In 2000 Proceedings of the International Conference in Artificial Intelligence

Air Traffic Control Agents: Landing and Collision Avoidance

Henry Hexmoor and Tim Heng
University of North Dakota
Grand Forks, North Dakota, 58202
{hexmoor,heng}@cs.und.edu

Abstract. *This paper presents an example of share autonomy between a human air traffic controller and two agents that assist the human operator. The agents monitor and interact with the dynamic situation of flights around an airport tower. When a situation arises that requires an air traffic response, the agents notify the human operator about the situation along with a suggested action. If the human operator does not respond in time, the agent will take over and issue the needed response.*

Keywords. Alert Agents, Shared Autonomy, Simulation, Air Traffic Control

1. Introduction

Agents that detect and process possible collisions as well as landing requests are implemented that assist the human air traffic controller. The agents share the responsibility as well as autonomy with the human operator. Upon initial detection, the agents allow the human operator first chance to reply and assume minimal autonomy. Gradually, as the human operator does not take action, the agents assume greater autonomy. When sufficient time expires and the situation becomes critical, the agent autonomy reaches a maximum level at which point they take action. The collision agent instructs the pilots with maneuvers that maintain safe separation distances and collision aversion.

This program works along with a tower simulator we have developed [1,2]. The tower simulator simulates 6 planes flying in a 20 by 20 miles grid around the Grand Forks airport tower. The simulated planes can fly in a straight line, turn left or right, ascend, descend, land or circle the tower. Figure 1 shows the main panel of simulator. The human operator has a simulated view out of tower.

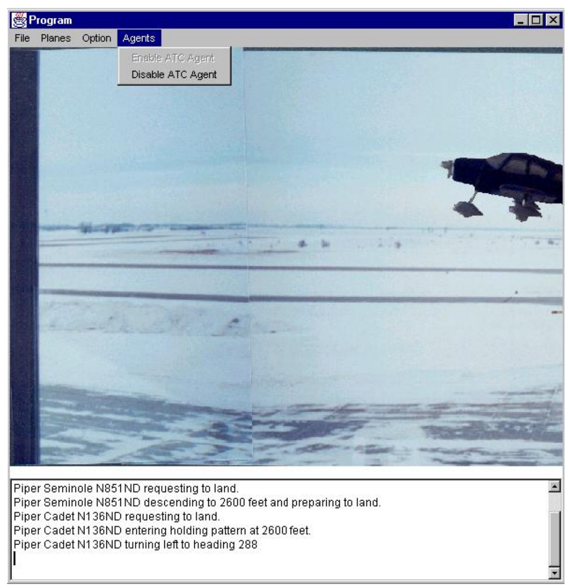

Figure 1. The main simulator panel

The following data about each plane on the radar is made available to our ATC agent. All ATC data is current and the latest being used by the simulator.

(a) *the plane location*: This is recorded in a three dimensional coordinate system.
(b) *the flight direction*
(c) *the flight speed*

(d) *the plane ascend/descend pitch*
(e) *the plane intention to land or their status in holding pattern*

The two main functions of our ATC agent are *to detect and prevent collisions*, and *to process landing*. Agent will display dialog boxes for collision detection and when there are requests for landing. The windows will stay open until either it is timed out or the human operator overrides the agent decision. Our agent maintains three queues. The first queue is the landing request queue. Along with a request to land, each pilot generate a landing priority; a number between 0 and 4. 0 is the default priority when here are no unusual circumstances, whereas 4 is the highest priority number corresponding to the most urgent emergency. The landing request queue is arranged with the ascending priority so the most urgent priorities are handled first.

The second queue is collision queue. Planes near enough to pose a collision threat are assigned a priority from 1 to 4. 1 signifies a near miss; whereas, 4 signifies a collision that is about to happen. Later in this paper we will explain how these priorities are a function of distances and flight projections.

The third queue maintains the order of planes that have requested landing but due to high traffic are in the holding pattern.

The ATC agent maintains a combined priority queue of all planes in its radar screen using the range of priorities from 0 to 4. ATC agent may use the pilot's landing priority when there are no other concerns. However, the ATC agent will use other cues for reprocessing its priority queue. In general, the following are the factors that contribute to the agent's determination of priority:

(a) pilot's requests and landing priority.
(b) perception of possible collision threats
(e) projection of paths for planes initially detected to be on a collision course
(c) landing requests from different pilots with priorities that are conflicting with prior requests
(d) planes in various landing stages
(f) status of planes in holding pattern

The agent will issue one of the following commands:
(a) clear to land
(b) go to the holding pattern,
(c) come out of the holding pattern
(d) avoid a collision by a maneuver (turn, ascend, descend)

In the remainder of this paper we will describe how our agent maintains these three queues and determines the priorities.

