ABSTRACT

NEXT GENERATION ACAS X SIMULATION

Sanberk DAVER
Master of Science, Computer Engineering Department
Supervisor: Assist. Prof. Dr. Adnan ÖZSOY
January 2021, 60 pages

In the near future, when air traffic will gain a new dimension, many light weight platforms like unmanned aerial vehicles and more advanced technologies for existing aircraft will appear in the sky in addition to the current aircraft traffic. Efforts to keep air traffic safe and support growth under control will accelerate and meet more demand than ever before. Collision Avoidance System will definitely take place in the works to be carried out within this scope.

In this context, this study tries to improve the operational efficiency and searching for a logical balance between operational efficiency and safety of flight criteria. This work analyzes encounter models of air platforms, generation of model data, utility functions, the effect of pilot response delays and proposes alternative maneuver selection costs.

Some safety critical avionic systems that are being developed today need to work on real-time platforms. Therefore,they are developed with real time C code to meet the limited hardware and high performance needs. In that regard, study for a simulation that can solve a partially observable markov decision process in order to verify the collision avoidance system is needed.

Motivation of this simulation work is to understand failure conditions given in a probabilistic model of the environment.

In this study, encounter models of air platforms, generation of model data, utility functions, maneuver selection costs and the effect of pilot response delays are analyzed. In addition, the generated look up table for the collision avoidance system is analyzed using simulated sensor data.

Keywords: Aircraft Collision Avoidance System, POMDP, Simulation, Sequential Decision Making, Pilot Behavior, Probabilistic Verification, Probabilistic Data Synthesis

ÖZET

YENİ NESİL ACAS X SİMÜLASYONU

Sanberk DAVER
Yüksek Lisans, Bilgisayar Mühendisliği
Danışman: Dr. Ögretim Üyesi Adnan ÖZSOY
Ocak 2021, 60 sayfa

Yakın gelecekte çok sayıda insansız hava aracı ve daha ileri teknoloji hava taşıtları gökyüzünde kendilerini gösterecektir. Mevcut uçakların oluşturduğu yoğunluk ile birlikte hava trafiği yeni bir boyut kazanacaktır. Hava trafiğinin güvenliğini sağlamak ve büyümesini kontrol altında tutmak için harcanan çabalara her zamankinden daha fazla ihtiyaç duyulacaktır. Bu kapsamda yapılacak çalışmalarda çarpışma önleme sistemleri şüphesiz çok önemli bir yer alacaktır.

Bu bağlamda, bu çalışma operasyonel verimliliği artırmaya ve operasyonel verimlilik ile uçuş emniyeti arasında mantıksal bir denge aramaktadır. Bu çalışma, hava platformlarının karşılaşma modellerini, model verilerinin üretilmesini, fayda fonksiyonlarını, pilot müdahale gecikmelerinin etkisini analiz etmekte ve alternatif manevra seçim maliyetleri önermektedir.

Bugün geliştirilmekte olan bazı güvenlik açısından kritik aviyonik sistemlerin gerçek zamanlı platformlarda çalışması gerekmektedir. Bu nedenle, sınırlı donanım ve yüksek performans ihtiyaçlarını karşılamak için gerçek zamanlı C kodu ile geliştirilirler. Bu bağlamda, çok fazla simülasyon çalışması gerektiren çarpışma önleme sistemlerini doğrulamak için kısmi gözlemlenebilir markov karar süreci çözebilen bir simülasyon çalışması gerekliliği doğmuştur.

Bu simülasyon çalışmasının amacı, ortamı ifade eden olasılıksal model üzerindeki çarpışma olaylarını tespit etmek ve anlamlandırmaktır.

Bu çalışmada, hava platformlarının karşılaşma modelleri, model verilerinin oluşturulması, fayda fonksiyonları, manevra seçim maliyetleri ve pilot müdahale gecikmelerinin çarpışma önleme sistemi üzerine etkisi analiz edilmiştir. Ek olarak, çarpışma önleme sistemi için oluşturulan model tablosu, simüle edilmiş sensör verileri kullanılarak analiz edilir.

Anahtar Kelimeler: Uçak Çarpışma Önleme Sistemi, POMDP, Simülasyon, Sıralı Karar Verme, Pilot Davranışı, Olasılıksal Doğrulama, Olasılıksal Veri Sentezi

CONTENTS

Page

FIGURES

Page

TABLES

1. INTRODUCTION

In order to develop and use a safety critical system, manufacturers are required to get certification for the software from authorities. The software development and verification processes are audited carefully by the authorities for compliance to DO-178 [3] standards. In this context, various simulations and analysis are useful to verify correctness of newly developed collision avoidance system.

ACAS X is the next generation aircraft collision avoidance system and it's being standardized for use on every large aircraft in the world [4]. Interest in aircraft collision avoidance systems arose during the 1950s. There was a mid-air collision over the Grand Canyon involving United Airlines and TWA. That collision led to the establishment of the *Federal Aviation Administration* (FAA) and with the establishment of FAA they put a number of procedures and introduced *Air Traffic Control* (ATC) [5].

Although ATC significantly improved the safety of the airspace, mid-air collisions continued to happen where more than hundred died [4]. MIT Lincoln Laboratory were tasked with developing the system known as TCAS or the *Traffic alert and Collision Avoidance System* currently mandated worldwide [6].

Basically, TCAS includes a system called Surveillance to detect air traffic around itself. This system transmits the traffic information it detects onto the advisory logic, which constitutes the intelligence component. Advisory logic decides what type of alert will be generated on the vertical axis. The decision can be climb, descend, or hold your position. For the TCAS the level of safety is fewer than; one collision per billion flight hours. Although this is a safety critical system, aircraft whose maximum takeoff weights are less than 5700 kg are not required to carry a TCAS [6]. TCAS is not designed for the maneuverability of small aircraft. Even though it is a useful equipment, it is expensive for small platforms. The smaller aircraft that do not carry TCAS have collisions about every month. On the other hand, there has never been a plane crash in the United States carrying TCAS since TCAS became mandatory.

Some safety critical avionic systems that are being developed today need to work on real-time platforms. Therefore, they are developed with real time C code to meet the limited hardware and high performance needs. In that regard, study for a simulation that can solve a partially observable Markov Decision process in order to verify the collision avoidance system is needed.

Motivation of this simulation work is to understand failure conditions given in a probabilistic model of the environment. Normally, in machine learning contexts, problem can be specified with a data set. For the collision avoidance system, however, there is a feedback loop and due to this, environment is able to update itself continuously.

Although there are better programming language options like Julia [7] in terms of readability, our study was conducted in C/C++. The method by which the collision avoidance system is dealt with, Partially Observable Markov Decision Process (POMDP) requires a lot of calculation and simulation work due to the size of the space.

Markov Decision processes are used for building a framework that helps us make selections in an non-deterministic environment. In case of an environment with some degree of uncertainty but does not have complete randomness, it is possible to build a framework to make selections. Markov Decision Processes express an approach that models the transition from one state to another with the action to be chosen. These transitions are calculated with various probabilities and supported by rewards [8].

There will be many studies concerning to make collision avoidance systems better. Since there are trade offs and a lot of parameters to decide and test, finding the best reasonable model or tuning the reward values is a very difficult task.

In this study, we deeply analyze encounter models of air platforms, generation of model data, utility functions, maneuver selection costs, and the effect of pilot response delays. In addition, the generated look up table for the collision avoidance system is populated using simulated sensor data. Finally, we propose alternative maneuver selection costs that give operational advantage and higher level of safety.

Contribution of this work can be listed as follows;

- Encounter models of air platforms are inspected and analyzed.
- An environment that can both generate and test model data is implemented.
- Utility functions explained and analyzed.
- Different sets of maneuver selection costs are tested and discussed.
- Effect of pilot response delays are analyzed.

- Generated look up table for the collision avoidance system is analyzed using simulated sensor data.
- Alternative reward values for maneuver selection costs are presented.
- Invalid transitions between generated resolution advisories are proposed and analyzed.

The rest of this thesis is organized as follows. Chapter 2. presents the existing literature in scope of collision avoidance simulations and discusses our framework. Chapter 3. discusses the collision avoidance problem about the reason why we need a system to prevent mid-air collisions and the first system that developed as a solution. In this section formulation of collision avoidance problem is taken at hand. At that part, in order to be able to work on the state space, parameters of Markov Decision Process are presented. Resolution advisories as a vital part of Collision Avoidance Systems are introduced and possible invalid transitions between them are denoted. Driving forces of dynamic model and programming methodologies are presented. Chapter 4. discusses the followed reward function outline, how maneuvers are interpreted, proposed transition penalties, how maneuvers are encouraged, and the trade-off between operational efficiency and level of safety. Chapter 5. analyzes simulation results in terms of both safety and operational suitability. Chapter 6. summarizes our conclusions.

2. RELATED WORK

Technologies that developed while needs are changing tend to be more complicated and more functional. However, when it comes to safety critical systems such as collision avoidance systems, validation and verification phases that must be carried out to ensure that the developing technologies are safe and suitable require a lot of effort and calculation that cannot be done with traditional simulation methods.

In order to define our research frame, we started our systematic literature review by setting our inclusion and exclusion criterias. Briefly, inclusion criteria indicates the characteristics of proposed topic. So that, a paper which takes place in our systematic literature review must have a relation with our prospective subject. On the other hand, exclusion criteria defines the unsuitable features or characteristics that should not take place in our work. In this manner, both inclusion and exclusion criterias are used in a positive way. In case of having a paper that met with our inclusion criteria, we put that paper in our paper pool. Contrary to this, in case of having a paper that met with our exclusion criteria, we take away that paper from the paper pool.

Articles searched by first search sentence and refined by *software* topic on Web of Knowledge [9]. In scope of that, by reading titles and abstracts we tried to find the meaningful articles. After that point, we have selected almost one hundred paper for the first phase.

We have detected three fundamental problems such as the compression of the encounter model data, mathematical verification of ACAS, and simulation studies on Markov Decision Process. Within the scope of this study, the compression of the encounter model data and mathematical verification of ACAS were excluded and the simulation studies were examined.

While conducting our literature review, we tried to focus on articles written after 2012. Related studies conducted in this context are presented below.

Lygeros and Lynch [10] worked on a simplified collision avoidance system for the verification of TCAS, the predecessor to ACAS X, which must be tested before being deployed to the aircraft platforms. Lygeros et al. [11] made some assumptions. Under these assumptions, they worked on high-level modeling and analysis methods to prove the usability of TCAS with an Hybrid I/O Automata. Our work for ACAS X includes aircraft encounter scenarios that are too many to be controlled by similar assumptions.

Essen and Giannakopoulou [12] examined the existing probabilistic analysis tools and developed a probabilistic model checking tool that targets ACAS X and similar systems due to the inadequacy of examined tools. Besides their formal probabilistic verification study, probabilistic data synthesis was also performed. Moreover, they implemented the ACAS X algorithm to enable effective verification. It is similar to our work in these aspects. It differs from black-box test methods in that it can access the internal parameters of the collision avoidance system and provides a comprehensive and effective environment for verification. On the other hand, Java language was preferred as the modeling language. In our study, C programming language was used. In this context, it will be possible to perform analysis and especially data synthesis faster.

In addition, with the model we offer, it has been made possible to test a wide variety of dynamics ranging from policy table used in data synthesis to pilot response. In this context, analysis results which enable probabilistic data synthesis with higher benefits are presented for ACAS X developers.

Gardner, Genin, McDowell, Rouff, Saksena, and Schmidt [13] developed a probabilistic model checking system to analyze behavior of ACAS X. Although they stated that differences in pilot behavior were analyzed in their study, the effect of pilot behavior was not clearly presented. At this point, our study reveals clear results on the impact of pilot behavior. Unlike this study, *Maintain*, one of the advisory types, was not included in our study because it is context dependent. On the other hand, our study includes *Strong Descend* and *Strong Climb* advisories. In addition, the times expressed as the average pilot response time were examined in a detailed way in our study, by taking into account of the reasons behind them.

Jeannin, Ghorbal, Kouskoulas, Schmidt, Gardner, Platzer, Mitsch, and Zawadzki [14, 15] used a hybrid system that could be used for validation of ACAS X under a set of assumptions and made graphical analysis. Although it is a good approach to examine both the discrete and continuous dynamics of the hybrid system they have worked on, it is not possible to do that study in a reasonable amount of time without simplifying the state space of ACAS X. Therefore, simulation studies are still needed. Our work is also beneficial for the detection of aircraft encounter scenarios that inevitably end up with collisions. At this point, unlike Jeannin et al. [14, 15] study, our study also explains the scenarios that inevitably end up in unsafe regions.

Lee, Kochenderfer, Mengshoel, Brat and Owen [16], on the other hand, carried out a study that can deal with the large space of ACAS in their study. In this study, scenarios end up with a collision and scenarios close to end up with collision are found with Monte Carlo Tree Search. This approach can be used with white-box simulators, as well as with black-box simulators, which provide much more comprehensive simulation opportunities. They worked with a binary ACAS X prototype that they got from FAA. Aircraft encounters and pilot behaviors were modeled with Lincoln Laboratory Correlated Aircraft Encounter Model (LLCEM) [17]. Consequently, they were able to examine more complex scenarios than in the PMC studies. For the aircraft encounter scenarios that contains two planes, they mentioned the problems experienced due to the *Crossing Time*, *High Turn Rates* and *Initially Moving Against RA* reasons. In the study we presented, the effects of these reasons on the model were examined.

Egorov, Sunberg, Balaban, Wheeler, Gupta, and Kochenderfer [18] worked on an open source framework that can solve Partially Observable Markov Decision Process. Since it is a study for both education and research purposes, they preferred the Julia language for easy and fast prototyping. In our study, C programming language is preferred. Similar frameworks made with C/C++ [19–21] are expected to have higher performance. On the other hand, there are no studies directly for ACAS X in these studies.

Unlike Lygeros and Lynch [10]; and Lygeros et al. [11] too many encounter scenarios that cannot be taken under control by certain assumptions have been tested. Unlike Essen and Giannakopoulou [12] all code implemented in C/C++ which is going to provide faster data synthesis. Additionally, with the model we offer, it has been made possible to test a wide variety of dynamics ranging from policy table used in data synthesis to pilot response. Unlike Gardner, Genin, McDowell, Rouff, Saksena, and Schmidt [13] our study reveals clear results on the impact of pilot behavior. Moreover, *Maintain*, one of the advisory types, was not included in our study because it is context dependent. On the other hand, our study includes *Strong Descend* and *Strong Climb* advisories. Unlike Jeannin, Ghorbal, Kouskoulas, Schmidt, Gardner, Platzer, Mitsch, and Zawadzki [14, 15] it is possible to validate ACAS X in a reasonable amount of time without simplifying the state space of ACAS X. Our work is also beneficial for the detection of aircraft encounter scenarios that inevitably end up with collisions. Lee, Kochenderfer, Mengshoel, Brat and Owen [16] mentioned the problems experienced due to the *Crossing Time*, *High Turn Rates* and *Initially Moving Against RA* reasons. In the study we presented, the effects of these reasons on the model were examined.

Unlike Egorov, Sunberg, Balaban, Wheeler, Gupta, and Kochenderfer [18] C programming language is preferred. And finally unlike similar frameworks made with C/C++ [19–21] this study directly tailored for ACAS X.

3. BACKGROUND

The first recorded plane crash in 1956 caused 128 deaths. The Federal Aviation Administration, which was established as a result of this collision, has made a series of regulations to make air transportation safer, but it has not been able to completely prevent aircraft accidents.

Thereupon MIT Lincoln Laboratory were tasked with developing a system known as *Traffic alert and Collision Avoidance System* currently mandated worldwide [6].

In the following subsections, TCAS as the ancestor of ACAS X, the limitations of TCAS that led to the development of ACAS X, and finally ACAS X as a modern approach to collision avoidance systems are going to be introduced.

3.1. Traffic Alert and Collision Avoidance System

Given the problem, the position and state of the aircraft in danger of collision had to be measurable in order to solve this problem, as in every engineering problem. However, the methods considered in the early studies were not very successful because of inconsistent sensor data or increasing air traffic. After ten years with advancing technology and effort, the TCAS system, which is now mandatory for all passenger aircraft, was developed. Roughly TCAS works as follows: it has a *surveillance system* that detects and tracks the intruders and then it passes the sensor measurements onto an *advisory logic*. The advisory logic tells the system exactly when to produce an alert, and what type of alert to provide to the pilot in terms of the vertical rate.

These types of alerts, which notify avoidance warnings on the vertical axis, consist of traffic advisory, which warns the pilot to be prepared for a possible avoidance maneuver, and resolution advisory to perform an avoidance maneuver.

TCAS can update its avoidance maneuvers according to changing conditions. These updates can reverse, strengthen or weaken the current maneuver. List of possible maneuvers are given in Section 3.4..

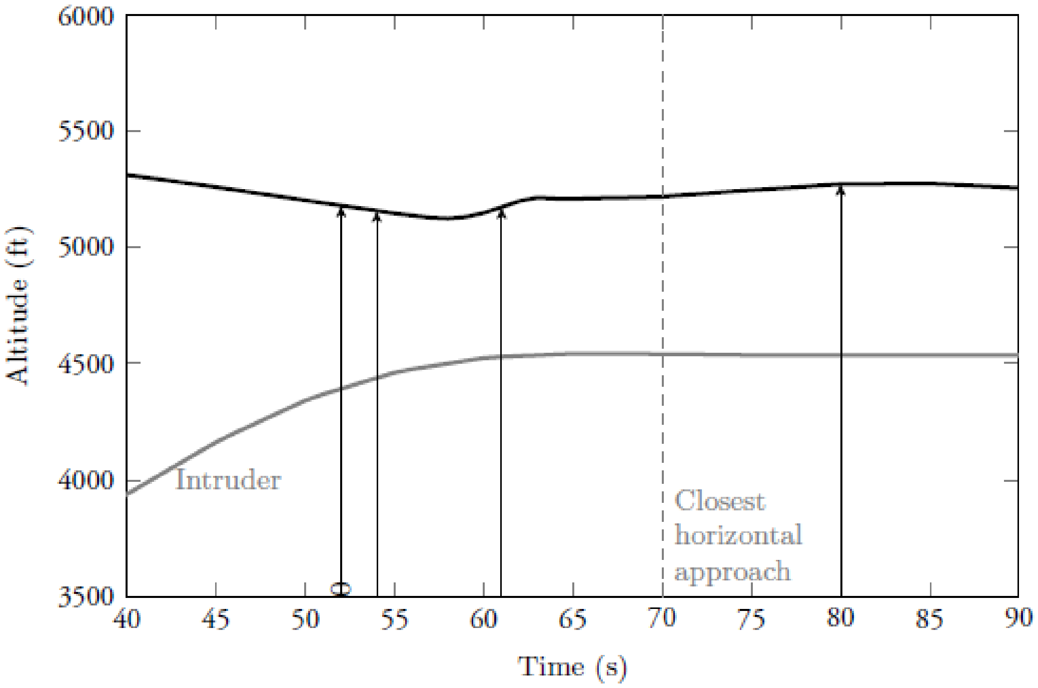
It follows established rules to determine exactly which alert will be generated and when the alert will be generated. Basically, it decides whether a warning is necessary or not by considering the values of time to closest approach and miss distance. If it is determined that the maneuver is necessary, an avoidance maneuver to be applied with an acceleration of 0.25 G will be chosen. The up or down direction of the maneuver is chosen to go the furthest from the threatening aircraft [6]. At this point unit ”G” should not be confused with gravitational constant. Unit G is out of scope of the International System of Units (SI) and it is used to measure amount of acceleration.

