2007

Modeling the Human Operator, Part II: Emulating Controller Intervention

Scott Bolland

Andrew Neal

Selina Fothergill

Michael Humphreys

Peter Lindsay

MODELING THE HUMAN OPERATOR, PART II: EMULATING CONTROLLER INTERVENTION

Scott Bolland, Andrew Neal, Selina Fothergill, Michael Humphreys, Peter Lindsay
The University of Queensland
Brisbane, Australia

Air traffic control workload prediction algorithms require a trajectory modeler to assess the flow of traffic through the sector. An accurate model of controller intervention is required to predict these trajectories. This paper overviews such a project aimed at emulating important controller actions in the Australian airspace.

Introduction

Common approaches to predicting the cognitive workload of air traffic controllers rely on extracting variables that are sensitive to the trajectories of aircraft traveling through a given sector. For example, Kopardekar and Magyaritis' (2003) Unified Dynamic Density (UDD) metrics include variables that are sensitive to the number of aircraft in a sector, the horizontal and vertical proximity of aircraft, the number of descending aircraft, the number of heading changes, and the degrees of freedom that an aircraft can move in a conflict situation. Although such variables can be extracted from historical data, aircraft trajectories are affected by controller intervention (such as the issuing of descents, the resolution of conflicts and the management of aircraft flow into airports). Thus, in order to accurately predict the workload associated with hypothetical situations (such as in the evaluation of new sector designs) accurate models of controller interventions (that affect the trajectories and flow of aircraft) are essential. This paper presents an overview of an empirically derived controller model aimed at mimicking the performance of human operators controlling the Australian airspace.

General Approach

At a high level, there are several controller interventions that affect workload metrics. Such interventions include the accepting and handing off of aircraft (as they affect the aircraft count metrics), issuing clearances to climb and descend (as they affect proximity metrics and potential conflicts), the resolution of conflicts, aircraft sequencing (i.e. providing adequate spacing between aircraft arriving into an airport), and weather avoidance. Although conflict resolution and flow optimization is regarded as a very complex mathematical problem (e.g., Durand & Alliot, 1997), human controllers have a limited cognitive capacity that makes an exhaustive search of the solution space for an optimal solution impossible. Instead, we have found through empirical observation, that rather than providing the "optimal solution" in such cases, controllers rely on a relatively small number of heuristics that, in combination, result in a potentially wide range of working solutions. Thus, we argue that the human operator can be considered (and modeled as) a complex system, in which a range of complex patterns of behavior emerge from the interplay between the constraints and complexity of the situation at hand and a fairly simplistic heuristically driven search engine.

In attempting to accurately mimic human problem solving, we have incorporated a number of psychologically plausible mechanisms and properties into our models. Firstly, the heuristics utilized by our models are empirically derived from experiments and self-reports from human controllers. Secondly, the general design of the cognitive architecture was motivated by psychological constraints. For example, as with other similar attempts to model controller behavior (such as the Operator Choice Model by Wicks et. al, 2005), we have assumed that the high level decision making involved in air-traffic control can be categorized as a controlled (as opposed to automatic) process, being under conscious awareness and serial in nature. For example, rather than attending to all aircraft on the screen simultaneously, it is evident that controllers "scan" the displays in a systematic manner, checking one aircraft at a time for potential problems or interventions that are required. Likewise, in resolving potential conflicts, mental exploration through the set of alternatives is serial and affected by time and working memory limitations. Because of such working memory limitations, we further assume that conflict resolution and planning is achieved through a limited search through the problem space. As such, we assume that the solutions generated can be far from optimal – a property that is in striking contrast to many of the automated techniques that have been proposed. However, we also assume that the mental heuristics utilized by controllers do lead to outcomes that are safe and most often expeditious.

In incorporating the above constraints, we have modeled the human operator using two separate, but interacting systems. Firstly, we have constructed a generic “problem solving engine” that emulates the process of mental deliberation involved in solution formation. For example, when given a conflict scenario, this module will generate a potential solution, taking into account the constraints on the situation. In addition, we have also implemented a related “action selection module” that emulates the scanning process, identifies current problems to be addressed, and selects the time at which proposed actions (generated by the problem solving engine) are taken. The following sections describe our general approach in detail.

Action Selection Model

The main flow of control of the agents in our simulations is driven by the “Action Selection Model” shown in Figure 1. This module reflects the serial mental processing involved in identifying and acting upon required interventions. Broadly speaking, aircraft are scanned one-at-a-time to determine if they require any intervention, what intervention is suitable and whether or not the action should be taken at that point. The following sections detail each of these processes.

Aircraft Selection

As mentioned earlier, controllers process aircraft in a fairly serial manner, using systematic scanning patterns to ensure that all aircraft are attended to. However, due to a lack of empirical data on the timing of such saccades, we have not attempted to accurately model the timing of such aircraft saccades. Also, as we are more interested in capturing when overt interventions are issued to aircraft, accurate modeling of such low-level behaviors is not crucial. As such, in our models we approximate this process by randomly selecting an aircraft on which to focus attention every 5 seconds (the rate at which the operator’s interface is updated).

Evaluation of Potential Interventions Required

Once an aircraft is selected, a “problem list” is generated, describing factors that relate to potential work that needs to be done on the aircraft by the controller. Such a problem list includes:

- whether an aircraft requires accepting into the sector
- whether an aircraft is not at a standard assignable level
- whether an aircraft is in conflict with other aircraft
- whether an aircraft is off track
- whether an aircraft requires sequencing

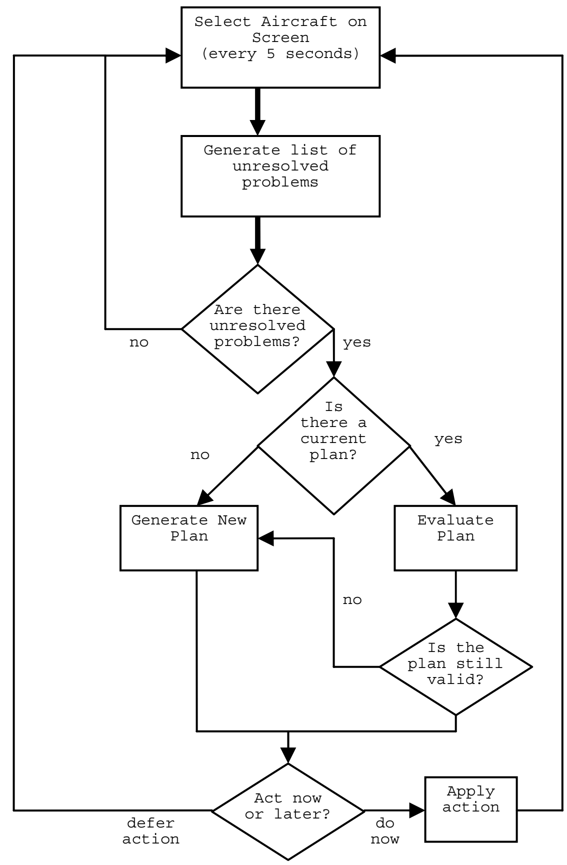

Figure 1. *Controller flow of control.* The controller model is a general serial framework for identifying and acting upon problems that need to be resolved.

Development of a Plan of Action

Instead of simply reacting to a given problem, developing a plan of action requires a form of mental deliberation. For example, when two aircraft are in conflict, changing the altitude of one to avoid the problem may lead to further conflicts. An integral part of problem resolution therefore is the mental evaluation of the consequences of various alternatives in order to select an appropriate action to take. As mentioned earlier, the mental deliberation required to generate such plans is achieved by our “problem solving engine”, described later in this paper. The output of this process is a “plan” containing a set of proposed actions on one or more aircraft. Such actions include:

- accepting an aircraft
- changing the level of an aircraft
- vectoring an aircraft
- changing the speed of an aircraft
- providing requirements on altitude changes
- returning an aircraft to track

The model contains an explicit memory for existing plans, so that replanning does not occur each time a given aircraft is focused on. However, these plans are evaluated for relevance, and are only used if the context has not changed (i.e. if the same set of problems still exist and the given solution still resolves the problem).

In terms of conflict resolution, plans of action often contain only partial solutions, containing suggestions of what to do in the immediate future. That is, rather than containing a set of temporally segregated behaviors, such complex solutions are handled implicitly by this system, by the creation of additional "problems" that need to be dealt with later. For example, in vectoring an aircraft to avoid a conflict, the vector itself will create an "off track" problem that will be dealt with on subsequent scanning cycles, returning the aircraft to its path once the conflict has been resolved. Such a methodology allows for more flexible and context dependent set of actions to occur.

Plan Execution

Once a plan of action has been formulated, it may or may not be executed immediately. Including delays in responses allows one to better model the distribution of actions taken by human operators. For example, when an aircraft needs accepting into a sector, there is a Gaussian-like distribution of acceptance times from actual controllers, rather than accepting them at the first possible instance. Assuming that the "need accepting" problem is detected early, one can explicitly write an acceptance time algorithm that closely matches the distribution exhibited by actual controllers.

