels of sophistication in numbers of hazards, hazard types, hazard geometries, conflict geometries, and hazard dynamics will be fleshed out and used to verify and validate software of increasing complexity.

6.0 REFERENCES

- [BK97] de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O., *Computational Geometry: Algorithms and Applications*, Springer Verlag., Berlin, Germany, 1997.
- [B196] Bloch, I., "Information Combination Operators for Data Fusion: A comparative Review with Classification," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 26, No. 1, pp. 52-67, Jan., 1996.
- [BM93] Bushman, J.B., Mitchell, C.M., Jones, P.M., and Rubin, K.S., "ALLY: An Operator's Associate for Cooperative Supervisory Control Systems," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, No. 1, pp. 111-128, Jan./Feb., 1993.
- [CMJ95] Chu, R.W., Mitchell, C.M., and Jones, P.M., "Using the Operator Function Model and OFMspert as the Basis for an Intelligent Tutoring System: Towards a Tutor/Aid Paradigm for Operators of Supervisory Control Systems," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 25, No. 7, pp. 1054-1075, July, 1995.
- [DAG99] Advanced Air Transportation Technologies DAG TM Team, *Concept Definition for Distributed Air/Ground Traffic Management (DAG TM)*, Version 1.0, NASA Ames Research Center, Sept., 1999.
- [F94] Franklin, W.R., *Triangulated Irregular Network to Approximate Digital Terrain*, Tech. Report, Dept. of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, 1994.
- [FL79] Fowler, R.J., and Little, J.J., "Automatic Extraction of Irregular Network Digital Terrain Models", *Computer Graphics*, Vol. 13, No. 2, pp. 199-207, Aug., 1979.
- [Ge74] Gelb, A., *Applied Optimal Estimation*, The MIT Press, 1974
- [G94] Geddes, N.D., "A Model for Intent Interpretation for Multiple Agents with Conflicts," *IEEE Intern. Conf. on Systems, Man, and Cybernetics*, San Antonio, TX, Oct., 1994.
- [HG89] Hoshstrasser, B. and Geddes, N., "OPAL: Operator Intent Inferencing for Intelligent Operator Support Systems," *Proc. of the IJCAI-89 Workshop on Integrated Human-Machine Intelligence in Aerospace Systems*, Detroit, MI, Aug., 1989.
- [H98] Honeywell Technology Center, *Weather Avoidance Using Route Optimization as a Decision Aid: An AWIN Topical Study*, Phase I Report, No. NCC-1-291, Minneapolis, MN, Dec., 1998.
- [KaA86] Kautz, H. and Allen, J., "Generalized Plan Recognition", *National Conf. on Artificial Intelligence*, Philadelphia, PA, Aug., 1986.
- [Kn98] Knuth, D., *The Art of Computer Programming: Sorting and Searching*, Vol 3, 2nd Ed, June 1998.
- [K92] Krozel, J., *Intelligent Path Prediction for Vehicular Travel*, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Purdue University, W. Lafayette, IN, May, 1992.
- [K99] Krozel, J., "On Conflict Detection," presented at the Joint Eurocontrol/RTCA CD&R Working Group Meeting, NASA Ames Research Center, Moffett Field, CA, Jan. 26-28, 1999.
- [KA91] Krozel, J., and Andrisani, D., "Predicting Mission Plans for an Observed Land Vehicle", *DARPA Symposium on Associate Technology: Opportunities and Challenges*, George Mason University, Fairfax, VA, June, 1991.
- [KA93] Krozel, J., and Andrisani, D., "Intelligent Path Prediction for Vehicular Travel", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 23, No. 2, March, 1993.
- [KA95] Krozel, J., and Andrisani, D., "Intelligent ϵ -Optimal Path Prediction for Vehicular Travel", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 25, No. 2, Feb., 1995.
- [KP97] Krozel, J., and Peters, M., "Strategic Conflict Detection and Resolution for Free Flight," *IEEE Conf. on Decision and Control*, San Diego, CA, Dec., 1997.