3.2. Limitations of Existing System

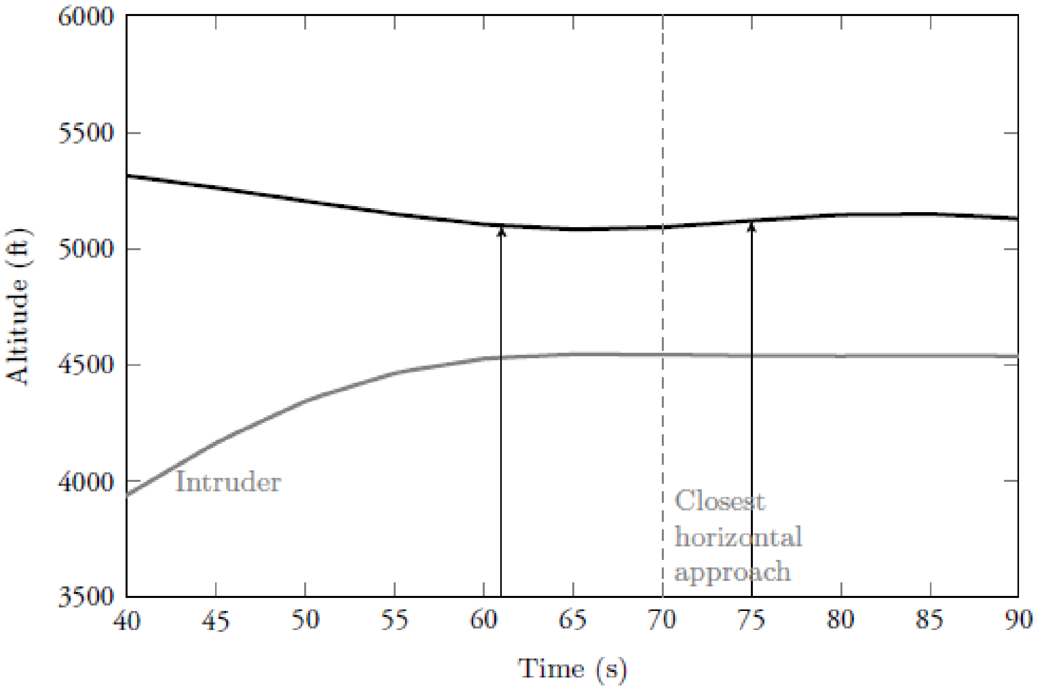
Over the years, there have been developments that could help keep increasing air traffic under control. However, it did not adapt very well to the evolving conditions. Dense air traffic was generating excessive amounts of warning. Moreover, since TCAS was not suitable for all aircraft of various sizes and capabilities, it could not be used in unmanned aerial vehicles and light weight platforms used in civil aviation, which we often hear nowadays. Since required effort to update TCAS is too much, it is considered to be more meaningful to develop a new system than to spend effort on updating the existing TCAS for these lightweight platforms. The new generation collision avoidance system is named as *Airborne Collision Avoidance System X* as known as ACAS X. ACAS X is pronunced as *ay.cas, eks*. ACAS X is supported by Federal Aviation Administration and it uses new technologies such as dynamic programming.

3.3. Airborne Collision Avoidance System X

In scope of Airborne Collision Avoidance System X studies, performance and memory requirements that TCAS could not keep up with have been enhanced by using dynamic programming. The solution to the collision avoidance system is applied by taking the problem as *Partially Observable Markov Decision Process* (POMDP). Current simulation studies indicate that operational performance and level of safety have been improved according to TCAS. This new system, which is also compatible with unmanned aerial vehicles and light platforms, is called *Airborne Collision Avoidance System X*. Unlike TCAS, ACAS X evaluates the space in which the plane is located by looking at the look-up table instead of making a decision according to the rule set determined by the experts. ACAS X provides better performance while preventing generation of too many alerts.



(a) TCAS.



(b) ACAS X.

FIGURE 3.1: TCAS vs. ACAS X. *Courtesy of* M. J. Kochenderfer, Decision Making Under Uncertainty [1]

Kochenderfer declared the comparison of operational characteristics of TCAS and ACAS X in the following Figure 3.1 on a level off example in his work named Decision making under uncertainty [1].

Figure 3.1 shows the operational difference between ACAS X and its predecessor, TCAS, in practice. As can be seen in part *a* (above), when the time axis is in the range of 50-55 seconds, where the intruder and own aircraft approach each other, TCAS has been over protective by producing a corrective advisory. TCAS recommended crossing for own aircraft which is descending. After a short while, TCAS changed the maneuver suggestion to climb. When sufficient vertical separation was achieved, it reduced the maneuver intensity and gave a clear of conflict message at the 80th second respectively. In part *b* (below), it can be observed that ACAS X awaits for a little bit longer to recommend an advisory. While time axis is in between 60-65 seconds, ACAS X has produced a preventive advisory to prevent descending of own aircraft. When it comes to the 75th second, it gave a clear of conflict message. By considering the altitude differences, it is noticed that the ACAS X has lower vertical separation, which allows it to manage denser air traffic.

3.4. Collision Avoidance Problem Formulation

As mentioned in the previous section, it is possible to solve the problem handled as POMDP with different methods. Basically, we can make observations from an environment, with those observations our beliefs are updated, and these updated beliefs are passed to a policy. At the end of this cycle, the policy performs an action which is going to change the environment. Since the environment is evolving, it is not possible to use a static data set. In other words, it has to react to the things that your system is doing. In order to do that, createDT algorithm given in Algorithm 2 calls two major functions. First it calls calculateUValues function to update beliefs and perform policy operations. Then it calls saveLUTFile function to generate a lookup table that is produced as a binary file. We will mention about resolution advisories which is a critical part of collision avoidance systems and models that should be known for digitizing the problem. In order to understand this problem addressed as POMDP, fundamentals of the Markov Decision process should be understood.

Algorithm 1 createDT

```
Call calculateUValues() function.
Call saveLUTFile() function for saving state-action values in a lookup table that are produced through dynamic programming.
```

Markov Decision Process

Markov Decision Processes can be applied to many real world problems. In terms of the collision avoidance system, Markov Decision Processes consist of states and actions correspond to relative situations, positions and maneuvers. Actions might provide transitions between the states that make up the MDP space. In the collision avoidance system, these actions might be climb, descend or other variations.

At this point, Markov Decision Process was tried to be visualized. Different states named as A, B and C and they can be observed from the Figure 3.2. States are used for making sense of the environment we are in. In order to recommend maneuvers, Collision Avoidance System needs information. For example the altitude, speed, and velocities of both own aircraft and the other aircraft around us are taken into account. Required information is going to be introduced in the following subsections. As can be understood from these, the collision avoidance maneuver to be produced may differ according to the environment and conditions we are in. Possible actions we can choose according to the environment we are in are shown as one and two. These type of actions are generated by collision avoidance systems and called as Resolution Advisories. These actions are going to be introduced in the upcoming subsection.

The MDP space to be expressed may contain more states, as well as more actions for each state. The figure is expressed with 3 states and 2 possible actions for each state. The floating point numbers next to the arrows indicate that state transition will occur as a result of the selected action with that probability distribution. For example, if we suppose we are in State C, in case of performing action one, state transition is going to end up in State B with probability 0.7. On the other hand, there is a probability of remaining at State C with a probability of 0.3.

The numbers shown with plus and minus symbols represent the reward or penalty points to be taken as a result of the preferred action. For example, suppose we are in State A. If we choose action one, 5 reward points will be received as a result of the state transition that

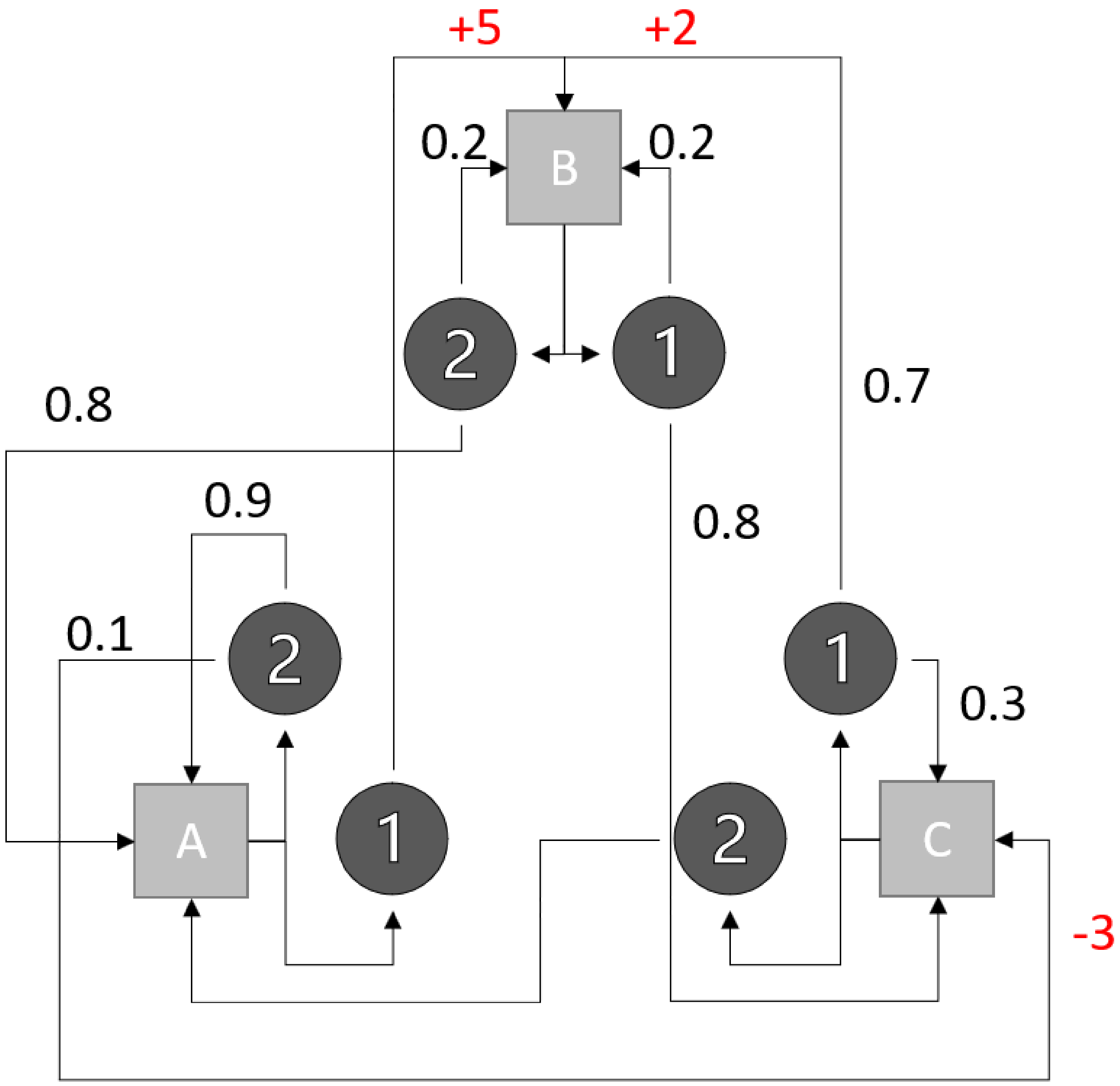


FIGURE 3.2: Markov Decision Process

will take place. But if we choose action two, the state transition that may end up in State C with a probability of 0.1. In case of this state transition, 3 penalty points will be taken. This approach, which includes probability calculation and reward points, forms the decision mechanism in Markov Decision Processes.

In Partially Observable Markov Decision Processes on the other hand, the states named as A, B, and C should be considered as nameless. In this case, although the states cannot be observed directly, it is possible to make a set of probabilistic estimation. At this point errors or fluctuation of sensors may be considered and with the help of recursive Bayesian estimation, it is possible calculate probability distribution for the state we are in. Aim of this approach is to collect maximum reward points from the route with the minimum cost.

While trying to solve a Markov Decision Process, from time to time we may need to ignore easily accessible reward points. Only in this way it is possible to collect higher reward points. This logic is provided with the Bellman equation. With the help of Bellman equation, algorithm gathers higher reward values.

Resolution Advisories

Maneuvers produced to avoid danger may suggest changing the speed in the vertical axis. These warnings are same with TCAS and they are called as advisories [2, 6]. Possible warnings that are generated by TCAS are given below and these warnings are used to generate the state space of Partially Observable Markov Decision Process. Moreover, it treats Maintain Climb and Maintain Descend maneuvers as a single maneuver rather than processing them separately. With the help of this combination, size of the space is reduced.

TCAS may issue a variety of different vertical advisories [22], including :

- *Clear of Conflict* (1)
- *Do not Climb* (2)
- *Do not Descend* (3)
- *Maintain*
- *Descend 1500 ft/min* (4)
- *Climb 1500 ft/min* (5)
- *Strong Descend 1500 ft/min* (6)
- *Strong Climb 1500 ft/min* (7)
- *Strong Descend 2500 ft/min* (8)
- *Strong Climb 2500 ft/min* (9)

TABLE 3.1: Invalid Transition Map

		to								
		Clear of Conflict	***Do not Climb***	***Do not Descend***	***Descend 1500 ft/min***	***Climb 1500 ft/min***	***Strong Descend 1500 ft/min***	***Strong Climb 1500 ft/min***	***Strong Descend 2500 ft/min***	***Strong Climb 2500 ft/min***
from	***Clear of Conflict***						X	X	X	X
	Do not Climb						X	X	X	X
	Do not Descend						X	X	X	X
	Descend 1500 ft/min								X	X
	Climb 1500 ft/min								X	X
	Strong Descend 1500 ft/min									
	Strong Climb 1500 ft/min									
	Strong Descend 2500 ft/min									P
	Strong Climb 2500 ft/min								P	

If the maneuver produced does not require changing the speed on the vertical axis, it is expressed as preventive. On the other hand, *Climb 1500*, *Descend 1500*, *Strong Climb 1500*, *Strong Descend 1500*, *Strong Climb 2500* and *Strong Descend 2500 feet/min* maneuvers are expressed as corrective [22]. *Do not Climb* and *Do not Descend* might be preventive as well as corrective.

On the other hand, there are some inconvenient maneuver transactions between preventive and corrective advisories. A map for inconvenient transactions are given in Table 3.1. Rows and columns are enumerated and their corresponding are given in parentheses at the list of advisories. Cells that are represented with X character means that the cell is inconvenient, and P character means the transition might be marked as inconvenient. As can be understood from the map, new advisory to be produced depends on the current one. For example, the *Strong Climb 2500 ft/min* advisory cannot be produced before *Climb 1500 ft/min*.

Dynamic Model

Geographical differences between Europe and America (high mountain ranges, etc.), frequency of control centers, differences in intercontinental flights can change the characteristics of air traffic. That's why air traffic controls may vary by region. An effort has been made to ensure that the model is as simple as possible but sufficient to avoid being affected by these differences. An analogy between this effort and handling effort for overfitting problem of neural network studies can be made. The model should not be created for a specific region. Another advantage of the model to be created in this way is that the model is not going to be too complex.

There are six state variables in the POMDP formulation:

- h_0, the altitude of the intruder relative to the own aircraft,
- $\acute{h}_0$, the vertical rate of the own aircraft,
- $\acute{h}_1$, the vertical rate of the intruder aircraft,
- T, the time to potential collision,
- s_{adv}, the current advisory, and
- s_{res}, whether the pilot is responding to the advisory.

Algorithm 2 calculateUValues

```
for $time = 1, 2, \ldots, T_{MAX}$ do
    Call calculateChunk() function for $time$ time steps, with 0 start to $N_{RALTITUDES} - 1$ end.
end for
```

While creating this dynamic model, the time axis is modeled with one-second intervals as can be seen from Algorithm 2. In the algorithm T_{MAX} stands for the time value until collision. Often referred as time to the closest approach or encounter time. This function calls *calculateChunk* Algorithm 3 for each second from the time to the closest approach back to the initial second.

Algorithm 3 calculateChunk

```
for $i_{iteration} = 0, 1, \ldots, N_{RALTITUDES}$ do
    Loop for relative altitudes.
    for $j_{iteration} = 0, 1, \ldots, N_{VSPEEDS}$ do
        Loop for own aircraft speeds.
        for $k_{iteration} = 0, 1, \ldots, N_{VSPEEDS}$ do
            Loop for intruder aircraft speeds.
            for $l_{iteration} = 0, 1, \ldots, N_{ADVISORIES}$ do
                Loop for current advisory types.
                for $m_{iteration} = 0, \ldots, N_{PR}$ do
                    Loop for pilot responses.
                    Call calculateUValue() function for $time$ time steps, with $i_{iteration}$ relative altitude, $j_{iteration}$ own aircraft speed, $k_{iteration}$ intruder aircraft speed, $l_{iteration}$ current advisory type, and $m_{iteration}$ pilot response.
                end for
            end for
        end for
    end for
end for
```

Pseudocode for the usage of other state variables in the POMDP formulation is given in Algorithm 3. *calculateChunk* function calls a calculation function which is named as *calculateUValue* Algorithm 4 for each possible state value that is possible to address by using all state variables in the POMDP formulation except time to potential collision: In the algorithm $N_{\mathrm{RALTITUDES}}$ stands for the altitude of the intruder relative to the own aircraft, N_{VSPEEDS} stands for the vertical rate of the own aircraft, second N_{VSPEEDS} stands for the vertical rate of the intruder aircraft, $N_{\mathrm{ADVISORIES}}$ stands for the current advisory, and finally, N_{PR} stands for the cases where pilot is responding or not. Iteration counts of these loops can be inspected from *State Count* column of Table 3.3.

In the following sections, it will be explained that each probability to be calculated is depend on only a second before of itself. Probability of responding an advisory that is going to be performed by a pilot is modeled by using previous response of the pilot (s_{res}), current advisory (s_{adv}), and new advisory (a). These three values are going to be taken into account in Equation 1.

$$P(s_{\mathrm{res}}, s_{\mathrm{adv}}, a) = \frac{1}{1 + Elapsed\,Time\,Till\,Response} \tag{1}$$

TABLE 3.2: Advisory Response Probabilities

		to								
		Clear of Conflict	***Do not Climb***	***Do not Descend***	***Descend 1500 ft/min***	***Climb 1500 ft/min***	***Strong Descend 1500 ft/min***	***Strong Climb 1500 ft/min***	***Strong Descend 2500 ft/min***	***Strong Climb 2500 ft/min***
from	***Clear of Conflict***	1	.16	.16	.16	.16	.16	.16	.16	.16
	Do not Climb	1	1	.16	.25	.16	.25	.16	.25	.16
	Do not Descend	1	.16	1	.16	.25	.16	.25	.16	.25
	Descend 1500 ft/min	1	.25	.16	1	.16	.25	.16	.25	.16
	Climb 1500 ft/min	1	.16	.25	.16	1	.16	.25	.16	.25
	Strong Descend 1500 ft/min	1	.25	.16	.25	.16	1	.16	.25	.16
	Strong Climb 1500 ft/min	1	.16	.25	.16	.25	.16	1	.16	.25
	Strong Descend 2500 ft/min	1	.25	.16	.25	.16	.25	.16	1	.16
	Strong Climb 2500 ft/min	1	.16	.25	.16	.25	.16	.25	.16	1

The probabilities given in the Table 3.2 were calculated according to above function which models a geometric distribution. To clarify this issue, delay times should be considered. According to the table, first column and diagonal line represent assumptions that made clear of conflict will always be answered, and the current advisory will continue to be applied until it is expired respectively. So these two probabilities are considered as one. For the following probabilities are calculated by following assumptions. The time it takes to implement the first advisory, and reversing the current advisory to the opposite direction is assumed to take about five seconds. The last possibility given in the table is made with the assumption that the time it takes to weaken or strengthen the current maneuver will take approximately three seconds. It may be possible to make these assumptions better. But we can state that the marginal-benefit is maximized [23].

The acceleration of the own aircraft depends on the probability of the pilot's response in upcoming second, and the advisory to be produced.

In cases where the pilot responds, advisories can be performed with an acceleration of 0.25

G or $0.\overline{33}$ G. These acceleration values correspond to $8.0435 ft/s^2$ and $10.7247 ft/s^2$ respectively and these values can vary $\pm 1 ft/s^2$. The advisories that the pilot does not respond to, on the other hand, are considered as non-accelerated movement and can vary $\pm 3 ft/s^2$.

Acceleration of other aircraft on the other hand, is going to be considered as non-accelerated movement with $\pm 3 ft/s^2$ vary.

Dynamic Programming

In order to be able to keep track of the environment we are in, we have to know the positions of the aircraft and velocities of both own and other aircraft. According to the possible actions a probability distribution calculated. In our example these actions might be Climb or Descend and as a result of performing an action a reward is given. By using this methodology we can select reasonable trajectories. In order to be able to implement this methodology Bellman equation is used and explained in Equation 2.

As stated under the Dynamic model section, it is stated that the values of acceleration of the own aircraft and acceleration of the other aircraft can vary. The values of acceleration of the own aircraft and acceleration of the other aircraft are determined with a method which is called as sigma-point sampling [1, 24]. The reason they vary mentioned here should be considered as standard deviation of a normal distribution with zero mean. Acceleration of the own aircraft and acceleration of the other aircraft are specified as continuous probability densities. Upcoming time value to potential collision on the other hand, is specified as density. At that point, integration performed in the calculation of continuous probability distribution will not be reasonable. [1] It would be more appropriate to calculate it with discrete sum rather than continuous sum with integration.

