Dimperio, E., Gunzelmann, G., & Harris, J. (2008). An Initial Evaluation of a Cognitive Model of UAV Reconnaissance. In J. Hansberger (Ed.), *Proceedings of the Seventeenth Conference on Behavior Representation in Modeling and Simulation* (pp. 165-173). Orlando, FL: Simulation Interoperability Standards Organization.

An Initial Evaluation of a Cognitive Model of UAV Reconnaissance

Eric Dimperio
Indiana University
1101 East Tenth Street Bloomington, IN 47405
812-856-4678
edimperi@indiana.edu

Glenn Gunzelmann
Jack Harris
Air Force Research Laboratory
6030 South Kent Street, Mesa, AZ 85212
480-988-6561 x-674; x-675
glenn.gunzelmann@mesa.afmc.af.mil, jack.harris@mesa.afmc.af.mil

Keywords:
Cognitive Model; Synthetic Task Environment; Reconnaissance; Flight Simulation; ACT-R; UAV

ABSTRACT: ***Training simulators have a specific need for accurate human behavior representation to best simulate conditions a trainee will encounter in the real world. This paper outlines a cognitive model of human pilots flying a Predator Unmanned Aerial Vehicle (UAV). The model was implemented to interact directly with a synthetic task environment (STE) built upon a validated flight dynamics model of the Predator RQ1A System 4 UAV. The model flies reconnaissance missions in the STE with the goal of maximizing the amount of surveillance footage that is obtained by maneuvering the Predator over a small hole in a layer of clouds. At the same time, the model must minimize the time spent in violation states (e.g., exceeding altitude restrictions or entering no-fly zones). Initial analysis shows that the model can complete simulated missions with performance characteristics similar to those observed in real pilots.***

1. Introduction

In addition to helping develop our understanding of the human mind, cognitive modeling can serve many practical purposes. One of the most notable areas of contribution is in the augmentation of training environments by simulating interactive and cognitively plausible synthetic teammates. Many complex tasks performed by doctors, pilots, machine operators, and others are done in coordination with other people. Sophisticated training programs require humans of various levels of expertise to play the peripheral roles. Unfortunately, as jobs get more specialized and the interactions become more complex, it often becomes difficult and costly to coordinate training events where large numbers of people are needed to fill all the roles. Having simulated players that can accurately recreate the behaviors of human experts and interact with a trainee can make training more convenient and cost effective.

Despite the obvious applications of cognitively realistic behavior representations, such models have rarely been transitioned to application environments. This is because of both the cost associated with creating detailed representations of human performance as well as existing limitations in our understanding of human perception, action, and cognition. The research described here is aimed at making progress on these issues. The model described flies reconnaissance missions within a Predator Unmanned Aerial Vehicle (UAV) Synthetic Task Environment (STE). The goals are to use the model to better understand the cognitive processes associated with performance, to continue to increase the psychological validity of such models, and to apply computational cognitive modeling methodologies to increasingly complex and naturalistic tasks.

1.1 Predator Synthetic Task Environment

The UAV STE is built upon a realistic simulation of the flight dynamics of the Predator RQ1A System 4 UAV. This core component of the STE has been used in other contexts and applications to train Air Force Predator pilots at Creech AFB. Additionally, Schreiber, Lyon, Martin, & Confer (2002) found that performance on the tasks in the UAV STE was better for experienced Predator pilots as compared to other expert pilots, indicating that the STE taps into skills that are unique to flying the Predator, in addition to more general piloting skills.

The STE builds upon the core aerodynamics model with three research tasks. First, there is a set of basic aircraft maneuvers, requiring precise, constant rate changes to the UAV's speed, altitude, and/or heading. Gluck, Ball, Krusmark, Rodgers, & Purtee (2003) present an ACT-R model that performs this set of maneuvers with close correspondence to the human data. Important insights from that modeling effort have been applied in the model described here, particularly the instrument "crosscheck" that the model uses to monitor and maintain the UAV's flight characteristics. Gluck, Ball, and Krusmark (2007) provide a comparison of alternative strategies for this process that vary considerably in their effectiveness.