- [Ku96] Kuchar, J.K., "Methodology for Alerting-System Performance Evaluation," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, pp. 438-444, March/April, 1996.
- [KH95] Kuchar, J., and Hansman, R. Jr., *A Unified Methodology for the Evaluation of Hazard Alerting Systems*, Report ASL-95-1, Aeronautical Systems Laboratory, MIT, Cambridge, MA, Jan., 1995.
- [KY97] Kuchar, J.K. and Yang, L.C., "Survey of Conflict Detection and Resolution Modeling Methods," *AIAA Guidance, Navigation, and Control Conf.*, New Orleans, LA, pp. 1388-1397, Aug., 1997.
- [OBS92] Okabe, A., Boots, B., and Sugirara, K., *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*, John Wiley and Sons, New York, 1992.
- [PS85] Preparata, F.P. and Shamos, M.I., *Computational Geometry*, Springer Verlag, NY, 1985.
- [RGC88] Rouse, W., Geddes, N., and Curry, R., "An Architecture for Intelligent Inferences: Outline of an Approach to Supporting Operators of Complex Systems," *Human-Computer Interaction*, Vol. 3, 1988.
- [RJM88] Rubin, K. S., Jones, P. M., and Mitchell, C. M. "OFMspert: Inference of Operator Intentions in Supervisory Control Using a Blackboard Architecture" *IEEE Transactions on Systems, Man, and Cybernetics*. 18(4), 618-637, 1988.
- [RTCA95] Radio Technical Commission for Aeronautics, *Report of the RTCA Board of Directors' Select Committee on Free Flight*, RTCA, Inc., Washington, DC, Jan., 1995.
- [RTCA186] Radio Technical Commission for Aeronautics Special Committee-186, *Minimum Operational Performance Standards for Cockpit Display of Traffic Information*, Draft 1, Version 4, May 28, 1998.
- [RTCA195] Radio Technical Commission for Aeronautics Special Committee-195, *Minimum Operational Performance Standards (MASPS) for Flight Information Services-Broadcast (FIS-B) Data Link*, Draft, Version 5.5, Oct, 1999.
- [RTCA232] Radio Technical Commission for Aeronautics Special Committee-169, *Operations Concepts for Data Link Applications of Flight Information Services*, RTCA/DO-232, March 14, 1996.
- [RTCA239] Radio Technical Commission for Aeronautics Special Committee-169, *Minimum Operational Performance Standards for Traffic Information Service (TIS) Data Link Communications*, RTCA/DO-239, April, 1997.
- [RTCA242] RTCA Special Committee-186, *Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)*, RTCA/DO-242, Feb., 1998.
- [SK99a] Schleicher, D., and Krozel, J., "RTO 21/RTO 30 ADS-B Message Content Specification," Seagull Technology, Inc., Tech. Report, September 1999.
- [SK99b] Schleicher, D., and Krozel, J., "RTO 21/RTO 30 TIS-B Message Content Specification," Seagull Technology, Inc., Tech. Report, September 1999.
- [SK99c] Schleicher, D., and Krozel, J., "RTO 21/RTO 30 FIS Message Content Specification," Seagull Technology, Inc., Tech. Report, September 1999.
- [SSG78] Schmidt, C.F., Sridharan, N.S., and Goodson, J.L., "The Plan Recognition Problem," *Artificial Intelligence*, Vol. 11, pp. 45-83, 1978.
- [SMK95] Silva, C.T., Mitchell, J.S.B., and Kaufman, A.E., "Automatic Generation of Triangular Irregular Networks using Greedy Cuts," *IEEE Conf. on Visualization*, pp. 201-208, 1995.
- [S93] Stengel, R.F., "Toward Intelligent Flight Control", *IEEE Systems, Man, and Cybernetics*, Vol. 23, No. 6, Nov./Dec., 1993.
- [TR94] Tambe, M. and Rosenbloom, P., "Event Tracking in Complex Multi-agent Environments", *4th Conf. on Computer Generated Forces and Behavioral Representations*, Orlando, FL, May, 1994.

- [TR95] Tambe, M., and Rosenbloom, P.S., "RESC: An Approach for Real-Time, Dynamic Agent Tracking," Intern. Joint Conf. on Artificial Intelligence, Montreal, Canada, Aug., 1995.
- [TR96] Tambe, M., and Rosenbloom, P.S., "Event Tracking in a Dynamic Multi-Agent Environment", *Computational Intelligence*, Vol. 12, No. 3, 1996.
- [YK98] Yang, L.C. and Kuchar, J.K., "Using Intent Information in Probabilistic Conflict Analysis," *AIAA Guidance, Navigation, and Control Conf.*, Boston, MA, pp. 797-860, Aug., 1998.
- [ZHH98] Zhao, Y., C. Haissig, and M. Hoffman, "Analysis of Pilot Intent in Air Traffic Management," *Proc of the American Control Conf*, Philadelphia, June 24-26, 1998.

7.0 APPENDICES

Appendix A: Functional Design Requirements

The following is a list of the Functional Design Requirements (FDRs) for the intent inference, confidence assessment, and hazard prioritization modules.

- FDR1.1: Software will be written in C or C++ programming languages.

All software shall be written in the C++ programming language. Existing software written in the C programming language may be adapted and incorporated. Conformance to the C++ and C language standards should be closely maintained to ensure portability across platforms. Seagull's C++ coding standard [ST99a] will be used as guideline for the creation of C++ source code.