In case of having lack of knowledge about the current state, a set of probability calculated to learn about the environment. In other words, the equation helps us to make a set of sequential decisions for a partially observable environment and additive rewards which is called as Partially Observable Markov Decision Process (POMDP) for the fundamentals of the airborne collision avoidance logic. However, performing a POMDP over a problem like an airborne collision avoidance logic, requires a lot of time and memory. In terms of computational complexity theory, worst case of this problem is going to be PSPACE-hard.

In order to develop a memory friendly system with a good performance, dynamic programming is used. At the problem at hand here, there is no need to remember the entire state space, since the states are determined by only considering the data of a second before. Excessive memory consumption problem [25] was solved with the single iteration Gauss-Seidel methodology. [1]

With the set of next states (s) and actions (a) are now finite, the Bellman equation [26] becomes

$$\underbrace{U^*(s)}_{\text{Utility value}} = \underset{\substack{a \\ \text{Maximum} \\ \text{overall the actions}}}{\max} \left(\overbrace{R(s,a)}^{\substack{\text{Reward for leaving} \\ \text{state with action}}} + \underset{\text{Discount factor}}{\gamma} \sum_{s'} \underbrace{T(s,a,s')}_{\substack{\text{Transition} \\ \text{function}}} \overbrace{U^*(s')}^{\substack{\text{U-Value} \\ \text{End up in}}} \right) \tag{2}$$

Discount factor shown here is important. Gamma as one essentially means there is no discounting. A lower value encourages short-term thinking. A higher value emphasizes long-term rewards [27]. Here we also use one. In other words, we want previous states to be as decisive as possible.

Pseudocode for calculateUValue function is given in Algorithm 4. In this part of the code, usage of dynamic programming and Bellman equation can be inspected.

Since this is a real world problem, data should be converted from analog to discrete. While doing this, multidimensional interpolation technique is used [28]. Relative altitude of the aircraft are presented with 61 intervals. Details of selecting these intervals can be explained as follows. Discrete state space can be inspected from the Table 3.3.

The relative heights should be equal in the positive and negative zones. If the determined maximum height value is chosen as 3000, it means the relative heights are going to be in the range of ± 3000 feets. So, if we select the intervals as 100 feet, we have 30 quantized relative heights for positive zone, 30 for negative zone and one for the same flight level. In terms of speed values there is a similar approach. For 2500 feet/min maximum relative speed, we have 5 quantized speed value in the positive zone, 5 for negative zone and one for the same speed.

Algorithm 4 calculateUValue

```
$State_{Current} \leftarrow$ {Relative Altitude, Own Aircraft Speed, Intruder Aircraft Speed, Current Advisory, Pilot Response}
$U_{Previous} \leftarrow time - 1$
$U_{Current} \leftarrow time$
for $i_{iteration} = 0, 1, \ldots, N_{ADVISORIES}$ do
    $RA \leftarrow i_{iteration}$
    $PR_{Current} \leftarrow$ Pilot Response
    $PR_{Prob} \leftarrow$ Calculated probability of occurrence assigned. (See Table 3.2)
    Bernoulli process: probability of occurrence.
    $respProbs[0] \leftarrow PR_{Prob}$
    Bernoulli process: probability of not occurrence.
    $respProbs[1] \leftarrow 1 - PR_{Prob}$

    In cases where the pilot responds, advisories can be performed with an acceleration of $8.0435 ft/s^2$ and $10.7247 ft/s^2$ and these values can vary $\pm 1 ft/s^2$. Following two for loops test all possibilities.
    for $j_{iteration} = 0, \ldots, 3$ do
        for $k_{iteration} = 0, \ldots, 3$ do
            Call calculateDynamics() function for $State_{Current}$, with all possible maneuvers that own aircraft can perform while intruder does not generate a resolution advisory.
            $State_{Next} \leftarrow$ {Relative Altitude, Own Aircraft Speed, Intruder Aircraft Speed, Advisory, Pilot Response}
        end for
    end for

    $Reward_{Current} \leftarrow$ Call getReward() function for $State_{Current}$, with $RA$ and $time$.

    if $time > 0$ then
        for $j_{iteration} = 0, 1 \ldots, N$ do
            $TotalRewards_{Previous} \leftarrow$ Interpolated reward value.
            $U \leftarrow$ Bellman equation applied with discount factor one.
        end for
    end if
    The state-action values produced through dynamic programming are saved in a lookup table.
    if $U > U_{Maximum}$ then
        $U_{Maximum} \leftarrow U$
    end if
end for
Utility for $State_{Current} \leftarrow U_{Maximum}$
```

TABLE 3.3: State space variables

Variable	**Minimum**	**Maximum**	**State Count**	**Metric**
Symbol	*Value*	*Value*	*#ofvalues*	*Unit*
h_0	-3000	3000	61	Feet
$\acute{h}_0$	-41.66	41.66	11	Feet/sec
$\acute{h}_1$	-41.66	41.66	11	Feet/sec
T	0	40	41	Second
s_{adv}	1	9	9	N/A ↓
				Clear of Conflict
				Do not Climb
				Do not Descend
				Descend 1500 ft/min
				Climb 1500 ft/min
				Strong Descend 1500 ft/min
				Strong Climb 1500 ft/min
				Strong Descend 2500 ft/min
				Strong Climb 2500 ft/min
s_{res}	0	1	2	N/A ↓
				Pilot responding to the advisory
				Pilot not responding to the advisory

The height and speed ranges can be chosen narrowly to keep the size of the decision table small and to shorten the run time of generation. Different resolutions can be selected according to the needs. In order to produce the decision table fast, all relative heights and speeds are now homogeneously distributed.

In the studies for ACAS X, the intervals determined for relative height and speed grow as they approach to the minimum and maximum boundary values. Intervals are smaller since they are more sensitive when they are closer to the same flight level and same speed.

On the other hand, the time axis is modeled with one second intervals. For the scenarios which are longer than 40 seconds, it may not be beneficial to consider the decision table. Therefore, the time axis is limited with 40 seconds.

4. REWARD FUNCTION

As stated in the previous section, to be able to keep track of the environment we are in, we have to calculate a probability distribution. In our example these actions might be Climb or Descend and as a result of performing an action a reward is provided to the Bellman Equation 2. By using this methodology we can select reasonable trajectories.

The actions chosen in the solution of the problem that we will deal with the Partially Observable Markov Decision Process can change the existing state space and a reward is given according to the result of the selected action [29]. This approach is used to identify possible maneuver that would provide the greatest benefits in terms of providing enough separation with the minimum cost.

On the other hand, providing sufficient separation is not decisive alone. At this point, the Figure 4.1, Figure 4.2, and Figure 4.3 given in Airborne Collision Avoidance System Guide [2] present the decision-making process. Amount of separation that aims to make encounter models safe is referred as ALIM. ALIM stands for *Altitude Limit*.

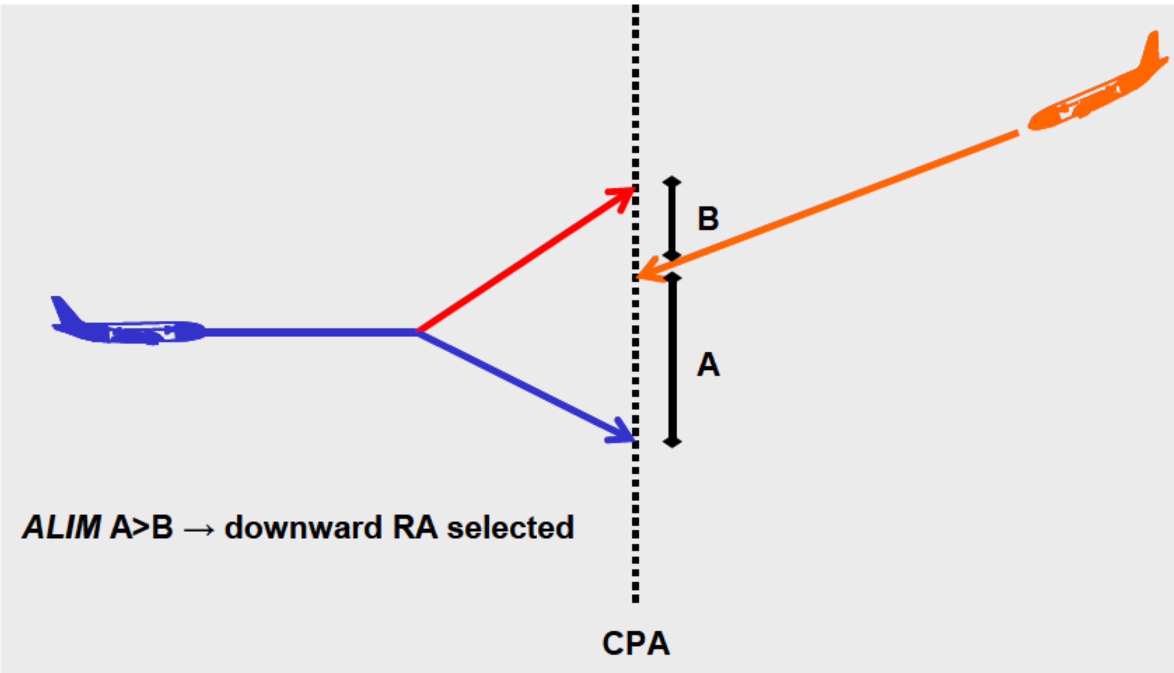


FIGURE 4.1: Deciding to sense. *Courtesy of* EUROCONTROL, Airborne Collision Avoidance System Guide [2]

In the simulation developed, this approach was implemented by using penalty points instead of reward points. A penalty point was set for each maneuver, considering that the best option

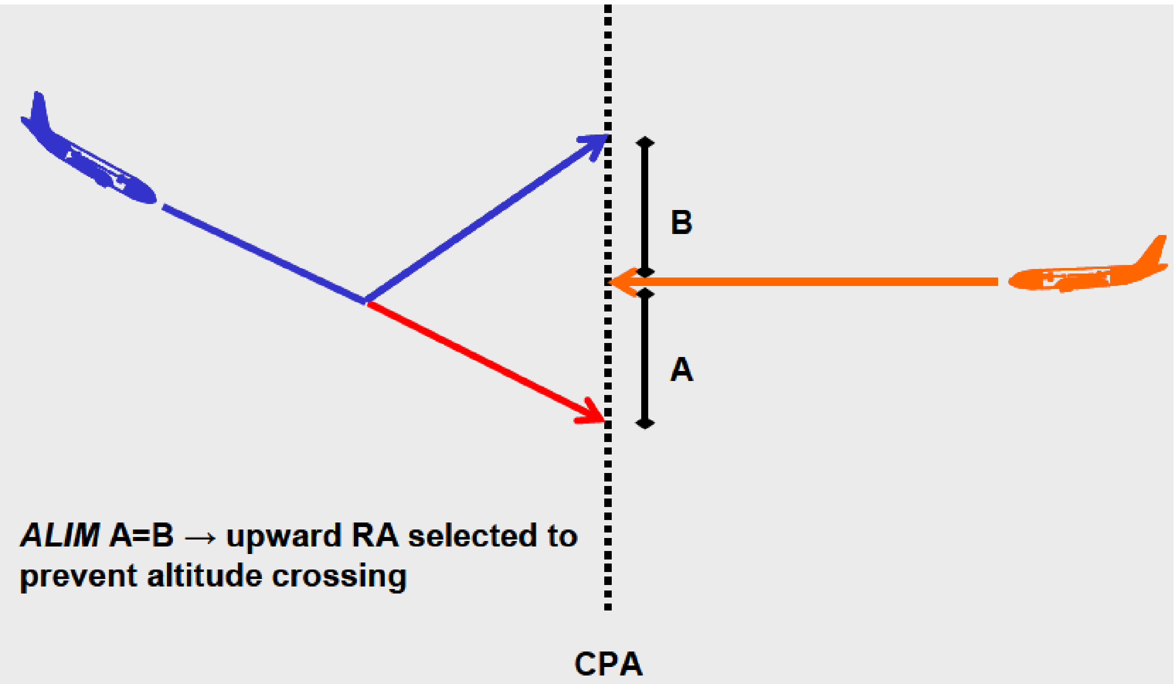


FIGURE 4.2: No Altitude Crossing Resolution Advisory. *Courtesy of* EUROCONTROL, Airborne Collision Avoidance System Guide [2]

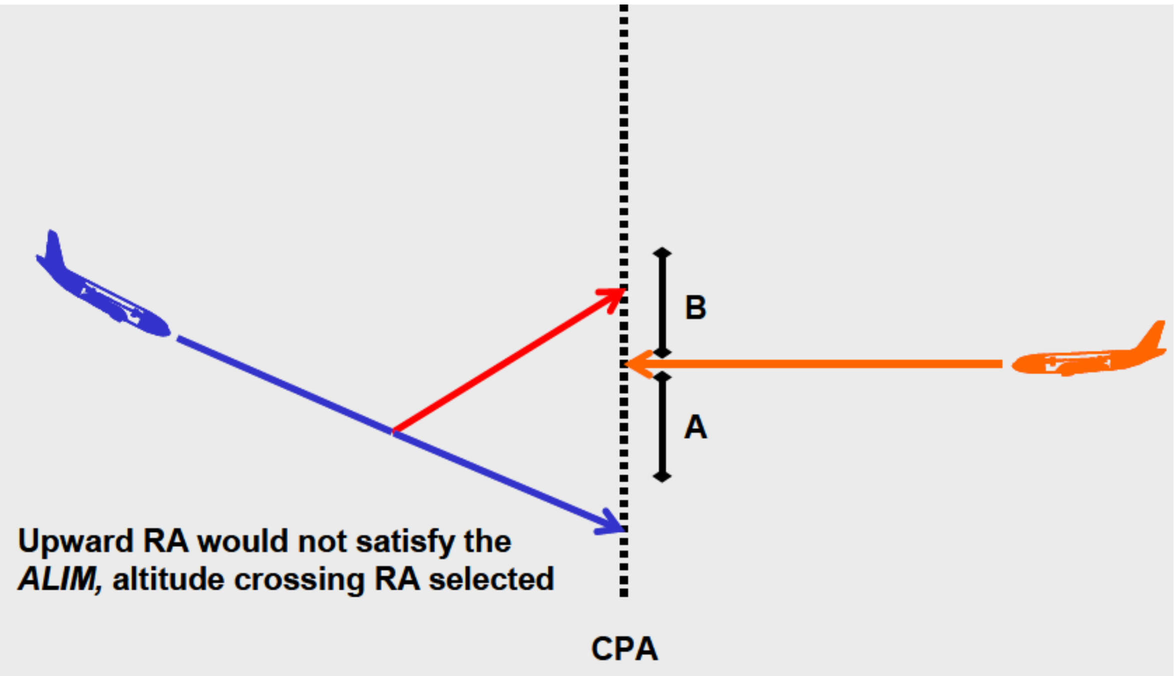


FIGURE 4.3: Altitude Crossing Resolution Advisory. *Courtesy of* EUROCONTROL, Airborne Collision Avoidance System Guide [2]

is to maneuver as little as possible. Therefore, the reward values given in Table 4.1 and Table 4.2 are shown as negative. Penalty points were determined by expert opinion [30], and these points have less or higher weights according to the relativity within themselves. Clear of Conflict advisory, which expresses only situations where collision is prevented, is positively valued.

Some rewards are calculated in relation to the speed of Own Aircraft in the vertical axis. Since these values can change wildly, the relative speed values are considered independently and their coefficients are evaluated as reward points.

In order to get a reward point, relative distance between own aircraft and the intruder in terms of vertical axis, vertical speed and advisory type and conditions can be determinant. In cases where the variables and conditions are met, the penalty or reward points matching the relevant parameters will be deserved and the deserved points will be added up with previous points. However, in order to deserve a point, all parameters such as relative distance and vertical speed may not have to be taken into account. Only a reversal of a current advisory might be enough to deserve a point. There might also be other possibilities that are sufficient to deserve a point like intensifying an advisory to be performed with stronger G value, or simply producing an advisory.

The conditions followed for the eligibility of penalty points for crossing advisories can be interpreted slightly different. For crossing advisory, the relative height is only considered as a sign. The main thing is whether the other plane is above or below us. In short, it is assumed that if you rise when the other plane is above and descend when it is below, there will be an intersection. This type of advisories are not preferred unless we have to.

It is stated that some absolute collisions can be prevented by crossing advisories. In order to see the effect of this, an alternative reward function is tested. In this alternative reward function, crossing advisories are not penalized. This approach will encourage the generation of crossing advisories. It was observed that higher performance was achieved with alternative reward function. By changing some parameters of the basic table, it is possible to compare the results with default values. These results are shared in evaluation section. In order to encourage crossing advisories, penalty points of event number 4, 8, and 15 have been set to zero in Table 4.1. With the help of this alternative reward function where we set these event numbers to zero, it is possible to observe the positive effect of crossing advisories. This table determines the event's story by using type of action to perform a transition from a state to another, and time value. Reward scores are used to see if the action is a good choice or not. In addition to the consideration of the event's story, both separation and closure values, which stand for relative height and relative speed respectively, are taken into account. When the relevant event story and, if appropriate, the separation and closure criteria are met, the reward score corresponding to the relevant maneuver is given. In this way, it is possible to determine reasonable trajectories. Unlike this table, there are no penalty points for maintaining advisories at neither default nor alternative reward functions.

While determining all these announced reward points, it is estimated that there is a trade-off between operational efficiency and safety of flight [31]. A number of empirical experiments have been carried out to find a better ratio to choose between these two important criteria. These experiments were carried out to try alternative reward points, as well as with the changes made in the ”prohibited advisory transition” table, which is one of the parameters that must be provided for the penalty point.

Outline of this reward values are given from Kochenderfer's study [1]. There might be some events to understood clearly. For example, although we think *clear of conflict* as a type of advisory, it is not covered in scope of *any advisory* event in Egorov et al's work [18]. Default version of get reward function is given in Algortihm 5.

Algorithm 5 getReward

```
$Reward \leftarrow 0$
$Story \leftarrow$ Detect story of current event given in Table 4.1 by considering next state, $RA$, and $time$.
if $Separation$ and $Closure$ applicable for the $Story$ then
    $Match \leftarrow$ Evaluate the $Story$ has a correspondence in the Table 4.1 with $Separation$ and $Closure$.
else
    $Match \leftarrow$ Evaluate the $Story$ has a correspondence in the Table 4.1.
end if
if $Match = TRUE$ then
    $Reward \leftarrow$ Reward value of the event.
end if
return $Reward$
```

It can be checked that less or more of these advisories will be produced by making changes with the corrective advisory scores given in the table. Decreasing a penalty score, given as negative values, to a smaller value will increase operational efficiency, but will make the system more vulnerable to collisions. On the other hand, giving this score higher will

TABLE 4.1: Rewards of events. *Courtesy of* M. J. Kochenderfer, Decision Making Under Uncertainty [1]

Event	**Event**	**Separation**	**Closure**	**Reward**
$\#$	$story$	$feet$	$feet/min$	
1	$T \leq 0$	≤ 175		-1
2	Maintain advisory with $\acute{h}_0 <$ 1500 ft/min			-1
3	Prohibited advisory transitions			-1
4	Preventive crossing advisory			-1
5	Corrective advisory	> 650	< 2000	-1×10^{-1}
6	Corrective advisory	> 1000	< 4000	-3×10^{-2}
7	Preventive advisory	> 650	< 2000	-1×10^{-2}
8	Crossing advisory	> 500		-1×10^{-2}
9	Reversal			-8×10^{-3}
10	Strengthening			-5×10^{-3}
11	Weakening			-1×10^{-3}
12	Non-MVS/LOLO		> 3000	-1.5×10^{-3}
13	Any advisory		> 3000	-2.3×10^{-3}
14	MVS/LOLO		> 3000	-5×10^{-4}
15	Crossing advisory when $\lvert\acute{h}_0\rvert >$ 500 ft/s and $\acute{h}_0$ is in opposite direction of advisory			-4×10^{-4} x $\Delta\acute{h}$
16	Maintain			-4×10^{-4}
17	MVS/LOLO			-1×10^{-4}
18	Any advisory			-3×10^{-5} x $\Delta\acute{h}$
19	Corrective advisory			-1×10^{-5}
20	Clear of Conflict			1×10^{-9}

strengthen the protection against the collision, but may cause generating too many warnings, but pilots do not want to hear these warnings too often.