A second task requires individuals to perform an approach and landing with the UAV, a particularly challenging task given the Predator's light weight and low clearance. The final task, and the one we focus on here, is a set of reconnaissance missions requiring the pilot to obtain surveillance footage of a target on the ground by maneuvering the UAV over a hole in a layer of clouds (see Figure 1). This requires that the plane be flown through the conic frustum over the cloud hole where the target can be filmed by using a gimbal camera located on the bottom of the plane. Individuals can toggle between the gimbal camera, which slews to focus on the target regardless of the position of the plane (Figure 2), and a nose camera view, which provides an out-the-cockpit view. In addition, a map of the area is provided, with the positions of the plane and target indicated (Figure 2).

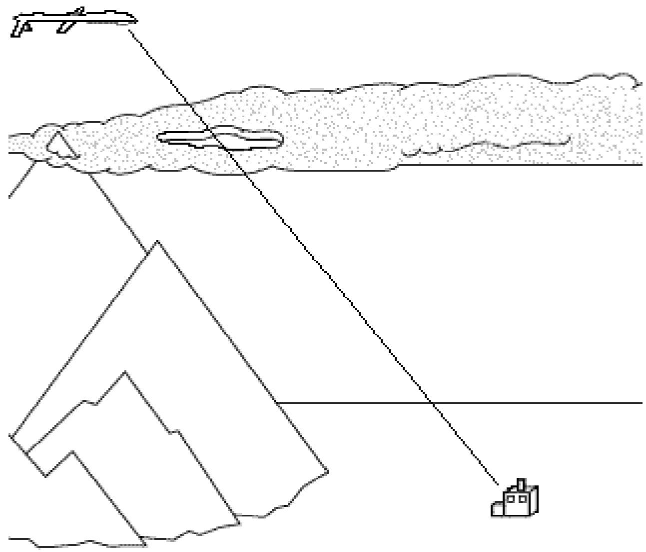

Figure 1. A cartoon depicting the Predator UAV viewing a target through a cloud hole during a reconnaissance mission.

The task is made more challenging by imposing restrictions on altitude, introducing no-fly zones in the airspace, and including the influence of wind on the plane's flight path. If restrictions are violated, or if the plane stalls, penalty time is assessed until the violation is corrected. In addition, when violations occur, the camera view goes blank, and information about the plane's location on the map is removed, requiring the pilot to

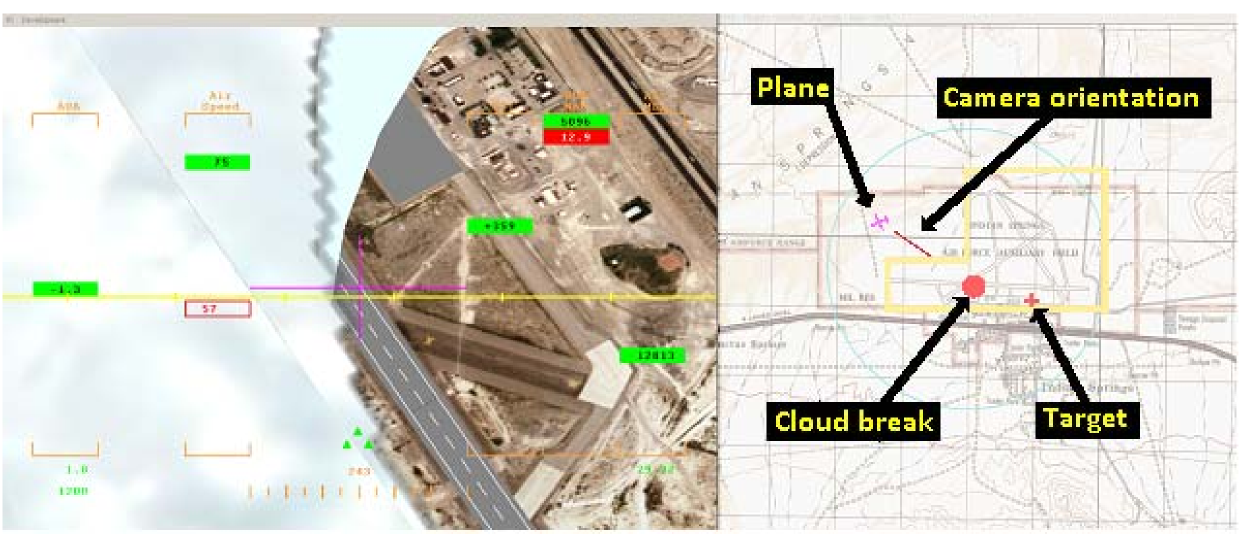

Figure 2. Screen shots illustrating the reconnaissance mission task within the UAV STE. Map and Gimbal camera view illustrating a situation where time on target is being obtained. The left side shows the target viewed through hole in the clouds in the gimbal camera view. The right side displays the map view.

