An Evolutionary Computational Analysis of Tactical Controller Tool

S. Alam, C.J. Lokan, H.A. Abbass,
Defence & Security Applications Research Centre
University of New South Wales,
Australian Defence Force Academy, Canberra, Australia
s.alam,h.abbass,c.lokan@adfa.edu.au

M. Ellejmi and S. Kirby
Cooperative Network Division (CND),
EUROCONTROL,
Brétigny-sur-Orge, Paris, France
mohamed.ellejmi,stephen.kirby@eurocontrol.int

Abstract — Eurocontrol has introduced a Tactical Controller Tool (TCT), for use by Tactical Air Traffic Controllers (ATCs) who require trajectory conflict information over the next 5 to 8 minutes to help them detect conflicts in their sector. Simulation trials indicate that the safety benefits of TCT may be limited because of large numbers of False Alarms. There is a need to tune the system to identify the best look-ahead time to reduce nuisance alerts while retaining genuine conflicts generated by the system. In this paper we quantitatively investigate the performance of TCT for different look-ahead timings (using evolutionary computation to evolve complex air traffic conflict scenarios), and we investigate the patterns in conflict alerts raised by TCT that resulted in False Alarms. We find that a 6-minute look-ahead time leads to TCT generating fewest False Alarms. Flights in climb phase and with wide convergence angle contribute to a large number of False Alarms. TCT predicted conflicts that have duration of less than 45 seconds, and are on the boundary of 5 nm separation also lead to high numbers of False Alarms.

I. INTRODUCTION

Eurocontrol's Air Traffic Management Strategy for 2000+ (ATM2000+) has identified that air traffic controller (ATC) workload is a major constraint to capacity improvement, and that improved automation tools will assist ATCs to handle more flights [1].

With the continued growth in air traffic, the usual peaks and troughs in the sector are gradually disappearing and are replaced by constant high traffic. As a result of this the ATCs are under constant pressure to deliver peak performance over more and longer time periods. This implies that alertness and traffic management skills of ATCs are coming under increasing pressure [2].

Advances in automation and their integration into ATC systems have the potential to assist ATCs in conflict detection and resolution [3, 4]. These advances include computer-based assistance tools including trajectory prediction, medium term conflict detection (MTCD) and highly interactive and advanced graphical interfaces.

In many sectors today, the Tactical Controller is overworked and most of his/her efforts are spent in monitoring traffic [5, 6]. The computer-based assistance tools mentioned above provide support, but mainly to the Planning Controller, mostly related to the aircrafts' planned trajectories. The Tactical Controller needs support in the near term to help him/her handle the dynamic and stressful situations in the sector.

At the most immediate level, Short Term Collision Alert (STCA) is a controller tool which detects short-term conflicts between aircraft, using only the information from their latest track state vector [7]. STCA makes no assumptions about anticipated manoeuvres, or any planned clearances. STCA needs an immediate reaction (time window of 2 minutes) by ATCs and can lead to major disasters [8]

At the other end of the tactical ATC level, Medium Term Collision Detection tool considers look-ahead times of 10–20 minutes. MTCD has been studied in [9] by the authors.

In between, for high traffic where the controller has little reaction time and needs immediate assistance, Eurocontrol has introduced a Tactical Controllers Tool (TCT). TCT is intended for use by the Tactical Controller who requires trajectory conflict information over the next 5 to 8 minutes to help them detect conflicts in their sector [10].

TCT has been under development for some time now. Real time simulation trials have been undertaken by taking data from Central Flow Management Unit (CFMU) in early 2009 [11]. Traffic samples were engineered by Eurocontrol to create particular conflict situations with different traffic load scenarios. Qualitative data was collected from questionnaires and interviews with the controllers. Quantitative data (capacity, safety, efficiency) was collected using INTEGRA tool [12]. The results indicated high confidence of ATC controllers in TCT, as it was able to identify potential future problems, so that the time and effort previously spent finding problems was available for controllers to concentrate on identifying appropriate solutions.

However, a number of False alarms (1.6 rates on a scale of

---

[1] The European Organisation for the Safety of Air Navigation EUROCONTROL) is an international organization whose primary objective is the development of a seamless, pan-European Air Traffic Management (ATM) system. The goal for EUROCONTROL is to develop, coordinate and plan for implementation of pan-European ATM strategies and their associated action plans.

5 points) were detected during the trials. Those nuisance alarms were detected mostly for climbing/descending geometries. It was recommended to tune the system to reduce nuisance alerts while avoiding losing genuine conflicts [11].

Tuning the system, and understanding the factors that still cause it problems, are the subject of this paper. We first characterise the performance of TCT (particularly with regard to false alarms) as the look-ahead time window varies from 5 to 8 minutes. Given the resulting best window size, we then seek to understand the nature of conflict characteristics that lead to False Alarms in TCT, so that we can gain a better insight into its performance and limitations in the given ATC environment.

The problem of assuring safety in TCT is one of evaluating that the system identifies all possible conflicts correctly, which is a very challenging task. This evaluation process must ensure the ability of TCT to cope with the most safety-critical situations and complex scenarios. Simply re-solving past problems is not enough. The new TCT system has to prove resilient to a wide variety of novel system challenges (conflict scenarios).

Generating a sufficient variety of novel challenges (conflict scenarios) is an interesting challenge in its own right. We approach it by using an evolutionary computational framework that evolves complex air traffic conflict scenarios [13] using the "Red Teaming" concept [14]. Red teaming is a concept, normally used in defence, which refers to studying a problem by anticipating adversary behaviours [15]. In this context the blue team represents the TCT, and the red team represents the adversaries (the conflict scenarios). By seeking conflict scenarios that cause problems for TCT, we aim to understand susceptibilities of the TCT in order to improve the overall performance of the system.

The paper is organized as follows. We first introduce the TCT followed by its algorithm, then we summarise the methodology of generating conflict scenario using evolutionary computation. We then present the methodology for evaluating TCT, evaluation metrics and experiment design. We conclude by analysing results and concluding remarks.

II. TACTICAL CONTROLLER TOOL

The primary role of the TCT is to manage and notify predicted losses of separation on the basis of the current aircraft track, the system flight plan, and the current and anticipated conformance to the aircraft's plan.

The standard MTCD conflict detection tool is used to predict conflicts along the system ground trajectory in the medium term (up to 10 - 20 minutes), providing the aircraft are in a relatively stable part of their route (e.g. cruise phase) and are following their flight plans. In cases where the flight is climbing or descending which is not in accordance with its flight plan, or when an aircraft starts to deviate from its flight plan at a critical time, the MTCD provides no help and could provide misleading conflict information. TCT is intended to improve this situation for the Tactical Controller for a look-ahead time of 5–8 minutes.

A. *TCT Trajectories*

TCT uses the concepts of tactical trajectory and state vector trajectory to identify conflicts between two aircraft.

1) Tactical Trajectory: It is generated by using an aircraft's actual position instead of its flight plan position. It is regularly updated and based on actual ground trajectory. It starts from the 4D position of the aircraft according to last update. If the aircraft is deviating from its trajectory it integrates a rejoin manoeuvre to the next available waypoint.. The vertical profile extends to the Cleared Flight Level (CFL) then on to the following altitude constraints. When climbing, an intermediary cleared flight level, between the actual flight level (AFL) and extended flight level (XFL), is effectively ignored. When descending, an intermediary CFL is obeyed before descending to the XFL.

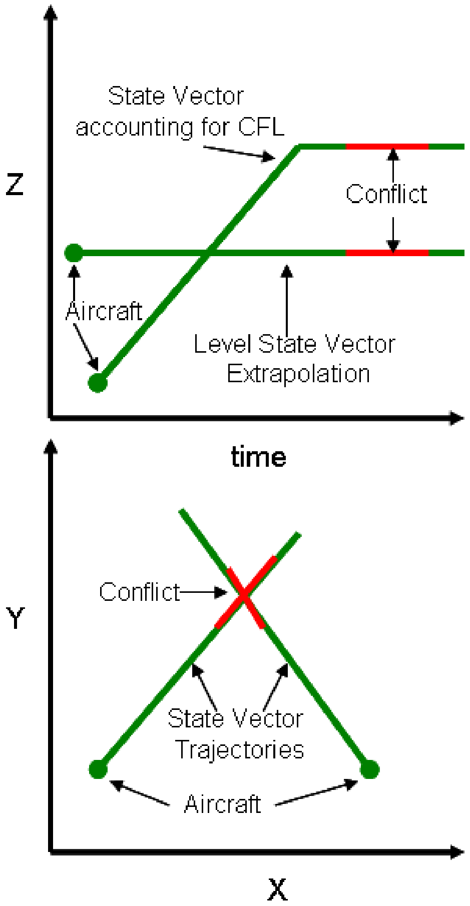

Figure 1. An illustration of State Vector Conflicts in TCT, accounting for Cleared Flight Level (CFL) above and a straight forward extrapolation of State vector (below).

2) State Vector Trajectory: State Vector Trajectory is a trajectory that follows the aircraft's current track, and extends to the CFL following a vertical profile that results from the trajectory predictor's model for the associated aircraft. It starts from the 4D position of the aircraft according to last update. The lateral route is a single segment extending ahead of the aircraft following the current aircraft track and for a duration equal to the TCT look-ahead time.

B. *TCT Conflicts*

By inspection of the state and vector trajectories, TCT produces the following conflict information:

1) State Conflict: This is a conflict detected between two State Vector Trajectories and for which there is no planned manoeuvre in either flight plan that might avoid it (Figure 1).

2) Tactical Conflict: This is conflict between two Tactical Trajectories. The Tactical Trajectory is updated whenever there is a small deviation from the expected position, so the Tactical Conflicts are updated to be relevant even when the aircraft is deviating from the route (Figure 2).

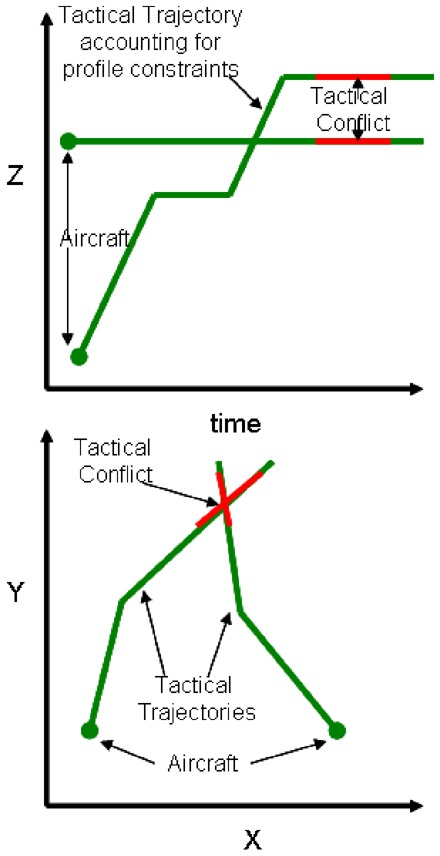

Figure 2. An illustration of Tactical Conflicts in TCT, accounting for climb profile (above) and a flight plan manouvre (below).

C. *TCT Conflict Algorithm formulation [16], [17]*

The TCT is based on MTCD algorithm which uses 3D vector line geometry to determine the closest approach between two line segments, and the points on the conflicting line segments where the minimum separation standard is met exactly (the points at which separation is lost and regained). These points will mark the closest approach point and the start and end points of the conflict.

To calculate the conflict intervals the relative position dx and speed dv of the intruder are calculated in Cartesian coordinates for the vector calculation. We use a right-handed reference frame with origin at the ownship position. The equation of relative motion x of an intruder with reference to the ownship is given by:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} dx_1 \\ dx_2 \\ dx_3 \end{pmatrix} + t \times \begin{pmatrix} dv_1 \\ dv_2 \\ dv_3 \end{pmatrix}$$

The time t in this equation is relative, implying $t = 0$ is now. First the vertical conflict interval $[t_{in-vert}, t_{out-vert}]$ is found by using vector calculation and solving for t

$$|x_3| = H \tag{1}$$

$$|dx_3 + t \cdot dv_3| = H \tag{2}$$

$$dx_3 + t \cdot dv_3 = H \bigcup dx_3 + t \cdot dv_3 = -H \tag{3}$$

$$t_1 = \frac{H - dx_3}{dv_3} \qquad t_2 = \frac{-H - dx_3}{dv_3} \tag{4}$$

which gives

$$t_{in-vert} = min(t_1, t_2) \tag{5}$$

$$t_{out-vert} = max(t_1, t_2) \tag{6}$$

Then the horizontal conflict interval is calculated as the intersection of line and circle in the horizontal plane. To find these times, the following equation is solved for t:

$$x_1 + x_2 = R^2 \tag{7}$$

$$(dx_1 + t \cdot dv_1)^2 + (dx_2 + t \cdot dv_2)^2 = R^2 \tag{8}$$

$$\begin{aligned} &(dv_1^2 + dv_2^2) \cdot t^2 + 2(dx_1 dv_1 + dx_2 dv_2)t + \\ &(dx_1^2 + dx_2^2 - R^2) = 0 \end{aligned} \tag{9}$$

This is a quadratic equation form with:

$$a = dv_1^2 + dv_2^2$$

$$b = 2(dx_1 dv_1 + dx_2 dv_2)$$

$$c = dx_1^2 + dx_2^2 - R^2$$

and discriminant $D = b^2 - 4ac$

If the discriminant is negative, there is no intersection and hence no conflict. If the discriminant is positive, the interval of horizontal conflict is given by:

$$t_{in-horz} = \frac{-b-\sqrt{D}}{2a} \tag{10}$$

and

$$t_{out-horz} = \frac{-b+\sqrt{D}}{2a} \tag{11}$$

If time is negative this refers to a time in the past (conflict already occurred).

The vertical and horizontal intervals are combined and checked for overlap. For the combined t_{in} the maximum of both values is used (conflict only if it has simultaneously intruded the protected zone horizontally and vertically)

$$t_{in} = max(t_{in-vert}, t_{in-horz}) \tag{12}$$

For the time of leaving the conflict the minimum of both values is used.

$$t_{out} = min(t_{out-horz}, t_{out-vert}) \tag{13}$$

If t_{out} is before t_{in} there is no overlap and hence no conflict. The time when conflict will happen can then be computed as:

$$t_{conflict} = t_{now} + t_{in} \tag{14}$$

III. CONFLICT SCENARIOS

We previously developed a methodology for algorithmically generating air traffic scenarios with desired conflict characteristics [13]. We break from the classical approach of pre-scripting conflict events in air traffic scenarios, and use evolutionary computation algorithm [18] instead to evolve conflicts. The objective of the evolutionary computation algorithm is to evolve increasingly complex conflict scenarios so that the TCT can incur maximum failure (in terms of evaluation metrics).

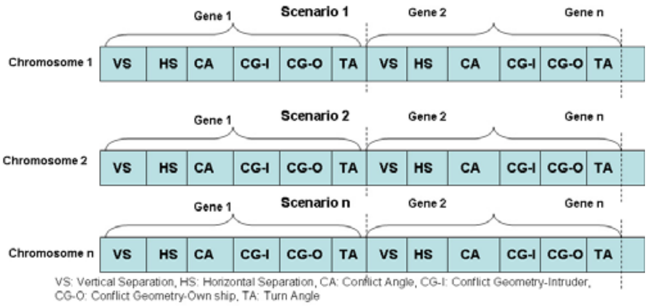

Figure 3. An illustration of chromosome data structure used for representing a two-aircraft conflict[9]

Based on [19], the following conflict characteristics at the closest point of approach (CPA) between two aircraft are encoded in the chromosomes data structure which is used by the evolutionary algorithm: Horizontal separation (HS) at CPA, Vertical separation (VS) at CPA, Conflict geometry Intruder (CGI) (climb, cruise or descent), