JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 2, March–April 2006

Intent Inference with Path Prediction

Jimmy Krozel*
Metron Aviation, Inc., Herndon, Virginia 20170
and
Dominick Andrisani II†
Purdue University, West Lafayette, Indiana 47906

In today's air-traffic management system, the intent of an aircraft is revealed in its flight plan and a pilot's adherence to published navigation routes. In a future free-flight environment, aircraft might be allowed to fly any route they choose. Intent will likely be broadcast in the form of trajectory change points (TCPs), for instance, up to four TCPs in an automatic dependent surveillance—broadcast message. However, if broadcast TCPs do not accurately represent intent, do not exist, or do not get received, then a nearby aircraft or ground-monitoring system has a need to infer the pilot's intent in real time. In this paper, a method of inferring intent, which is based on artificial intelligence models and a process for best fitting an intent model to observed aircraft motion, is investigated. Horizontal, vertical, and speed dimensions are first investigated independently, and then combined, including sequences of actions, to fully explain the three-dimensional guidance and navigation plan of an aircraft. Finally, the inferred intent is used as a basis to predict a path from the current location of the aircraft into the future.

Introduction

As new free-flight[1] procedures remove jetway routing, positive control, and other constraints, an added emphasis will be placed on distributed control techniques. In free flight, a data link technology, for instance, automatic dependent surveillance—broadcast (ADS-B), will be used to communicate state and intent data between aircraft. State and intent information [e.g., position, speed, and trajectory change points (TCPs)] will be exchanged between aircraft so that conflicts between aircraft can be detected and resolved in a distributed manner.

Real-time intent inference is needed in free flight because there is no guarantee, even if flight plans or intent are broadcast to nearby aircraft, that these flight plans will be followed. Even with acceptable required navigation performance levels which stipulate how closely the flight crew should locate the aircraft relative to the intent, there still is the possibility that the crew might change 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 because of poor wind predictions, weather avoidance, or emergency diversions.

Figure 1 illustrates the intent inference problem. In this situation, an aircraft has lateral deviations caused by flight technical errors while flying to a waypoint, 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 pointed that way), 2) flying directly to the next waypoint ignoring the weather (because the flight plan indicates this), or 3) flying south of the storm and skipping a waypoint (because airlines have policies to avoid storms and arrive on schedule). In general, the problem of intent inference is to determine: What is the intent of the aircraft?

While intent data is broadcast between aircraft, the following cases can occur: 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; or 3) the broadcast intent message is missing (because of data dropout or an unequipped aircraft), and the intent needs to be inferred. In each case, the intent inference algorithm designed in this paper produces a predicted intent of the aircraft and a prediction of the future flight path.

Related Literature

In the literature, Zhao et al.[2] specify several classifications of pilot intent models, including motive intent, objective intent, trajectory intent, and cost intent. Several studies, including our own, have found that many of these types of intent can be captured in some form of a knowledge database using artificial-intelligence models. Furthermore, these knowledge databases can be searched or managed in ways that allow for the operator's intent to be understood.

Next, we review the work of several researchers developing human–machine interaction theories and/or systems. First, we consider systems that reason about discrete actions taken by the pilot. In these situations, the system acts as an observer "over the shoulder of the pilot" watching the switches, modes, and discrete control choices made by the pilot:

1) The first system is operator plan analysis logic (OPAL),[3–5] in which plan-goal graphs are used to explain the causal relationship between the observed discrete actions and the goals (intent) of the operator. OPAL identifies conflicts that exist in goals as well as conflicts that exist with plans.

2) The next is the operator function model expert system (OFMspert),[6,7] in which an expert system uses a blackboard architecture, and operator function models (OFM), a heterarchic-hierarchic network of nodes that represent operator control functions, to replicate and infer an explanation of the actions of the human operator.

3) The third system is generalized plan recognition,[8] in which a deductive inference system performs intent inference by recognizing the plan an operator is implementing, based on observations of actions, and an action taxonomy (an exhaustive set of sequences of discrete admissible actions).

4) Next is event tracking using Soar,[9] in which the Soar knowledge-based system reasons about the decisions made in the task of the operator, triggering rules that create problem space, goals, and subgoals that lead to an explanation.

5) The last system is crew activity tracking system,[10,11] in which a normative model based on the OFM captures the discrete human–machine interactions between the pilot and the control system. Each activity is expressed in terms of binary states and an and/or tree.

Received 10 December 2004; revision received 2 June 2005; accepted for publication 3 June 2005.

*Chief Scientist, Research and Development, 131 Elden Street, Suite 200. Associate Fellow AIAA.

†Associate Professor, School of Aeronautics and Astronautics.

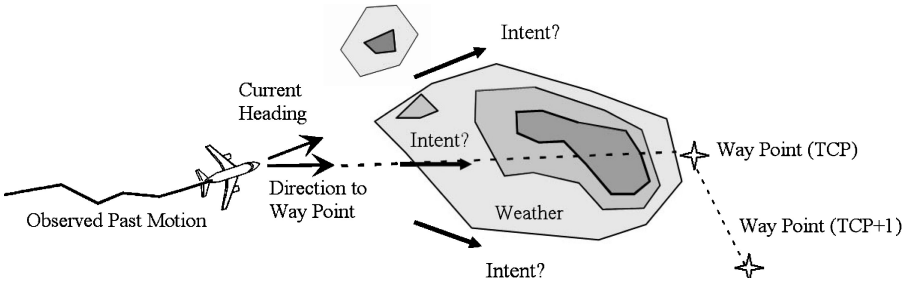

Fig. 1 What is the intent of this aircraft?

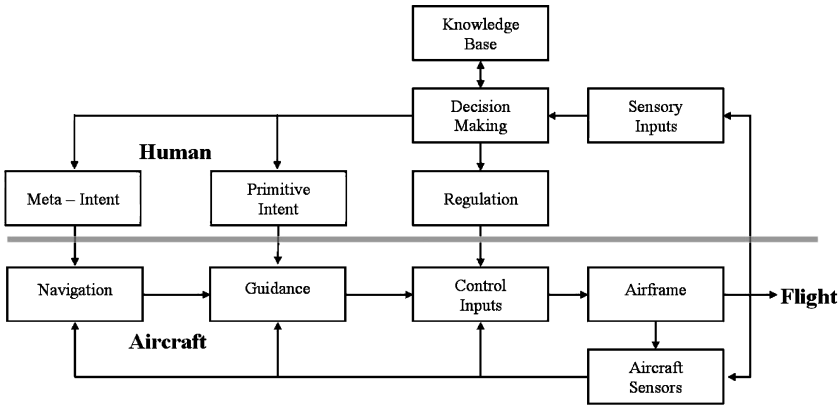

Fig. 2 Human decision-making process for flying an aircraft.

The intent inference method described in this paper is very similar to these discrete intent inference methods because discrete and/or type graphs are used to identify the intent. However, these discrete intent inference methods rely on the binary or discrete nature of the problem to classify the motion into a represented state of the and/or graph, plan/goal graph, or to an OFM. In the technique described in this paper, the match is made with a continuous (rather than discrete) matching, which is novel to the approach.

The second set of systems monitors the continuous motion of an aircraft from an external perspective (no pilot control inputs are observed, only the motion of the vehicle is observed); these methods are referred to as continuous intent inference:

1) The first is mission segment identification,[12,13] in which neural-network logic or fuzzy logic classifies the state inputs to linguistic maneuvers (outputs). When a neural-network approach is pursued, a sufficient amount of training data must be available spanning all possible classifications. When the fuzzy-logic approach is pursued, tuning of the parameters associated with fuzzy membership functions must be performed.

2) The second is conformance monitoring[14,15] in air traffic control (ATC), in which fault-detection techniques are used to identify when an aircraft is conforming (vs nonconforming) to the flight plan.

Note that neither of the continuous intent inference methods mentioned here reason about the causes of the change in vehicle motion. In contrast, the continuous intent inference method presented in this paper links the geometry of hazardous weather, special-use airspace (SUA), and other factors directly to the inferred intent.

In short, the intent inference technique described in this paper captures the most salient properties of both discrete and continuous intent inference methods. Tracking filter-based prediction, for instance, with a Kalman filter, although useful for short time periods ahead of an aircraft, does not address the intent inference problem when it comes to explaining factors such as avoiding weather, SUA, and turbulence, which are factors not easily incorporated into the aircraft equations of motion modeled in most tracking filters. Using a tracking filter is most applicable when *coupled* with the intent inference method as we discuss later in this paper.

The next sections of this paper detail the theory behind intent inference and the method of path prediction, present some examples, and state our conclusions.