correct the violation before this information is made available again. The goal of the task is to maximize the amount of time spent getting surveillance footage (time-on-target, or TOT) during each 10 minute scenario, while minimizing time spent in violations. Scenarios vary in terms of the location of the cloud hole, the wind speed and direction, and the characteristics of the no-fly zone (size, shape, and position). In the next section we describe the model that has been developed to perform this task.

2. Model Design

Our model for this task was developed within the ACT-R cognitive architecture (Anderson, 2007). ACT-R represents central cognition as a production system where knowledge is represented as production rules consisting of condition-action pairs. At any point in the model's performance of a task, rules are matched against the current state of the system and a single applicable rule is selected and executed (fired). The current state in ACT-R is represented in a set of buffers that contain information regarding different components of cognitive functioning. For instance, there is a retrieval buffer that holds the current item that has been requested from declarative memory. There are also perceptual and motor buffers to represent information encoded from the environment and for executing actions. These buffers serve as the interface between central cognition and functional modules, which contain the mechanisms for performing particular types of processing.

Because of ACT-R's implementation, it is able to interact directly with software-based tasks. However, this capability is not universal, and ACT-R cannot directly interact with the UAV STE. As a result, ACT-R flies the UAV STE by interacting with a reimplementation of the task interface that is written in Lisp (the same language as ACT-R). ACT-R sends commands to this reimplementation, which are then passed over a socket to the UAV STE, resulting in requested actions being performed within the simulation. Changes in the STE are then passed back through a set of variables that are used to maintain the state of the Lisp-based reimplementation. This setup allows us to utilize the detailed flight dynamics model embedded within the STE and collect detailed performance data for the model that is generated automatically by the STE. The cost, however, is that there is a substantial infrastructure required to manage the communication between applications and the interface with which ACT-R interacts.

2.1 Model Strategy: Flying in Stages

At the highest level, the process of flying a mission is broken into *stages* based on qualitatively different subtasks. The model's behavior during a particular stage is influenced by a set of goals unique to each stage. All stages are associated with a set of ideal flight performance characteristics. These may take a quantitative form with a specified range of altitudes and a specific airspeed, or they may take a more qualitative form specifying relative changes such as to "go higher" or "go very fast." Although the model flies well using either (or both) types of knowledge, the data described below is based upon knowledge of the qualitative nature of what should be done.

This representation seems to capture human performance better since, unlike the basic maneuvering tasks mentioned above, precise control of the UAV's performance characteristics is not required to perform the task successfully. Instead, more general constraints like "go slower when getting TOT" or "Turn around as fast as possible" are more appropriate.

The model utilizes eight separate stages to control the goals of the simulated pilot. The model determines what stage it is in based on its current knowledge of the target viewing area (TVA), which is the conic frustum over the cloud hole within which the target is visible. Here we will describe the function of the stage specific procedures of the model.

- Search for TVA: If the model has no information about the location of the TVA and the cloud hole is not visible through the nose camera view, the model enters this stage where it tries to locate the hole in the cloud layer. This involves banking to the side to bring more of the space into view through the limited field-of-view nose camera.

- Orient to TVA: When the model can see the hole in the cloud layer through the nose camera view, it enters the "Orient to TVA" stage. Here, its primary goal is to turn until the plane is flying steadily toward the center of the cloud-break. Although the TVA may not perfectly coincide with the cloud hole if the target is not directly under it, there is generally some overlap.

- Find TVA: Once the model is oriented toward the cloud hole, it enters the "Find TVA" stage where it switches to the gimbal camera and flies until it encounters the cloud hole and flies through the TVA. Once the gimbal camera is selected, the model cannot rely on the egocentric nose camera for information about the location of the hole. Thus, it relies on the information it encoded to identify a vector on the map along which the cloud hole must be located. During the “Find TVA” stage, the model flies along this vector to locate the hole’s position. Along the way, it tries to maximize its speed to maximize the amount of time available for getting TOT once at the TVA. The model also increases its altitude, since flying higher leads to more TOT per pass because the diameter of the TVA increases as altitude increases. Thus, this is a strategy to maximize TOT.