Note that software that is used to develop or investigate engineering concepts independent of the software deliverables of RTO 30 may be written in languages other than C or C++. In particular, Matlab code that is used to investigate engineering tradeoffs, as with the tradeoff investigations associated with the Kalman filter being investigated in Task 2, will not be delivered and will not be written in C or C++.

- FDR1.2: Software will be compliant with the Orbix 3.0 network communication protocol.

The software shall be able to use Iona Technologies Orbix 3.0 C++ implementation of the Common Object Request Broker Architecture (CORBA) specification [OMG99] for distributed object communication.

- FDR1.3: Software will be executable from the FFSim environment.

The software shall be able to operate within the FFSim environment. This requires that the software shall be able to communicate with the FFSim infrastructure software using the FFSim specified software protocols and procedures [ST99b].

- FDR1.4: Software will be capable of running native under Windows NT 4.0, IRIX 6.5, and Solaris 2.6 and 7.0 after compilation.

All source code shall be capable of being compiled and executed on the operating systems and associated compilers listed in Table A.1. Native operating system calls shall be encapsulated to simplify switching platforms.

Table A.1. Required Operating Systems and C/C++ Compilers.

Operating System	Compiler
Windows NT 4.0 (Service Pack 5)	Visual C++ 6.0 (Service Pack 3)
Solaris 2.6 and 7.0	EGCS 1.1.2 [EGCS99], SPARC C++ Compiler 4.2 (if using Orbix)
IRIX 6.5	EGCS 1.1.2

- FDR1.5: Software will be Year 2000 compliant.

All software shall be Year 2000 date compliant. Software shall be able to correctly handle times and dates before, on, or after January 1, 2000. All software shall be tested for Year 2000 compliance. The following company Year 2000 compliance statement will apply to this RTO 30:

Seagull Technology, Inc. hereby warrants that Seagull's aviation and transportation related software products that have been developed and released since June 1997 have been tested to ensure Year 2000 compliance in order to determine that the product(s) will record, store, process, manage and present calendar dates (and data or functions involving or based on calendar dates) falling on or after January 01, 2000 in the same manner and with the same functionality, accuracy, data integrity and performance as the Seagull software records, stores, processes, manages and presents calendar dates (and data involving or based on calendar dates) falling on or before December 31, 1999.

- FDR1.6: The solution will be compatible with the ADS-B message content defined by RTCA SC-186 (MASPS [RTCA98]), including broadcast intent.

ADS-B is a function on an aircraft or a surface vehicle operating within the surface movement area that periodically broadcasts its state vector and other information. An ADS-B message is a packet of formatted data that convey information used in the development of ADS-B reports. Message contents and formats are specific to the ADS-B data link; the MASPS does not address message data fidelity and data structures; thus, RTO 21 and RTO 30 will specify definitions as needed. An ADS-B report is information provided by ADS-B messages received from a transmitting participant. These information elements are available for use by applications external to the ADS-B system.

In this RTO 21 and RTO 30, ADS-B messages and reports will be modeled following the RTCA SC-186 (MASPS [RTCA98]) standards.

- FDR1.7: The solution will be compatible with FFSim Flight Information Service (FIS) specifications.

The FIS message content defined for RTO 21 will be used in RTO 30.

- FDR1.8: The solution will be compatible with FFSim Traffic Information Service (TIS) specifications,.

The TIS message content defined for RTO 21 will be used in RTO 30.

- FDR1.9: The solution will incorporate intent models that represent reasonable cases of potential flight crew deviations from the planned or nominal flight path or data linked intent (next two way points),

Intent models will be coordinated with NASA as well as with RTO 22: Scenario Generation for FFSim. The process will be to enumerate the flight crew deviations and insure that it includes the same flight crew deviations that are being investigated by Matt Jardin at NASA Ames, researching intent inference for CTAS, and the RTO 22 team. All intent models will be fully explained and documented in the Final Report.

- FDR1.10: Models of intent will be capable of independent verification of correctness and completeness,

The plan is to have a pilot, a NASA research engineer, and an aerospace engineer (not currently working on the RTO 30 project) review the models of intent to have a human verification of the intent models. At NASA Ames, we plan to have Matt Jardin review the models. Additionally, a mathematical or experimental verification method will be implemented based on theoretical analysis or data analysis (from synthetic or real data).

- FDR1.11: The solution will gracefully degrade in situations where nearby aircraft flight trajectory information cannot be reliably inferred.

For Task 1, the most recent information about nearby aircraft is used for intent inference whenever this information becomes available. The data does not have to be synchronous nor on time, rather, it simply has to have the appropriate time stamp on the data. For Task 1 and Task 2, the solution will gracefully degrade because the tracking filter (both the Kalman filter and the low pass filter) have the property that they can handle asynchronous or missing data by providing near-term estimates of missing data.