Therefore, it is avoided to change the specified default values too much. In the study, 6 penalty points were determined as the basis to present a comparative analysis and all combinations are tested with a distribution that does not exceed half of the reward points. These tested minimum and maximum values were determined by doing a set of arithmetic operation for the value expressed as *DEFAULT* and shown in pseudocode. Pseudocode for value generation process is given in Algorithm 6. Since Table 4.1 contains 20 different events, a nested for loop structure is required. There is an if-else condition for each reward value to compose combination of the value by using range of reward values from fifth reward to the last reward value. The reward points for the first four events are excluded because these events are crucial for system safety or point out abnormal stories such as invalid advisory transition or negative amount of time to potential collision.

Algorithm 6 valueGenerator

```
$afData[Event_{Number}] \leftarrow$ Rewards of events are assigned according to Table 4.1 as default values.
$COUNT[Event_{Number}] \leftarrow$ How many different value is going to be tested for event.
$DEFAULT[Event_{Number}] \leftarrow$ Default reward value is assigned.
$MIN[Event_{Number}] \leftarrow 0.5 * DEFAULT[Event_{Number}]$
$MAX[Event_{Number}] \leftarrow 1.5 * DEFAULT[Event_{Number}]$
$ACCURACY[Event_{Number}] \leftarrow (MAX[Event_{Number}] - MIN[Event_{Number}]) \div COUNT[Event_{Number}]$

$afData[1,2,3,4] \leftarrow -1$

for $i_5 = 0, \ldots, COUNT_5$ do
    $afData[5] \leftarrow$
    if $COUNT_5 = 0$ then $DEFAULT_5$ else $MIN_5 + (i_5 * ACCURACY_5)$

    for $i_6 = 0, \ldots, COUNT_6$ do
        $afData[6] \leftarrow$
        if $COUNT_6 = 0$ then $DEFAULT_6$ else $MIN_6 + (i_6 * ACCURACY_6)$
        ⋮
        for $i_{19} = 0, \ldots, COUNT_{19}$ do
            $afData[19] \leftarrow$
            if $COUNT_{19} = 0$ then $DEFAULT_{19}$ else $MIN_{19} + (i_{19} * ACCURACY_{19})$

            $threadParams[nextThreadID].inputData \leftarrow$ Values$_N$.txt
            $threadParams[nextThreadID].outputData \leftarrow$ ACASXDT$_N$.bin
            $threadParams[nextThreadID].resultTxt \leftarrow$ Results$_N$.txt
            Call writeDataToFile() function for $threadParams[nextThreadID]$ input data, with $afData[Event_{Number}]$ to write input data values to file for this thread.
            Call threadRun() function for $threadParams[nextThreadID]$ data.
        end for
        ⋮
    end for
end for
```

In order to be able to present a comparative analysis with all combinations that are tested with a distribution which does not exceed half of the reward points with 3 different values for each requires extreme amount of time. Generation of a binary file specified at Operation 1 with a single set of reward values takes almost three minutes with an Intel Core i7-4790 CPU @ 3.60GHz, 64-bit, 16.0 GB RAM, Windows 10.

Since there are too many parameters to be tested in an empirical way, finding the best combination of reward values requires huge amount of man-hour. Therefore, in this case 6 penalty points were determined as the basis to be analyzed and execution time is reduced by running our program in parallel with 5 thread. Since this development environment has only 4 physical (8 threads) cores thread number is limited by considering requirements of the operation system as well.

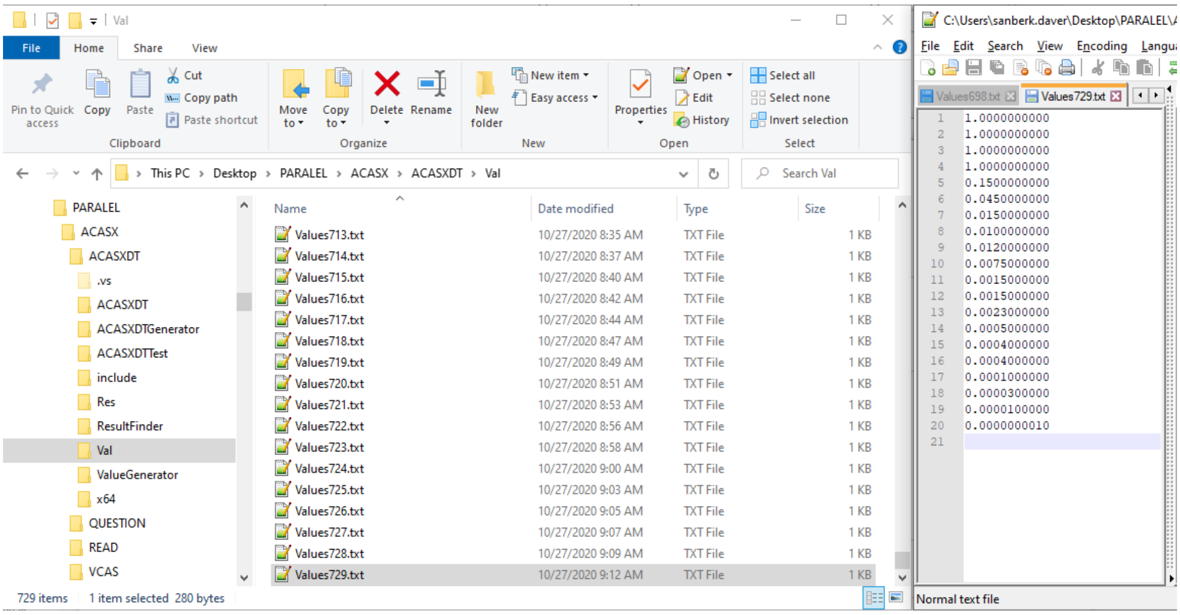
Since each value determined is going to be tested with three different combinations, 729 iterations are required. The number 729 is obtained by the sixth power of three. Computation of a single set of reward values takes almost three minutes with the presented system properties. With a linear approach, all combinations would take more than 36 hours to test. Duration of this period is important in simulation studies that require a lot of repetition. Therefore, parallelization is important. With the preferred approach, the time has been reduced to 8 hours.

Algorithm 7 threadRun

```
Start ACASXDTGenerator.exe with $threadParams[nextThreadID].inputData$ to generate $threadParams[nextThreadID].outputData$ binary file.
Start ACASXDTTest.exe with $threadParams[nextThreadID].outputData$ to get $threadParams[nextThreadID].resultTxt$ simulation results for $threadParams[nextThreadID].inputData$.
Delete $threadParams[nextThreadID].outputData$.
```

Pseudocode for thread operations are given in Algorithm 7. Each thread initiates *ACASXDTGenerator.exe* to generate an output binary file that is created by using a different combination of reward values. This data synthesis part ends up with an output binary file with a size of almost 380 MB. After that each tread is responsible to test its own output binary. Therefore each thread initiates *ACASXDTTest.exe* by taking its own output binary as input to get simulation results and saves the results of it. Since storing all generated output binary files is a very memory consuming task, each tread deletes its own output binary file at the end of test operation.

Operation 1, that is given in thread run Algorithm 7, takes reward values as input. A sample input file and folder structure can be inspected from Figure 4.4. On the other hand, the result file which is the output file of Operation 2 can also be observed from the Figure 4.5.

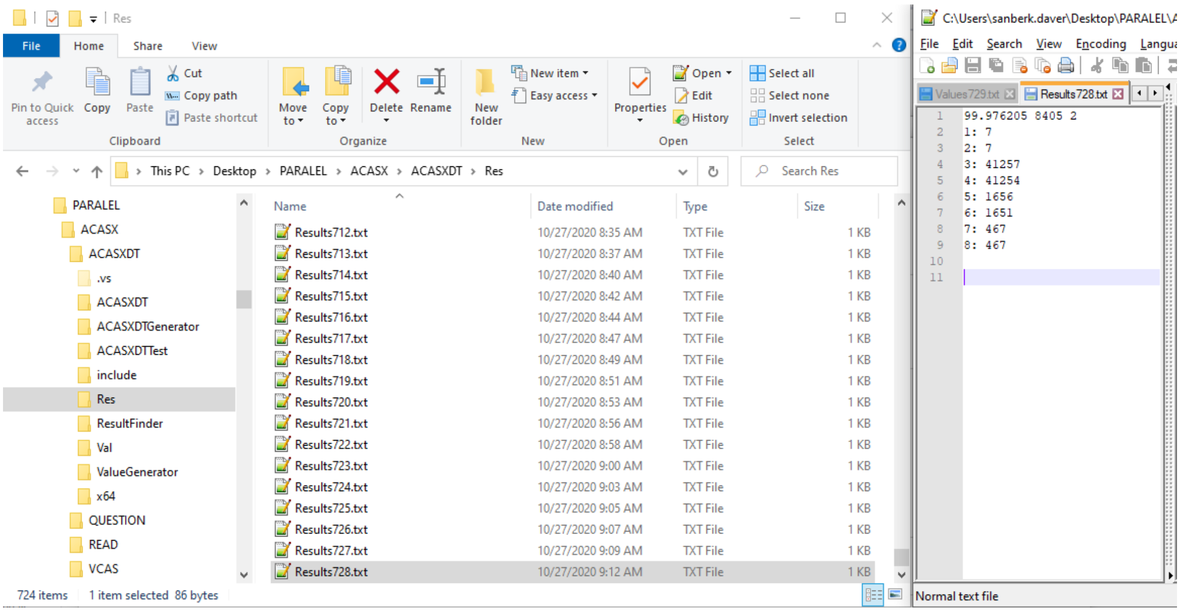


Name	Date modified	Type	Size
Values713.txt	10/27/2020 8:35 AM	TXT File	1 KB
Values714.txt	10/27/2020 8:37 AM	TXT File	1 KB
Values715.txt	10/27/2020 8:40 AM	TXT File	1 KB
Values716.txt	10/27/2020 8:42 AM	TXT File	1 KB
Values717.txt	10/27/2020 8:44 AM	TXT File	1 KB
Values718.txt	10/27/2020 8:47 AM	TXT File	1 KB
Values719.txt	10/27/2020 8:49 AM	TXT File	1 KB
Values720.txt	10/27/2020 8:51 AM	TXT File	1 KB
Values721.txt	10/27/2020 8:53 AM	TXT File	1 KB
Values722.txt	10/27/2020 8:56 AM	TXT File	1 KB
Values723.txt	10/27/2020 8:58 AM	TXT File	1 KB
Values724.txt	10/27/2020 9:00 AM	TXT File	1 KB
Values725.txt	10/27/2020 9:03 AM	TXT File	1 KB
Values726.txt	10/27/2020 9:05 AM	TXT File	1 KB
Values727.txt	10/27/2020 9:07 AM	TXT File	1 KB
Values728.txt	10/27/2020 9:09 AM	TXT File	1 KB
Values729.txt	10/27/2020 9:12 AM	TXT File	1 KB

```
1.0000000000
1.0000000000
1.0000000000
1.0000000000
0.1500000000
0.0450000000
0.0150000000
0.0100000000
0.0120000000
0.0075000000
0.0015000000
0.0015000000
0.0023000000
0.0005000000
0.0004000000
0.0004000000
0.0001000000
0.0000300000
0.0000100000
0.0000000010

```

FIGURE 4.4: A sample Values.txt file



Name	Date modified	Type	Size
Results712.txt	10/27/2020 8:35 AM	TXT File	1 KB
Results713.txt	10/27/2020 8:37 AM	TXT File	1 KB
Results714.txt	10/27/2020 8:40 AM	TXT File	1 KB
Results715.txt	10/27/2020 8:42 AM	TXT File	1 KB
Results716.txt	10/27/2020 8:44 AM	TXT File	1 KB
Results717.txt	10/27/2020 8:47 AM	TXT File	1 KB
Results718.txt	10/27/2020 8:49 AM	TXT File	1 KB
Results719.txt	10/27/2020 8:51 AM	TXT File	1 KB
Results720.txt	10/27/2020 8:53 AM	TXT File	1 KB
Results721.txt	10/27/2020 8:56 AM	TXT File	1 KB
Results722.txt	10/27/2020 8:58 AM	TXT File	1 KB
Results723.txt	10/27/2020 9:00 AM	TXT File	1 KB
Results724.txt	10/27/2020 9:03 AM	TXT File	1 KB
Results725.txt	10/27/2020 9:05 AM	TXT File	1 KB
Results726.txt	10/27/2020 9:07 AM	TXT File	1 KB
Results727.txt	10/27/2020 9:09 AM	TXT File	1 KB
Results728.txt	10/27/2020 9:12 AM	TXT File	1 KB

```
99.976205 8405 2
1: 7
2: 7
3: 41257
4: 41254
5: 1656
6: 1651
7: 467
8: 467


```

FIGURE 4.5: A sample Results.txt file

The list of tried advisories is given below;

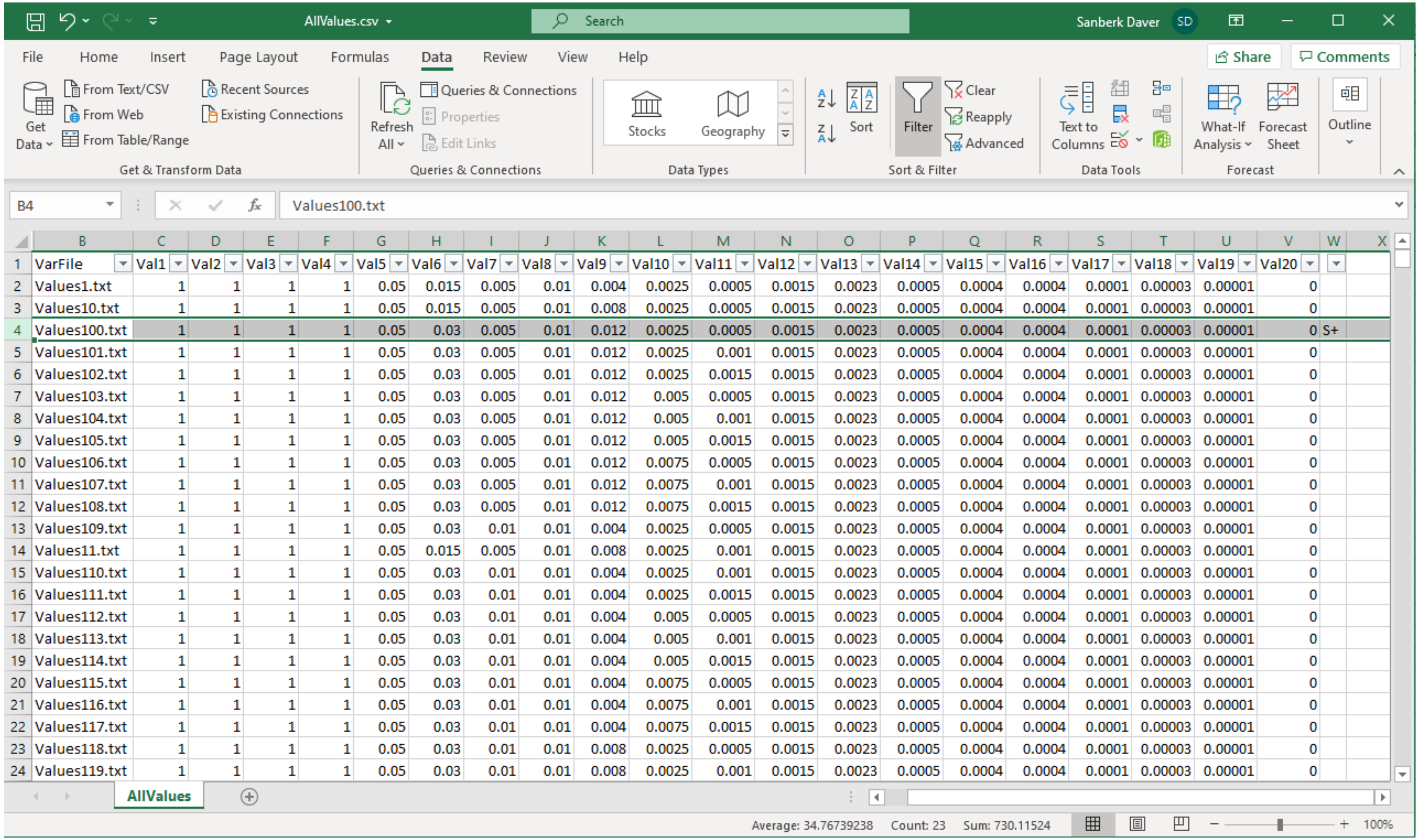
- *Corrective Advisory* with more than 650 ft. separation,
- *Corrective Advisory* with more than 1000 ft. separation,
- *Preventive Advisory* with more than 650 ft. separation,
- *Reversal*,
- *Strengthening*,
- *Weakening*

Complex computations and computation time required by the process does not allow the examination of entire encounter scenarios. Therefore, a sample has been chosen from the possible scenarios. Scenarios that have 17 seconds left to encounter determined as sample and all combinations of the 6 parameters mentioned above were tested. Among the 729 results obtained, the reward values that have resulted with the least advisory generation without compromising the level of safety were selected and expressed as *operational plus (O+)*. In addition, the values that have resulted with less collisions were also examined. Among the scenarios that have less collisions, reward points that generates the least advisory generation were named as *safety plus (S+)*.

Algorithm 8 is given for result finder pseudocode. Result finder accesses to all tested reward values and test results that are achieved with the combination of reward values. Result finder also accesses the number of generated advisories and performance data for each test that is performed with a look-up table which is synthesized with a different combinations of reward points.

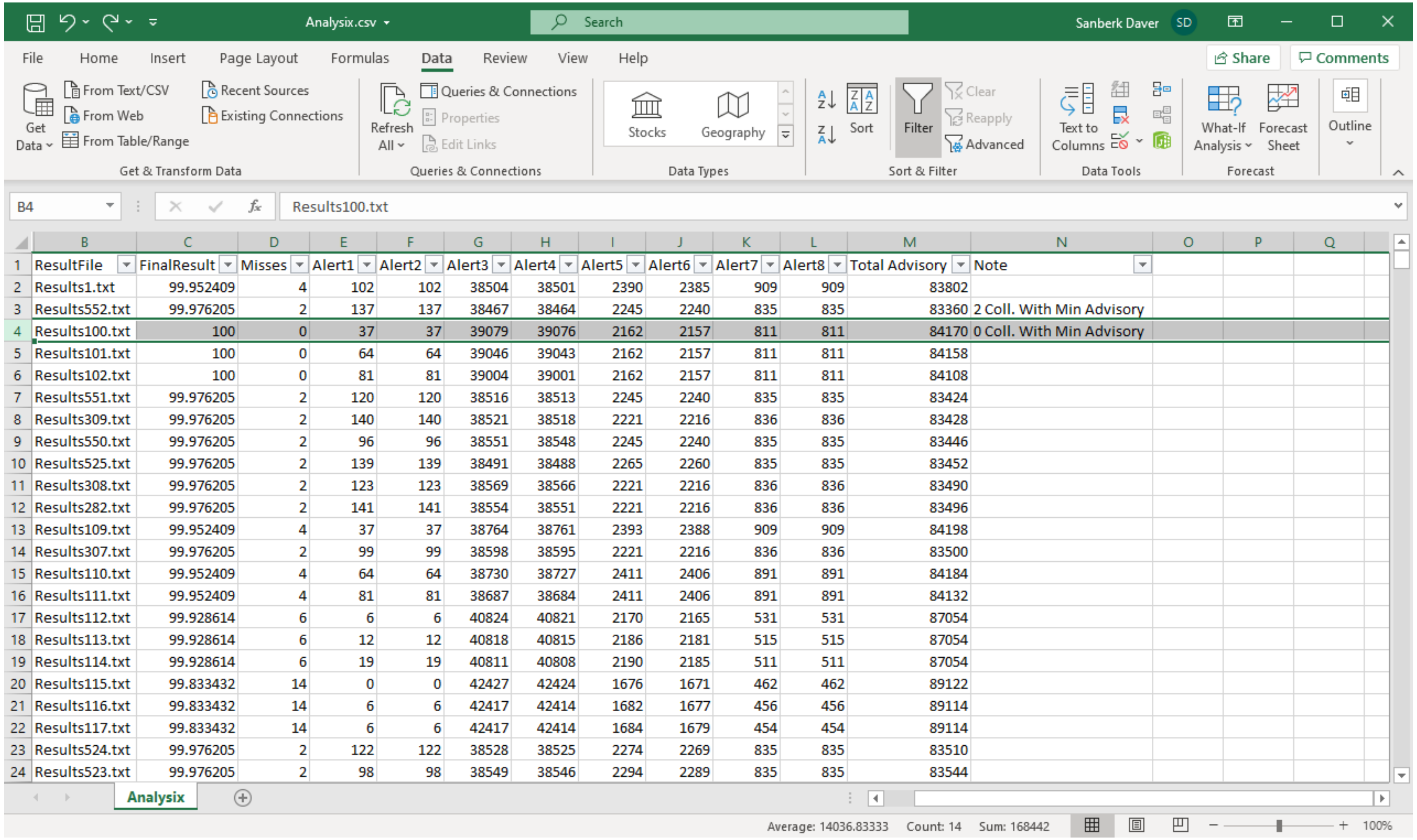
Produced Analysis.csv file consist of columns that holds the tested scenario names, percentage of collision avoidance success, number of collisions, and number of generated advisories except clear of conflict advisories. At the second part, AllValues.csv file consist of columns that holds the tested scenario names, and reward values which are used to synthesize look-up tables.

