AIAA Guidance, Navigation, and Control Conference and Exhibit
16 - 19 August 2004, Providence, Rhode Island

AIAA 2004-4992

# Automated Conflict Resolution for Air Traffic Management Using Cooperative Multiagent Negotiation

Steven Wollkind* John Valasek† Thomas R. Ioerger‡

*Texas A&M University, College Station, TX, 77845*

**The National Airspace System in its current incarnation is nearing its maximum capacity. The Free Flight initiative, which would alter the current system by allowing pilots to select more direct routes to their destinations, has been proposed as a solution to these problems. Allowing pilots to fly anywhere, as opposed to being restricted to planned jetways, greatly complicates the problem of ensuring separation between aircraft.**

**In this paper we propose using cooperative, multi-agent negotiation techniques in order to efficiently and pseudo-optimally resolve air traffic conflicts. Our system makes use of software agents running in each aircraft that negotiate with one another to determine a safe and acceptable solution when a potential air traffic conflict is detected. The agents negotiate using the Monotonic Concession Protocol and communicate using aircraft to aircraft data links.**

**There are many benefits to using such a system to handle the resolution of air traffic conflicts. Automating CD&R will improve safety by reducing the workload of air traffic controllers. Additionally, the robustness of the system is improved, as the decentralization provided by software agents running in each aircraft reduces the dependence on a single ground based system to coordinate all aircraft movements. The pilots, passengers, and carriers benefit as well due to the increased efficiency of the solutions reached by negotiation.**

## I. Introduction

Maintaining separation between aircraft is an important task of air traffic controllers. Controllers are assisted in this task by national network of jetways to which air traffic is currently confined. These jetways can be thought of as predefined routes, or highways, through the national airspace system. Unfortunately, the jetway system is an inefficient use of airspace, and current research indicates that with continuing increases of air traffic it will cease to be viable. In order to more efficiently use the national airspace the system will need to become more flexible and open.

A NASA initiative currently being researched, known as free flight, would allow pilots to chose their own direct routes rather than relying on air traffic control and the jetway network. Allowing pilots to plot their own preferred courses, free from external interference from ground control, will allow a more efficient use of airspace. Additionally, when aircraft are allowed to take advantage of local weather or traffic conditions the efficiency of their own flight plans will be greatly increased.[1]

These benefits come at the cost of a significant increase in the difficulty of ensuring safe separation between the aircraft in the airspace. The jetway network, while inefficient, greatly reduces the complexity of the task of maintaining aircraft separation. As the national airspace is shifted from a constrained network of highways in the sky to a system where pilots select their own routes, human air traffic controllers may no longer be able to manually provide this separation.

*Research Assistant, Department of Computer Science, Mail Stop 3112, AIAA Student Member.
†Associate Professor, Department of Aerospace Engineering, Mail Stop 3141, AIAA Associate Member.
‡Associate Professor, Department of Computer Science, Mail Stop 3112

For this and other reasons, there has been a great deal of research into automating the detection and resolution of air traffic conflicts. The majority of this research has approached the problem from a non-cooperative, game theoretic angle. These approaches often solve for solutions that will work in the worst case scenario. While these methods produce viable solutions, they are far from optimal.

We propose applying multiagent negotiation techniques to the problem of air traffic conflict resolution. By treating the aircraft as agents that are competing for the limited resource of airspace, we can use well-known negotiation techniques to achieve more efficient solutions than those produced by other automated CD&R systems. By allowing aircraft to cooperate and negotiate to resolve their conflicts, two goals will be accomplished. First, the aircraft will be able to resolve impending conflicts without the assistance of a central ground control system. Secondly, the inefficient worst case scenario planning which is common in currently proposed systems can be avoided.

This paper will demonstrate that multi-agent negotiation techniques can be effectively applied to resolve air traffic conflicts and that the solutions generated by these methods are more efficient than those created by traditional methods or alternative non-cooperative solutions.

## II. Background

The current systems and methods in place to manage the National Airspace System (NAS) do not scale well. The inefficiencies are due to the fact that air traffic is confined to predetermined jetways that are rarely the optimal path from the origin to the final destination. Additionally, research has indicated that the current hub and spoke model will not be able to handle the increased demand for air travel that is anticipated over the next 10 years.[1]

The Free Flight concept has been introduced as a potential solution to these problems. Under Free Flight, pilots are given more freedom to select their own routes, altitudes and speeds. Implementation of such a system would greatly increase the efficiency of airspace usage, as well as reducing travel time and fuel costs for carriers.

Increased usage of airspace comes at a cost, however. As the airspace system moves away from predetermined jetways towards pilot defined flight plans, the task of ensuring safe aircraft separation becomes much more complex. Planes can be anywhere at any time. The existing system of ground based radar and air traffic controllers relies heavily on the rigid guidelines and procedures that confine aircraft to certain regions. When pilots can set their own flight, plans the entire airspace must be monitored to prevent losses of separation.

We believe that the success of Free Flight depends heavily on an efficient, on-board conflict detection and resolution system. As the number of potential conflicts rises, there is more pressure on the system to be as automated and decentralized as possible to prevent unreasonable workloads for air traffic controllers. Towards this end, we propose the use of a multi-agent architecture to distribute the workload of detecting and resolving conflicts among the aircraft themselves with minimal intervention from ground based facilities.