- In TVA: Once the target can be seen through the hole in the cloud layer (i.e., the UAV enters the conic frustum that defines the TVA) the model enters the “In TVA” stage where it slows down as much as possible and levels out its flight to get as much TOT as possible during each pass. The first time this stage occurs, the model encodes an estimate of the location of the TVA that provides a reference point to fly to on future passes. This estimate gets refined in subsequent passes.

- Fly out: After the vehicle has flown out of the TVA, the goal is to get back to the TVA as quickly as possible. It will also try to gain altitude while trying not to violate altitude restrictions. During the “Fly out” stage, the aircraft quickly speeds up to get some distance away from the target. This is necessary because the model is only given control of the flaps to adjust bank and cannot control the rudder to adjust yaw, resulting in wider turns. Consequently, to completely turn around and then return to straight and level flight requires the model to place some distance between itself and the target before beginning the turn.

- Turn-around: Here, the model slows down, which allows it to turn more quickly, and turns the vehicle until it is flying steadily back toward the reference point it encoded to represent the location of the TVA. The model begins the turn by maximally adjusting its bank, and then eases off progressively as it approaches the desired heading (i.e., facing directly at the estimated location of the TVA).

- Return to TVA: This stage signals the model to increase speed and fly at the TVA as quickly as possible.

- Approach TVA: To maximize time over the target it is necessary to fly more slowly while over the TVA. Thus, the model makes an attempt to decrease airspeed as it approaches the estimated location of the TVA so that it is going more slowly when the TVA is reached.

When the model encounters the TVA again, it should go back to the “In TVA” stage and cycle through the last five stages until time runs out. All stages utilize a similar set of productions that handle transitions into the stage. This includes recalling the appropriate goals for the stage and making immediate control adjustments.

Outside of the stages, there are two sets of high-level behaviors. The first set of behaviors allows the model to maintain flight characteristics and make adjustments to control settings (i.e., stick and throttle positions) to accommodate changes in desired performance characteristics (i.e., speed, altitude, and heading). When a pilot is flying toward a specific location with preferences for certain altitudes and airspeeds, flight is guided by what were referred to as the *crosscheck* procedures above. The goal of the crosscheck is to monitor instruments and make any adjustments to the controls necessary to keep the plane “on track.” The other set of procedures is used to avoid violations. If the simulated pilot notices that the aircraft is too high and might violate the altitude restriction, or is traveling too slow and might stall, etc., this set of procedures augments the immediate goals of the pilot in order to prevent a violation from occurring. These sets of high-level behaviors of the model are described in the next sections.

2.2 The Crosscheck

The model’s division of the task into stages provides a framework for making decisions about what flight characteristics are appropriate given the current state of knowledge. The process of monitoring the aircraft’s state and making adjustments to control settings to achieve desired performance is referred to as the crosscheck. The crosscheck productions have a lower priority than other productions, meaning that they represent the background activity of the model when more specific actions are not required.

The crosscheck productions allow for monitoring of airspeed, bank angle, altitude, pitch, and engine power (i.e., rpm). After the model has made its first pass through the TVA and an estimate of its location is known, the crosscheck will include checking the map to note the orientation of the UAV icon relative to the TVA (this impacts the stage that the model is in – for instance, *Fly Out* versus *Return to TVA*). This information is also used to update the desired heading over the course of the trial, which varies depending on the stage of flight and the plane’s location and orientation relative to the target.

The model follows a specific process during the crosscheck. First, it probabilistically chooses a flight instrument according to ACT-R's conflict resolution mechanism. Currently, all productions for selecting a particular instrument have an equal probability of being selected. It recalls from memory where that instrument is located, shifts visual attention to that location, and encodes the current information from that instrument. At this point, the model compares the instrument values to the flight goals. Any discrepancies are encoded as qualitative difference magnitudes. The model then probabilistically (again via ACT-R's conflict resolution mechanism) decides whether it wants to adjust a current control position (relative update) or set either the stick or the throttle to an expected ideal control position (absolute adjustment). The motor system makes the appropriate changes to the controls. The system then returns to a state where the crosscheck can start over again.