Intent Inference Theory

Intelligent inference is based on the work of Krozel and Andrisani.[16–18] Our solution incorporates any broadcast or data-link information from nearby aircraft being tracked. Furthermore, when domain knowledge data are available, for example, traffic, weather, terrain, standard terminal arrival routes, and SUA, these data are taken into consideration in intent models. The intent inference system is able to verify that a tracked aircraft is following the broadcast intent, and, if the aircraft is seemingly following some other intent, then it will identify the most plausible intent(s) for the aircraft being tracked. Finally, the intent inference system uses the most plausible intent to predict the future path of the vehicle being tracked.

Basis of Theory

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

Consider a closed-loop model for intelligent flight control postulated by Stengel.[19] This model includes sensing, regulation, and decision making, as in Fig. 2. The sensory inputs to the human are associated with seeing (states from displays, aircraft out the window, weather, runways, etc.), feeling (forces felt by the body), and hearing (sounds correlated with speed, engine noise, and inner ear effects from gravity)—taste and smell are assumed negligible in the model. 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. Through the decision-making process the pilot plans and sets goals to form navigation and guidance command inputs. Such decision making is dependent on the pilot's knowledge base of both rules of flight and past experience.

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 instance, autopilots allow the pilot to command a constant heading angle or airspeed. The maneuvering

precision will differ based on whether flight is performed by a pilot or by automatic control.

The human decision-making model has a hierarchical structure. In the inner loops, stability augmentation and reflexive control are performed with fast update rates, perhaps 1–10 Hz. In the outer loops, guidance and navigation changes can be made every 1–10 min. The central nervous system supports such a hierarchy for decision making by providing a structure that includes declarative actions, procedural actions, and reflexive actions—the basis of the intelligent flight control system of Stengel.[19] In intent inference, this hierarchical structure and separation between declarative actions from reflexive actions are also modeled.

We consider the problem of predicting the intent and path of an aircraft being tracked by an outside observer. These concepts are similar but different:

1) Intent inference determines *what* the tracked vehicle is most likely attempting to do.

2) Path prediction determines *how* the tracked vehicle will most likely accomplish the intent.

The path of an aircraft is directly a result of the control inputs determined by regulation and/or guidance decisions. The pilot intent is to use guidance and control inputs to follow a flight plan. 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.

Because of the hierarchical nature of decision making, navigation is likely to remain constant for a long time as inner-loop guidance commands change more frequently. If path prediction is pursued with very short look-ahead times, for instance 1–5 min, then the outer-loop navigation commands can be assumed constant. However, as look-ahead times become longer, for instance, 5 to 20 min, one should expect that the navigation commands can change, depending on how far along the process the pilot is in terms of following the outer-loop 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 (e.g., over 5 min) to include declarative or procedural decisions. In our work, we call pilot intent at the guidance level a "primitive intent" and the pilot intent at the navigation level a "meta-intent," as illustrated in Fig. 2.

Intent inference and path prediction are solved simultaneously. Predicting a pilot's intent can be abstracted into the process presented in Fig. 3. Later in this paper, this process is extended to show how sequences of primitive intent guidance commands can be linked together to infer navigation tasks and to predict further in time according to the meta-intent models.

Intent Inference Knowledge Base

Knowledge engineering and operations analysis are used to build a knowledge base consisting of a set of plausible intent models.

The aircraft flight domain has a rich set of intent models that will reside in the intent inference knowledge base, including but not limited to the models listed in Table 1. A database of primitive intent models holds all of the associated constants and parameters needed to provide the direction that an aircraft should go given any arbitrary set of initial conditions. A unique index is also created to process each intent model, although some intent models are relatively

Table 1 Primitive intent models

Primitive intent model description	Dimension	Period	Index
Direct to TCP	H	Steady state	$I^H = 1$
Direct to TCP + 1	H	Steady state	$I^H = 2$
Direct to TCP + 2	H	Steady state	$I^H = 3$
Direct to TCP + 3	H	Steady state	$I^H = 4$
Hold course	H	Steady state	$I^H = 5$
Return to flight leg	H	Steady state	$I^H = 6$
Avoid SUA (horizontal)	H	Steady state	$I^H = 7$
Violate SUA (horizontal)	H	Steady state	$I^H = 8$
Follow jet route (jet route from a database)	H	Steady state	$I^H = 9$
Direct to airport (destination, alternate, or database airport)	H	Steady state	$I^H = 10$
Direct to fix (fix selected from a fix database)	H	Steady state	$I^H = 11$
Direct to navaid (Navaid from a Navaid database)	H	Steady state	$I^H = 12$
Hold coordinated turn (1.5 deg/s or 3 deg/s; left or right)	H	Transient	$I^H = 13$
Avoid aircraft (horizontal)	H	Transient	$I^H = 14$
Violate aircraft protected airspace zone (horizontal)	H	Transient	$I^H = 15$
Avoid weather cell (horizontal)	H	Transient	$I^H = 16$
Hold TCP altitude (altitude given)	V	Steady state	$I^V = 1$
Climb/descend to TCP altitude (climb rate given)	V	Steady state	$I^V = 2$
Hold TCP + 1 altitude (altitude given)	V	Steady state	$I^V = 3$
Climb/descend to TCP + 1 altitude	V	Steady state	$I^V = 4$
Hold TCP + 2 altitude (altitude given)	V	Steady state	$I^V = 5$
Climb/descend to TCP + 2 altitude	V	Steady state	$I^V = 6$
Hold TCP + 3 altitude (altitude given)	V	Steady state	$I^V = 7$
Climb/descend to TCP + 3 altitude	V	Steady state	$I^V = 8$
Return to flight plan altitude	V	Transient	$I^V = 9$
Descend to airport	V	Steady state	$I^V = 10$
Avoid aircraft conflict (vertical)	V	Transient	$I^V = 11$
Violate aircraft protected airspace zone (vertical)	V	Transient	$I^V = 12$
Avoid SUA (vertical)	V	Transient	$I^V = 13$
Avoid weather (vertical)	V	Transient	$I^V = 14$
Hold planned speed	S	Steady state	$I^S = 1$
Return to flight plan speed	S	Transient	$I^S = 2$
Speed to meet required time of arrival restriction	S	Transient	$I^S = 3$
Speed to meet TCP time-to-go (TTG) requirement	S	Transient	$I^S = 4$
Speed to meet TCP + 1 TTG requirement	S	Transient	$I^S = 5$
Speed to meet TCP + 2 TTG requirement	S	Transient	$I^S = 6$
Speed to meet TCP + 3 TTG requirement	S	Transient	$I^S = 7$
Speed to meet miles-in-trail (MIT) restriction	S	Transient	$I^S = 8$
Decelerate to airport	S	Transient	$I^S = 9$
Avoid aircraft protected airspace zone (speed)	S	Transient	$I^S = 10$
Avoid weather cell (speed)	S	Transient	$I^S = 11$

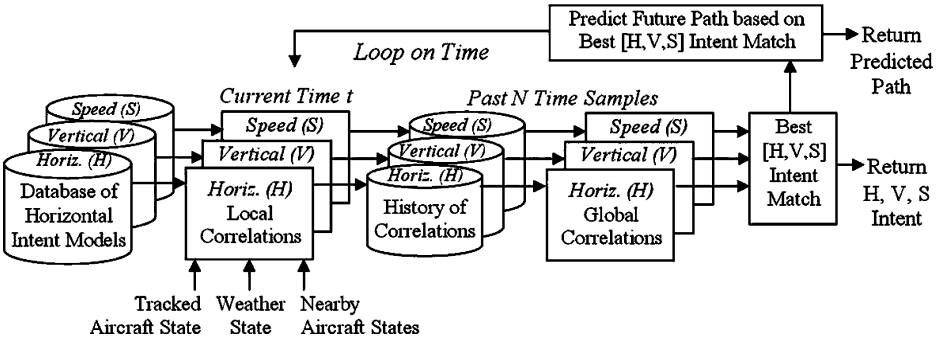

Fig. 3 Identifying intent models that best match observed motion.

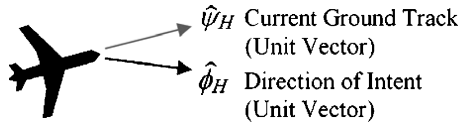

Fig. 4 Local path correlation—horizontal dimension.