### A. Multi-agent Systems

An 'agent' is a piece of code or a program that is capable of analyzing input from its environment to make decisions and use those decisions to take actions. This idea of rational agents was developed in the artificial intelligence field, and they are used to solve problems that involve individuals that make decisions and take actions. One example of an agent is a robot control program that analyzes sensor data and controls the motion of the robot.

Combining several agents together creates a multi-agent system in which complex interactions and behaviors become possible. Agents can communicate, collaborate, and compete with each other to achieve their goals. Some multi-agent systems feature a team of agents with a common goal, but systems of individual agents are also common. In these systems it is typical for each agent to have its own set of goals, which may conflict with the goals of other agents in the system. In these cases, the agents use automated negotiation protocols to resolve their conflicts in such a way that the outcome is acceptable to all parties involved. The possibility for cooperative negotiation is one of the strengths of multi-agent systems. Agents are free to resolve their conflicts by cooperatively developing and implementing solutions that are more efficient than either agent could pursue on its own.

The NAS can very naturally be regarded as a multi-agent system in which each aircraft is an agent with its own goals (destination, time frame of arrival, service standards, etc) that are independent of the goals of the other aircraft. With this approach, it is possible to apply the negotiation techniques that have been developed for such systems. These techniques will allow the aircraft to cooperate to resolve airspace conflicts, and through this cooperation, resolve their conflicts as efficiently as possible.

Due to the natural parallels between the National Airspace System and a multi-agent system, as well as the benefits that can be realized from the use of such a system, we feel that this is an ideal approach for solving the air traffic conflict problem. We propose that each aircraft carry an on-board computer which would run a simple software agent like those described above. These agents will continuously monitor the airspace around them using current communications technologies to determine if a conflict is imminent. Agents will be free to communicate with one another and negotiate, using the simple negotiation system known as the Monotonic Concession Protocol to calculate safe and efficient solutions to the conflict. Once a resolution has been agreed upon by the agents, the pilots of the aircraft involved would be alerted that a course change had been negotiated and would be presented with a modified flight plan. The negotiation process itself would take place automatically between the agents, without the participation of the pilots.

### B. Conflict Detection and Resolution Systems

There has been a great deal of research in the last few years focused on automating the CD&R process. Kuchar and Yang provide an overview of a number of these systems.[2] While there have been many methods proposed, few have applied the techniques of distributed artificial intelligence and multi-agent systems to the problem, and fewer have applied cooperative methods.

Tomlin, Pappas and Sastry have published a number of papers on automated air traffic conflict resolution. They have developed a non-cooperative distributed system based on game theory. When a potential conflict is detected, the aircraft use a worst case scenario analysis based on the pursuer/evader problem to plot a trajectory that preserves separation regardless of the actions of the other plane.[3–5]

While the game theoretic worst case scenario approach does ensure separation between the aircraft, it does so at the cost of efficiency. Any system which is adversarial in nature cannot produce optimal resolutions, as the solutions are based on worst case analysis. In order to maintain the flexibility to respond to any possible course change by the pursuer, the evader must trade away efficiency. Viable conflict resolutions that are reached through cooperative methods will often be of higher utility to the aircraft involved.

Wangerman and Stengel have presented a solution that treats aircraft as negotiating agents, but this system does not take advantage of the possibilities for agent to agent communication.[6] Their principled negotiation system involves agents negotiating indirectly via a centralized controller, which is responsible for ensuring that suggested trajectories do not conflict.

Menon, Sweriduk and Sridhar have developed a method which uses cost functions similar to those found in agent based systems.[7] However, their system is based on quasilinearization optimization methods rather than agent to agent negotiation. In their system, aircraft trajectories are defined as sequences of four dimensional waypoints (three spatial dimensions and a time dimension) and various parameterization methods. Using the parameterized representation of initial aircraft trajectories, appropriate cost functions, and the sequential quadratic programming method, optimal trajectories are computed. These new trajectories minimize the overall cost of the system subject to the constraint that the aircraft not violate protected zones of other aircraft. This method can handle any number of conflicting aircraft.

Prior work at Texas A&M has focused on the idea of using agent systems in air traffic management to resolve conflicts and perform other pilot advisory tasks.[8–10] These methods have primarily utilized agents as advisers which monitor situations and provide pilots with warnings or recommendations. In general, they do not communicate with other agents or entities aside from the pilot.

### C. Monotonic Concession Protocol

The monotonic concession protocol (MCP) is a simple protocol developed by Zlotkin and Rosenschein for automated agent to agent negotiations.[11,12] The MCP captures the incremental bargaining process that takes place between negotiating parties. The agents incrementally make proposals and counter proposals of progressively less value to themselves until a middle ground is reached that can be agreed upon by both agents.

To begin an explanation of the MCP we must first introduce some notation. We will denote the negotiating agents as $A$ and $B$. At any time each agent has some plan that it is following. We will denote the agent's plans as $P_A$ and $P_B$. A *Deal* will be defined as a pair of plans $(P_A, P_B)$.