Operational plus (O+) and *safety plus (S+)* values are realized by sorting AllValues.csv and Analysis.csv files. Contents of AllValues.csv and Analysis.csv files can be inspected from Figure 4.6 and Figure 4.7 respectively.



	VarFile	Val1	Val2	Val3	Val4	Val5	Val6	Val7	Val8	Val9	Val10	Val11	Val12	Val13	Val14	Val15	Val16	Val17	Val18	Val19	Val20	
2	Values1.txt	1	1	1	1	0.05	0.015	0.005	0.01	0.004	0.0025	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
3	Values10.txt	1	1	1	1	0.05	0.015	0.005	0.01	0.008	0.0025	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
4	Values100.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.0025	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	S+
5	Values101.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.0025	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
6	Values102.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.0025	0.0015	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
7	Values103.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.005	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
8	Values104.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.005	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
9	Values105.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.005	0.0015	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
10	Values106.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.0075	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
11	Values107.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.0075	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
12	Values108.txt	1	1	1	1	0.05	0.03	0.005	0.01	0.012	0.0075	0.0015	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
13	Values109.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.0025	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
14	Values11.txt	1	1	1	1	0.05	0.015	0.005	0.01	0.008	0.0025	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
15	Values110.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.0025	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
16	Values111.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.0025	0.0015	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
17	Values112.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.005	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
18	Values113.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.005	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
19	Values114.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.005	0.0015	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
20	Values115.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.0075	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
21	Values116.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.0075	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
22	Values117.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.004	0.0075	0.0015	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
23	Values118.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.008	0.0025	0.0005	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	
24	Values119.txt	1	1	1	1	0.05	0.03	0.01	0.01	0.008	0.0025	0.001	0.0015	0.0023	0.0005	0.0004	0.0004	0.0001	0.00003	0.00001	0	

FIGURE 4.6: View from AllValues.csv file



Results100.txt

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	ResultFile	FinalResult	Misses	Alert1	Alert2	Alert3	Alert4	Alert5	Alert6	Alert7	Alert8	Total Advisory	Note			
2	Results1.txt	99.952409	4	102	102	38504	38501	2390	2385	909	909	83802				
3	Results552.txt	99.976205	2	137	137	38467	38464	2245	2240	835	835	83360	2 Coll. With Min Advisory			
4	Results100.txt	100	0	37	37	39079	39076	2162	2157	811	811	84170	0 Coll. With Min Advisory			
5	Results101.txt	100	0	64	64	39046	39043	2162	2157	811	811	84158				
6	Results102.txt	100	0	81	81	39004	39001	2162	2157	811	811	84108				
7	Results551.txt	99.976205	2	120	120	38516	38513	2245	2240	835	835	83424				
8	Results309.txt	99.976205	2	140	140	38521	38518	2221	2216	836	836	83428				
9	Results550.txt	99.976205	2	96	96	38551	38548	2245	2240	835	835	83446				
10	Results525.txt	99.976205	2	139	139	38491	38488	2265	2260	835	835	83452				
11	Results308.txt	99.976205	2	123	123	38569	38566	2221	2216	836	836	83490				
12	Results282.txt	99.976205	2	141	141	38554	38551	2221	2216	836	836	83496				
13	Results109.txt	99.952409	4	37	37	38764	38761	2393	2388	909	909	84198				
14	Results307.txt	99.976205	2	99	99	38598	38595	2221	2216	836	836	83500				
15	Results110.txt	99.952409	4	64	64	38730	38727	2411	2406	891	891	84184				
16	Results111.txt	99.952409	4	81	81	38687	38684	2411	2406	891	891	84132				
17	Results112.txt	99.928614	6	6	6	40824	40821	2170	2165	531	531	87054				
18	Results113.txt	99.928614	6	12	12	40818	40815	2186	2181	515	515	87054				
19	Results114.txt	99.928614	6	19	19	40811	40808	2190	2185	511	511	87054				
20	Results115.txt	99.833432	14	0	0	42427	42424	1676	1671	462	462	89122				
21	Results116.txt	99.833432	14	6	6	42417	42414	1682	1677	456	456	89114				
22	Results117.txt	99.833432	14	6	6	42417	42414	1684	1679	454	454	89114				
23	Results524.txt	99.976205	2	122	122	38528	38525	2274	2269	835	835	83510				
24	Results523.txt	99.976205	2	98	98	38549	38546	2294	2289	835	835	83544				

FIGURE 4.7: View from Analysis.csv file

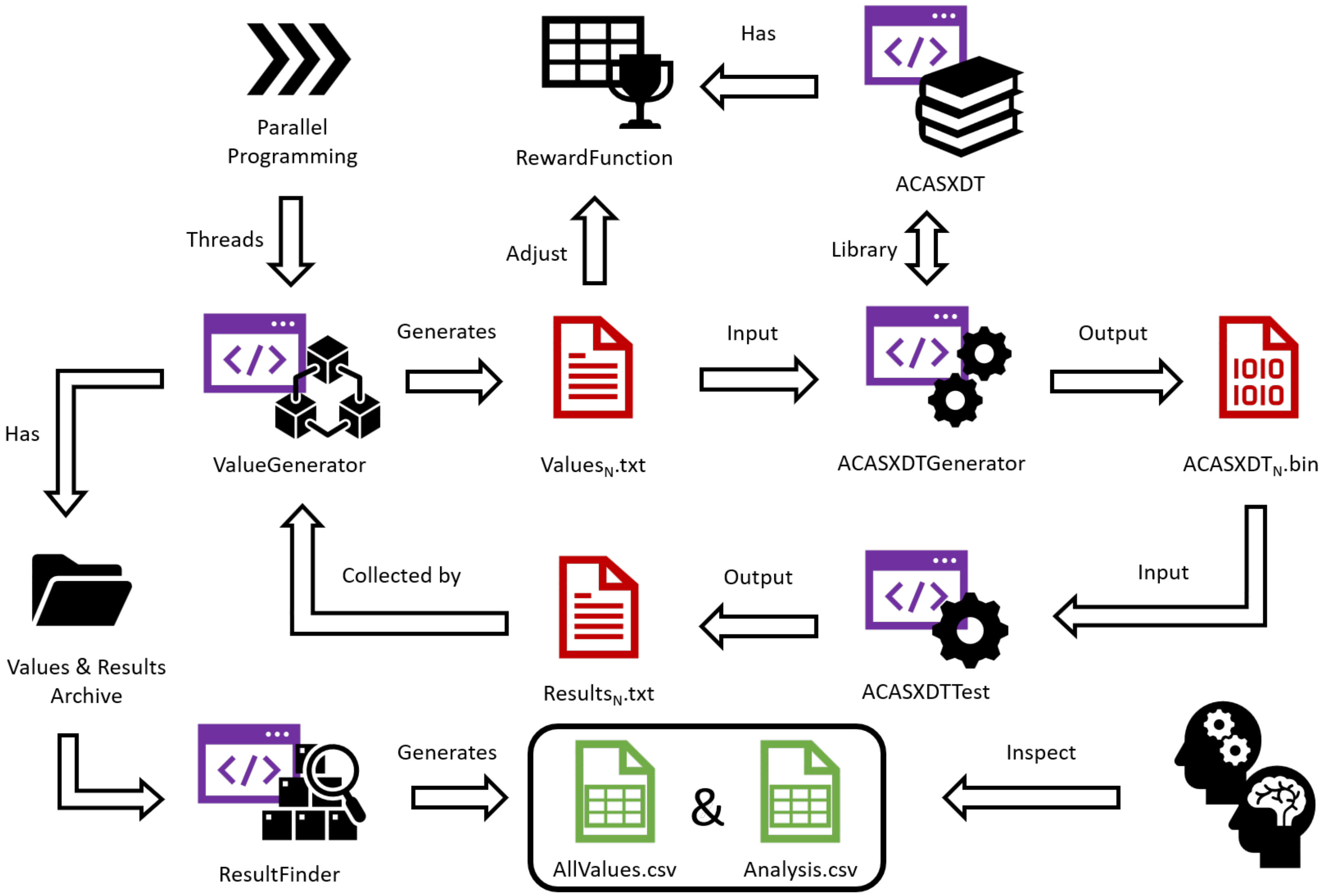


FIGURE 4.8: Overall system architecture

Algorithm 8 resultFinder

```
$Directory_{Result}$ ← Path that contains $\text{Results}_\text{N}$.txt files.
$Directory_{Values}$ ← Path that contains $\text{Values}_\text{N}$.txt files.

Creates a $Analysis.csv$ file to store result details.
Defines attributes as $ResultFileName, FinalResult, Misses, Alert1, \ldots, Alert8$ respectively.

Call $FindFirstFile()$ with $Directory_{Result}$
do
    $Analysis.csv$ ← Extract current $file$ content into CSV.
while (Call $FindNextFile()$ with $Directory_{Result}$) $\neq NULL$

Creates a $AllValues.csv$ file to store details of reward value combinations.
Defines attributes as $VarFileName, Value1, Value2, \ldots, Value20$ respectively.

Call $FindFirstFile()$ with $Directory_{Values}$
do
    $AllValues.csv$ ← Extract current $file$ content into CSV.
while (Call $FindNextFile()$ with $Directory_{Values}$) $\neq NULL$
```

It will be useful to examine the system architecture from Figure 4.8 to understand how the analyzed test results are obtained. Proposed values and related events are presented in Table 4.2.

TABLE 4.2: Details of Proposed Rewards

Event	**Event**	**Separation**	**Closure**	**Reward**	**Reward (O+)**	**Reward (S+)**
#	*story*	*feet*	*feet/min*	*Default*	*Proposed#1*	*Proposed#2*
5	Corrective advisory	> 650	< 2000	-1×10^{-1}	-1.5×10^{-1}	-1.5×10^{-1}
6	Corrective advisory	> 1000	< 4000	-3×10^{-2}	-1.5×10^{-2}	-1.5×10^{-2}
7	Preventive advisory	> 650	< 2000	-1×10^{-2}	-1.5×10^{-2}	-1.5×10^{-2}
9	Reversal			-8×10^{-3}	-8×10^{-3}	-12×10^{-3}
10	Strengthening			-5×10^{-3}	-2.5×10^{-3}	-2.5×10^{-3}
11	Weakening			-1×10^{-3}	-1.5×10^{-3}	-1.5×10^{-3}

These reward points, which were determined by inspecting the encounter scenarios that have 17 second left to encounter, were analyzed under two groups, for all possible simulation scenarios between 6 and 39 seconds, and scenario between 17 and 39 seconds. While TAU

6-39 provides general projection including stress scenarios, TAU 17-39 focuses on scenarios that have resulted with less number of collisions.

As it can be seen from the Table 4.3, it has been observed that higher level of safety and less advisory are produced with proposed reward points. At the projection of TAU 6-39, number of generated advisory is reduced by 11035 without causing further collisions with *operational plus* reward values. With *safety plus* reward values 24 less collisions occurred and 26723 less advisory generated.

TABLE 4.3: Effect of Proposed Rewards

	Reward	**Reward (O+)**	**Reward (S+)**
	$Default$	$Proposed\#1$	$Proposed\#2$
Number of Advisory Generated (TAU: 6-39)	4,825,709	4,814,674	4,798,986
Number of Advisory Generated (TAU: 17-39)	3,432,513	3,418,554	3,410,136
Number of Missed Collision (TAU: 6-39)	35,610	35,584	35,586
Number of Missed Collision (TAU: 17-39)	16	16	10

Although the proposed reward values provide higher level of safety and operational advantage, it has been observed that the ratio between generated and applied advisory regressed from 0.562000527 to 0.526221998.

5. RESULTS & ANALYSIS

ACAS X must accommodate many operational goals and constraints while meeting the established safety requirements. It is important that the system provide effective collision protection without unnecessarily disrupting pilots and the air traffic control system. In addition to producing as few alerts as possible, it must issue advisories that resolve encounters in a manner deemed suitable and acceptable by pilots and the operational community. This section discusses the process of safety and operational performance analysis, tuning of the logic, and flight tests of an ACAS X prototype.

Since we can neither predict nor control what we cannot measure, we have to make our problem measurable first. Since outcomes of this study are going to be used for preventing the collision of the planes, we have decided to inspect two different metrics. These metrics has been examined under two separate sections, these are ”Safety Analysis” and ”Operational Efficiency”.

Of course there are more subjects to examine. For example, Pilot Acceptability might be another subject to analyze but that would not be end up with an objective conclusion. As Fenton said ”Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.” [32] Since there will be some ambiguity to define pilot acceptability, it would not be a feasible metric.

5.1. Safety Analysis

In studies evaluating how safe the system is, the fundamental safety metric is expressed as *collision risk ratio*. This ratio is expressed as number of collisions encountered when collision avoidance system is active divided to the number of collisions encountered when collision avoidance system is deactivated [33]. Therefore, interpretation of collision can change this ratio. How to define the encounter situation, which is expressed as collision, can change this ratio. In similar studies, the collision situation of the aircraft was mathematically determined as the separation distance between the intruder and the own aircraft to be 500 ft horizontally and 100 ft vertically, and this cylindrical area created around the own aircraft was expressed as Near Mid-Air Collision (NMAC) [15].

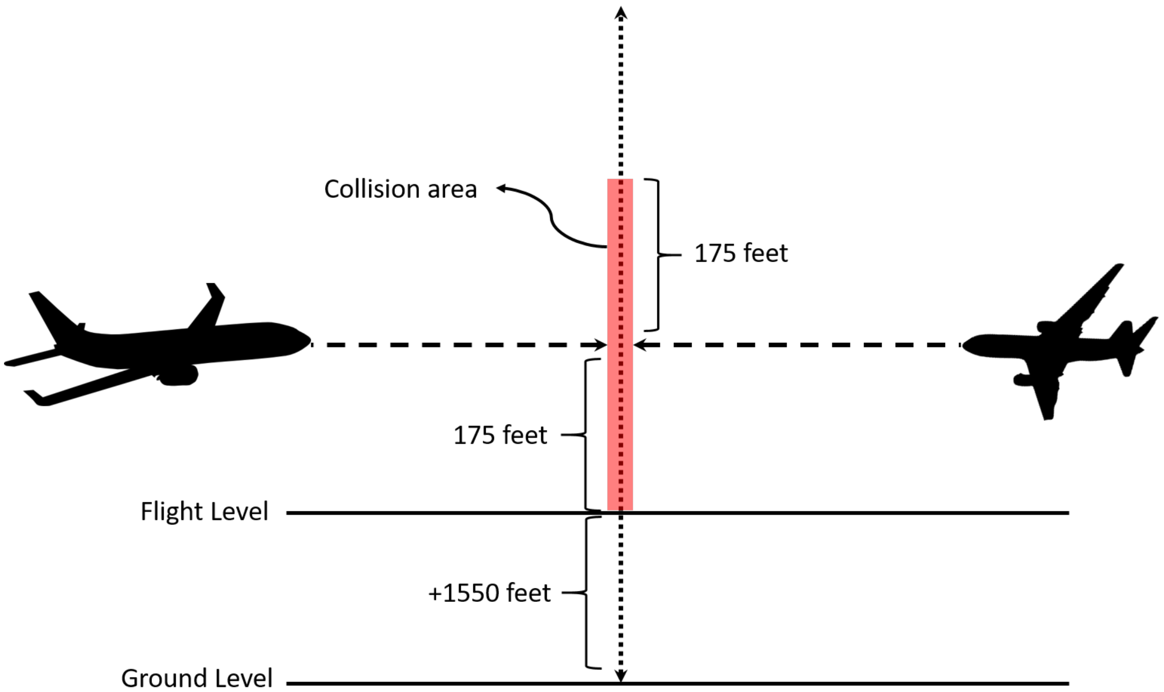


FIGURE 5.1: Critical amount of vertical separation

The NMAC range accepted in the study conducted here was accepted as 175 ft on the vertical axis and depicted in Figure 5.1. This value was kept higher than the similar studies, considering the large aircraft that are possible to be encountered.

A state space composed from possible encounter scenarios. Some of these encounters end up with a collision and some are not. The scenarios resulting with a collision in this space are identified. In the study conducted here, we have 8405 encounter scenarios that are resulting with a collision. It was analyzed how much of the existing collisions could be prevented by activating the collision avoidance system.

Number of prevented collisions differs according to the time left to collision (TAU). Scenarios with less than 5 seconds to collision have not been examined, since the pilot's response to the avoidance maneuvers produced can occur with a delay of 5 seconds. On the other hand, it would not be very meaningful to comment on scenarios with more than 40 seconds to collision. Detailed information on the selection of these duration can be found in Section 3.4..

There is an inverse proportion between TAU and number of collision. Performance is decreasing when the time left to collision is low. Because there are fewer options in terms of vertical speed and separation to prevent collision in maneuvers. Therefore, it can be observed that the number of collisions is less in cases where the time remaining to collision is more or close to 40 seconds. The results we obtained can be followed from the Figure 5.2 and a list of samples are given in Table 5.1 according to the seconds left to encounter. Results are gathered when transition between SDES2500 and SCL2500 is valid and crossing advisories are allowed. Each TAU value has been analyzed with 8405 possible collisions in the space. In the simulation studies in case of having 17 seconds to encounter, the level of safety is almost 100%. In cases where there are more seconds left to encounter, we have not observed a collision.

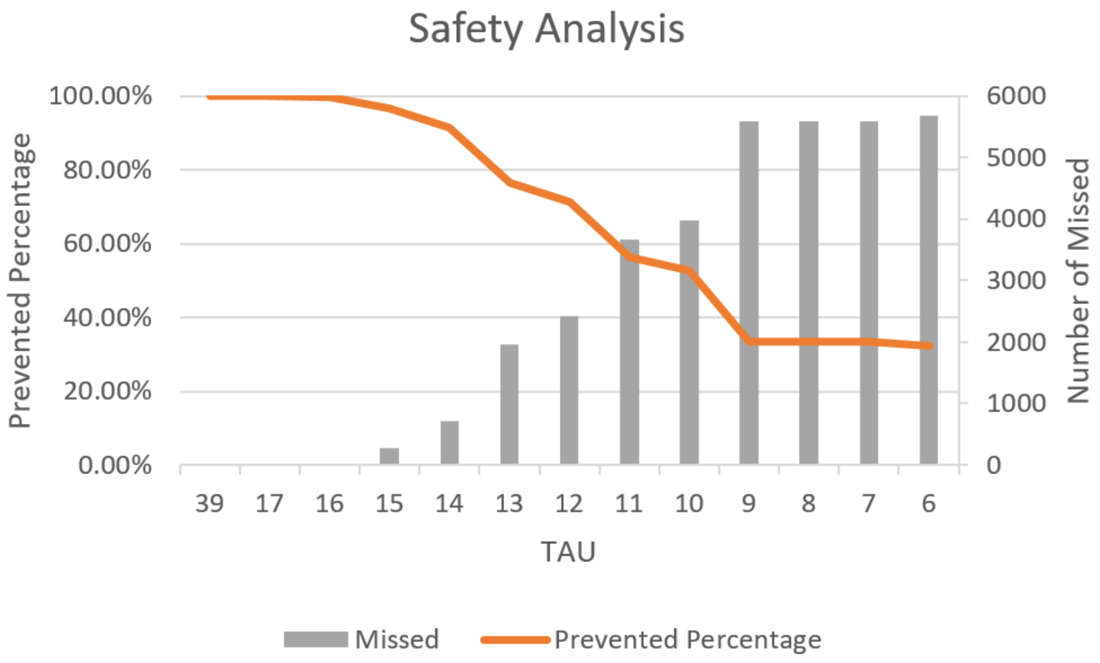


FIGURE 5.2: Safety Analysis

TABLE 5.1: Comparison of collisions

TAU $Seconds$	**Collision** $\%of prevented$	**Collision** $\#of misses$
6	32.4%	5681
17	99.9%	8
39	100.0%	0

All 8 collisions that occurred when the time left to encounter was 17 seconds was examined, and it was observed that the collisions that occurred could not be prevented by the current set of advisories with descending/climbing capabilities.

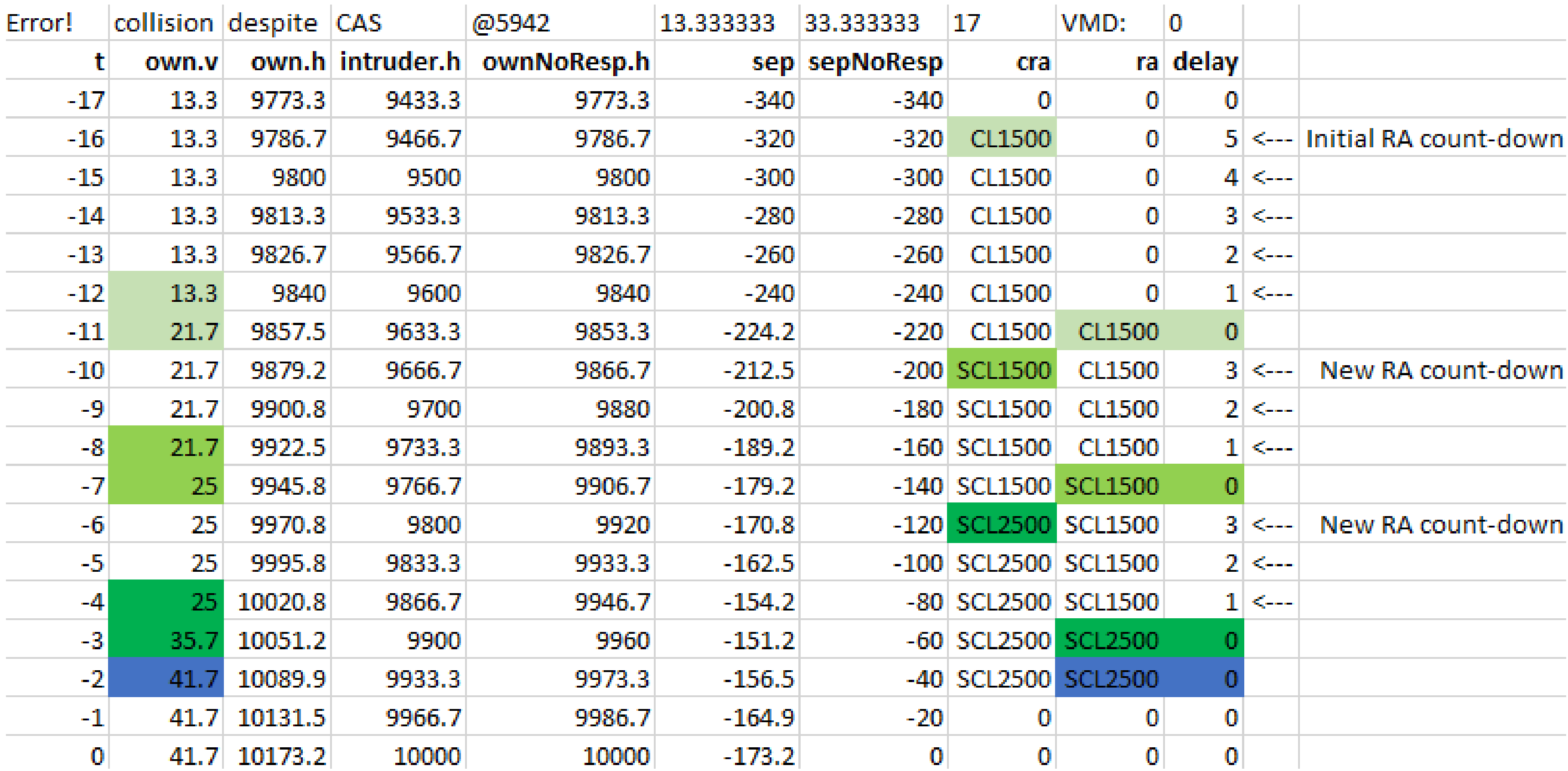
Two scenarios resulting in an absolute collision as a result of the simulation are given in the Figure 5.3 and Figure 5.4.

Columns of these tables are named as following. Explanation of the column titles are presented in the list below.

- ***t***: Remaining time to collision, a.k.a. TAU.
- ***own.v***: Own aircraft's relative vertical speed per second.
- ***own.h***: Own aircraft's altitude (AGL).
- ***intruder.h***: Intruder aircraft's altitude (AGL).
- ***ownNoResp.h***: Own aircraft altitude (AGL) position at TAU while collision avoidance logic is inactive.
- ***sep***: Relative altitude separation.
- ***sepNoResp***: Own aircraft relative altitude separation at TAU while collision avoidance logic is inactive.
- ***cra***: Currently selected resolution advisory at TAU. (This variable may not be seen in every simulation result figures. It is added to monitor flicker advisories.)
- ***ra***: Resolution advisory to be performed.
- ***delay***: Pilot response delay to perform the resolution advisory.

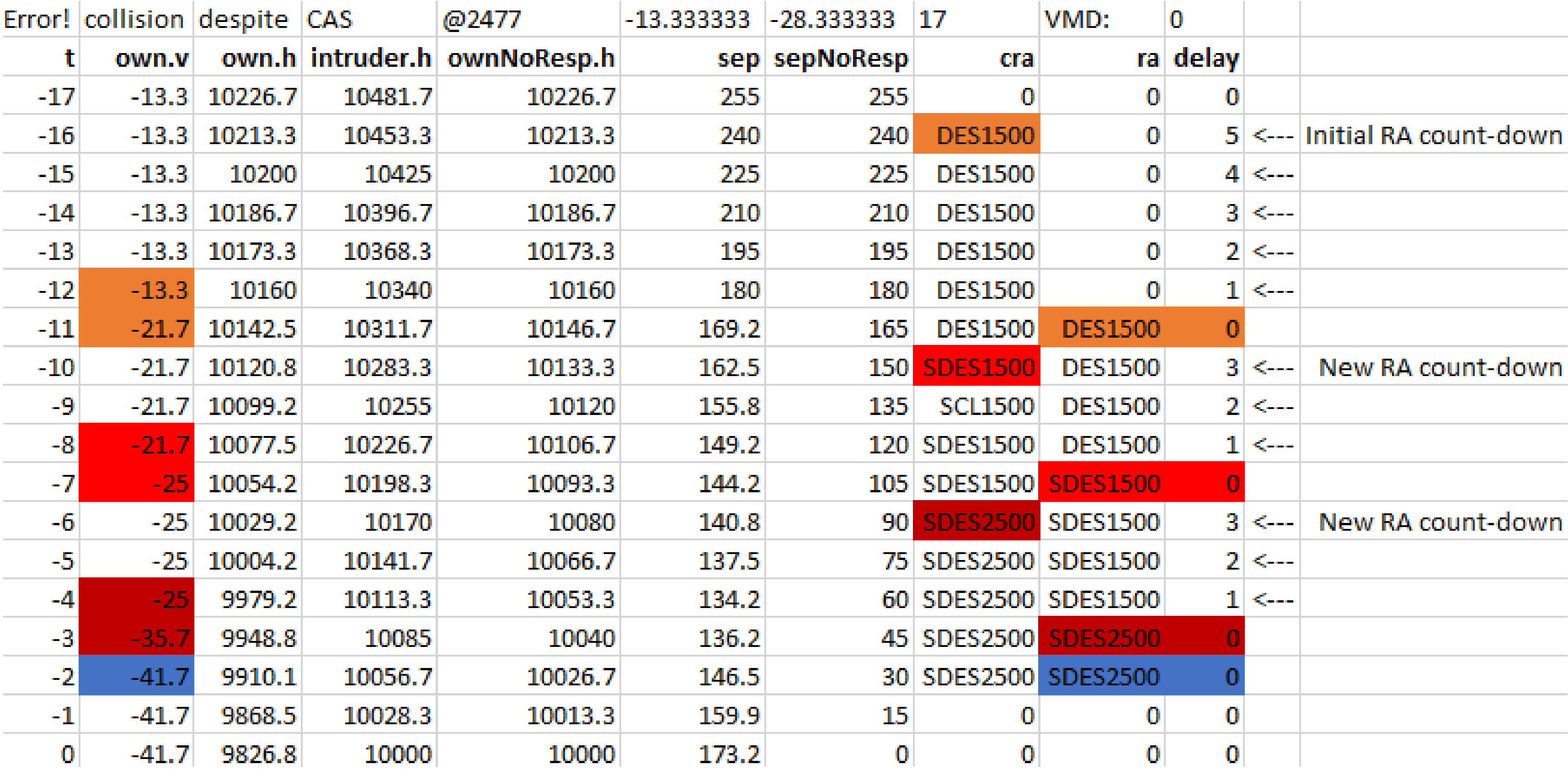
Simulation printouts are colored to increase readability. As seen in Figure 5.3, although the maneuver was started as quickly as possible with the least pilot response delay possible, the minimum required separation could not be achieved and the mid-air collision occurred. As can be seen, own aircraft reached the limit climb value as soon as possible. In Figure 5.4, a similar situation occurred between the descending aircraft. It can also be seen that the flicker advisory that occurred at 9 seconds before the collision was detected and did not cause additional delay.

Collision scenarios during the climbing at scenario number 5942 and scenario during descending at scenario number 2477 can be inspected from Figure 5.5 and Figure 5.6, respectively.



Error!	collision	despite	CAS	@5942	13.333333	33.333333	17	VMD:	0		
t	own.v	own.h	intruder.h	ownNoResp.h	sep	sepNoResp	cra	ra	delay		
-17	13.3	9773.3	9433.3	9773.3	-340	-340	0	0	0		
-16	13.3	9786.7	9466.7	9786.7	-320	-320	CL1500	0	5	<---	Initial RA count-down
-15	13.3	9800	9500	9800	-300	-300	CL1500	0	4	<---	
-14	13.3	9813.3	9533.3	9813.3	-280	-280	CL1500	0	3	<---	
-13	13.3	9826.7	9566.7	9826.7	-260	-260	CL1500	0	2	<---	
-12	13.3	9840	9600	9840	-240	-240	CL1500	0	1	<---	
-11	21.7	9857.5	9633.3	9853.3	-224.2	-220	CL1500	CL1500	0		
-10	21.7	9879.2	9666.7	9866.7	-212.5	-200	SCL1500	CL1500	3	<---	New RA count-down
-9	21.7	9900.8	9700	9880	-200.8	-180	SCL1500	CL1500	2	<---	
-8	21.7	9922.5	9733.3	9893.3	-189.2	-160	SCL1500	CL1500	1	<---	
-7	25	9945.8	9766.7	9906.7	-179.2	-140	SCL1500	SCL1500	0		
-6	25	9970.8	9800	9920	-170.8	-120	SCL2500	SCL1500	3	<---	New RA count-down
-5	25	9995.8	9833.3	9933.3	-162.5	-100	SCL2500	SCL1500	2	<---	
-4	25	10020.8	9866.7	9946.7	-154.2	-80	SCL2500	SCL1500	1	<---	
-3	35.7	10051.2	9900	9960	-151.2	-60	SCL2500	SCL2500	0		
-2	41.7	10089.9	9933.3	9973.3	-156.5	-40	SCL2500	SCL2500	0		
-1	41.7	10131.5	9966.7	9986.7	-164.9	-20	0	0	0		
0	41.7	10173.2	10000	10000	-173.2	0	0	0	0		

FIGURE 5.3: Collision No. 5942 Details



Error!	collision	despite	CAS	@2477	-13.333333	-28.333333	17	VMD:	0		
t	own.v	own.h	intruder.h	ownNoResp.h	sep	sepNoResp	cra	ra	delay		
-17	-13.3	10226.7	10481.7	10226.7	255	255	0	0	0		
-16	-13.3	10213.3	10453.3	10213.3	240	240	DES1500	0	5	<---	Initial RA count-down
-15	-13.3	10200	10425	10200	225	225	DES1500	0	4	<---	
-14	-13.3	10186.7	10396.7	10186.7	210	210	DES1500	0	3	<---	
-13	-13.3	10173.3	10368.3	10173.3	195	195	DES1500	0	2	<---	
-12	-13.3	10160	10340	10160	180	180	DES1500	0	1	<---	
-11	-21.7	10142.5	10311.7	10146.7	169.2	165	DES1500	DES1500	0		
-10	-21.7	10120.8	10283.3	10133.3	162.5	150	SDES1500	DES1500	3	<---	New RA count-down
-9	-21.7	10099.2	10255	10120	155.8	135	SCL1500	DES1500	2	<---	
-8	-21.7	10077.5	10226.7	10106.7	149.2	120	SDES1500	DES1500	1	<---	
-7	-25	10054.2	10198.3	10093.3	144.2	105	SDES1500	SDES1500	0		
-6	-25	10029.2	10170	10080	140.8	90	SDES2500	SDES1500	3	<---	New RA count-down
-5	-25	10004.2	10141.7	10066.7	137.5	75	SDES2500	SDES1500	2	<---	
-4	-25	9979.2	10113.3	10053.3	134.2	60	SDES2500	SDES1500	1	<---	
-3	-35.7	9948.8	10085	10040	136.2	45	SDES2500	SDES2500	0		
-2	-41.7	9910.1	10056.7	10026.7	146.5	30	SDES2500	SDES2500	0		
-1	-41.7	9868.5	10028.3	10013.3	159.9	15	0	0	0		
0	-41.7	9826.8	10000	10000	173.2	0	0	0	0		

FIGURE 5.4: Collision No. 2477 Details

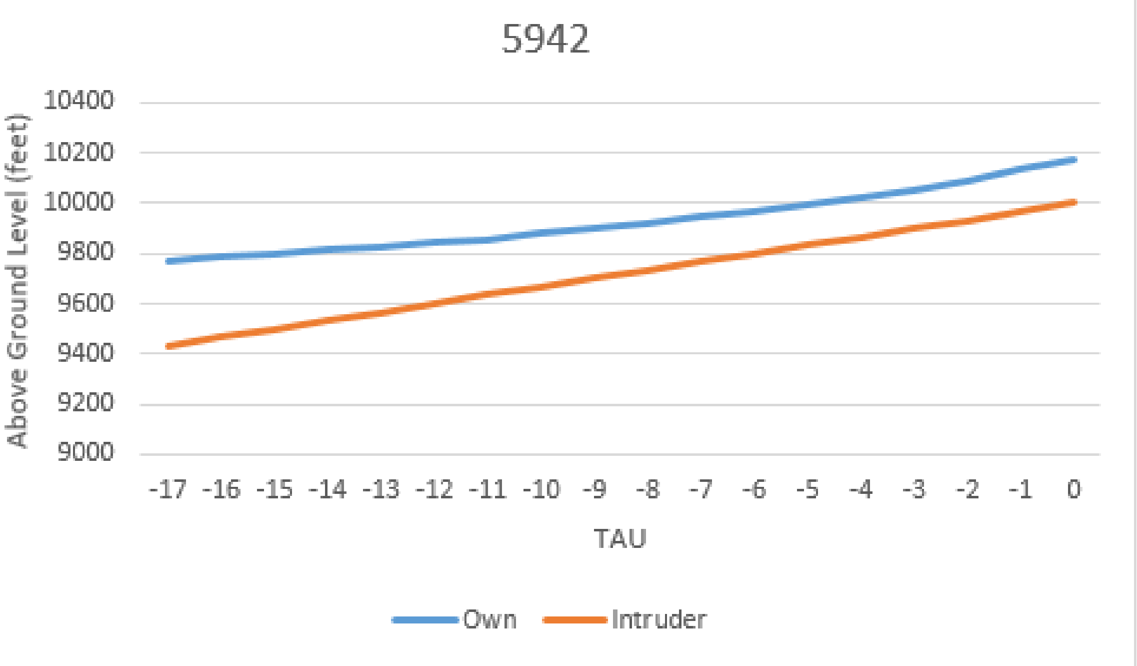


FIGURE 5.5: Collision No. 5942 while Climbing

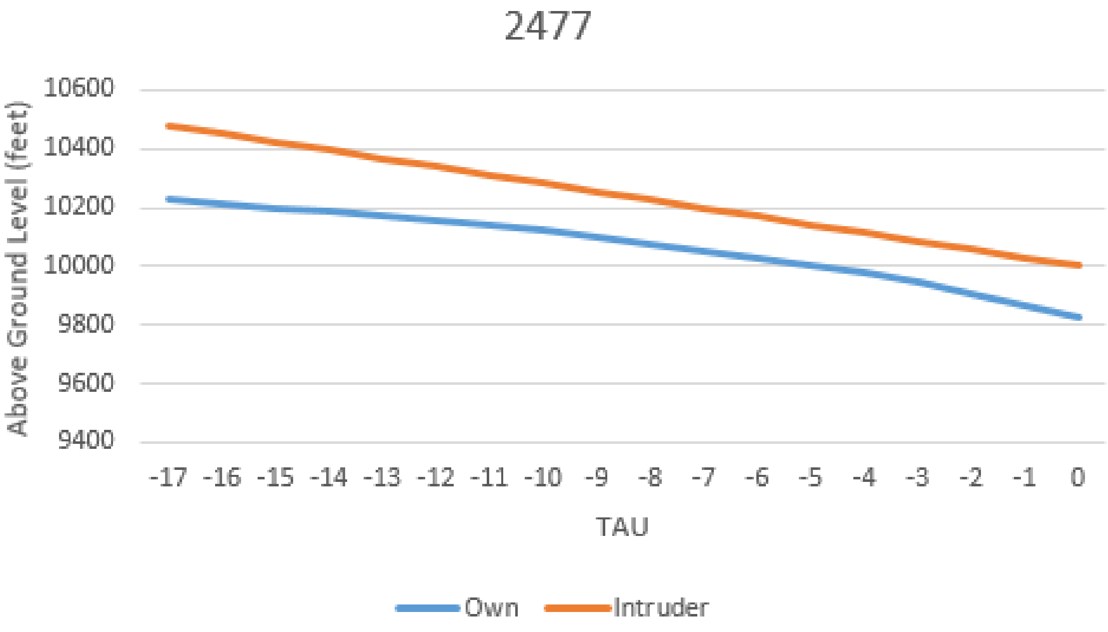


FIGURE 5.6: Collision No. 2477 while Descending

Of course there are some superior manned and unmanned aircraft that can perform better climb or descent rates that may prevent collision. For example, a Global Hawk UAV can perform a 3500 feet per minute climb or 4000 feet per minute descent rate [34–36]. Similarly, a Boeing 737 Max 7 which is widely used in civil aviation can perform 3000 feet per minute as maximum climb rate but most of the aircraft can not perform these climb and descent rates. With the light of this, having more collisions in scenarios that have less time to collision are interpreted as an expected result. Therefore, it would not be possible to prevent 100% of the collisions with 17 seconds or less TAU.

A couple of inconvenient transactions were proposed in Table 3.1. According to that, transactions between *Strong Climb 2500 ft/min* and *Strong Descend 2500 ft/min* maneuvers proposed as inconvenient. Results of analyses are given in the Table 5.2. There are 285770 possible collisions in this space.

TABLE 5.2: Comparison of Proposed Invalid Transition Map

Transition	**TAU**	**Collision**	**Collision**	**Advisories**
$Between2500feet/min$	$Seconds$	$\%ofprevented$	$\#ofmisses$	$\#ofgenerated$
Invalid	6-39	87.5%	35,608	4,830,063
Valid	6-39	87.5%	35,610	4,825,709

As can be seen, inhibition of the transition between these two maneuvers that are performed with $0.\overline{33}$ G does not improve the level of safety. Also, it was not very effective and did not contribute to the operational effectiveness.

With the values given in the Table 5.2, it can be observed that the operational performance has slightly decreased, but level of safety has almost not changed. It is surprising that; penalizing the transition between these strong maneuvers did not decrease level of safety in scenarios where the TAU was small. The scenarios in which the TAU is 6, 10, 17 and 39 seconds were examined separately, and no different situation was observed other than the reflected in the table.

On the other hand, another parameter that can be changed to prevent existing collisions may be related to the pilot delays that are explained in Section 3.4.. Effect of pilot response delay on the success of the system is presented in Section 5.2..

5.2. Operational Suitability and Acceptability

Increasing the level of safety as a result of the work done will not make sense alone. In order to be able to claim an improvement, operational efficiency and level of safety should be considered together. There is a trade-offs between these two topics. Of course it is possible to improve the level of safety but while trying to improve the safety, operational performance might be drastically decreased because of too many alerts generated.

In scope of operational efficiency, frequency of the alerts generated in situations that do not involving exposure to danger is determined as the base metric. If the pilot is exposed to too many warnings that he thinks are unnecessary, he may start to ignore the warnings.

Also, there might be some maneuvers that pilots are hesitant to do or avoid. These type of maneuvers are also effect the operational performance. Although we do not analyze the pilot acceptability, pilots are absolute parts of collision avoidance systems.

In Figure 5.7, the gray columns on the bar chart show how many advisory were produced for 8405 possible collisions in the respective TAU, the yellow colored parts of the columns show how many of these advisory were applied. Finally, the blue line shows the ratio of the applied advisories to the produced advisories.

As can be seen from the Figure 5.7, as the time left to the collision decreases, the ratio of applied advisories decreases. Scenarios where TAU is relatively small can also be called as surprise advisory generated scenarios. Some of these scenarios, where TAU is 6 seconds, may not be very realistic. Most of these scenarios duration will be passed with a delay of 5 seconds before the pilot begins to execute the maneuver. Maneuver can be applied for the remaining one second. We can explain the decrease in the rate of advisory applied by considering the generated advisory may also change in the 5-second period until the pilot starts to apply the maneuver. In case of a single change in the advisory, scenario will be ended without applying any advisory. Based on these results, the most important issue affecting the performance of the system can be considered as the delay of the pilot.

In order to observe the effect of delays caused by the pilot on the success of the system, trials have been made with different amount of delays and the results are presented. Advisory response probabilities that are given in Table 3.2 are recalculated with different amount of

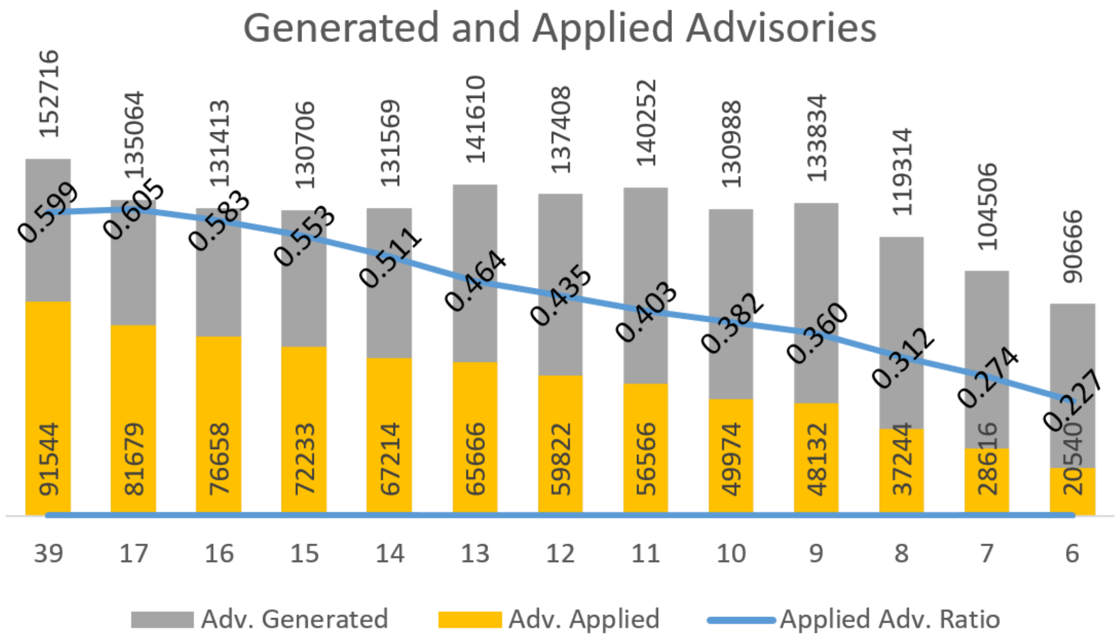


FIGURE 5.7: Operational Analysis

TABLE 5.3: Effect of Pilot Delays

Initial Delay *Seconds*	**Reversal Delay** *Seconds*	**Regular Delay** *Seconds*	**Collision** *prevented*	**Collision** *missed*	**Advisories** *generated*	**Applied Advisory Ratio** *Applied/Generated*
0	0	0	99.10%	2,623	2,584,745	1.00
5	3	3	87.50%	35,620	4,836,075	0.56246088
5	5	3	87.50%	35,610	4,825,709	0.562000527

delays which are given in Table 5.3. As can be seen, even if there is no pilot delay, the performance did not exceed 99.1% with the current aircraft capabilities.

The results we obtained can be followed from the Table 5.4. In default reward function reversal type transitions between *Strong Climb 2500* and *Strong Descend 2500* are allowed and range of TAU values are examined together. There are 285770 possible collisions in this space. On the other hand, in alternative reward function we have encourage to perform any crossing advisories to inspect effect of them.

When the system wants to terminate a maneuver due to force majeure and initiate an opposite maneuver, the pilot may be concerned about this situation. Therefore, it is desired that such

TABLE 5.4: Difference between Default and Alternative Reward Functions

Function $Type$	**TAU** $Seconds$	**Collision** $\%of prevented$	**Collision** $\#of misses$	**Advisories** $\#of generated$	**Advisories** $\#of applied$
Alternative	6-39	88.0%	34,384	5,163,855	3,049,753
Default	6-39	87.5%	35,610	4,825,709	2,712,051

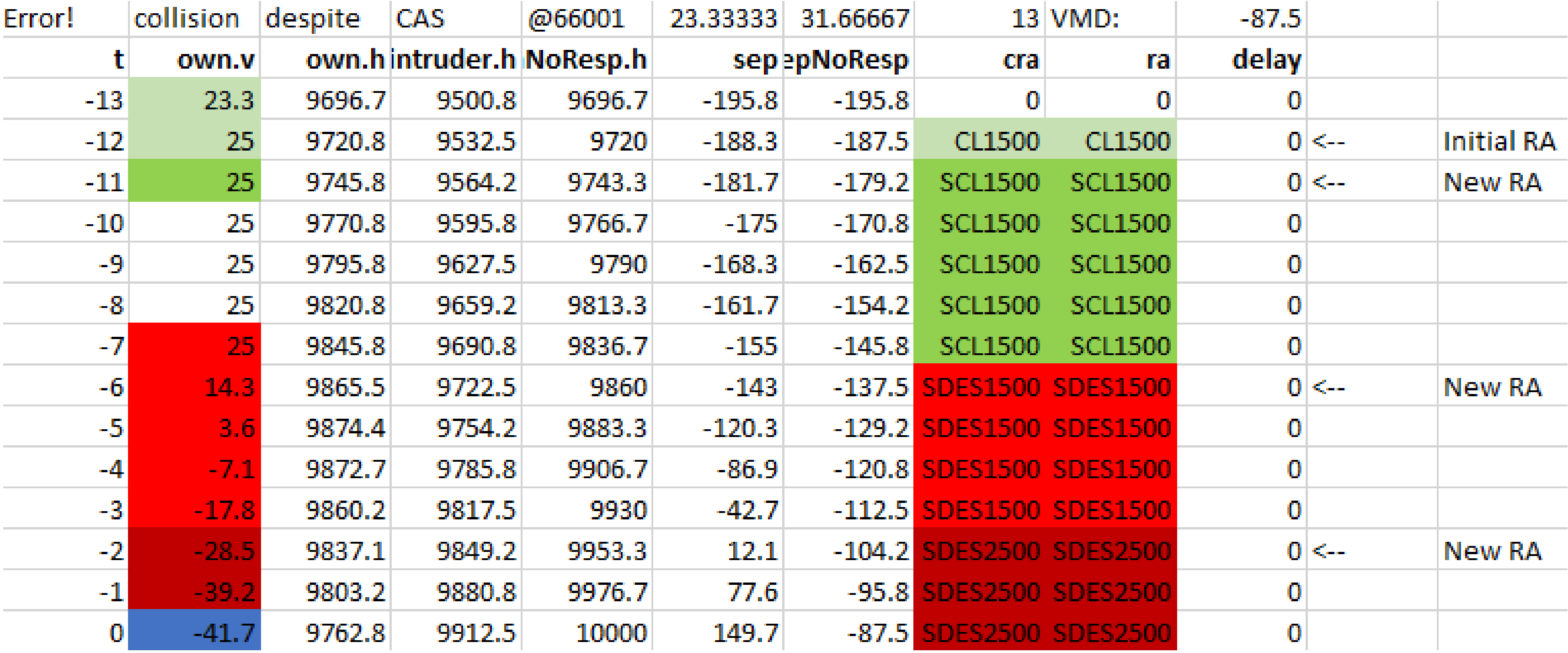
reversals occur as little as possible. Similarly, crossing advisories should only be produced when there is not a better possible maneuvers to perform [30].

Two scenarios resulting with collision as a result of the simulation are given in the Figure 5.8 and Figure 5.9. Simulation printouts are colored to increase readability. Pilot response delays are ignored in these two scenarios. In these scenarios both own aircraft and intruder have a climb trend. Details of scenario number 66001 can be seen from Figure 5.8. Although it was recommended to accelerate the climb with the advisory of *Climb 1500*, the relative speed did not increase enough to provide the separation. Six seconds before the collision, own aircraft was recommended with a reversal advisory to end his climb and start to descend. As can be seen from the details, it continued its acceleration until own aircraft reached the limit value of $41.7 feet/sec$ which is equal to $2500 feet/min$. Own aircraft performed the crossing maneuver, which is not preferred by pilots, except in compulsory situations. Despite all this, it could not provide the required separation and ended with a collision.

Details of scenario number 25135 can be seen from Figure 5.9. In this scenario, own aircraft is below the intruder in terms of altitude. Own aircraft first tries to descend in a way that does not allow crossing as the altitude, but since it cannot reduce the relative speed to sufficient level in the required time, it changes the direction of maneuver action with a reversal advisory and decides to crossing.

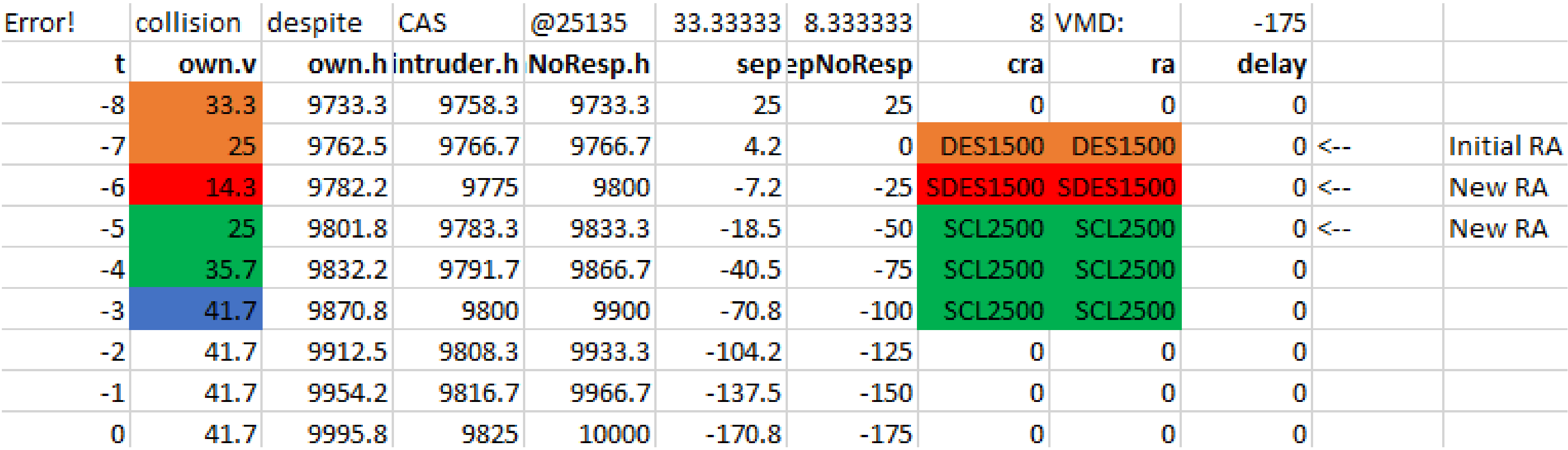
Projection of these collision scenarios can be inspected from Figure 5.10 and Figure 5.11. As can be seen from this figure, crossing type advisories are quite stressful advisories for the pilots.

Although pilot may concern about crossings, as can be seen from the Table 5.4, encouraging crossing advisories is helpful in preventing some collisions, but causes more advisory which reduce operational efficiency. It would not make sense to completely prevent crossing advisories. It is reasonable to keep crossing advisories in the system with some penalty points.



Error!	collision	despite	CAS	@66001	23.33333	31.66667	13	VMD:	-87.5		
t	**own.v**	**own.h**	**intruder.h**	**NoResp.h**	**sep**	**epNoResp**	**cra**	**ra**	**delay**		
-13	23.3	9696.7	9500.8	9696.7	-195.8	-195.8	0	0	0		
-12	25	9720.8	9532.5	9720	-188.3	-187.5	CL1500	CL1500	0	<--	Initial RA
-11	25	9745.8	9564.2	9743.3	-181.7	-179.2	SCL1500	SCL1500	0	<--	New RA
-10	25	9770.8	9595.8	9766.7	-175	-170.8	SCL1500	SCL1500	0		
-9	25	9795.8	9627.5	9790	-168.3	-162.5	SCL1500	SCL1500	0		
-8	25	9820.8	9659.2	9813.3	-161.7	-154.2	SCL1500	SCL1500	0		
-7	25	9845.8	9690.8	9836.7	-155	-145.8	SCL1500	SCL1500	0		
-6	14.3	9865.5	9722.5	9860	-143	-137.5	SDES1500	SDES1500	0	<--	New RA
-5	3.6	9874.4	9754.2	9883.3	-120.3	-129.2	SDES1500	SDES1500	0		
-4	-7.1	9872.7	9785.8	9906.7	-86.9	-120.8	SDES1500	SDES1500	0		
-3	-17.8	9860.2	9817.5	9930	-42.7	-112.5	SDES1500	SDES1500	0		
-2	-28.5	9837.1	9849.2	9953.3	12.1	-104.2	SDES2500	SDES2500	0	<--	New RA
-1	-39.2	9803.2	9880.8	9976.7	77.6	-95.8	SDES2500	SDES2500	0		
0	-41.7	9762.8	9912.5	10000	149.7	-87.5	SDES2500	SDES2500	0		

FIGURE 5.8: Collision No. 66001 Details



Error!	collision	despite	CAS	@25135	33.33333	8.333333	8	VMD:	-175		
t	**own.v**	**own.h**	**intruder.h**	**NoResp.h**	**sep**	**epNoResp**	**cra**	**ra**	**delay**		
-8	33.3	9733.3	9758.3	9733.3	25	25	0	0	0		
-7	25	9762.5	9766.7	9766.7	4.2	0	DES1500	DES1500	0	<--	Initial RA
-6	14.3	9782.2	9775	9800	-7.2	-25	SDES1500	SDES1500	0	<--	New RA
-5	25	9801.8	9783.3	9833.3	-18.5	-50	SCL2500	SCL2500	0	<--	New RA
-4	35.7	9832.2	9791.7	9866.7	-40.5	-75	SCL2500	SCL2500	0		
-3	41.7	9870.8	9800	9900	-70.8	-100	SCL2500	SCL2500	0		
-2	41.7	9912.5	9808.3	9933.3	-104.2	-125	0	0	0		
-1	41.7	9954.2	9816.7	9966.7	-137.5	-150	0	0	0		
0	41.7	9995.8	9825	10000	-170.8	-175	0	0	0		

FIGURE 5.9: Collision No. 25135 Details

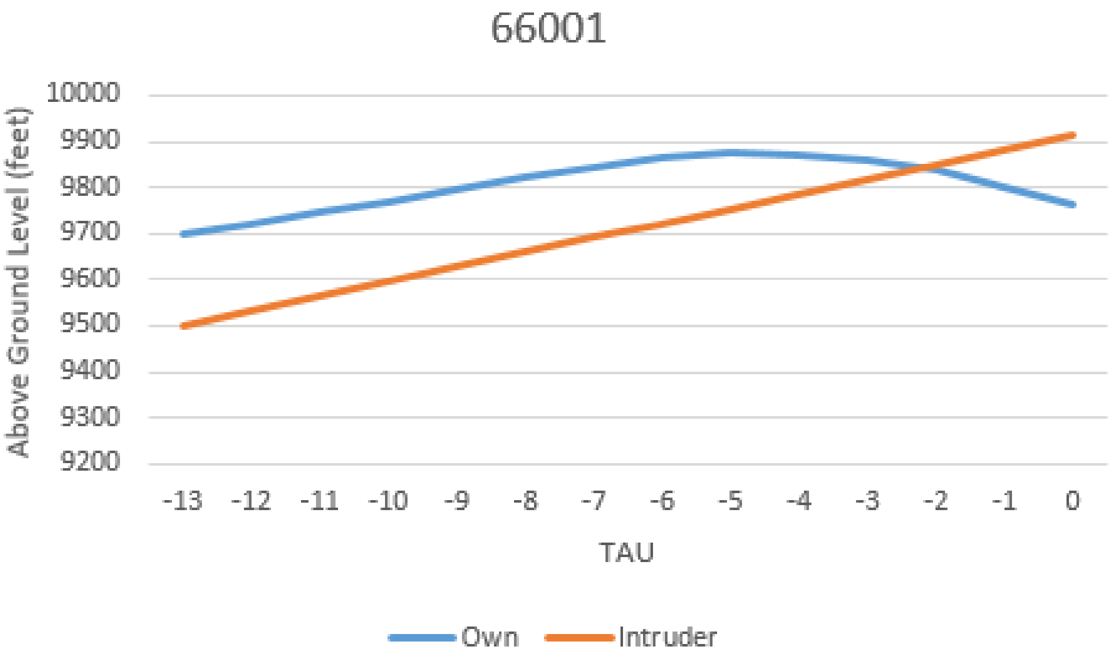


FIGURE 5.10: Collision No. 66001 with crossing while above intruder

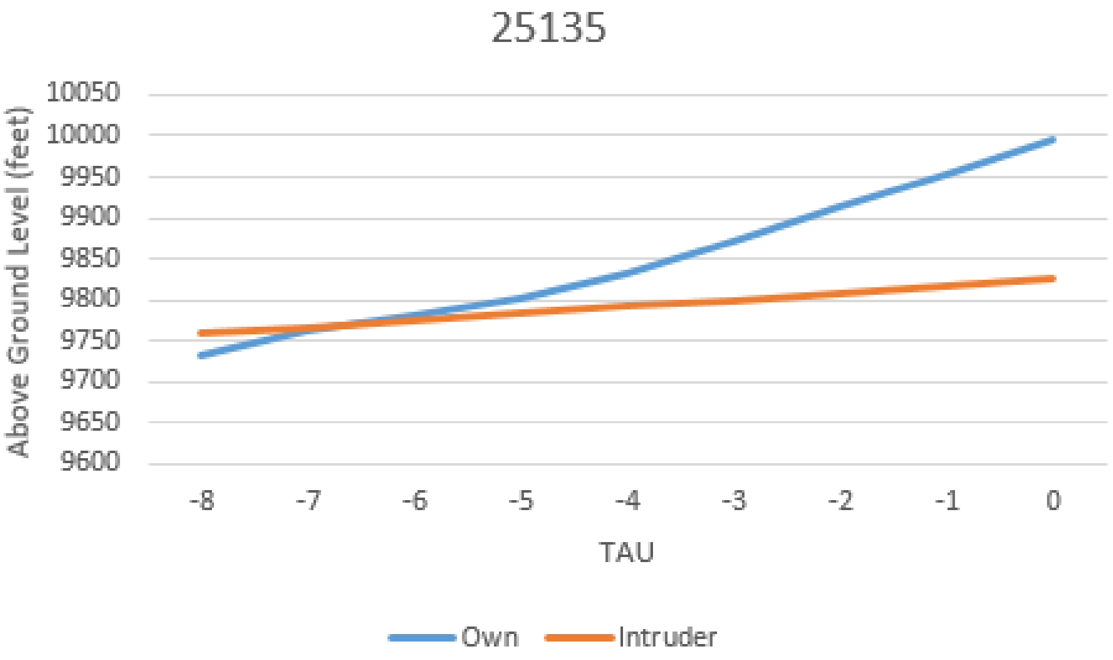


FIGURE 5.11: Collision No. 25135 with crossing while below intruder

By considering two scenarios in which the performance is 100% in terms of Level of Safety, studies can be carried out for a more effective model in operational terms. Also, there will be another study might be performed by using TSIM scenarios, instead of synthetic test data. Since it is very important to validate the performance of collision avoidance system, a further study should be conducted by using RTCA's TSIM [37]. TSIM scenarios have also multiple encounter scenarios and are used to validate parts of TCAS studies. Additionally, it may be possible to create action plots to see which avoidance maneuvers will be produced in a specific point in that state space. With the help of those encounter scenarios, it would be possible to make analyzes on those plot graphs.