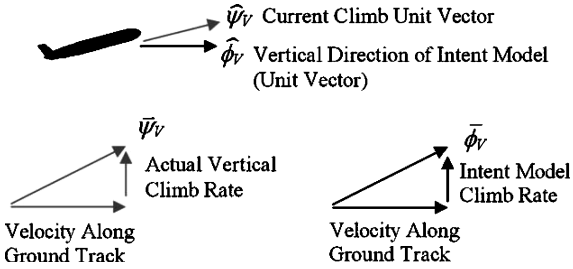

Fig. 5 Local path correlation—vertical dimension.

simple, for instance, the model to hold heading, and others are more complex, for instance, the model to avoid weather. Intent models are defined for horizontal (H), vertical (V), and speed (S) dimensions; we use the notation [H, V, S] to refer to these dimensions.

The intent inference knowledge base also requires domain knowledge. Domain knowledge is used to build a situation assessment model—a map of the situation—which includes the factors that might influence the pilot's decision making. Our particular application includes inputs from the following:

1) ADS-B[20] includes the broadcast of state and intent data in [H, V, S] dimensions. We focus on a data link, which provides up to four TCPs, although, in general, the approach also applies to the case of n TCPs (TCP, TCP + 1, ..., TCP + n − 1).

2) Flight information services[21] includes weather and SUA constraints.

3) Onboard databases provide static data, including airport, navigation, SUA, and terrain elevation data.

Intent Correlation

The actions of a decision maker can be observed and analyzed both locally (instantaneously) and globally (over a time window) in the [H, V, S] dimensions.

As shown in Fig. 4, the horizontal (H) variables that determine the local path correlation are a unit vector $\hat{\psi}_H$ in the ground track direction and a unit vector $\hat{\phi}_H$ in the optimal direction to proceed according to the horizontal intent model. The local path correlation is defined by the dot product:

$$L(t) = \hat{\psi}_H(t) \cdot \hat{\phi}_H(t) \tag{1}$$

The local correlation will indicate (locally) if the aircraft is heading toward the direction of the intent model (dot product of 1) or in some sense opposite of the intent model (dot product of −1). This dot product acts as a good measure of correlation between the intent to fly according to the logic of the intent model and the current aircraft state as indicated by the current ground track.

The global analysis considers the correlation between a history of state variables with a series of decisions that would support a consistent intent. For a global correlation ρ, one integrates (sums up) the local correlation function L over the observed flight-path data:

$$\rho(N) = \frac{1}{N}\sum_{i=1}^{N} L(i) = \frac{N-1}{N}\left[\rho(N-1) + \frac{1}{N-1}L(N)\right] \tag{2}$$

where at time $t = N\Delta$ (the Nth time increment of duration Δ) the global correlation is computed from the local correlation function $L(i)$. The right side of Eq. (2) represents a recursive form of the summation. The global correlation has a range: $-1 \le \rho \le 1$ for the horizontal dimension.

As shown in Fig. 5, the vertical (V) variables that determine the local path correlation are a unit vector $\hat{\psi}_V$ in the direction of the climb vector $\bar{\psi}_V$ from the ground track and a unit vector $\hat{\phi}_V$ in the optimal climb direction from the ground track to proceed according to the vertical intent model. The local path correlation is defined by the dot product:

$$L(t) = \hat{\psi}_V(t) \cdot \hat{\phi}_V(t) \tag{3}$$

The global correlation is defined by Eq. (2); it has a range $-1 \le \rho \le 1$ for the vertical dimension.

Instead of a dot product, the speed (S) dimension is processed with a speed ratio. Let the current aircraft speed be V_{actual}. Each intent model dealing with speed will derive an intent speed V_{intent}. Both of these speeds will be a function of where the aircraft is on a path. The local speed correlation is defined by the ratio:

$$L(t) = V_{\text{actual}}/V_{\text{intent}} \tag{4}$$

This function will always be positive because speeds are always greater than zero. Because Eq. (4) can be greater or less than one, the global correlation will be a number in a range between zero and infinity, with one indicating that the vehicle has a speed consistent with the intent model. The global correlation is defined by Eq. (2).

In making this choice for local correlation, we have recognized that speed is inherently different than the other variables used to define horizontal or vertical path correlation functions. Speed is a scalar function, while the quantities used in the horizontal or vertical path correlation functions were vectors. With vectors, a dot product is appropriate to capture if two path directions are aligned. With scalars, the simple division is appropriate to indicate if the scalar is at or near the intent value. The dot product provides normalization to ±1. With a scalar there is still a normalization to 1, but the local correlation can be greater or less than 1 and is always positive.

Weighting Historical Data

In computing global correlation functions in real-world applications, control over the time history of data is important as well as the weighting given to past data. Two methods are as follows:

Fading Memory

Past data points are weighted less and less. The fading memory factor f ($f = 1$ for no fading, $0 < f < 1$ for some fading) is used to exponentially reduce the weighting on past data points:

$$\rho_f(N) = \frac{1}{G_N}\sum_{i=1}^{N} f^{N-i}L(i) = \frac{G_{N-1}}{G_N}\left[f\rho_f(N-1) + \frac{L(N)}{G_{N-1}}\right] \tag{5}$$

where

$$G_N = \sum_{i=1}^{N} f^{N-i} = G_{N-1} + f^{N-1}$$

and initially $G_1 = 1$. A fading memory can be useful, for instance, in military applications where the intent might be changing quite frequently.

Moving Window

Only data points between time $(N - M + 1)\Delta$ and $N\Delta$ are included in the summation (M terms in the summation).

$$\rho_M(N) = \frac{1}{M}\sum_{i=N-M+1}^{N} L(i) = \frac{N}{M}\rho(N) - \frac{N-M}{M}\rho(N-M) \tag{6}$$

A moving window can be useful, for instance, in ATC, where pilots hold their intent constant for a considerable amount of time while flying flight legs.

A general strategy for handling these forms is to compute $\rho_f(N)$ recursively and to store past values in a circular buffer. Then, if required, the moving window correlation $\rho_M(N)$ can be computed from $\rho_f(N)$ and the stored $\rho_f(N-M)$. In this strategy, each intent model must set up and manage its own circular buffer and compute its own local correlation function.

Naïve Intent Model

As a default, the naïve intent model is the intent model that is invoked when all other intent models fail. It is designed to be simple and robust, applicable to a wide variety of aircraft types, and essentially, is on all the time ready to provide a simple extrapolation of the observed motion. This model encompasses each of the three [H, V, S] degrees of freedom. Estimated states from Kalman filters are the input to this intent model. Redundancy in the estimation results from a bank of six decoupled Kalman filters, exploited to produce better overall estimates of the aircraft velocity, heading, and vertical speed. Average quantities form the basis of a straight-line prediction algorithm. The prediction time is set to T_p seconds ahead (a user-defined parameter). The prediction time is fairly short (typically 30 s) because the model is so simple. Acceleration terms (second derivative terms) are ignored in the prediction, because the estimated acceleration terms should be highly sensitive to noise and thus not useful for long-term trajectory prediction. The purpose of the naïve intent model is to provide a straight-line prediction of the aircraft motion in the worst case that no other primitive intent or meta-intent model explains the motion of the aircraft. Primitive and meta-intent models are described next.

Primitive Intent Models

The local and global correlation functions operate on a comparison between the aircraft state data and intent models. Table 1 lists the primitive intent models that have been implemented. Each primitive intent model returns a heading, climb rate, or speed that the aircraft would most likely be using if the pilot were attempting to follow the logic of the primitive intent model. Although space limitations do not permit a full description of all of the primitive intent models implemented, an example primitive intent model is presented in the Appendix to illustrate how they are designed.

Next, to identify which primitive intent model best correlates with the observed aircraft motion, ranking functions are implemented. Ranking functions rank the global correlation results for each [H, V, S] dimension. All of the primitive intent models are compared to one another:

1) *Horizontal:* All horizontal intent model outputs are ranked at time $t = i\Delta$ according to

$$\rho^H_{\text{Best}}(i) = \max\left[\rho^H_1(i), \rho^H_2(i), \dots\right] \tag{7}$$

2) *Vertical:* All vertical intent model outputs are ranked at time $t = i\Delta$ according to

$$\rho^V_{\text{Best}}(i) = \max\left[\rho^V_1(i), \rho^V_2(i), \dots\right] \tag{8}$$

3) *Speed:* The aircraft is said to be exhibiting this intent if the speed ratio correlation function is close to 1. Closeness to 1 is ranked highest, and so the ranking function involves the subtraction of 1 and the application of the absolute value. All speed intent model outputs are ranked at time $t = i\Delta$ according to

$$\rho^S_{\text{Best}}(i) = 1 + \min\left[\left|\rho^S_1(i) - 1\right|, \left|\rho^S_2(i) - 1\right|, \dots\right] \tag{9}$$

Note that the horizontal correlation functions, vertical correlation functions, and speed ratio correlation functions used in Eqs. (7–9) can take any of the forms $\rho(\,)$, $\rho_f(\,)$, or $\rho_M(\,)$. In the event of a tie, we give preference to an intent model that is associated with TCP first, then TCP + 1, then TCP + 2, etc.