Each agent $i$ has a utility function $Utility_i$ which relates a plan of action to a value which represents the desirability of that plan for the agent. As it is also useful to discuss the utility of deals for the various agents, we define a utility function that operates on deals. Given a Deal $D = (P_A, P_B)$ then $DealUtility_i(D) = Utility_i(P_i)$. For example, $DealUtility_A(D)$ is the value of $U_A(P_A)$.

We will also define the negotiation set $NS$ to be the set of all Deals which are under consideration during the negotiation. The deals in the negotiation set have an additional property that they are *pareto optimal*. A deal $D = (P_A, P_B)$ is pareto optimal if there is no other deal $D' = (P'_A, P'_B)$ for which $DealUtility_A(D') > DealUtility_A(D)$ and $DealUtility_B(D') > DealUtility_B(D)$. Note that $D'$ is preferred over deal $D$ by *both* agents. Since both agents prefer $D'$ to $D$, the deal $D$ is not pareto optimal and can be safely left out of the negotiation process. Restricting the $NS$ to pareto optimal deals ensures that we need not be concerned at the end of the negotiation that there was another deal preferred by both agents that was overlooked. The $NS$ also contains the *conflict deal*, the deal that will be implemented if no agreement is reached. This is usually the set of plans that the agents were operating under before the negotiation began but this need not be the case (in fact, we will see that this is *not* an acceptable option in the realm of air traffic conflicts).

The protocol begins (at $t = 0$) with each agent $i$ proposing the deal from the $NS$ that maximizes the deal utility function for that agent. The proposals made by the agents are simultaneous on each step. Once the proposals for a step have been made, the agents have several options.

First, an agent may accept the deal proposed by the other agent. Let $D_A$ be the deal proposed by agent $A$ and let $D_B$ be the deal proposed by agent $B$. Agent $B$ would choose to accept the deal offered by $A$ if $DealUtility_B(D_A) \geq DealUtility_B(D_B)$. This inequality holds if agent $A$ offers a deal that is better for $B$ than the deal that $B$ suggested. If both agents would be willing to accept the deal proposed by the other, one of the two deals is chosen at random. The negotiation process ends when one of the agents accepts a deal.

If neither deal is acceptable to both parties, the negotiation continues. Each agent then decides whether to concede or stick to its last offer. This decision is made using a calculation of risk based on the utility values of the various deals that are under consideration. Intuitively, risk is defined as the ratio of how much utility is lost by accepting the offer of the other agent to the amount of utility that is lost by causing a conflict.

Formally, this is defined for agent $A$ as

$$Risk_A = \frac{DealUtility_A(D_A) - DealUtility_A(D_B)}{DealUtility_A(D_A) - DealUtility_A(D_{Conflict})}$$

and similarly for agent $B$. This relationship is derived in 12.

If $Risk_A > Risk_B$ then agent $B$ should concede on this step. Both agents calculate both risk values and decide accordingly whether to concede or hold. It is important to note here that this definition of risk requires that the agents are aware of each other's utility values for the various deals that are under consideration. This may lead to some concern that an agent could manipulate the negotiation system by misrepresenting its utility values, but it has been shown by Rosenschein and Zlotkin that no advantage can be gained in this manner.[11]

If an agent is to concede, it must determine which new deal from $NS$ it should propose. The appropriate strategy is to make the smallest concession that will force the other agent to concede on the next step.

The strategy of beginning the negotiation by proposing the deal of minimal utility to the other agent and proceeding with minimum sufficient concessions when $Risk_{self} < Risk_{other}$ is known as the Zeuthen Strategy. The Extended Zeuthen Strategy is the Zeuthen Strategy with some additional logic to take care of a special case that can arise at the end of the negotiation.[12]

It has been shown that the Extended Zeuthen Strategy will result in the agents selecting the deal of highest product of utilities from the negotiation set. It has also been shown that this strategy is in equilibrium, meaning that if agents $A$ and $B$ are negotiating, if agent $A$ uses this strategy then agent $B$ also prefers this strategy to any other.[12]

The fact that this protocol has an equilibrium strategy is of key importance. When a protocol has such a strategy, it is not possible for the agents to gain an unfair advantage by seeking alternate strategies. If any agent uses the equilibrium strategy, then other agents obtain maximal payoff by using the same strategy. All agents can safely use the equilibrium strategy without concern that another agent can abuse the system to obtain a better result for itself at the expense of the others.

## III. Applying the MCP to Air Traffic Conflict Resolution

We have seen that the Monotonic Concession Protocol captures the basic ideas of a bargaining process and involves relatively simple computation. The use of this protocol results in the selection of the deal from $NS$ which maximizes the system utility. Additionally, and most importantly, it has been shown that the protocol cannot be manipulated by one of the agents to gain an unfair advantage by providing false information or by changing negotiation strategies.

In order to apply the MCP to the air traffic conflict resolution problem, many issues must be resolved. These include details pertaining to utility functions for aircraft, conflict detection, negotiation set generation, and others. We have developed a simple model which addresses these concerns.