- FDR1.12: The intent inferencing function shall update trajectory intent inferences within the time interval to be defined by the AOP architecture,

At this time the algorithms are being designed to update with a one second time interval, subject to change, based on the update rate and availability of information from ADS-B, FIS, TIS, CPDLC, and other potential data sources.

- FDR1.13: The information confidence assessment shall be applicable to the size and memory constraints of a modern avionics system and in line with the trends in the applicable technologies.

While we really don't want to limit our system to current avionics cpu's, since Free Flight won't go mainstream for at least 5 years, we do want to keep the size and memory constraints in mind while designing this system. Essentially, this FDR will be considered a soft constraint, stating that the cpu requirements will be identified and checked for reasonableness given the current trends in technology.

A summary of the FDRs will also be included in a Final Report.

Appendix B: Algorithms to Convert Grid Weather Data to Triangulated Data

This Appendix describes algorithms suitable for triangulating weather data. While RTO 30 modules have been designed to operate on grid-based weather data, other AOP modules may require either grid-based or triangulated weather data, for example, to define polygon hazard regions for hazardous weather. This Appendix provides the process for this future work, if it is required at some future date.

Figure B.1 illustrates weather data in grid form. In a uniform 2D grid form, each grid cell represents a square region of airspace in a 2D plane. Weather data are typically stored within computers in a grid form, so the benefit of using a grid form is that there are no additional data structures to be created; the typical computer data structure to maintain weather data is an array. As weather moves and weather data are updated, the data structure need not change in size or shape. Finally, the data fusion process for weather data is greatly simplified, since data fusion for two grids with the same grid spacing is much easier than fusing two triangulations.

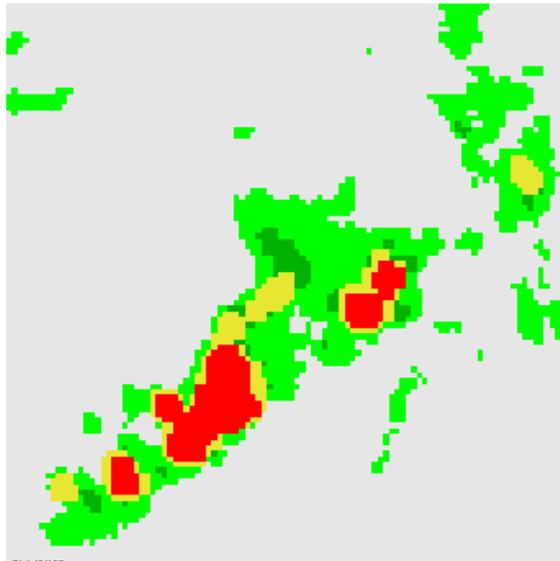


Figure B.1. Grid-based weather data with colors indicating NWS Level.

The triangulation method of Fowler and Little [FL79] is one of the first methods of automatic generation of triangulation networks. Their application was to convert digital terrain elevation map data into a triangular irregular network. The method selects feature points such as ridges or peaks to use in the triangulation. Given these points, they then use a Delaunay Triangulation [BK97, OBS92] of these points to establish a triangulation. The triangulation is established by introducing new feature points into the triangulation until the error between the triangulation surface and the digital elevation surface is below a specified minimum.

The triangulation method of Franklin [F94] uses a top-down construction to approximate a digital terrain map with a triangulation surface. Initially, the triangulation is approximated with two triangles. Then, within each triangle, the elevation point that is furthest above or below the triangle surface is used to create a set of three triangles that better approximates the surface.

When new triangles are created, a check is made between adjacent triangles forming a quadrilateral is made to see if a diagonal swap would better fit the terrain surface. The process repeats until the error between the triangulation surface and the digital terrain surface is below a specified bound.

Recently, a triangulation method by Silva et al [SMK95] has been developed based on greedy cuts. The algorithm works by surrounding the terrain region with a polygon and proceeds to take greedy cuts (or “bites”) out of the polygon. The cuts establish triangular facets which closely approximate the terrain surface. The cuts progress inward towards the part of the polygon which has yet to be processed. Three cuts are considered: ear cutting, greedy biting, and edge splitting. The algorithm may often be faster and require fewer triangles to approximate a given terrain surface.

For weather processing, a three step process similar to the method of Fowler and Little [FL79] is useful for creating a triangulation. First, contour lines for the weather data are created at level curves that correspond to the NWS levels. This is performed using a Laplacian of Gaussian filter for edge detection, as shown in Figure B.2. From these contour lines, a subsampling of points are chosen as feature points to be used in a triangulation, as shown in Figure B.3. These points are then triangulated using a Delaunay Triangulation (which is available as COTS software), and a unique color (weight or NWS level) is assigned to each triangle, as shown in Figure B.4.