Meta-Intent Models

Simply comparing (ranking) primitive intent models results in an inferred intent that is based on guidance information only (refer back to Fig. 2). The purpose of creating meta-intent models is to consider navigation information, which involves a sequence of guidance tasks. Additionally, these navigation tasks can couple [H, V, S] decisions in the meta-intent logic. A sequence of meta-intent required state transitions T_i are identified to correlate a meta-correlation, as illustrated in Fig. 6.

As an example, a meta-intent model for flying a flight plan would involve the primitive intent models in the sequence identified in

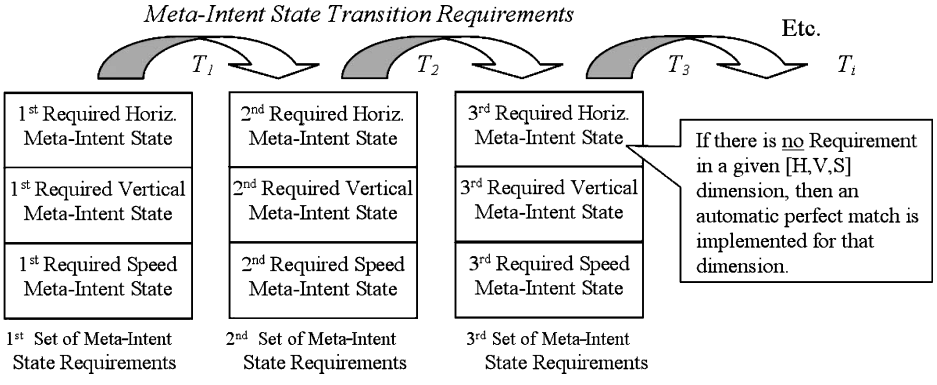

Fig. 6 Meta-intent models transition between sets of requirements.

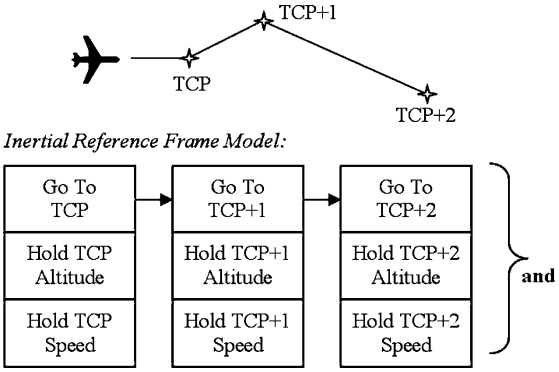

Fig. 7 Meta-intent model for fly flight plan.

Fig. 7. The and condition indicates that all three dimensions must be matched in order for the meta-intent model to correlate with the observed data. Later, a predicate logic will be described so that and and or conditions can be considered when constructing meta-intent models.

Next, a notation system is introduced for reasoning about primitive intent models and meta-intent models. A meta-intent model is defined by a sequence of requirements R_j. Given a meta-intent model with the following sequence of R_j,

$$R_1 = \begin{bmatrix} I_1^H & I_1^V & I_1^S \end{bmatrix} \rightarrow R_2 = \begin{bmatrix} I_2^H & I_2^V & I_2^S \end{bmatrix}$$

$$\rightarrow R_3 = \begin{bmatrix} I_3^H & I_3^V & I_3^S \end{bmatrix} \ldots \tag{10}$$

I_j^H with $j = 1, 2, 3, \ldots$ identifies the name or intent model index of one of the horizontal primitive intent models, defined as the jth horizontal meta-intent model requirement. For instance, $I_1^H =$ "Direct to TCP + 1," or equivalently $I_1^H = 2$ according to Table 1. Similarly for the vertical intent $I_1^V =$ "Avoid Weather Cell (Vertical)," or equivalently $I_1^V = 14$, and for speed intent $I_1^S =$ "Hold Planned Speed" or equivalently $I_1^S = 1$. In the sequence just given, the meta-intent model requirement denoted R_1 (which includes simultaneous primitive intent model requirements for I_1^H and I_1^V and I_1^S) needs to be followed by R_2, which needs to be followed by R_3 for this meta-intent model. The indexes of the [H, V, S] dimension primitive intent models are denoted: 1) kth horizontal primitive intent model index $I^H = k$, where $k = 1$ through 16; 2) lth vertical primitive intent model index $I^V = l$, where $l = 1$ through 14; and 3) mth speed primitive intent model index $I^S = m$, where $m = 1$ through 11. The aforementioned examples for $R_1 = [I_1^H \;\; I_1^V \;\; I_1^S]$ can be summarized using the notation of the primitive intent model indexes as $R_1 =$ ["Direct to TCP + 1," "Avoid Weather Cell (Vertical)," "Hold Planned Speed"] or with the notation $R_1 = [2 \;\; 14 \;\; 1]$.

As time moves forward, the global correlation functions are computed for each primitive intent model and ranked (Fig. 3). At any time denoted with index i, the [H, V, S] primitive intent models with the best global correlation functions are denoted $I_{\text{Best}}^H(i)$, $I_{\text{Best}}^V(i)$, and $I_{\text{Best}}^S(i)$. For example, if "Direct to TCP + 1" is the best primitive intent model at time i then $I_{\text{Best}}^H(i) = 2$.

The value of the primitive global correlation function is a function of the primitive intent model index I and the time index i. Two ways to represent this are possible as in the examples:

1) $\rho_2^S(i) = \rho(I^S = 2, i)$ represents the global correlation value for "return to flight plan speed" requirement at time i.

2) $\rho_{\text{Best}}^S(i) = \rho[I_{\text{Best}}^S(i), i]$ represents the global correlation value for the best speed primitive intent model at time i.

Both notations are used in the following text depending on convenience.

Local Meta-Correlation

Ideally, we would like the best (highest ranked) primitive intent model to perfectly match the requirement of the meta-intent model in all of the [H, V, S] dimensions. But what if the requirement is met by the primitive intent model that ranks second highest? The "match" condition takes this into consideration by creating a comparison between the requirement in the meta-intent model and the *relative* ranking of the primitive intent models. Two key factors are considered:

1) *Highest ranking primitive intent:* The highest ranking primitive intent models for the [H, V, S] dimensions are determined by Eqs. (7–9), respectively, at time i.

2) *Ranking relative to the meta-intent requirement:* Given requirement $R_1 = [I_1^H \;\; I_1^V \;\; I_1^S]$, then the local meta-correlation $M_1(i)$ equals zero if any winning intent model is naïve (that is, the winning intent model does not exceed a minimum threshold, e.g., 0.90); otherwise

$$M_1(i) = \frac{1}{3}\left\{ \frac{\rho\left(I_1^H, i\right)}{\rho\left[I_{\text{Best}}^H(i), i\right]} + \frac{\rho\left(I_1^V, i\right)}{\rho\left[I_{\text{Best}}^V(i), i\right]} + \frac{\rho\left[I_{\text{Best}}^S(i), i\right]}{\left|\rho\left(I_1^S, i\right) - 1\right| + 1} \right\} \tag{11}$$

Note: The local meta-correlation $M(i)$ collapses the three-dimensional [H, V, S] requirements of a meta-intent model into a scalar, that is, $\rho(I_1^H, i), \rho(I_1^V, i), \rho(I_1^S, i) \rightarrow M(i)$. Also, if there is "No Requirement" in any of the [H, V, S] dimensions, then the local meta-correlation $M(i)$ should use 1.0 for the H, V, or S term inside Eq. (11).