2. The Collision Queue

Distance between every plane is checked against every other plane in the radar range. Distances are checked in the horizontal plane and then comparisons are made in the vertical axis. See Figure two and three. There are two different sets of collision ranges for determining a collision threat. One range is for flight in the air space before they request landing. This range is two miles on the xy-axis and a 1000 feet between planes on the z-axis. The second range is when the plane or planes are going to be landing or going to be in a holding pattern. This range is one mile apart on the xy-axis and 500 feet between planes on the z-axis. If the separation distances between the two planes are violated, the program will assign a number to correspond to the severity of collision threat. Collision threat is a priority number similar to landing priorities. The numbers range from 1 to 4.

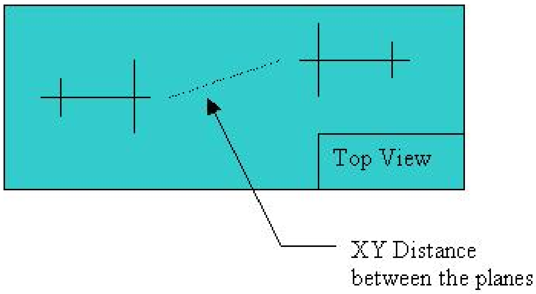

Figure 2. *The horizontal separation distance between two planes computed.*

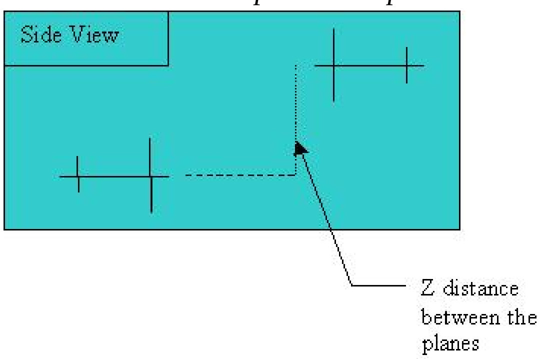

Figure 3. *The elevation distances between two planes computed*

Let's first consider the general flight conditions. 1 designates very low likelihood of a collision; the planes have just reached the boundaries of collision concern. I.e., if the planes were about 2 miles apart on the xy-axis and almost 1000 feet apart would have a number of 1 assigned to it. If the planes were about ½ mile apart on the xy-axis and about 300 feet apart on the z-axis, the system would assign a number of 4 to it, since they are very close to each other. Priorities in the near the tower flight conditions are divided with the same proportions.

The agent will make projections on the flight paths. Some collisions may be only near misses and continued paths will not lead to collision. The agent does this prediction for the next second of flight. If the original collision situation will worsen, the priorities are increased; otherwise, they are decreased.

The priority queue will contain the planes in descending order so the situation with the highest priority is addressed before the situations with lower priority. For example, if there is both a collision and a near miss, the agent will attend to the collision before solving the near miss.

If the distance between two planes is within 1 mile on the xy-axis and 500 feet away, the agent will command one of the planes to ascend or to descend. When the distances are closer than one mile and 500 feet the agent will command the planes to turn left or right.

The collision dialog window is shown in Figure 4. The timer is very important, if the air traffic controller does not handle the problem right away there could be an accident. So with the timer insures that either the human operator or the agent will handle the problem. The timer will be set to the severity of the collision threat. The closer the two planes are the less time will be used on the counter.

3. The Landing and Holding Pattern Queues

We will first present the process of landing and then discuss the agent's landing scheme. There are 5 different stages in the landing flight plan. A stage is a certain part of a flight plan for landing the plane. The plane is in stage zero when it receives a confirmation that it can proceed to land. The plane flies to the East side of the airport and aligns with a line that is heading toward south of the airport.

The plane will be in stage one when the plane is on southward line and heading south at about 170 degrees.