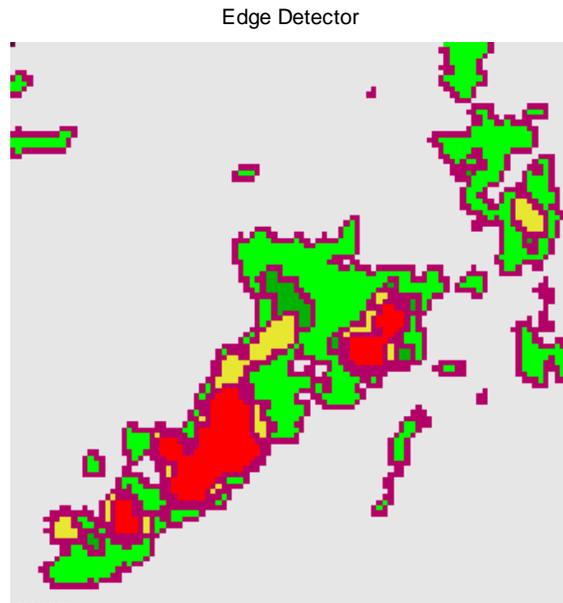


Figure B.2. The first step in the triangulation process is to identify the contour lines that separate each NWS Level.

Edge Sub-Sampling



Figure B.3. The second step in the triangulation process is to sample the contour lines to arrive at a set of feature points for a triangulation.

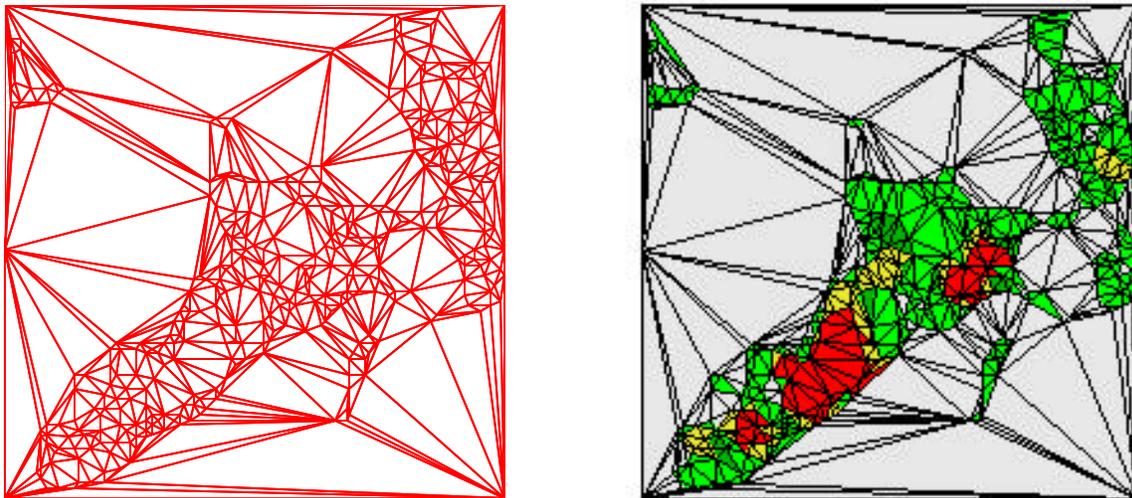


Figure B.4. The third step in the triangulation process is to create the Delaunay Triangulation of the set of feature points (left) and to assign the appropriate unique color (weight or NWS Level) to the triangle (right).

Appendix C: Derivations of CPA and Time-To-CPA for Hazard Detection

In Appendix C, we investigate hazard detection derivations that provide CPA and Time-to-CPA.

C.2 CPA and Time-to-CPA for Line Segment Hazards

In this Appendix we investigate the point-to-point hazard detection problem. We consider the relative motion of a point hazard as it passes by a point model of the ownship. Both stationary and moving point hazards are considered. We start by defining the mathematical terms for relative motion information. The ownship is modeled as point A and the hazard is modeled as point B. The position of the hazard relative to the ownship is located by the vector \vec{r} , the velocity of the ownship is \vec{v}_A , and the velocity of the hazard is \vec{v}_B . The motion of the hazard relative to the ownship is described by the relative velocity \vec{c} :

$$\vec{c} = \vec{v}_B - \vec{v}_A, \quad (\text{C.1})$$

and it is useful to our analysis to use the unit vector \hat{c} in the direction of \vec{c} , defined as:

$$\hat{c} = \frac{\vec{c}}{\|\vec{c}\|}. \quad (\text{C.2})$$

Given these relative motion variables, the time-to-CPA τ and the vector \vec{r}_f pointing to the CPA derived in [KP97] as:

$$\text{Time of CPA:} \quad \tau = -\frac{\vec{r} \cdot \hat{c}}{\hat{c} \cdot \hat{c}}, \quad (\text{C.3})$$

$$\text{Location of CPA:} \quad \vec{r}_f = \hat{c} \times (\vec{r} \times \hat{c}) \quad (\text{C.4})$$