3) *Or logic in a meta-intent requirement:* The logic for the meta-intent model requirement R_i can contain an *or* condition in a [H, V, S] dimension. For example, given requirement $R_1 = [(2 \, or \, 4 \, or \, 6), (1 \, or \, 5), (3 \, or \, 7)]$ we could write this as $R_1 = [(2, 4, 6), (1, 5), (3, 7)] = [(I_{1,1}^H, I_{1,2}^H, I_{1,3}^H), (I_{1,1}^V, I_{1,2}^V), (I_{1,1}^S, I_{1,2}^S)]$. The requirement is carried through in the meta-intent correlation as follows:

a) If R_1 contains an *or* in the H dimension, for example, primitive A *or* B, then use the following term in the meta-intent correlation for the H dimension:

$$\frac{\max\left[\rho\left(I_{1,1}^H, i\right), \rho\left(I_{1,2}^H, i\right)\right]}{\rho\left[I_{\text{Best}}^H(i), i\right]}$$

b) If R_1 contains an *or* in the V dimension, for example, primitive A *or* B, then use the following term in the meta-intent correlation for the V dimension:

$$\frac{\max\left[\rho\left(I_{1,1}^V, i\right), \rho\left(I_{1,2}^V, i\right)\right]}{\rho\left[I_{\text{Best}}^V(i), i\right]}$$

c) If R_1 contains an *or* in the S dimension, for example, primitive A *or* B, then use the following term in the meta-intent correlation for the S dimension:

$$\frac{\rho\left[I_{\text{Best}}^S(i), i\right]}{\min\left[\left|\rho\left(I_{1,1}^S, i\right) - 1\right|, \left|\rho\left(I_{1,2}^S, i\right) - 1\right|\right] + 1}$$

Given these individual conditions, the local meta-correlation is computed:

$$M_1(i) = \frac{1}{3}\left\{ \frac{\max\left[\rho\left(I_{1,1}^H, i\right), \rho\left(I_{1,2}^H, i\right), \rho\left(I_{1,3}^H, i\right)\right]}{\rho\left[I_{\text{Best}}^H(i), i\right]} + \frac{\max\left[\rho\left(I_{1,1}^V, i\right), \rho\left(I_{1,2}^V, i\right)\right]}{\rho\left[I_{\text{Best}}^V(i), i\right]} + \frac{\rho\left[I_{\text{Best}}^S(i), i\right]}{\min\left[\left|\rho\left(I_{1,1}^S, i\right) - 1\right|, \left|\rho\left(I_{1,2}^S, i\right) - 1\right|\right] + 1} \right\} \tag{12}$$

Global Meta-Correlation

Global meta-correlation is a path correlation in the [H, V, S] dimensions that correlates a *sequence* of primitive intent models with meta-intent model state transitions T_i. Recall that the basic recursive form of the global correlation function $\rho(N)$ is computed from Eq. (2). Because our pilot decision-making model is hierarchical, we re-use this equation format for the meta-global correlation function $P(\,)$ as follows:

$$P(N) = \frac{1}{N}\sum_{i=1}^{N} M(i) = \frac{N-1}{N}\left[P(N-1) + \frac{1}{N-1}M(N)\right] \tag{13}$$

where $M(\,)$ is the local meta-correlation function. The local meta-correlation $M(i)$ and global meta-correlation $P(i)$ are purposely defined to be analogous to the local correlation $L(i)$ and the global correlation $\rho(i)$. Fading memory and moving window functions can also be defined as analogous to the local and global correlation functions defined earlier.

The process for computing the global meta-correlation considers two models: 1) a singleton (R_1 is not followed by any other requirement) and 2) a sequence of requirements ($R_1 \rightarrow R_2 \rightarrow \cdots$). In both of these cases, we rely on a requirement R_i to match the meta-intent model with sufficient strength [level of correlation $P(\,)$ of 0.90 or above] as well as to match consistently [k consecutive times ($k = 5$)]; otherwise, we consider the situation weakly correlated or noisy.

A comparison between $P(i)$ for the meta-intent models selects and outputs the best global meta-intent model at time increment i that explains the intent of the aircraft. If no result has a high enough correlation, then no result is output. In this case, the best primitive intent models can be output for that instant of time. If no primitive intent model is ranked high enough, then the naïve intent predictor is invoked (a straight-line prediction is made based on the best state estimate for the tracked aircraft).

Prediction Theory

Prediction is based on three cases: 1) naïve intent, 2) primitive intent, and 3) meta-intent.

Naïve Predictor

If there is not sufficient evidence (i.e., a global correlation value above 0.9) that a primitive intent or meta-intent model can match the observed motion, then the naïve predictor is used. As described earlier for the naïve intent model, a suite of Kalman filters is used to predict to a short time period T_p (30 s in our applications) ahead of the vehicle using a straight-line prediction.

Primitive Intent Predictor

If there is not sufficient evidence that a meta-intent model can match the observed motion, yet there is sufficient evidence (i.e., a global correlation value above 0.9) for a match of a [H, V, S] primitive intent, then the primitive intent predictor is used. The primitive intent predictor extends a straight-line prediction out as far as possible. For example, prediction for the "Avoid Weather Horizontal" primitive intent will extend a straight line out to the corner of a hazardous weather cell, but will not be able to "turn around the corner."

Meta-Intent Predictor

If there is sufficient evidence (i.e., a global correlation value above 0.9 and k consecutive matches of the requirement R) that a meta-intent model can match the observed motion, then the meta-intent predictor is used. Recall that meta-intent models include a sequence of requirements R_j, for example, $R_1 \rightarrow R_2 \rightarrow R_3$. In this case, the process of matching the meta-intent model can be at any of the following stages:

1) R_1 is matched, and $R_2 \rightarrow R_3$ remains; the prediction includes the primitive intent prediction to finish R_1 and then the navigation through the remaining $R_2 \rightarrow R_3$ navigation.

2) $R_1 \rightarrow R_2$ is matched, and R_3 remains; the prediction includes the primitive intent prediction to finish R_2 and then the navigation through the remaining R_3 navigation.

3) $R_1 \rightarrow R_2 \rightarrow R_3$ is matched; the prediction includes the primitive intent prediction to finish R_3.

In general, the meta-intent logic is used to guide the intent inference prediction forward in time based on TCPs or the terminal conditions of the meta-intent requirements. Note that the primitive intent model predictors can be iteratively invoked to predict forward in time.

Examples

Primitive Intent Example

A total of 41 primitive intent models (see Table 1) compete to explain the motion of an aircraft in each of the [H, V, S] dimensions. Discrete intent inference equations are applied with a 1-s update rate for aircraft state data and a 1-min update rate for weather cell motion. As illustrated in Fig. 8, an aircraft is flying around a hazardous weather cell while holding altitude. Intent inference does not assume that the aircraft will avoid weather until the pilot actually begins to turn to avoid weather. Thus, initially, even though the predicted path (dashed line) passes through hazardous weather, the prediction is directly to the TCP. While the aircraft is avoiding the weather cell, the predicted "go to" point moves with the weather cell—it is not a stationary point. Primitive intent predictions do not extend around the corner of the weather cell until the aircraft actually turns around the corner of the weather cell.

Figure 9 demonstrates the ranking of horizontal primitive intent models for this example. The global correlation value for a moving window history [Eq. (6)] is plotted as a function of time. At each instant of time, the maximum global correlation value for each intent model is compared, and the highest ranking, according to Eqs. (7–9), determines the predicted intent. These predicted intent outputs are labeled at the top of Fig. 9. In the case of a tie, preference is given to Go To TCP, as is the case initially.

Meta-Intent Example

Continuing on with the preceding example, meta-intent models are added to the intent inference algorithm to further explain the motion of the vehicle. The two key meta-intent models in this example are the "Fly Flight Plan" meta-intent model (Fig. 7) and the "Avoid Weather and Return to Flight Plan" meta-intent model, as illustrated in Fig. 10. The meta-intent requirements are used in the path prediction, as shown in Fig. 11, to extend the tactical prediction made by the primitive intent models to extend further out in time by exploiting the logic of the meta-intent model requirements. Thus, the "Fly Flight Plan" meta-intent model allows the predictor to extend the prediction from TCP to TCP + 1 $\cdots$ to TCP + n. The "Avoid Weather and Return to Flight Plan" meta-intent model assumes that the pilot will return to the flight plan because the pilot started on the flight plan prior to weather avoidance.

Application

This theory was first applied to real-time data from today's ATC system and second to a simulated free-flight environment representing a possible future application.

Real-world data were used to compare the output of intent inference algorithms to the intent revealed through pilot-controller communications at the Chicago Air Route Traffic Control Center, otherwise called Chicago Center (ZAU). An on-site visit to ZAU was performed in September 2002, to observe how pilots and controllers interact in the transition airspace leading to and from Chicago O'Hare International Airport (ORD). Weather was in the vicinity of the airport causing aircraft to request deviations around hazardous weather cells. The precise date and aircraft call sign are removed in these data to eliminate the implication of any performance evaluation of the pilot or controller. Table 2 shows the dialog between pilot and controller taken from ZAU voice recordings; time stamps on the ZAU recordings provided Zulu Time (an absolute standard). Aircraft track data at a 1-min update rate were retrieved from the Federal Aviation Administration (FAA)'s Enhanced Traffic Management System data source. Weather data at a 5-min update rate were recorded from the National Convective Weather Diagnostic data from the National Weather Service.