### A. System Overview

Figure 1 provides an overview of the CD&R process described in the following sections. The process begins with agents constantly monitoring the traffic situation in their areas and uses nominal state propagation to check for impending conflicts. The agents look 20 minutes into the future when predicting conflicts. If the agent determines that a conflict is going to occur, the agent initiates negotiation with the aircraft in question.

The agent initiates negotiation by sending a message to the other aircraft that an impending conflict has been detected. The agent then uses a prescribed procedure to generate a number of possible alternate trajectories. Each of these trajectories is then evaluated using the flight plan cost function of the agent to determine a cost, which is then compared to the cost of the fallback trajectory to determine a utility score.

Each alternate trajectory and its associated utility score is then sent to the aircraft with which conflict is impending. At the same time, the agent receives the other aircraft's possible trajectory alternatives and utility functions. Each of these trajectories is paired with each of the agent's own trajectories to produce 36 potential deals.

It is possible that some or all of these deals will not be conflict free. Therefore, each deal is checked, using a nominal state propagation method, to ensure that no conflict will arise. Any deal that is not conflict free is then rejected. Once this process is complete, the negotiation set has been generated.

After the negotiation set is finalized, the agents execute the monotonic concession protocol to select one of the deals from the negotiation set. If no deal can be agreed upon, the conflict deal is selected.

Finally, the agents present the selected flight plan to the pilot for approval and implementation. In the simulations used for this paper, the agents played the role of pilot, so the selected flight plan was simply executed. In real life, the pilots would have veto power and it is possible that the resulting plan would also need to be submitted to an overseeing air traffic controller for approval before implementation.

### B. Conflict Detection

The system implemented uses a simple conflict prediction tool. Nominal state propagation is used to determine the future location of aircraft using current state information. On each agent update cycle, the current position, speed, and heading of each other aircraft in the area are used to look ten minutes into the future. If another aircraft's protected zone will overlap with that of the agent during that time period, then a conflict resolution process will be initiated. The model defines a conflict as any overlap of the protected zones of two aircraft. The protected zone is a cylinder with a 1.5 nautical mile radius and a height of 1000 feet, centered on the aircraft. The model uses all three dimensions of state information.

During the process of state propagation and conflict detection, the times at which the impending conflict would begin and end are stored. The first time and last time of protected zone overlap are used in the process of generating the alternate trajectories that comprise the negotiation set.

### C. Negotiation Set Generation

When the agent detects that a conflict is going to occur with another aircraft, it uses a predefined process to generate six alternate trajectories. The six prescribed deviations are left, right, up, down, speed up and slow down. Each pair of opposite trajectories is generated with a given process. In terms of the taxonomy outlined by Kuchar and Yang, this resolution process allows for climbs, turns, and speed changes, but not for combinations thereof.