2.2 Avoiding and Recovering From Violations

The crosscheck in the model represents the "business as usual" piloting of the plane. However, when the monitoring process detects an impending violation, a different set of knowledge is engaged to alter flight characteristics in order to avoid violations. As this description implies, this knowledge is currently highly reactive. The model engages in little or no planning up front to avoid these situations, but instead relies on corrective actions to avoid or minimize the time spent in violation states.

Restrictions concerning altitude and airspeed (to avoid stalling) have straightforward solutions. When the model notices that it is approaching an altitude restriction, it reacts with a quick change of the control stick (pull back to climb or push forward to dive) and modifies any altitude goals to be safely within the appropriate range. Similarly the model increases throttle and modifies goals to keep itself safely away from the stall point when it notices that its airspeed is too low.

Other restrictions require more complicated solutions. While attempting to maneuver toward the estimated TVA, the model may encounter a no fly zone (NFZ). Currently, the model deals with this situation in a reactionary manner. If the UAV is heading toward the NFZ, the model does not act until just before it believes it cannot avoid the NFZ and makes a hard turn away from the NFZ. In most cases, this is enough to avoid it completely. Occasionally, the model will slip into the NFZ for a few seconds, depending on the estimate and the details of performing the maneuver.

When in the "Find TVA" stage, the model remembers its position and heading when it last saw the hole in the clouds through the front camera. If it avoids a NFZ in this stage, it will return to tracking that same line once on the other side of the NFZ.

Even when not immediately threatened by a NFZ, the model's default behavior is to turn away from the NFZ. This behavior leads to a pattern of flying in figure-eights parallel to the nearest perpendicular to the line connecting the TVA and the center of the NFZ, which is observed in the performance of some participants.

Occasionally, the model and actual pilots do cause a violation by entering a NFZ, violating altitude restrictions, or stalling. In all of these conditions, the simulator causes the camera view to go blank and the UAV icon is no longer visible on the map. One implementational limitation that still exists is that the instrument values are still visible in the STE during a violation, but these values are not updated for the model until the violation is corrected. Therefore, the model relies on knowledge of its own state before the violation occurred to react, which seems to lead to appropriate behavior in most circumstances.

Most procedures that were designed to avoid violations will also work to get back to an appropriate state. This does not hold true for the NFZ, but this does seem somewhat appropriate. If the model cannot see itself on the map, then it cannot know it is in a NFZ. Currently, it is very rare for the model to not see a NFZ before it enters it. This situation is most prevalent when high winds push the aircraft into the NFZ even though it was not directly facing the NFZ.

2.4 Accounting for Wind

Many of the scenarios include wind. Wind speed and direction remain constant throughout each 10 minute scenario. It impacts the flight path of the Predator aircraft, even though it does not affect the location of the hole in the cloud (this remains constant throughout the trial). The model can accurately fly a mission without knowledge of the wind by regularly making corrections in heading during the crosscheck. However, this does not appear to be the appropriate cognitive strategy based upon evidence from expert pilots.

A study done using Unites States Air Force reserve pilots (Park, 2006) had the pilots think aloud as they flew a number of reconnaissance missions. Transcripts of those flights were studied to identify the strategies used by the pilots to deal with the force of wind. It seemed very common for pilots to consciously make adjustments to a target heading to adjust for wind. Some typical comments included:

> "I'll do a little bit of crab left for the winds"

> "So three-zero-zero looks like my heading but my ground track's going to have to be two-eight-zero with the winds"

> "Accounting for the winds, I should go about five-zero, that should be good"

The model was, therefore, given knowledge about how to update an ideal heading based on the wind speed and the difference in direction between the wind and the ideal heading. A current limitation of this mechanism is that the model still estimates the direction it is traveling by monitoring the direction it is facing. The consequences of this are most apparent in the context of avoiding no-fly-zones (NFZs). This issue was alluded to earlier – sometimes the model fails to notice that the wind is pushing it into a NFZ because the NFZ is not in front of the plane. This is one key area for improvement in the model currently.

3. Model Validation

3.1 Strategic comparison

The ACT-R model presented here is intended to capture human-like performance. This includes making human-like mistakes and being able to express human-like variability. Some pilots failed to locate the TVA during a pass and had to reinitiate a search for it. Some flew in a figure-eight pattern over the TVA while others made only left turns to circle back to the TVA. The current model produces such differences in performance with changes to declarative knowledge. Pilots also produced a variety of errors in flight control, including stalling the Predator and over-steering. The current model can account for some of these failures simply based on the stochastic nature of selecting procedures. We have not yet explored the capacity for such variations in the model to account for individual differences, but this is one potential avenue for future research. In addition, so far we have treated each 10 minute run as an isolated trial. Another avenue for improvement is to explore learning across trials in the model to capture improvements stemming from experience with the task.