Figure C.1 illustrates the relationships of these vectors. Note that the CPA occurs on the zero range rate line, and the relative motion of the hazard is perpendicular to the zero range rate line. As shown, the normal convention for a coordinate system is to have the point model of the aircraft at the origin of a system with y-axis in the direction of \vec{v}_A and the x-axis in the direction of the aircraft right wing. This assumes that the aircraft has no sideslip angle and flies in the horizontal plane.

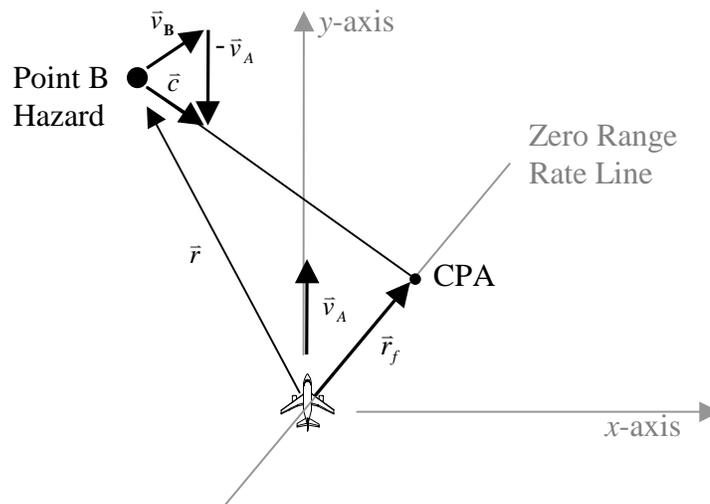


Figure C.1. Geometry of the point hazard relative to the point model of the ownship.

Note that the Time-to-CPA and CPA equations (C.3) and (C.4) apply to both the case where the point hazard is stationary ($\bar{v}_B=0$) and moving with constant velocity. While these equations constitute the solution to the point-to-point hazard detection problem, they also form the basis for the solutions to the point-to-segment, point-to-polygon, and point-to-polyhedron hazard detection problems discussed next.

C.2 CPA and Time-to-CPA for Line Segment Hazards

In this section we investigate the point-to-segment hazard detection problem. We consider the relative motion of a line segment hazard as it passes by a point model of the ownship. The line segment hazard is assumed not to rotate and to have all points moving with the same constant horizontal velocity \bar{v}_B ; thus, all points have the relative velocity \bar{c} with respect to the point model of the ownship. The geometry of the problem is described by Figure C.2. Most of the variables are the same as the point-to-point hazard detection problem, but now the hazard is described by two endpoints located at \bar{r}_1 and \bar{r}_2 relative to the ownship position. Because of the constant velocity assumption with no rotation, there is only one zero range rate line for the entire hazard segment, as shown in Figure C.2.

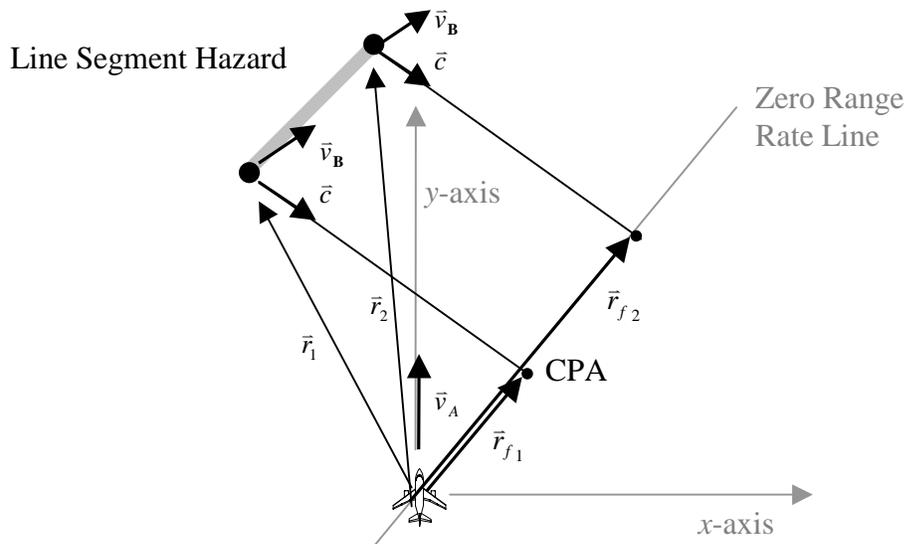


Figure C.2. Geometry of the line segment hazard relative to the point model of the ownship.