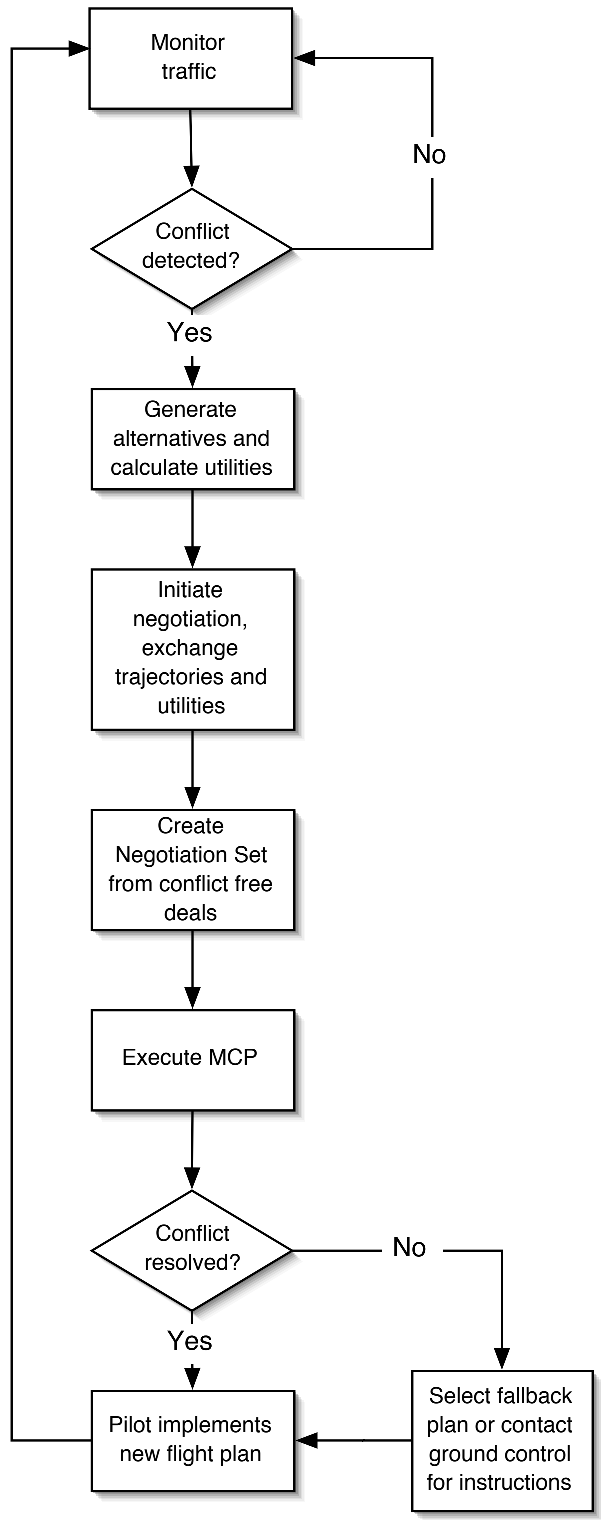


**Figure 1. Overview of the Conflict Resolution Process**

The generation of the left turn, right turn pair (see figure 2) is based upon the predicted times of the beginning and end of the conflict. The agent determines the location it would have occupied at the time the protected zones first would have overlapped. This point is used to generate two temporary waypoints; at right angles to the current heading, the aircraft projects three nautical miles to the left and to the right of the point it would have occupied at the start of the conflict. A similar procedure is used to generate two waypoints using the position the agent aircraft would have occupied at the end of the conflict. Finally, a fifth temporary point is created on the original path several minutes after the conflict would have ended. This point is called the rejoin point and is used by the agent to return to the original path after the conflict evasion maneuver has been completed. These five waypoints, two to the left of the original path, two to the right, and the rejoin point allow two new trajectories to be defined. The left alternative involves an immediate turn to fly towards the first left waypoint, to the second left waypoint, and then rejoining the original path at the rejoin point. Similarly, the right turn alternative is the path from the current position to the first right waypoint, then to the second right waypoint, and finally to the rejoin point.

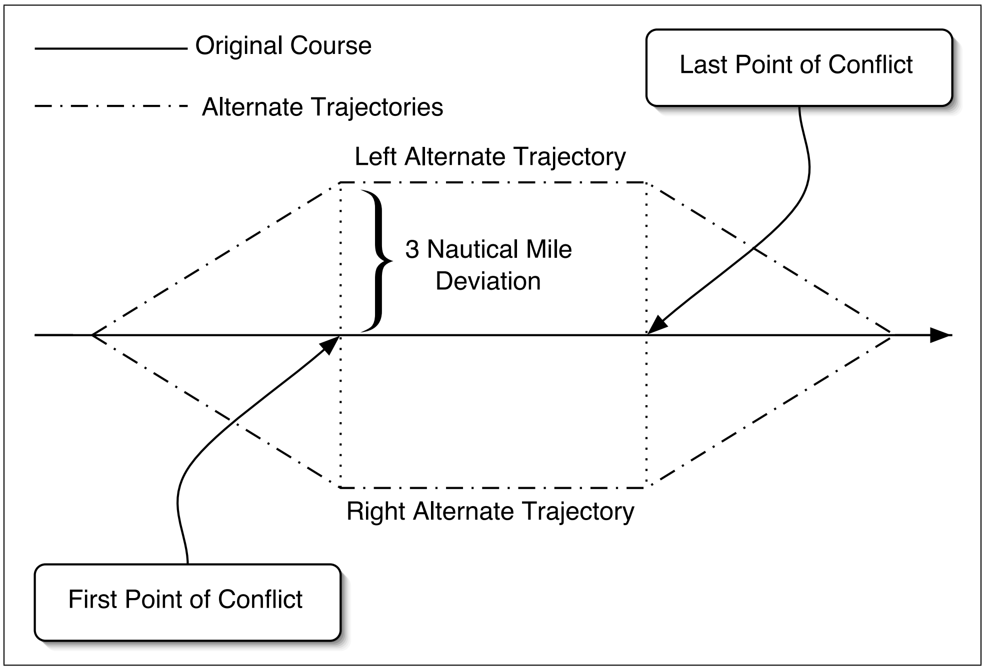


**Figure 2. Generation of left and right alternate trajectories (top view).**

The climb and descend alternatives (figure 3) also make use of the conflict times recorded during the conflict prediction step. The climb alternative is comprised of an immediate climb to an altitude 500 feet above the current altitude. The agent continues to fly its original course at the new altitude until it passes the point at which the conflict is over, at which point it descends back to the original altitude. The descend option is the opposite; the agent descends 500 feet and maintains that altitude until crossing the end of conflict point and climbing back to the original altitude.

Finally, the speed up and slow down options are generated. They are less complicated and simply require the agent to fly the previously intended path at slightly faster or slower speeds. No other course or altitude changes are required. A value of 50 knots was arbitrarily chosen for the amount of speed up and slow down that were used in this model, but this value could easily be altered. It is also likely that in a real world application of this process, this value would be dependent on the class of aircraft in question.

Once all six options have been generated, the agent processes each one using its utility function, as described below. This produces a utility score for each potential conflict resolution. The agent then sends messages to the other aircraft involved in the potential conflict. Each message includes a request to initiate the negotiation process and then all of the flight plan data required to completely describe the six alternate trajectories and their associated utility scores. The message concludes with a statement that all the information has been sent.