In this paper we focus on how stochasticity influences the performance of the model. One aspect of the model's performance where stochasticity is relevant is the crosscheck procedure. At the beginning of each cycle of the crosscheck, each instrument is equally likely of grabbing the attention of the model. One side effect of this is that it is possible for the model to ignore one or more of the instruments for a relatively long period of time, resulting in errors like over-steering. While turning around, the model may see that there is a large deviation in heading, which causes it to push the stick all the way to the left. It is possible for the model to subsequently consider altitude and make adjustments to the pitch and/or check airspeed and make adjustments to the throttle. By the time the model goes back to check the heading or the UAV icon on the map, the aircraft may already be facing the target position. Unfortunately, it takes several seconds for the model to return to straight and level flight after the stick is returned to a neutral position. The next time the model checks the icon, it will see that it has turned past the target and make an appropriate correction. Similar patterns can happen for other controls. These types of behaviors were observed on some occasions for some of the human pilots.

3.2 Matching Human data

Preliminary comparisons have been made between the model performance and the performance of human pilots. The pilots were USAF pilots selected as candidates for Predator training (Schreiber, Lyon, Martin, and Confer, 2002). An important difference to note is that, in the empirical study, participants had access to rudder controls allowing them potentially to make tighter turns than the model.

Figure 3 displays sample flight paths during a run through two scenarios for both the model and for a sample pilot. We have selected an easier scenario with a nontrivial NFZ as well as a more difficult scenario (because of the high winds and placement of NFZ relative to target) to compare flight paths. The illustrations in Figure 3 are intended to show that, in both cases, the path produced by the model is qualitatively similar to the path produced by human pilots. Of course, these are data from a single model run and from a single human participant. There is variability, both in the model runs and in human performance, although much more so in human performance across individuals. These examples were selected to emphasize the general consistency of the model with human performance on this task. In informal presentations, people knowledgeable of both the model and the human data were unable to reliably differentiate between the model and human flight paths.

At the time of writing, the model has been run through a single trial of each of the 30 scenarios flown by human participants. Because of the dependence on the external simulator, the model can only be run in real time. Consequently, ACT-R parameters have not been optimized for this task. Instead, we have used default ACT-R parameters to control the timing of perceptual, cognitive, and motor events. From these runs, we have been able to perform some initial qualitative comparisons between the model and human data, presented in Table 1. These comparisons are still at a high level, and further refinements to the model will be required to capture the human performance data at even finer degrees of resolution.