Generality of the Approach

Although the general action selection process follows a fairly rigid structure, the framework was designed to also be open-ended, allowing any of a number of algorithms to be used for each decision in the control diagram. For example, in selecting an aircraft on which to attend, an aircraft may be selected randomly (i.e. with uniform distribution) from the set of aircraft on the screen. However, to more accurately capture the actual scanning utilized by controllers, a more complex algorithm could also be substituted, utilizing unequal selection probabilities based on such factors as screen saliency (i.e. there are visual cues given for aircraft that require certain tasks to be performed), how recently the aircraft has been scanned before, and whether or not the aircraft is being monitored closely. It is our aim to update each specific function as our empirical understanding of each of the subtasks grows.

The Problem Solving Engine

The aim of the problem solving engine is to model the process of mental deliberation and solution formation in a psychologically plausible manner. The development of a general framework for problem solving was driven by constraints identified from empirical studies with air traffic controllers, and other fundamental research. From our studies on problem solving where controllers verbalize their cognitions, there are a number of properties about their thought processes that are interesting to note:

(1) Mental exploration through the alternatives in novel situations is a serial process. That is, controllers identify conflicts using mostly pair-wise comparisons, and express a serial search and evaluation of the possible solutions.
(2) The problem space is discrete, with the degrees of freedom that an aircraft is generally given in a conflict situation being fairly limited and governed by a relatively small set of heuristics
(3) Complex patterns of solutions across a range of controllers can be understood as an emergent result from these simple heuristics and a fairly basic stochastic search process.
(4) Both time and working memory capacity is limited, making it infeasible to search through all options seeking the optimal solution. Instead, a range of "good" solutions are generated through a fairly short search through the possibilities using a set of simple yet powerful heuristics.

In capturing the above properties, our model uses an architecture inspired mainly by a class of psychologically validated models of human problem solving developed by the Fluid Analogies Research Group (Hofstadter & FARG, 1995). These models provide powerful insights into how to capture human capacity limits in terms of the breadth and depth of the controllers' search among decision alternatives.

In our approach, problem solving is modeled as a probabilistic search through a discrete solution space. Initially, there is a single state, representing the situation at hand, explicitly indicating the features requiring intervention (such as pairs of aircraft currently in conflict). Using a set of simple heuristics, possible courses of action are generated, with a single action being probabilistically chosen to evaluate. The consequence of this action is evaluated in a new explicit state, and checked to see if all the noted problems have been resolved. If not, courses of action to consider from this state are proposed, with the next action to evaluate being chosen probabilistically from the entire list.

In selecting which action to evaluate next in calculating a solution, not all alternatives are weighted equally. Firstly, alternatives are given a selection penalty relating to how far away their corresponding state is from the root state. Such a penalty promotes the exploration of simple solutions (i.e. solutions that can be achieved in a few steps) over complex ones.

A second form of penalty is given to actions based upon the number of problems in their corresponding state. Thus, if an action is evaluated and creates a new state that increases the number of problems, this "train of thought" is readily abandoned. In contrast, if an action is found that resolves one or more problems, this line of investigation is likely to continue. This selection bias promotes a relatively quick (as opposed to thorough) search of the problem space.

A third set of biases that is aimed at minimizing the disruption to current aircraft trajectories has also been implemented. For example, if the proposed new flight level deviates from the aircraft's preferred trajectory, a penalty is applied to the weight. Likewise, if an aircraft needs to vector off track, a further penalty is also applied. Such penalties in weight allow for options that are more highly preferred by the aircraft to be explored first. However, if these preferred options lead to further problems (e.g. additional conflicts), the less preferred options may also be explored.

From our experiments, like humans, the model is often (but not always) able to find a simple solution to a problem without an exhaustive search of the decision space. This process is aided through the use of simple, yet powerful heuristics that provide a specific and small amount of options that a controller is likely to contemplate, that generally yield safe and expedient solutions. The following section details examples of such heuristics.

Heuristics

The aim of the heuristics are to specify the degrees of freedom that aircraft are able to move to potentially avoid the given conflict. It is then up to the problem solving engine to explore the various options to see if they are viable. For example, it may be discovered that a particular aircraft, due to its performance characteristics, cannot reach the given level specified in time, or doing so may place it in conflict with a third aircraft.

The heuristics that we have implemented in our model were empirically derived and are simple in nature, yet provide viable options in a range of cases. An example heuristic is detailed below:

Example Vertical Heuristic

Generally speaking, in the Australian Airspace, aircraft must be separated by at least 5nm laterally or 1000ft vertically. Thus, controllers often resolve conflicts by changing the altitude of one or more aircraft. A general heuristic that can be used to resolve a conflict is to assign a standard level to one of the aircraft that is at least 1000ft above the predicted altitude of the second aircraft, or at least 1000ft below. As the second aircraft may be on climb or descent, the predicted altitude of the aircraft may be a range of altitudes given the best and worst case climb or descent rates (as in Loft, Bolland, & Humphreys, 2007).