Because all of the agents in the simulation are using the same process as described above, the second agent

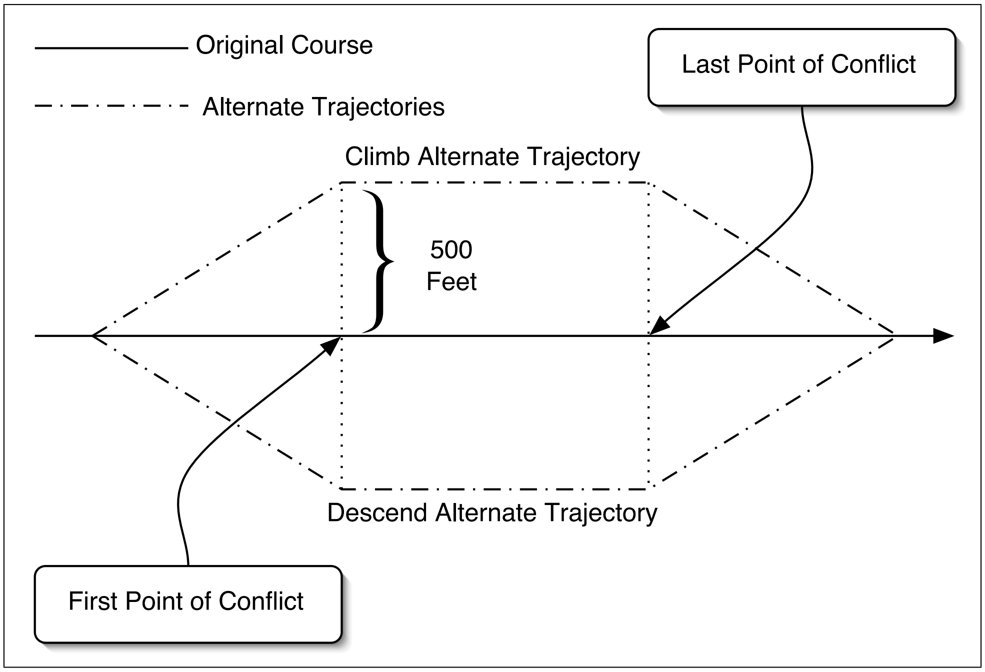


**Figure 3. Generation of climb and descend alternate trajectories (side view).**

involved need not wait for the initiation of negotiation message to begin generating its course alternatives. The process described above can be thought of as occurring simultaneously in both affected agents. Similarly, the following steps, taken by an agent after receiving the trajectory alternatives and their utilities, can also be considered to be occurring simultaneously for all parties.

After receiving the trajectory alternatives, an agent pairs them up with its own generated alternatives. Each received trajectory is paired with each generated trajectory for a total of 36 potential solutions. In the terminology of the MCP, each of these 36 trajectory pairs is a deal, and together they form a preliminary negotiation set.

There is an additional constraint that must be met, however, before the process can go forward. In the deal validation step, each of the 36 potential deals must be checked to determine if it is a viable solution. In order to do this, each deal is analyzed using a process similar to that used for conflict detection. Nominal state propagation is applied to each leg of the intended trajectories to determine if the deal is truly a conflict-free solution. Any deal which contains a future conflict is rejected and discarded from the negotiation set. It should be noted that while this validation occurs twice, once in each agent that is involved in the conflict, the results obtained should be symmetric.

### D. The Conflict Deal

As outlined in the discussion of the MCP in the previous section, a conflict deal must exist in the negotiation set. This deal serves two purposes. First, it is a baseline for the calculation of utility scores. The utility of an option for an agent is defined as the cost of the conflict deal minus the cost of the alternative. Second, the conflict deal provides a fallback that will be used if the negotiation ends in conflict or does not end at all.

In most multi-agent applications of the MCP, the conflict deal defined as the continuation of the agent's pre-negotiation plans. The air traffic domain does not allow for this possibility, however, as both agents following their initial plans will lead to a violation of the safety constraints of the system. Special considerations must be made for the conflict deal in this domain.

In this system, the conflict deal must consist of a conflict-free pair of trajectories available to both agents at the start of the negotiation process. There are several possibilities for the generation of this deal. First, prior to the negotiation process, the aircraft could contact air traffic control and obtain the pair of trajectories that air traffic control would have given to the aircraft in the event that they could not negotiate. This is a poor choice, as it does not allow for the full automation of the process. Second, the pilots of the aircraft in question could manually communicate at the start of the negotiation process and determine a pair of conflict free trajectories that they would be willing to fly and input this into their flight computers. The agreed upon pair could then be used as a conflict deal for the MCP process. This possibility significantly adds to the workload of the pilot, however, and is therefore not preferred. Finally, one of the previously described conflict resolution methods based on worst case analysis could be used by the agents prior to the negotiation process to determine the fallback conflict deal. Each agent would first run this method to determine what course it would fly to avoid the other aircraft. The agents would then exchange these courses to form the conflict deal. Since these courses are created using a worst case analysis, they can safely be used in the event that this initial communication cannot be made and the negotiation cannot take place. This method also preserves the full automation of the process and does not require pilot intervention until the final agreed-upon resolution is presented to the pilot for implementation. This approach is the one used in the system we present here. We have implemented the method discussed by Tomlin, Papas, and Sastry in 4 to use as our baseline worst case analysis system.