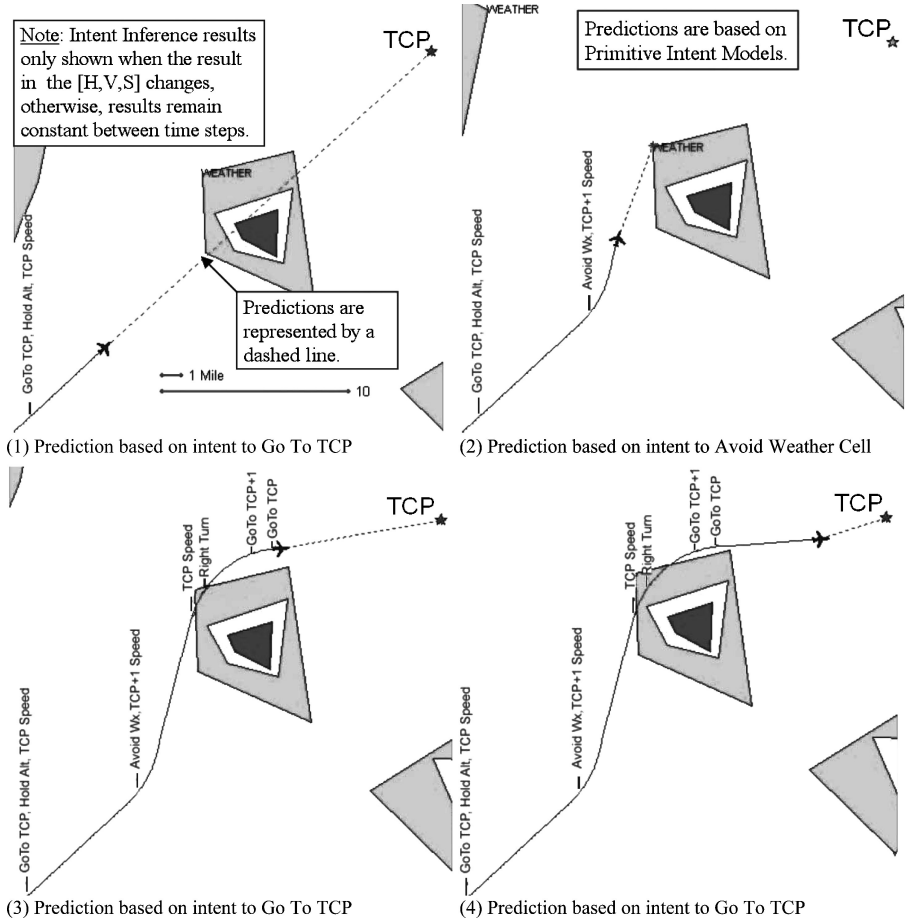

Fig. 8 Primitive intent models explain pilot intent.

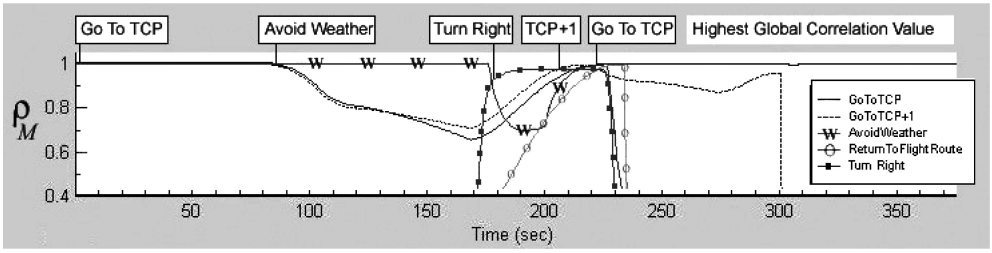

Fig. 9 Global correlation values ranked determine horizontal primitive intent.

Table 2 Pilot-controller dialog

Zulu time	Dialog	Speaker
21:41:10	Would like to deviate right of course for weather	Pilot
21:41:16	Course approved, BEARZ O'Hare when able	Controller
21:41:18	Descend to flight level 190, speed of 280 kn	Controller
21:44:15	Descend to 1-7000	Controller
21:41:16	(Repeat) descend 1-7000	Controller
21:41:18	Descend and maintain 1-1000 as soon as you can turn left 270, do so	Controller
21:44:15	Negative on the 270 down to 1-1000 we can come left a little bit	Pilot
21:41:16	As left as possible and as soon as you can fly 270 do so	Controller
21:41:18	{Call sign omitted} is on a 270 heading	Pilot
21:44:15	On a 270 heading intercept 130 radial O'Hare and fly inbounds	Controller
21:41:16	Contact Chicago approach	Controller

Figure 12 illustrates a comparison between the pilot-controller dialog and intent inference results time synchronized along an aircraft trajectory. Initially, the pilot is flying a direct-to route to the OXI fix. Next, when the pilot requests to make a deviation for weather, the controller approves the request, and the intent inference algorithm detects a weather deviation intent approximately 60 s later. Later, the controller requests the pilot return to the flight plan as soon as possible, and within 20 s the intent inference algorithm reports a return to flight plan intent. Although these results are dependent on the chosen algorithm parameters (in particular, a 0–10 n mile expected clearance around hazardous weather and a 45-deg angle of return to flight plan), they support a proof of concept for intent inference. A formal validation of the concept would require an operational assessment with many scenarios comparing and contrasting the various intent models to real-world data and possibly pilot and controller interviews.

Figures 13–15 demonstrate meta-intent prediction based on real-world aircraft and weather data of Fig. 12. Figure 13 shows the

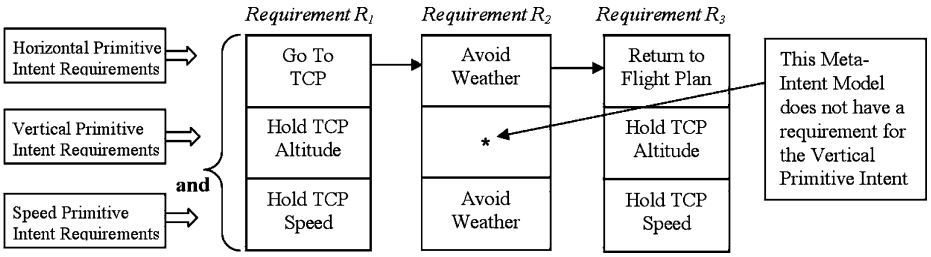

Fig. 10 Avoid weather and return to flight plan meta-intent model.

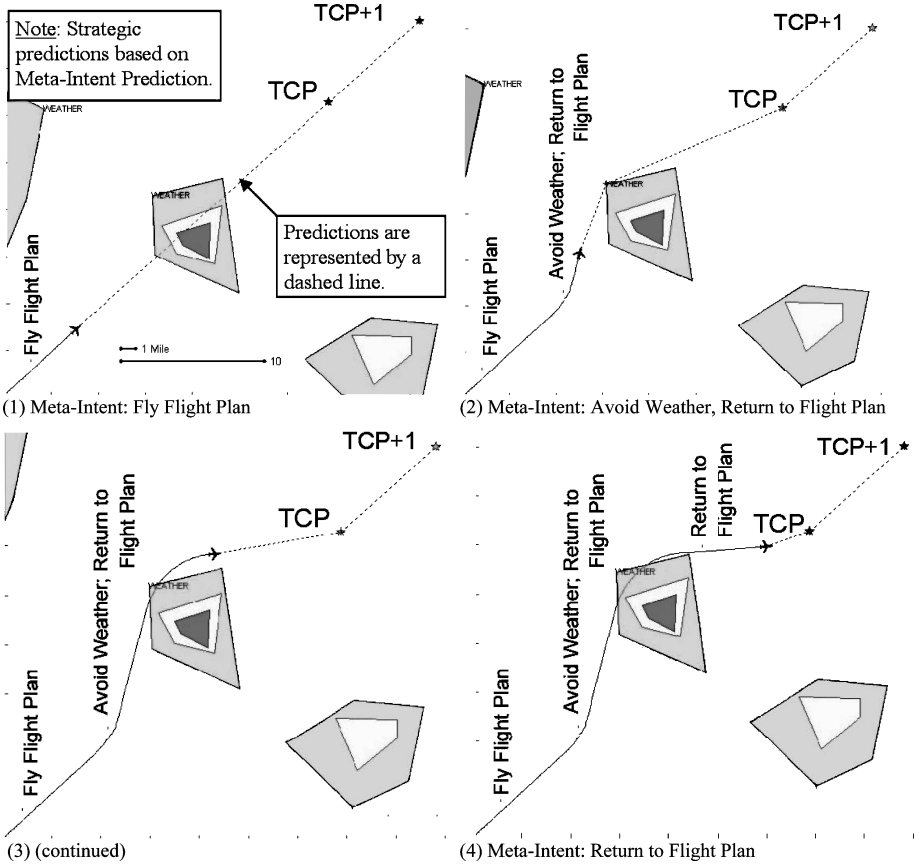

Fig. 11 Meta-intent models explain pilot intent.