As shown by figure 3, such a heuristic emulates the solutions given by controllers in a number of cases:

Both aircraft are flying level Given that both aircraft are flying level, possible degrees of freedom to avoid the conflict is to descent either by 1000ft or to climb either aircraft by 1000ft (see figure 3a).

Aircraft A is level, whereas aircraft B is descending At this particular case, the altitude at which B will be at the point of conflict is uncertain, and is represented by a range of possibilities (calculated using B's worst and best descent rates). In this case, aircraft A can either be moved to the next highest standard level 1000ft above the predicted range for B, or the next lowest standard level 1000ft below the predicted range of B (see figure 3b).

Aircraft A is descending, whereas aircraft B is level Similar to the first case above, the degrees of freedom for A is to make sure that A is either 1000ft above or below B at the point of conflict. In this first case, this may involve cutting off the descent of A (if it is descending from an altitude higher than 1000ft above B), or asking the aircraft to climb over the level (if its current altitude is less than this amount). If attempting to fly under the level of B, and the

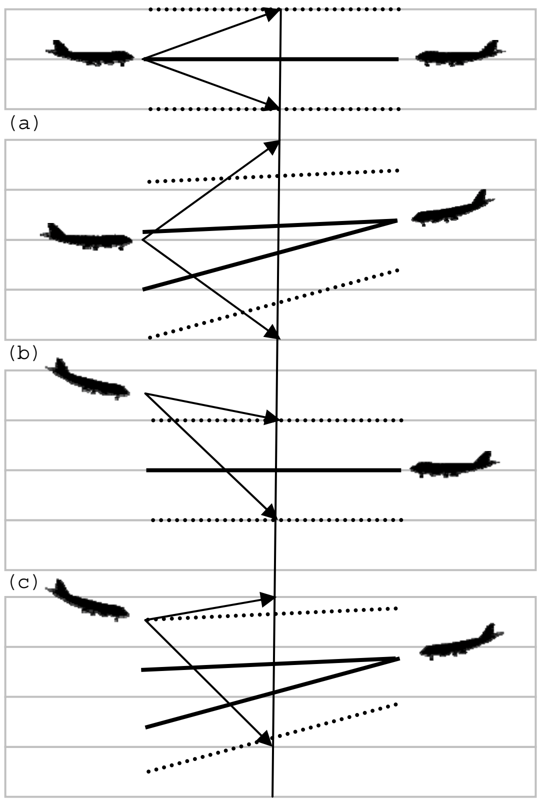

Figure 3. *Resolving a conflict through a single level change*. Given the point of predicted separation violation, a conflict can be resolved by issuing an aircraft a standard level at least 1000ft above or below the predicted range of altitudes at which the second aircraft may lie. This simple heuristic works in a number of cases: (a) when both aircraft are flying level; (b) when the first aircraft is level, and the second is descending or ascending; (c) when the first aircraft is ascending or descending and the second is level; and (d) when both aircraft are ascending or descending.

cleared flight level for A is below 1000ft under B, this may be achieved by issuing a "requirement" for the aircraft to reach the level by the given point.

The above heuristics are also applicable to cases where both aircraft are either descending or ascending (there is a separate rule below for when one aircraft is on descent and the other is climbing).

As is demonstrated by this case, a simple heuristic can be used to provide a constrained set of potential actions in a range of situations. Part of our ongoing research is to specify the heuristics that are used by controllers, that lead to solutions being found in a small number of mental steps.

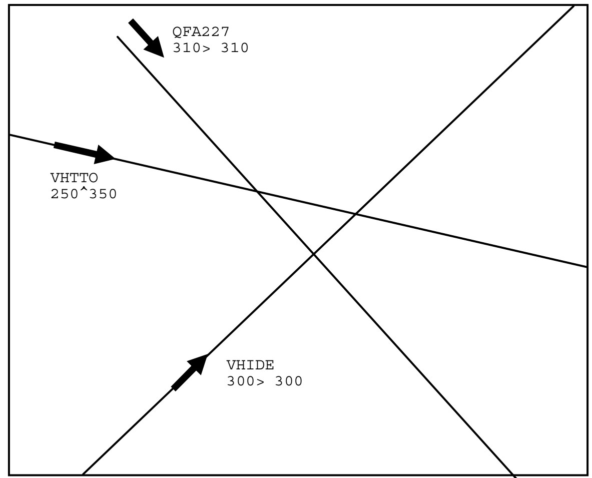

Figure 4. *A typical conflict scenario.* In this case, VTTO is in conflict with QFA227, with the solutions being constrained by the additional aircraft.