Equations (C.3) and (C.4) apply to each of the endpoints of the line segment hazard, providing the following range and time information:

$$\text{Time to zero range rate line for hazard end point 1:} \quad \tau_1 = -\frac{\bar{r}_1 \cdot \bar{c}}{\bar{c} \cdot \bar{c}}, \quad (\text{C.5})$$

$$\text{Location on zero range rate line for hazard end point 1:} \quad \bar{r}_{f1} = \hat{c} \times (\bar{r}_1 \times \hat{c}), \quad (\text{C.6})$$

$$\text{Time to zero range rate line for hazard end point 2:} \quad \tau_2 = -\frac{\bar{r}_2 \cdot \bar{c}}{\bar{c} \cdot \bar{c}}, \quad (\text{C.7})$$

$$\text{Location on zero range rate line for hazard end point 2: } \bar{r}_{f_2} = \hat{c} \times (\bar{r}_2 \times \hat{c}). \quad (\text{C.8})$$

There are two possible cases that determine the CPA for the line segment. First, the line segment may pass ahead of or behind the ownship point model. Second the line segment may collide with the ownship point model. These two cases are mathematically determined by:

Case 1. Segment Passes In Front or Behind Ownship ($\bar{r}_{f_1} \cdot \bar{r}_{f_2} = 1$)

CPA: \bar{r}_f direction is the same as \bar{r}_{f_1} (or \bar{r}_{f_2}) and magnitude $\|\bar{r}_f\| = \min \{ \bar{r}_{f_1}, \bar{r}_{f_2} \}$

Time-to-CPA: If $\|\bar{r}_f\| = \|\bar{r}_{f_1}\|$, then $\tau = \tau_1$, otherwise $\tau = \tau_2$.

Case 2. Segment Collides with Ownship ($\bar{r}_{f_1} \cdot \bar{r}_{f_2} = -1$)

CPA: $\bar{r}_f = 0$

Time-to-CPA: $\tau = \tau_1 + \frac{\|r_{f_1}\|}{\|r_{f_1} + r_{f_2}\|} (\tau_2 - \tau_1)$

The point-to-segment hazard detection problem is valid for both moving hazard segments and stationary hazard segments.

4.1.2.3 CPA and Time-to-CPA for Polygon Hazards

In this section we investigate the point-to-polygon hazard detection problem. We consider the relative motion of a polygon hazard as it passes by a point model of the ownship. The polygon hazard is assumed not to rotate and to have all points moving with the same constant horizontal velocity \bar{v}_b ; thus, all points have the relative velocity \bar{c} with respect to the point model of the ownship. The geometry of the problem is described by Figure C.3. The hazard is described by a list of n vertices v_n located at $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_n$ relative to the ownship position. Because of the constant velocity assumption with no rotation, there is only one zero range rate line for the entire polygon, as shown in Figure C.3.

Equations (C.3) and (C.4) apply to each of the endpoints of the line segments that compose the polygon hazard, and the CPA and Time-to-CPA equations for the line segment hazards apply as well. Each of the vertices can be investigated to determine a time and location when they cross the zero range rate line:

$$\text{Time to zero range rate line for hazard vertex point } i: \quad \tau_i = -\frac{\bar{r}_i \cdot \bar{c}}{\bar{c} \cdot \bar{c}}, \quad (\text{C.9})$$

$$\text{Location on zero range rate line for hazard vertex point } i: \quad \bar{r}_{f_i} = \hat{c} \times (\bar{r}_i \times \hat{c}) \quad (\text{C.10})$$

The polygon hazard detection algorithm for determining the CPA for the polygon computes the CPA for n polygon edges (line segments) in order to identify the closest CPA:

$$\text{CPA} = \min \{ \text{CPA}_1, \text{CPA}_2, \text{CPA}_3, \dots, \text{CPA}_n \} \quad (\text{C.11})$$

and additionally, the corresponding Time-to-CPA is determined. The point-to-polygon hazard detection equations are valid for both moving polygon hazards as well as stationary polygon hazards.