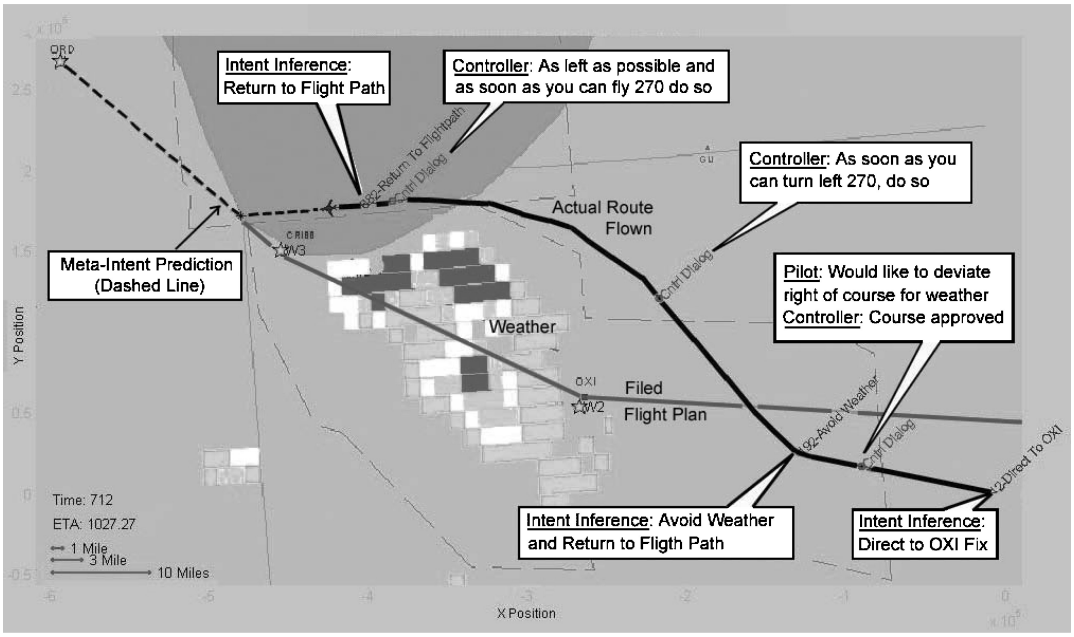

Fig. 12 Intent inference results compared to pilot-controller dialog.

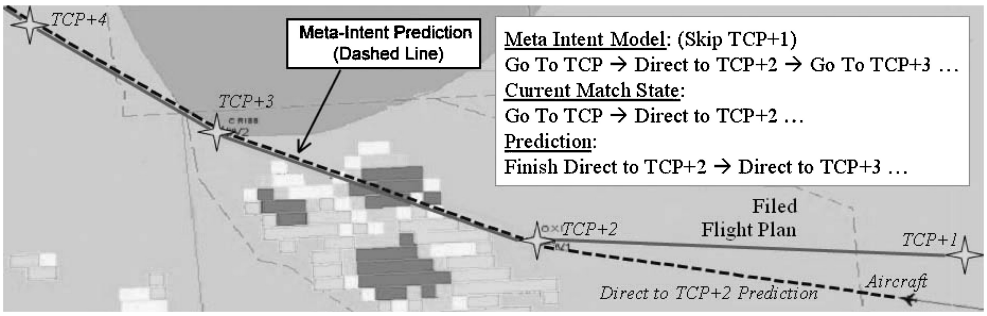

Fig. 13 Meta-intent prediction for direct to TCP + 2 then follow flight plan.

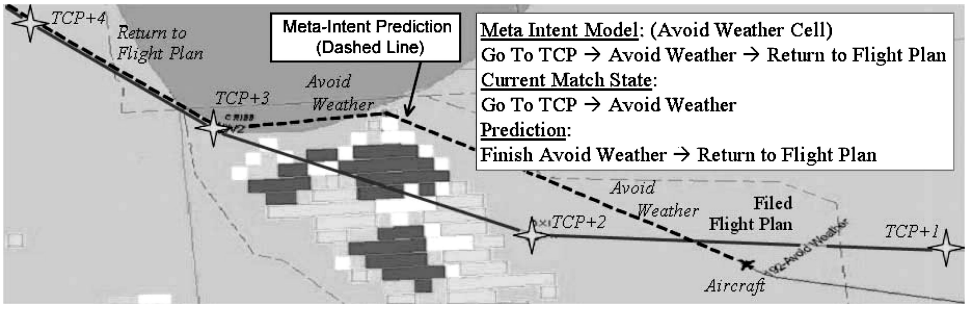

Fig. 14 Meta-intent prediction for avoid weather then follow flight plan.

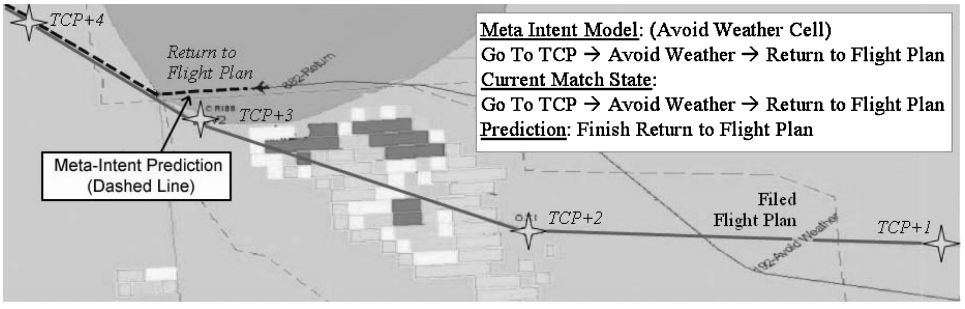

Fig. 15 Meta-intent prediction for return to flight plan.

present location of the aircraft with the aircraft symbol. The aircraft has skipped TCP + 1 and appears to be heading to TCP + 2. The meta-intent model for flying the points TCP → TCP + 2 → TCP + 3 → TCP + 4 is invoked. The meta-intent prediction (dashed line) shows the aircraft moving from its present location directly to TCP + 2, then to TCP + 3 and to TCP + 4. Figure 14 shows what happens when the aircraft deviated from its flight path to avoid weather. The meta-intent for avoiding a weather cell is invoked. The meta-intent predictor determines that the aircraft will move toward the corner of the weather cell, turn the corner, and return to its broadcast flight path (i.e., TCP + 3 → TCP + 4). Figure 15 shows that the aircraft is returning to its broadcast flight path but takes a path that that skips TCP + 3. The meta-intent predictor places the aircraft on a trajectory that intercepts the flight path between TCP + 3 and TCP + 4 on a 45-deg intercept angle.

For a future free-flight system, a series of 19 movies were created and shown to two current commercial airline pilots (9 and 13 years commercial experience) to discuss intent inference results. Although this is a weaker test than the previous pilot-controller dialog proof of concept, it was necessary to test out scenarios that could not be observed in real-world on-site observations. For instance, in the case of flying around SUA, the "violate SUA" intent model can be easily tested in simulation but would be difficult to find a real-world example. Movies were created by simulating the flight of one or more aircraft using a 6-degrees-of-freedom simulator (B-747 aircraft); 1-s ADS-B updates were simulated with 1-min weather updates.

The general feedback from the pilot critique was positive. Minor problems were identified with primitive intent models; however, the pilots generally believed that the approach taken to intent inference was a solid baseline for modeling aircraft path behavior.