### E. Utility Functions

In order to make use of the MCP, each agent must have a utility function. This function is used to encode the agent's preferences for certain flight plans. The power of the cooperative negotiation approach comes from these functions. By encoding preferences for certain flight plan attributes in a utility function, the agents are able to inject their desires into the conflict resolution process in a way that is not possible with most standard systems.

Utility functions in the MCP are built on top of cost functions. The utility of a deal is defined as the reduction in cost of that deal as compared to the conflict deal. The cost functions themselves are where the preferences are actually encoded. There has not been extensive work in the air traffic domain dealing with utility functions; some simple examples were formulated for this work. The important feature of these functions is not their precise formulation, but the fact that any preference can be encoded into them. This allows maximal flexibility for the agents to express their desires for some flight plans over others. Any conceivable function that maps flight plan data to a number can be used.

Encoding flight plan preferences into the MCP utility functions allows the MCP to select the deal that is of greatest value to the two agents. The fact that the MCP takes these agent cost functions into account is what gives it an advantage over non-cooperative methods. A system which does not incorporate agent preferences can never satisfy those preferences. The MCP is able to provide solutions which are more acceptable to all parties due to this feature.

## IV. Experiments and Results

The conflict detection and resolution system was evaluated using randomly generated scenarios. Each scenario consisted of two aircraft whose initial headings and speeds would cause a conflict. Figure 4 provides a visual overview for the scenario generation process.

Random scenarios were created with the following process. First, a conflict point was selected. This point is the location at which the aircraft would sustain a collision without intervention. Random velocities were selected for the aircraft from a nominal range. Headings were also randomly generated for the aircraft, with the restriction that they not be within 30° of one another. Finally, a conflict time was selected. This number is how many minutes after the start of the scenario the conflict will occur, and was drawn from the range 10 to 20.

Using the selected headings, velocities, and time, aircraft starting positions were calculated. The flight plans were completed by creating waypoints on the aircrafts' headings one hour forward from the starting position.

The most important function performed by any conflict detection and resolution system is the maintenance of safe separation between the aircraft involved. This was qualitatively verified by running several hundred trials of these randomly generated scenarios. In these trials, the agents were presented with a scenario, and negotiated a resolution. The resolution was analyzed to verify that no conflict occurred. In all trials conflict was avoided; the protocol, therefore, meets the minimum safety standard.

The second goal of the system is increased efficiency through cooperative conflict resolution. The efficiency

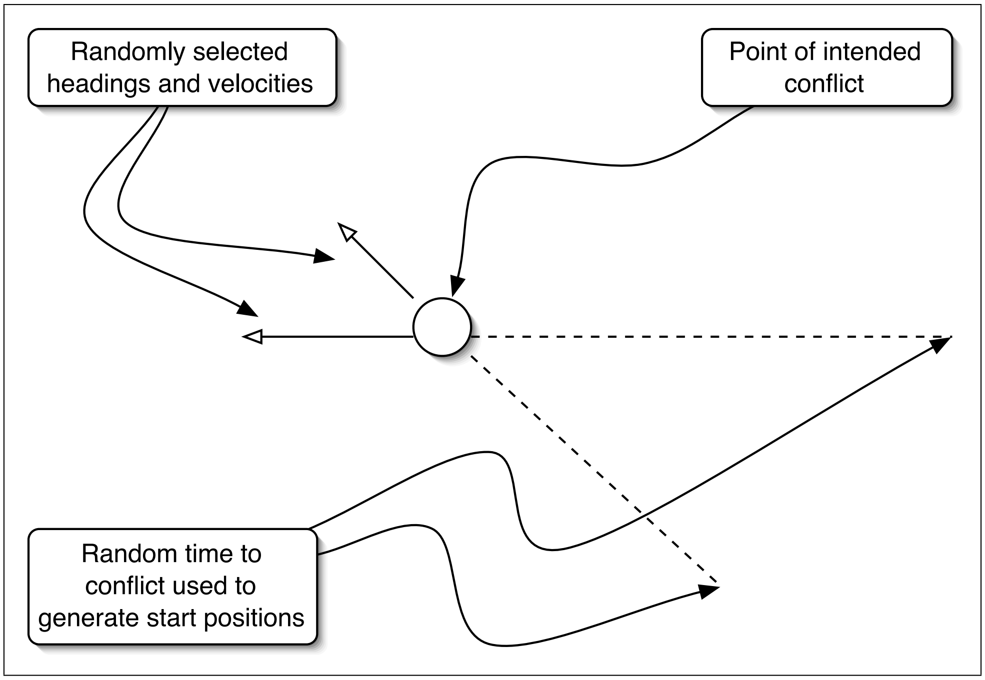


**Figure 4. Random scenario generation overview.**

of the system was evaluated using cost functions as described previously. A number of basic cost parameters were defined and combined to create cost functions for use in these tests.

$$D = \text{total distance traveled}$$
$$\Delta A = \text{total altitude changes during flight plan}$$
$$\Delta H = \text{total heading changes during flight plan}$$
$$Cost = D \quad (1)$$
$$Cost = D + \Delta A \quad (2)$$
$$Cost = D + \Delta H \quad (3)$$
$$Cost = D + \Delta A + \Delta H \quad (4)$$

Equation 1 is the simplest cost function tested, in which the cost of the flight plan is equal to the distance in nautical miles that is flown. Equations 2, 3, and 4 add penalties for altitude and heading changes. The altitude penalty is equal to the gross altitude change over entire the flight plan. For example, a plan which includes a 500 foot climb and a 1000 foot descent will incur an altitude penalty of 1500. The heading change penalty is defined similarly. These penalties are intended to capture the desire to minimize pilot workload and maximize passenger comfort.