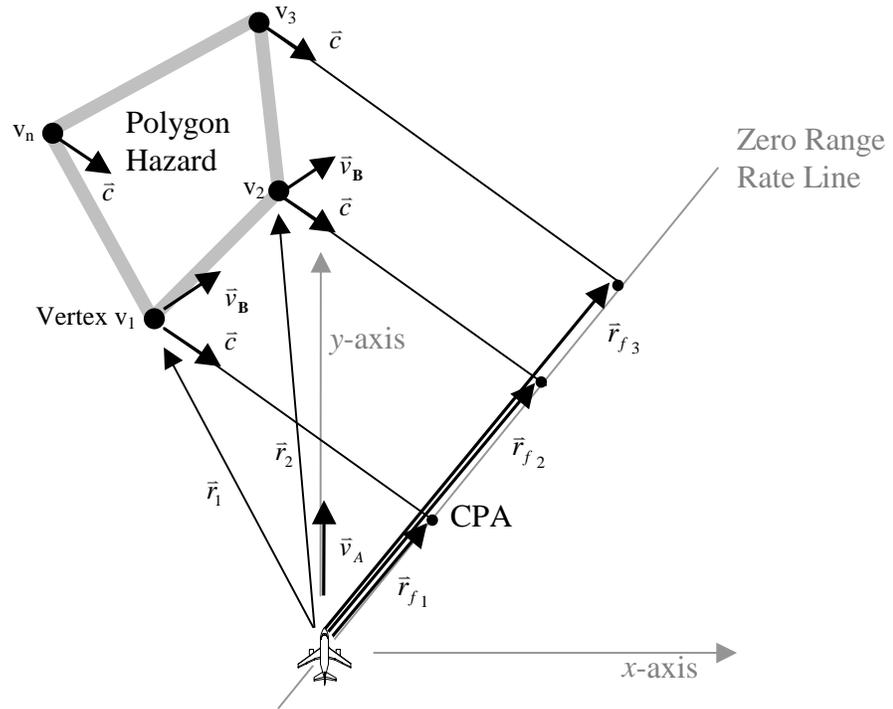


Figure C.3. Geometry of the polygon hazard relative to the point model of the ownship.

4.1.2.4 CPA and Time-to-CPA for Polyhedron Volume Hazards

In this section we investigate the point-to-polyhedron hazard detection problem. We consider the relative motion of a polyhedron hazard as it passes by a point model of the ownship. The polyhedron hazard is assumed not to rotate and to have all points moving with the same constant horizontal velocity \bar{v}_B ; thus, all points have the relative velocity \bar{c} with respect to the point model of the ownship. We only consider polyhedron volumes of a particular type, since the problem of determining the point-to-polyhedron hazard problem for arbitrary polyhedrons is not needed for the AOP domain. In particular, the type of polyhedron of interest for the AOP domain has a horizontal top and bottom face with identical shape, and vertical faces that connect the two. The geometry of such a polyhedron is shown in Figure C.4. The hazard is described by a list of n vertices v_n located at $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_n$ which identify the location of the top horizontal polygon relative to the ownship position, and a height (thickness) h . Finally, we consider only cases where the aircraft is flying at an altitude between the bottom hazard polygon and the top hazard polygon surface flight levels. When the aircraft is holding an altitude above the top horizontal polygon hazard surface, the algorithm returns the height above the top horizontal hazard surface. Likewise, when the aircraft is holding an altitude below the bottom horizontal polygon hazard surface, the algorithm returns the depth below the bottom horizontal hazard surface. When the aircraft is climbing or descending towards the polygon hazard, then a cross section (slice) of the polygon hazard is computed, and the conflict detection problem is processed as a 2D point-to-polygon type problem; the details of this process are not outlined here.

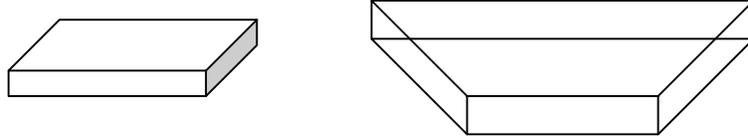


Figure C.4. Example polyhedrons modeled for the point-to-polyhedron hazard detection problem.

The algorithm for the point-to-polyhedron problem implements the point-to-polygon hazard detection solution and the point-to-segment hazard detection solution. The point-to-polygon hazard detection solution is applied to the top (or bottom) set of n vertices that describe the top (or bottom) polygon of the polyhedron:

$$\text{Closest Top Point: } CPA_T = \min\{CPA_1, CPA_2, CPA_3, \dots, CPA_n\} \quad (\text{C.12})$$

Next, the n line segments that connect the top polygon face to the bottom polygon face are run through the point-to-segment hazard detection algorithm:

$$\text{Closest Side Point: } CPA_S = \min\{CPA_1, CPA_2, CPA_3, \dots, CPA_n\} \quad (\text{C.13})$$

From these two results, the CPA is identified:

$$CPA = \min\{CPA_T, CPA_S\} \quad (\text{C.14})$$

This solution is valid because of the type of polyhedron hazard defined for this problem, and the constant velocity, no rotation, and horizontal motion assumptions.