The following sections detail an example problem and corresponding solution generation that is representative of those typically faced by controllers.

Example Scenario

Figure 4 illustrates a typical conflict scenario faced by air traffic controllers. In this example, VHTTO, currently at flight level 250, is on climb to flight level 350. However, in doing so, it is in conflict with QFA227, currently at flight level 310.

Figure 5 illustrates a typical run of the problem solving engine. Firstly, a range of possible solutions is generated based upon the heuristics mentioned earlier. In this case, either of the two aircraft could be assigned 1000ft above or below the other aircraft. However, although it is possible to change the cleared flight level of QFA227, this option is penalized because it would require QFA227 to deviate considerably from its preferred trajectory. Based upon probabilistic selection, in this run, the alternative of assigning an intermediate level of 300 to VHTTO is explored. In the resulting state however, it is noted that this places it in conflict with VHIDE. As the issue is not yet resolved, a second action is evaluated, this time, exploring the issuing of a requirement for VHTTO to be above level 320 before it violates lateral separation with QFA227. This option is evaluated, leading to a state in which no further problems are identified. Thus, the possible solution in this case is issuing a requirement to VHTTO.

The problem solving engine has been tested against human decision making on a range of problems, displaying a promising fit to the data.

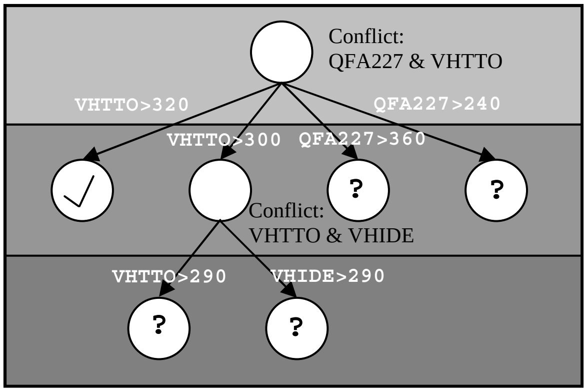

Figure 5. *Problem Space Sampling.* Given the original problem state, actions are sampled probabilistically (creating new states), until a solution is found.

Ongoing Work and Future Directions

Our general framework for modeling controller intervention has been tested on a number of tasks (including the acceptance of aircraft, the issuing of standard levels, and the resolution of conflicts), showing a promising fit to empirical observation. For example, in one of our experiments, we issued human controllers with static scenarios involving multiple conflicts, and asked them to verbalize the solution making process. In most of the scenarios, our model well mimics the solutions provided by controllers, as well as the sequence and number of solution states explored. However, in a few cases, the model exhibits unrealistic mental behavior, exploring too many possibilities in the formation of a solution. Through re-exploring the comments provided by human controllers, we have noted a range of heuristics that have not yet been implemented in our system. For example, in a multiple conflict situation, there is a bias to focus on aircraft that are causing the most number of problems, and "get them out of the way first" in order to simplify the rest of the problem solving process. It is our current objective to identify and implement such heuristics that allow acceptable solutions to be generated in a short amount of time. Work to better model the temporal distribution of action execution is also underway.

References

Anderson, J.R., & Lebiere, C. (1998). *The atomic components of thought*. Lawrence Erlbaum associates, Mahwah, NJ.

Durand, N., & Alliot J. (1997). Optimal Resolution of En Route Conflicts, *Laboratoire d'Optimisation Globale*, CENA at ENAC, Toulouse

Hofstadter, D. R. & The Fluid Analogies Research Group (1995). *Fluid Concepts and Creative Analogies*. Basic Books, New York.

Kopardekar, P., & Magyarits, S. (2003). Measurement and Prediction of Dynamic Density. In *Proceedings of the 5th USA/Europe ATM 2003 R & D Seminar*, Budapest, Hungary.

Loft, S., Bolland, S., & Humphreys, M. (2007). Modeling the human air traffic controller, Part I: Expert-trainee differences in conflict detection. To appear in *Proceedings of the 14th International Symposium on Aviation Psychology*.

Wicks J, Connelly S, Lindsay P, Neal A, Wang J, Chitoni R (2005). Simulation of Air Traffic Controllers' Behaviour Using the Operator Choice Model. In Zerger, A. and Argent, R.M. (eds). 2005 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2005, pp. 3023-3029.