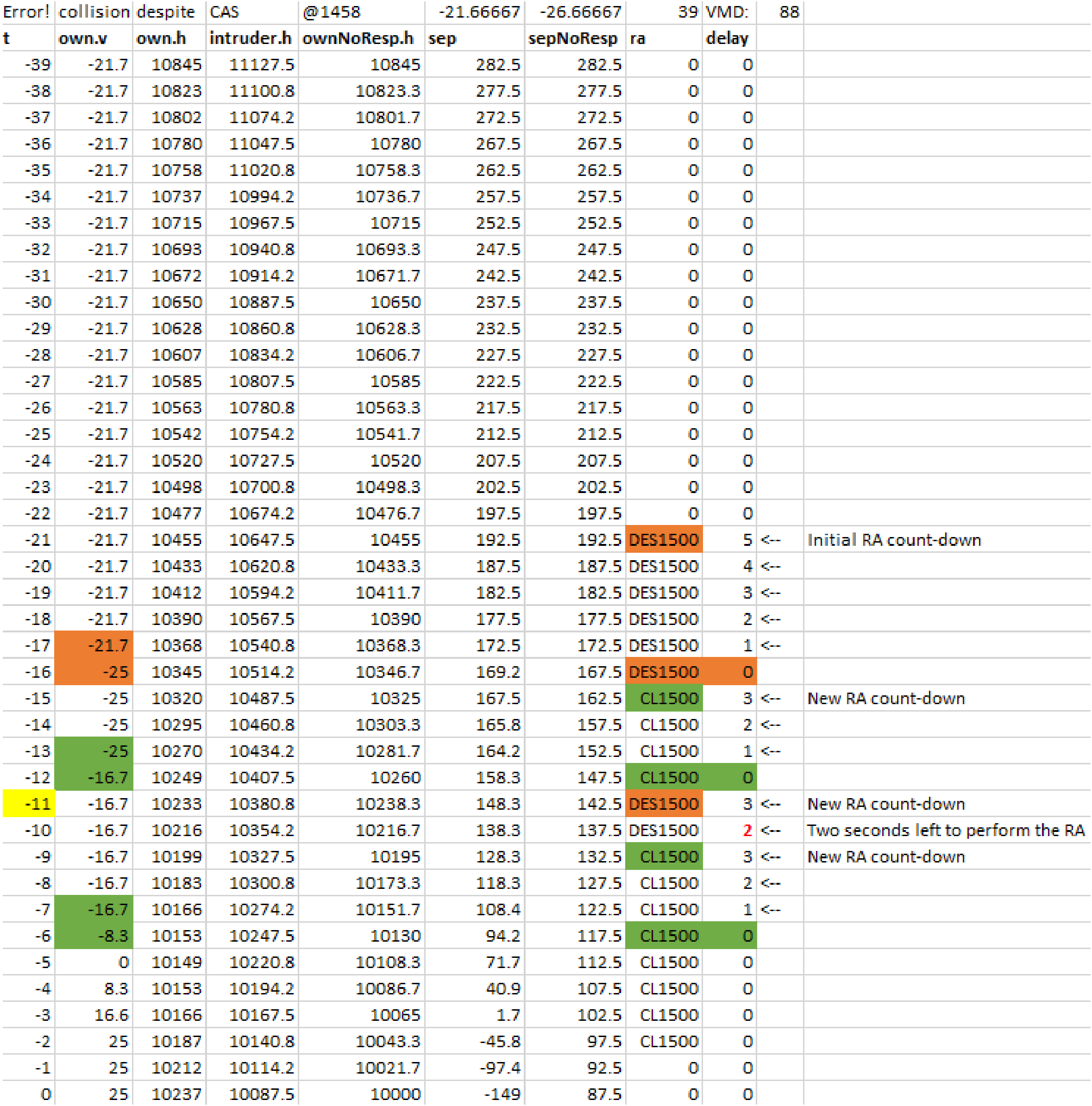
5.3. An Example Run

In the early version of the simulation we developed, it was concluded that some encounter scenarios ending with a collision could actually be prevented. As explained in Section 3.4., the update of the current maneuver with the newly generated advisory takes place with a 3-5 second delay depending on the current situation and the type of generated advisory. This delay is implemented by setting the count-down timer to the new value.

If we express this situation with an example which is shown in Figure 5.12 of the analyzed collision scenarios, the *Climb 1500*, which is applied when the time remaining to the collision is 11 seconds, is updated to the *Descend 1500*. The new maneuver should not be applied within the next 3 seconds. The current maneuver should not change until the count-down timer is zero. The latest *Climb 1500* maneuver is recommended again, without the count-down timer of the *Descend 1500* maneuver yet to be zero.

Similar situations occurring within 1-2 seconds, which can also be observed between different maneuvers, were expressed as *flicker advisory*.

We set the count-down timer to wait 3 seconds again every second that an advisory flickers. In this way, we were facing too many collision scenarios. As a matter of fact, returning to a maneuver that we did not fully change was subject to repeated 3-second count-downs, and it could take a lot of time to perform an avoidance maneuver. At this point, we can say that simulation and reality differ from each other.



Error!	collision	despite	CAS	@1458	-21.66667	-26.66667	39	VMD:	88	
t	own.v	own.h	intruder.h	ownNoResp.h	sep	sepNoResp	ra	delay		
-39	-21.7	10845	11127.5	10845	282.5	282.5	0	0		
-38	-21.7	10823	11100.8	10823.3	277.5	277.5	0	0		
-37	-21.7	10802	11074.2	10801.7	272.5	272.5	0	0		
-36	-21.7	10780	11047.5	10780	267.5	267.5	0	0		
-35	-21.7	10758	11020.8	10758.3	262.5	262.5	0	0		
-34	-21.7	10737	10994.2	10736.7	257.5	257.5	0	0		
-33	-21.7	10715	10967.5	10715	252.5	252.5	0	0		
-32	-21.7	10693	10940.8	10693.3	247.5	247.5	0	0		
-31	-21.7	10672	10914.2	10671.7	242.5	242.5	0	0		
-30	-21.7	10650	10887.5	10650	237.5	237.5	0	0		
-29	-21.7	10628	10860.8	10628.3	232.5	232.5	0	0		
-28	-21.7	10607	10834.2	10606.7	227.5	227.5	0	0		
-27	-21.7	10585	10807.5	10585	222.5	222.5	0	0		
-26	-21.7	10563	10780.8	10563.3	217.5	217.5	0	0		
-25	-21.7	10542	10754.2	10541.7	212.5	212.5	0	0		
-24	-21.7	10520	10727.5	10520	207.5	207.5	0	0		
-23	-21.7	10498	10700.8	10498.3	202.5	202.5	0	0		
-22	-21.7	10477	10674.2	10476.7	197.5	197.5	0	0		
-21	-21.7	10455	10647.5	10455	192.5	192.5	DES1500	5	<--	Initial RA count-down
-20	-21.7	10433	10620.8	10433.3	187.5	187.5	DES1500	4	<--	
-19	-21.7	10412	10594.2	10411.7	182.5	182.5	DES1500	3	<--	
-18	-21.7	10390	10567.5	10390	177.5	177.5	DES1500	2	<--	
-17	-21.7	10368	10540.8	10368.3	172.5	172.5	DES1500	1	<--	
-16	-25	10345	10514.2	10346.7	169.2	167.5	DES1500	0		
-15	-25	10320	10487.5	10325	167.5	162.5	CL1500	3	<--	New RA count-down
-14	-25	10295	10460.8	10303.3	165.8	157.5	CL1500	2	<--	
-13	-25	10270	10434.2	10281.7	164.2	152.5	CL1500	1	<--	
-12	-16.7	10249	10407.5	10260	158.3	147.5	CL1500	0		
-11	-16.7	10233	10380.8	10238.3	148.3	142.5	DES1500	3	<--	New RA count-down
-10	-16.7	10216	10354.2	10216.7	138.3	137.5	DES1500	2	<--	Two seconds left to perform the RA
-9	-16.7	10199	10327.5	10195	128.3	132.5	CL1500	3	<--	New RA count-down
-8	-16.7	10183	10300.8	10173.3	118.3	127.5	CL1500	2	<--	
-7	-16.7	10166	10274.2	10151.7	108.4	122.5	CL1500	1	<--	
-6	-8.3	10153	10247.5	10130	94.2	117.5	CL1500	0		
-5	0	10149	10220.8	10108.3	71.7	112.5	CL1500	0		
-4	8.3	10153	10194.2	10086.7	40.9	107.5	CL1500	0		
-3	16.6	10166	10167.5	10065	1.7	102.5	CL1500	0		
-2	25	10187	10140.8	10043.3	-45.8	97.5	CL1500	0		
-1	25	10212	10114.2	10021.7	-97.4	92.5	0	0		
0	25	10237	10087.5	10000	-149	87.5	0	0		

FIGURE 5.12: Flicker Advisory

We anticipate that these 3-5 seconds delays may occur for two reasons. First choice; Preparing the pilot for maneuver in order to maneuver the plane takes time with complex cockpit systems. The second option is; As the pilot tends to preserve the existing maneuver by human-nature, he is skeptical about the changed maneuver and it takes time to understand the situation. These delays are explained as pilot's guard is down due to human fatigue and having poor judgment and understanding about situational awareness in Pilot's Handbook [38].

In the example given above, the time from 11 seconds until the time left to collision to 6 seconds is a wasted time without applying an avoidance maneuver.

In the light of these findings, our simulation studies are performed by preventing delays due to the flicker advisories.

6. CONCLUSION

This study demonstrates the need for collision avoidance systems and provides guidance for improving advanced collision avoidance systems.

Since TCAS is mandatory in all large aircraft that are used for commercial purposes today, and it lays foundation of next generation collision avoidance systems, working principles of it and the limitations of the current system is presented. Resolution Advisories, the advisory output of collision avoidance systems for the vertical axis, is introduced.

ACAS X, which is the next generation collision avoidance system, is introduced and the operational difference between ACAS X and its predecessor, TCAS, in practice is shown.

Within the scope of this study, ACAS X simulation studies were deeply analyzed. While conducting our literature review, we tried to focus on articles written after 2012. Important studies conducted in this context are presented.

Unlike other studies all code implemented in C/C++ which is going to provide faster data synthesis. Additionally, with the model we offer, it has been made possible to test a wide variety of dynamics ranging from policy table used in data synthesis to pilot response. Our study reveals clear results on the impact of pilot behavior. Moreover, Maintain, one of the advisory types, was not included in our study because it is context dependent. On the other hand, our study includes Strong Descend and Strong Climb advisories. In our study it is possible to validate ACAS X in a reasonable amount of time without simplifying the state space of ACAS X. Our work is also beneficial for the detection of aircraft encounter scenarios that inevitably end up with collisions. Problems experienced in previous studies due to the Crossing Time, High Turn Rates and Initially Moving Against RA reasons are presented, the effects of these reasons on the model are examined.

It is explained how ACAS X, which provides better operational performance and flight safety, models the problem as a Partially Observable Markov Decision Process. Observations from an environment, beliefs updates, and policy operations performed within the POMDP cycle have been introduced.

Existing inconvenient maneuver transactions between preventive and corrective advisories have been introduced, and new inconvenient maneuver transactions have been propose for better operational performance and flight safety. These proposed maneuvers are tested and their results are examined.

State variables of the POMDP state space are introduced and the calculation for each variable is shown with pseudocodes.

Probability of responding an advisory that is going to be performed by a pilot is introduced mathematically and analyzed with different elapsed time until response values. In this way, the effects of the pilot on the system is examined.

In order to develop a memory friendly system with a good performance, ACAS X uses dynamic programming. There is no need to remember the entire state space, since the states are determined by only considering the data of a second before. Excessive memory consumption problem was solved with the single iteration Gauss-Seidel methodology. In scope of this, Bellman equation is explained and usage of these methods are presented with pseudocodes.

In this work the height and speed ranges can be chosen narrowly to keep the size of the decision table small and to shorten the run time of generation. Different resolutions can be selected according to the needs. In order to produce the decision table fast, all relative heights and speeds are homogeneously distributed.

On the other hand, the time axis is modeled with one second intervals. For the scenarios which are longer than 40 seconds, it may not be beneficial to consider the decision table. Therefore, the time axis is limited with 40 seconds.

Additionally, in order to identify possible maneuver that would provide the greatest benefits in terms of providing enough separation with the minimum cost, methodology of reward function and its values are deeply analyzed. It is stated that some absolute collisions can be prevented by crossing advisories. In order to see the effect of this, an alternative reward function where crossing advisories are not penalized is tested.

A number of empirical experiments have been carried out to find a better ratio to choose between operational efficiency and safety of flight.

In the study, 6 penalty points were determined as the basis to present a comparative analysis and all combinations are tested with a distribution that does not exceed half of the default reward points. Among the 729 results obtained, the reward values that have resulted with the least advisory generation without compromising the level of safety were selected and expressed as operational plus (O+). Additionally, reward points that generates the least advisory generation were named as safety plus (S+). And all possible simulation scenarios between 6 and 39 seconds, and scenario between 17 and 39 seconds were tested. Although the proposed reward values provide higher level of safety and operational advantage, it has been observed that the ratio between generated and applied advisory regressed.

Finally, examples of different maneuvers selected from our simulation study and flicker advisory and its solution, a problem we encountered in the early stages of our simulation study, are explained.

REFERENCES

[1] Mykel J Kochenderfer. *Decision making under uncertainty: theory and application*. MIT press, **2015**.

[2] EUROCONTROL. Airborne collision avoidance system guide. https://www.eurocontrol.int/publication/airborne-collision-avoidance-system-acas-guide, **2019**. Accessed: Nov 2, 2020.

[3] RTCA (Firm). SC 167. *Software considerations in airborne systems and equipment certification*. RTCA, Incorporated, **1992**.

[4] Mykel J Kochenderfer, Jessica E Holland, and James P Chryssanthacopoulos. Next-generation airborne collision avoidance system. Technical report, Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States, **2012**.

[5] HL Ammerman, ES Becker, GW Jones, WK Tobey, and MK Phillips. Federal aviation administration air traffic control operations concepts volume i: Atc background and analysis methodology. washington dc: Federal aviation administration report no. Technical report, DOTIFAAIAP-87-01, **1987**.

[6] Federal Aviation Administration. Introduction to tcas ii: Version 7.1, **2011**.

[7] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, **2012**.

[8] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley Sons, Inc., USA, 1st edition, **1994**. ISBN 0471619779.

[9] Clarivate Analytics. Web of science. https://apps.webofknowledge.com/. Accessed: Sep, 2019.

[10] John Lygeros and Nancy Lynch. On the formal verification of the tcas conflict resolution algorithms. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 2, pages 1829–1834. IEEE, **1997**.

[11] C. Livadas, J. Lygeros, and N. A. Lynch. High-level modeling and analysis of the traffic alert and collision avoidance system (tcas). *Proceedings of the IEEE*, 88(7):926–948, **2000**. doi:10.1109/5.871302.

[12] Christian von Essen and Dimitra Giannakopoulou. Probabilistic verification and synthesis of the next generation airborne collision avoidance system. *International Journal on Software Tools for Technology Transfer*, 18(2):227–243, **2016**.

[13] Ryan W Gardner, Daniel Genin, Raymond McDowell, Christopher Rouff, Anshu Saksena, and Aurora Schmidt. Probabilistic model checking of the next-generation airborne collision avoidance system. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, **2016**.

[14] Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan Gardner, Stefan Mitsch, and André Platzer. A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. *International Journal on Software Tools for Technology Transfer*, 19(6):717–741, **2017**.

[15] J. Jeannin, K. Ghorbal, Y. Kouskoulas, R. Gardner, A. Schmidt, E. Zawadzki, and A. Platzer. Formal verification of acas x, an industrial airborne collision avoidance system. In *2015 International Conference on Embedded Software (EMSOFT)*, pages 127–136. **2015**. doi:10.1109/EMSOFT.2015.7318268.

[16] Ritchie Lee, Mykel J Kochenderfer, Ole J Mengshoel, Guillaume P Brat, and Michael P Owen. Adaptive stress testing of airborne collision avoidance systems. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pages 6C2–1. IEEE, **2015**.

[17] Mykel J Kochenderfer, Leo P Espindle, James K Kuchar, and J Daniel Griffith. A comprehensive aircraft encounter model of the national airspace system. *Lincoln Laboratory Journal*, 17(2):41–53, **2008**.

[18] M. Egorov, Zachary Sunberg, E. Balaban, T.A. Wheeler, J.K. Gupta, and Mykel Kochenderfer. Pomdps.jl: A framework for sequential decision making under uncertainty. *Journal of Machine Learning Research*, 18:1–5, **2017**.

[19] Eugenio Bargiacchi. Ai-toolbox. https://github.com/Svalorzen/AI-Toolbox. Accessed: Nov 10, 2020.

[20] Trey Smith. Zmdp software for pomdp planning. https://github.com/trey0/zmdp. Accessed: Nov 10, 2020.

[21] Wee Sun Lee Hanna Kurniawati, David Hsu and Haoyu Bai. Approximate pomdp planning toolkit. http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/. Accessed: Nov 10, 2020.

[22] RTCA SC. 147, june 19, 2008, minimum operational performance standards for traffic alert and collision avoidance system ii (tcas ii)–version 7.1–volumes i and ii. Technical report, RTCA/DO-185B, Washington, DC, RTCA, Inc.

[23] J. P. Chryssanthacopoulos and M. J. Kochenderfer. Collision avoidance system optimization with probabilistic pilot response models. In *Proceedings of the 2011 American Control Conference*, pages 2765–2770. **2011**. ISSN 2378-5861. doi: 10.1109/ACC.2011.5990776.

[24] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, **2004**. doi:10.1109/JPROC.2003.823141.

[25] Mykel J Kochenderfer and Nicholas Monath. Compression of optimal value functions for markov decision processes. In *2013 Data Compression Conference*, pages 501–501. IEEE, **2013**.

[26] Richard Bellman. The theory of dynamic programming. Technical report, Rand corp santa monica ca, **1954**.

[27] Anders Ringgaard Kristensen. Textbook notes of herd management: Dynamic programming and markov decision processes, **1996**.

[28] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, and M. J. Kochenderfer. Policy compression for aircraft collision avoidance systems. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–10. **2016**. doi:10.1109/DASC.2016.7778091.

[29] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, **1997**. ISBN 9780071154673.

[30] Jessica E Holland, Mykel J Kochenderfer, and Wesley A Olson. Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance. *Air Traffic Control Quarterly*, 21(3):275–297, **2013**.

[31] Mykel J Kochenderfer and JP Chryssanthacopoulos. Robust airborne collision avoidance through dynamic programming. *Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371*, 130, **2011**.

[32] Norman E. Fenton. *Software Metrics A Rigorous Approach*. Chapman Hall, Ltd., London, UK, UK, **1991**. ISBN 0442313551.

[33] ICAO Annex. 10. vol. iv. aeronautical telecommunications-surveillance and collision avoidance systems, july 2007.

[34] Selim Temizer, Mykel Kochenderfer, Leslie Kaelbling, Tomas Lozano-Pérez, and James Kuchar. Collision avoidance for unmanned aircraft using markov decision processes. In *AIAA guidance, navigation, and control conference*, page 8040. **2010**.

[35] John Deaton and Michael P Owen. Evaluating collision avoidance for small uas using acas x. In *AIAA Scitech 2020 Forum*, page 0488. **2020**.

[36] R. E. Tompa, B. Wulfe, M. P. Owen, and M. J. Kochenderfer. Collision avoidance for unmanned aircraft using coordination tables. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–9. **2016**. ISSN 2155-7209. doi: 10.1109/DASC.2016.7777958.

[37] Thomas B Billingsley, LP Espindle, and J Daniel Griffith. Tcas multiple threat encounter analysis. *Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-359*, **2009**.

[38] Federal Aviation Administration and United States. Federal Aviation Administration. *Pilot's Handbook of Aeronautical Knowledge*. Skyhorse Publishing Inc., **2009**.