Conclusions

We present a method for inferring the intent of a pilot in real-time given aircraft and weather state information, domain knowledge (e.g., standard terminal arrival routes and jetways, special-use airspace region boundaries, and navaid locations), and broadcast intent information (e.g., TCP or waypoint information from a flight plan). Domain knowledge is used to build plausible models of intent in the horizontal, vertical, and speed dimensions, and through correlation measures the models of intent are ranked based on data that supports the plausible models of intent. In each of the horizontal, vertical, and speed dimensions, the highest ranking models of intent from the set of plausible models constitute the best estimate of the intent of the aircraft. Sequences of primitive intent matches are considered to infer a navigation task intent. This technique can be used for conformance monitoring of aircraft motion compared to the intent, or it can be used to infer an intent when there is no intent information available. Our method was tested using recorded data from today's air-traffic control system and was favorably evaluated by two

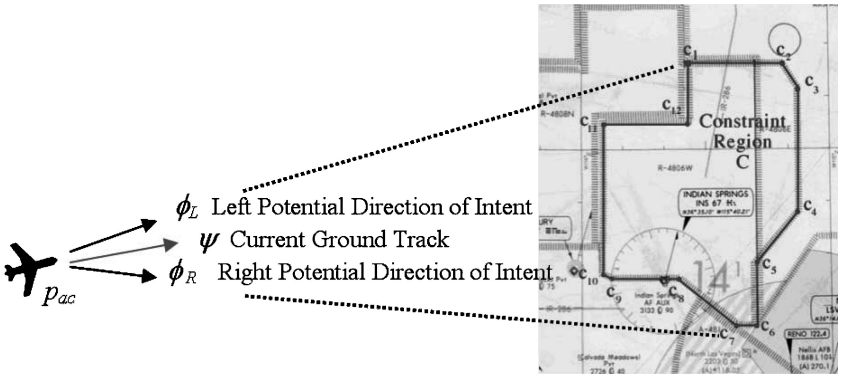

Fig. A1 Avoid constraint region (SUA example).

current commercial airline pilots. Simulated real-time results show that our method can determine the pilot's intent (e.g., avoid weather) and produce a long-range trajectory predictions (10–12 min ahead) based on that inferred intent (e.g., fly around the weather and then return to the flight plan).

Appendix: Example Primitive Intent Model

A total of 41 primitive intent models was created. This Appendix demonstrates one such intent model "Avoid SUA (Horizontal)." The model ignores the vertical dimension ["Avoid SUA (Vertical)" models the vertical dimension]. The algorithm is stated in terms of avoiding a single nonmoving polygon constraint region; if more than one constraint region is present, the algorithm does not solve the coupled problem, rather, each constraint region is investigated separately (thus, we are not considering the possibility that a constraint region might be immediately behind another constraint region).

Let the aircraft position be $p_{\text{ac}} = (x_{\text{ac}}, y_{\text{ac}})$, and the constraint region $C = c_1, c_2, \ldots, c_n$ be defined by a set of polygon vertices $c_1, c_2, \ldots, c_n$. Two cases are of interest, as illustrated in Fig. A1.

Case 1

Aircraft is inside the SUA. In this case, we assume that the aircraft has no intention of avoiding the SUA; $\phi = -1$ is returned to indicate there is no intent to avoid the SUA.

Case 2

Aircraft is outside the SUA. In this case, we assume that the aircraft is going to fly to one of two common tangent points to the SUA in order to avoid the SUA: 1) the left common tangent between the aircraft and the SUA and 2) the right common tangent between the aircraft and the SUA. A standard computational geometry algorithm is used to identify the left and right common tangent points between the aircraft position and the polygon. Given the aircraft position $(x_{\text{ac}}, y_{\text{ac}})$, we determine the left and right common tangent points c_L and c_R. The points c_L and c_R are used to define the left potential direction of intent and the right potential direction of intent:

$$\hat{\phi}_L = \frac{c_L - p_{\text{ac}}}{\|c_L - p_{\text{ac}}\|} \tag{A1}$$

$$\hat{\phi}_R = \frac{c_R - p_{\text{ac}}}{\|c_R - p_{\text{ac}}\|} \tag{A2}$$

Next, compare the left and right common tangent point directions to the current track direction, and return the common tangent point direction that is closest to the current track direction. The local correlation is computed for the left $L(i) = \hat{\psi} \cdot \hat{\phi}_L$ and right $L(i) = \hat{\psi} \cdot \hat{\phi}_R$ common tangent points, returning the higher of these two correlations.

Prediction involves determining the heading angle Ψ_p from the aircraft position p_{ac} to the common tangent point (c_L or c_R) and the range R_h from p_{ac} to the common tangent point. Prediction to range R_h is accomplished with the following equations:

$$X_{\text{Predict}} = x_{\text{ac}} + R_h \sin \Psi_p \tag{A3}$$

$$Y_{\text{Predict}} = y_{\text{ac}} + R_h \cos \Psi_p \tag{A4}$$

Because the aircraft has been determined to be traveling to the common tangent point but the path after this point is unknown, it is not reasonable to predict beyond the common tangent point unless meta-intent prediction logic is used to identify where the prediction should extend.

Acknowledgments

This research was funded by NASA Langley Research Center under the Small Business Innovative Research Program Contract NAS1-02101. The guidance of our NASA Technical Monitor E. Johnson is greatly appreciated; his guidance helped direct and focus the effort. The authors appreciate the diligent software engineering support by C. Schwalm, software analyst from Metron Aviation.

References

[1]"Free Flight Implementation," RTCA, Inc., Final Rept. for RTCA Task Force 3, Washington, DC, Oct. 1995.

[2]Zhao, Y., Haissig, C., and Hoffman, M. J., "Analysis of Pilot Intent in Air Traffic Management," *Proceedings of the American Control Conference*, Vol. 3, Inst. of Electrical and Electronics Engineers, New York, 1998, pp. 1789–1792.

[3]Geddes, N. D., "A Model for Intent Interpretation for Multiple Agents with Conflicts," *Conference on Systems, Man, and Cybernetics*, Vol. 3, Inst. of Electrical and Electronics Engineers, New York, Oct. 1994.

[4]Hoshstrasser, B., and Geddes, N., "OPAL: Operator Intent Inferencing for Intelligent Operator Support Systems," *Proceedings of the IJCAI-89 Workshop on Integrated Human-Machine Intelligence in Aerospace Systems*, edited by V. Shalin and G. Boy, Vol. 1, IJCAI, Detroit, MI, 1989, pp. 53–70.

[5]Rouse, W. B., Geddes, N. D., and Curry, R. E., "An Architecture for Intelligent Inferences: Outline of an Approach to Supporting Operators of Complex Systems," *Human-Computer Interaction*, Vol. 3, No. 2, 1988, pp. 87–122.

[6]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*, Vol. 18, No. 4, 1988, pp. 618–637.

[7]Thurman, D. A., Chappell, A. R., and Mitchell, C. M., "An Enhanced Architecture for OFMspert: A Domain-Independent System for Intent Inferencing," *Proceedings of the International Conference on Systems, Man, and Cybernetics*, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 1998, pp. 955–960.

[8]Kautz, H., and Allen, J., "Generalized Plan Recognition," *Proceedings of the 5th National Conference on Artificial Intelligence*, Vol. 1, AAAI Press, Menlo Park, CA, 1986, pp. 32–37.

[9]Tambe, M., and Rosenbloom, P., "Event Tracking in a Dynamic Multi-Agent Environment," *Computational Intelligence*, Vol. 12, No. 3, 1996, pp. 499–521.

[10]Callantine, T. J., "CATS-based Air Traffic Controller Agent," NASA CR-2002-211856, Oct. 2002.

[11]Callantine, T. J., "The Crew Activity Tracking System: Leveraging Flight Data for Aiding, Training, and Analysis," *Proceedings of the 20th Digital Avionics Systems Conference*, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 2001, pp. 5C3/1–5C3/12.

[12]Krishnamurthy, K., and Ward, D. T., "An Intelligent Flight Director for Autonomous Aircraft," AIAA Paper 2000-0168, Jan. 2000.

[13]Krishnamurthy, K., and Ward, D. T., "An Intelligent Inference Engine for Autonomous Aerial Vehicle," AIAA Paper 1999-4251, Aug. 1999.

[14]Reynolds, T. G., and Hansman, R. J., "Investigating Conformance Monitoring Issues in Air Traffic Control Using Fault Detection Techniques," *Journal of Aircraft*, Vol. 42, No. 5, 2005, pp. 1307–1317.

[15]Reynolds, T., and Hansman, R. J., "Investigating Conformance Monitoring Issues in Air Traffic Control Using Fault Detection Approaches," International Center for Air Transportation, Massachusetts Inst. of Technology, ICAT-2003-5, Cambridge, Nov. 2003.

[16]Krozel, J., "Intelligent Path Prediction for Vehicular Travel," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Purdue Univ., West Lafayette, IN, May 1992.

[17]Krozel, J., and Andrisani, D., "Intelligent Path Prediction for Vehicular Travel," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 2, 1993, pp. 478–487.

[18]Krozel, J., and Andrisani, D., "Intelligent ε-Optimal Path Prediction for Vehicular Travel," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 2, 1995, pp. 345–353.

[19]Stengel, R. F., "Toward Intelligent Flight Control," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 6, 1993, pp. 1699–1717.

[20]"Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)," RTCA Special Committee-186, RTCA, Inc., RTCA/DO-242A, Washington, DC, June 2002.

[21]"Operations Concepts for Data Link Applications of Flight Information Services," RTCA Special Committee-169, RTCA, Inc., RTCA/DO-232, Washington, DC, March 1996.