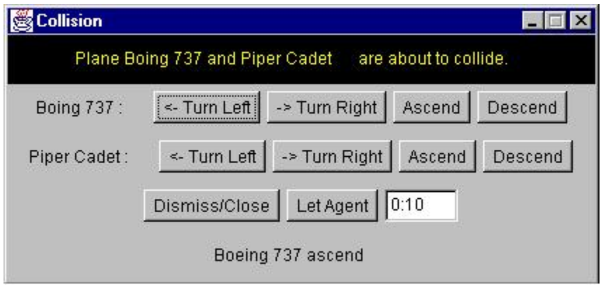

Figure 4. *The collision dialog window. Each plane can receive different commands: Turn left or right, ascend or descend. "Dismiss/Close" button will do nothing to the planes, whereas "Let Agent" button will allow the agent to perform the actions being suggested by the agent at the bottom of the screen. When the timer (10 seconds is shown) runs out, the agent will perform the suggested action.*

The plane will be in stage two when it turns right to start heading back to the runway. It will maintain stage two until it reached a heading of 350 degrees, at which point it would be lined up with the runway, then, the plane would be in stage three. Stage four is when the plane is on its final approach to the runway and it starts descending for the landing. Once the plane is on the ground, it would leave stage four and taxis over to the tarmac to park.

Only a single plane can be landing at a given time and will be given permission to land. All other request for landing will receive a command to enter holding pattern. Holding pattern is a circular flight at a specified altitude. The agent maintains a queue of planes in the holding pattern.

The agent monitors progress of planes along landing stages. The agent maintains a list of planes in the 5 flight stages.

Once the agent receives a request to land, the agent will check to see if they can land. It will have to check to see if any other planes are landing at that time. If a plane is in stage zero, one or two, the agent cannot send another plane to land; it will have to wait until the plane that is landing is past stage two. Then the agent will have to put the planes into a holding pattern around the tower until the landing path is clear. If the first three stages are clear, then the agent can tell the requesting plane it is clear to land. A flight in the holding pattern is given permission to land when the current landing plane enters stage three and is popped from the holding pattern queue.

The landing dialog window is shown in Figure 5. As shown in the Figure, there is two override options that the user can choose: clear to land, and enter a holding pattern. For the collision window, there are a few overriding choices. The human operator may command one plane to ascend, descend, and turn left or right. The human operator can tell the second plane to do a similar action.

Figure 5. *Landing dialog window. "Let Agent" allows the agent to take care of the landing. "Let Land" is used by the human operator to give permission to land. "Hold Landing" is used by the human operator to decline clearance for landing and to send the plane into holding pattern. The agent action is shown in the bottom of display.*

4. The Combined Queue

The combined queue is constructed by merging the three queues we have discussed with the ascending priorities.

Consider the following example: The landing request queue has two planes in it, plane ND239D wants to land and it does not have an emergency. Plane ND1002 also wants to land but there is a person on board that needs to have medical help, so there is an emergency on board and it will considered to have priority 2.0.

The landing queue: ND1002(2.0) ND239D(0)

Planes MN3230 and SD392 are about 1.32 miles away on the xy-axis and 950 feet apart on the z-axis, and the planes will be getting closer together in the next second. This situation will be assigned a priority number of 2.4.

The collision queue: MN3230/SD392 (2.4)

The queue for holding pattern is empty since there are no planes waiting to land.

The holding queue: NIL

After merging the queue we will have the following:

Combined queue: MN3230/SD392 (2.4)
ND1002(2.0) ND239D(0)

The agent will attend to the collision between MN3230 and SD392, then the emergency landing will be given a clear to land, and ND239D(0) will be told to enter holding pattern until ND1002 reaches stage 2 of its landing.

5. Conclusion

We have presented an agent program that shares its autonomy with the human user for air traffic control. We have shown an intuitive sharing of autonomy where the human is the supervisor. We believe our agent addresses a niche. Clearly, the added level of automation and assistance is welcome in the domain of air traffic control [3].

Acknowledgements

We acknowledge the ATC Group for their work on the simulator. Nathan Faiman implemented the dialog boxes and Mark Schroder refined our implementation of landing.

References

[1] Hexmoor, H. and Heng, T. 1999. ATC Tower Simulator: TACUND, Proceedings of **IASTED** on Applied Modeling and Simulation, Cairns, Australia.

[2] Hexmoor, H. and Heng, T. 2000. Air Traffic Control and Alert Agent, In Proceedings of Agents 2000, Barcelona.

[3] National Research Council, 1998. **The Future of Air Traffic Control.** National Academy Press.