Test 1 (see table 1) shows the average results from 100 trials in which both agents use cost function 1. As expected, when both agents use the same utility function their results are very similar. Agent 1 averaged a 6.1% improvement and agent 2 averaged 9.3% improvement when comparing the negotiated deals to the conflict deal.

**Table 1. Summary Results for Evaluation #1**

| Agent | Conflict Cost | Negotiated Cost | Average Improvement | Std. Deviation |
|---|---|---|---|---|
| 1 | 156.31 | 146.00 | 6.1% | 5.4% |
| 2 | 160.04 | 144.65 | 9.3% | 5.6% |

Test 2 (see table 2) shows the results of 100 trials in which agent 1 continued to use cost function 1, but agent 2 used cost function 4 which includes all three penalties. In this test agent 1 showed an average improvement of 8.5%, which is not statistically different from the result of test 1. Agent 2, however, showed a much greater improvement. This demonstrates that when an agent has strong preferences for certain solutions, the system allows those preferences to be expressed. This is the primary difference between this system and any non-cooperative solution; the negotiation process allows agents to select the plan that suits them the best.

**Table 2. Summary Results for Evaluation #2**

| Agent | Conflict Cost | Negotiated Cost | Average Improvement | Std. Deviation |
|---|---|---|---|---|
| 1 | 145.03 | 143.76 | 8.5% | 6.4% |
| 2 | 218.01 | 158.09 | 27.8% | 20.5% |

## V. Conclusion

As the National Airspace System becomes more crowded, a transition to Free Flight becomes more and more important. Free Flight requires an effective system for resolving airspace conflicts that does not place undue load on the air traffic controllers.

We proposed and implemented a simple conflict resolution system based on multi-agent cooperation and negotiation. The system was tested to assess both its safety and efficiency. In all test cases, the conflict was successfully avoided, demonstrating that minimum safety requirements can be met by such a system.

Additionally, the tests have confirmed the intuition that when agents are allowed to use cost functions to express preferences, they can reach resolutions that are of higher utility than those provided by non-cooperative methods.

We have demonstrated, that by considering the NAS as a system of individual agents with differing goals and utility functions, that well known cooperative negotiation techniques can be brought to bear on the air traffic conflict problem. These cooperative techniques will allow a group of aircraft to efficiently and safely resolve their conflicts without constant interaction with ground control.

## References

[1] Radio Technical Commission for Aeronautics, "Final Report of RTCA Task Force 3, Free Flight Implementation, RTCA/DO-242," Tech. rep., Washington, D.C., October 1995.

[2] Kuchar, J. K. and Yang, L. C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000, pp. 179–189.

[3] Tomlin, C., Pappas, G., Lygeros, J., Godbole, D., and Sastry, S., "A next generation architecture for air traffic management systems," Technical Report UCB/ERL M97/7, University of California, Berkeley, 1997.

[4] Tomlin, C., Pappas, G., and Sastry, S., "Noncooperative Conflict Resolution," *Proc. IEEE Int. Conf. on Decision and Control*, San Diego, California, December 1997.

[5] Tomlin, C., Pappas, G., and Sastry, S., "Conflict resolution for air traffic management : A study in muti-agent hybrid systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509–521.

[6] Wangermann, J. P. and Stengel, R. F., "Optimization and Coordination of Multiagent Systems Using Principled Negotiation," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 1, January-February 1999, pp. 43–50.

[7] Menon, P., Sweriduk, G., and Sridhar, B., "Optimal Strategies for Free-Flight Air Traffic Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, March-April 1999, pp. 202–211.

[8] Rong, J., Bokadia, S., Shandy, S., and Valasek, J., "Hierarchical Agent Based System for General Aviation CD&R Under Free Flight," *AIAA Guidance, Navigation, and Control Conference*, Monterey, California, 5-8 August 2002.

[9] Rong, J., Ding, Y., Valasek, J., and Painter, J., "Intelligent System Design with Fixed-Base Simulation Validation for General Aviation," *IEEE International Symposium on Intelligent Control*, Houston, TX, 5-8 October 2003.

[10] Shandy, S. and Valasek, J., "Intelligent Agent for Aircracft Collision Avoidance," *AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 6-9 August 2001.

[11] Rosenschein, J. S. and Zlotkin, G., *Rules of Encounter*, The MIT Press, Cambridge, Massachusetts, 2nd ed., 1994.

[12] Zlotkin, G. and Rosenschein, J. S., "Negotiation and Task Sharing Among Autonomous Agents in Cooperative Domains," *IJCAI*, 1989, pp. 912–917.