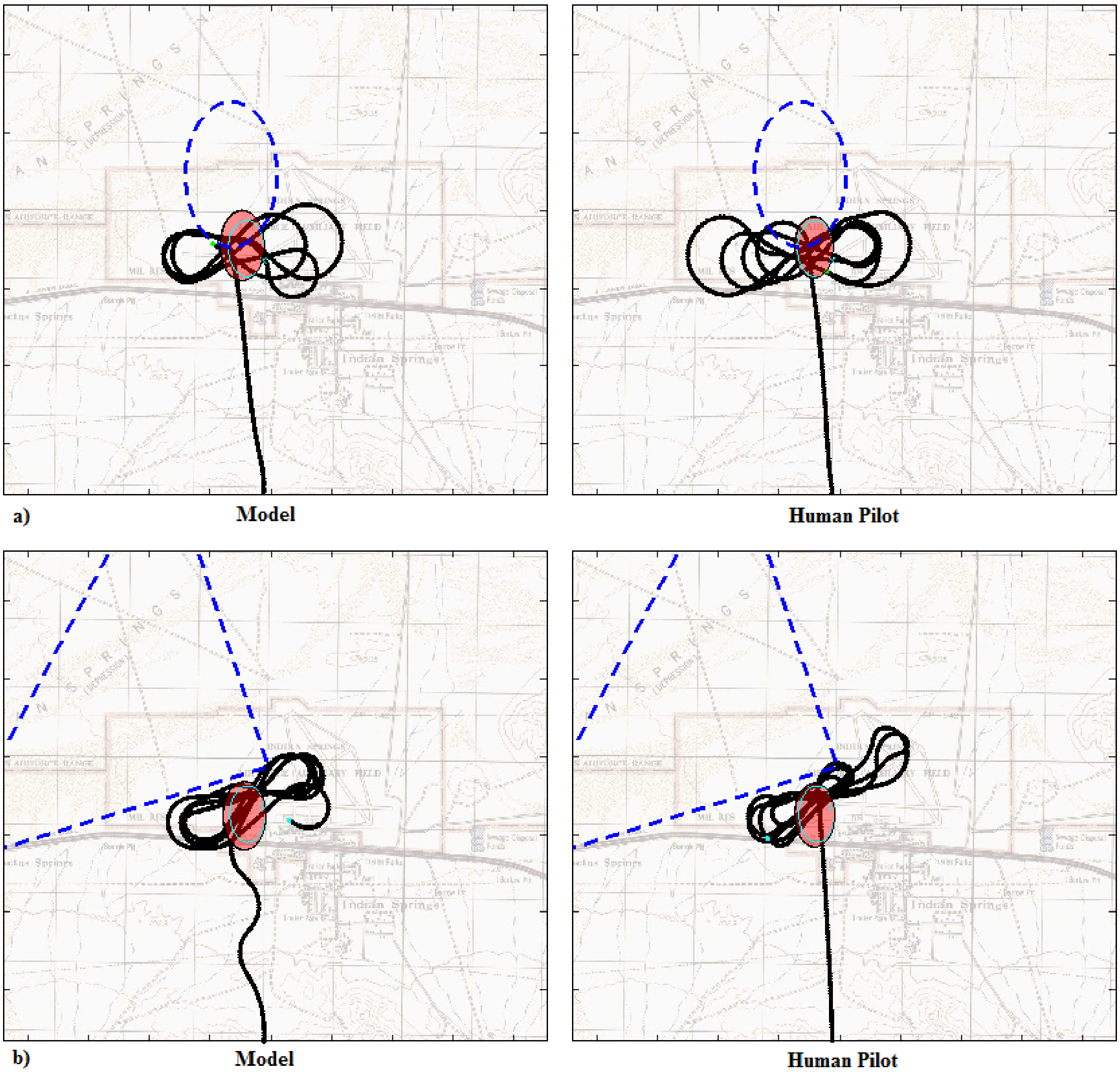

Figure 3: Overhead views of flight paths for the model and a sample subject on 2 different scenarios. The aircraft starts in the middle at the southern edge of the map and follows a path toward the hole in the clouds. The highlighted circle indicates the target viewing area. The first scenario a) has no wind and a circular no fly zone indicated by a dotted outline that overlaps part of the hole. The second scenario b) has a 15mph wind coming in from the south and a polygonal no fly zone.

Table 1 shows performance characteristics averaged across all 30 scenarios for the models and for the pilots. Data from the pilots were averaged across all 11 pilots who participated in the original study by Schreiber et al. (2002). The top section shows mean data across all 10 minutes of the scenario. The middle section shows data averaged across all times while the aircraft was in a position where it could view the target. The bottom section shows data averaged across all times when an aircraft has a bank angle greater than five degrees and is assumed to be turning. Bank angles are absolute deviations from 0 degrees regardless of direction.

Table 1: Performance measures from the model and human pilots averaged across all 30 scenarios. An * indicates the model datum falling within one standard deviation of the mean across scenarios. A ** indicates falling within half of a standard deviation.

		Model	**Pilots**
All	distance from target	1064.8 m **	969.51 m
	Time on target	99.12 sec **	109.14 sec
	Bank-angle violation	9.44 sec **	8.94 sec
	Stall violation	0.56 sec **	1.68 sec
	Altitude violation	0 sec **	2.62 sec
	Min altitude	11,998 ft	11,503 ft
	max altitude	13,695 ft	12,439 ft
	mean altitude	13,158 ft	11,954 ft
	final altitude	13,481 ft	11,919 ft
TVA	Bank	2.1° **	2.5°
	Pitch	2.09° **	1.74°
	Speed	70.7 kn	64.8 kn
Turn	Bank	28.5°	25.3°
	Pitch	1.4° *	2.4°
	Speed	75.1 kn	65.9 kn

It should be pointed out that these data do not capture some significant differences between the pilots and the current version of the model. In general, the model is actually getting more time-on-target in the easy scenarios, but the pilots are performing better on the more difficult scenarios.

The table demonstrates that at this point, the model performs within the range of human performance on this task overall. This also gives us some idea about where the model can be improved. These improvements can come in the form of changes to declarative knowledge (the model currently is more aggressive than participants about maintaining a high altitude), changes in procedural knowledge (e.g., the model does not plan based on an expected future position), and changes in parameter values. As more data is collected from the model more detailed analyses of each scenario will provide further insights into how the model can be improved.

4. Conclusions

Though the model described here is not yet able to stand in for another human in a cooperative training environment, we have demonstrated that cognitive models can be successfully implemented to operate in complex and naturalistic tasks. Our model is able to perform reconnaissance missions in the UAV STE, and performs similarly to human pilots at qualitative and aggregate levels.

This model extends other modeling work focused on maneuvering the Predator UAV (e.g., Gluck et al., 2003). The basic maneuvering tasks are difficult, both for humans and for the model. Whereas precise control of the Predator is not required to perform the reconnaissance missions, the task modeled here places additional demands on the architecture. For instance, control of the Predator must be performed while simultaneously reasoning about the reconnaissance task where decisions must be made regarding where to fly, how to maximize TOT, and how to efficiently avoid penalty time.

The model that we have presented here captures human performance on this task at a high level, but additional refinement will be needed to increase its cognitive validity. Specifically, we are working to refine the model to better capture more detailed flight characteristics across the 10-minute trials (e.g., the altitude profile). In addition, Park (2006) recorded eye movements from pilots performing the reconnaissance task. As ACT-R interacts with the task, it generates predictions about sequences of attention fixations that we can compare to human eye movement data. Such comparisons will provide a highly detailed means for evaluating the model.

5. Acknowledgements

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was sponsored by the Air Force Research Laboratory's (AFRL) Warfighter Readiness Research Division and by grant 02HE01COR from the Air Force Office of Scientific Research (AFOSR). ED's participation was sponsored by an AFRL internship administered by the Oak Ridge Institute for Science and Education.

6. References

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York, NY: Oxford University Press.

Gluck, K. A., Ball, J. T., & Krusmark, M. A. (2007). Cognitive control in a computational model of the Predator pilot. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 13-28). New York, NY: Oxford University Press.

Gluck, K. A., Ball, J. T., Krusmark, M. A., Rodgers, S. M., & Purtee, M. D. (2003). A computational process model of basic aircraft maneuvering. In *Proceedings of the Fifth International Conference on Cognitive Modelling*, 117-122. Bamberg, Germany.

Park, L. (2006). Identifying the knowledge and skills of expert pilots for validating a cognitive model of a Predator pilot flying a reconnaissance mission. *Unpublished master's thesis*, Arizona State University Polytechnic Campus.

Schreiber, B. T., Lyon, D. R., Martin, E. L., & Confer, H. A. (2002). Impact of prior flight experience on learning Predator UAV operator skills (AFRL-HE-AZ-TR-2002-0026). Mesa, AZ: Air Force Research Laboratory, Warfighter Training Research Division

Author Biographies

ERIC DIMPERIO is a researcher and doctoral candidate at Indiana University in the Department of Psychological and Brain Sciences. His research primarily focuses on the development of dynamic models of human learning and decision making. Eric was previously employed as a scientist for SPAWAR Systems Center San Diego in the Decision Support Systems branch. He earned a B.S. in computer science and a B.S. in psychology from Arizona State University (2001).

GLENN GUNZELMANN is a Research Psychologist with the Air Force Research Laboratory. His research is oriented around the primary interest of developing psychologically valid computational accounts of human cognition and performance. Research projects include using multiple tasks and contexts to identify and validate mechanisms for human spatial competence and for explaining the effects of fatigue on human cognitive performance. Dr. Gunzelmann received a B.A. in psychology from Albright College (1997), a M.S. in psychology from the University of Florida (1999), and a Ph.D. in cognitive psychology from Carnegie Mellon University (2003).

JACK HARRIS is a Computer Scientist at the Air Force Research Laboratory's Warfighter Readiness Research Division. His research interests include artificial intelligence for high performance and volunteer computing platforms and cognitive modeling in dynamic, complex, time-pressured domains, such as Predator reconnaissance missions. Jack earned his B.S. in Computer Science from the Georgia Institute of Technology in 2001.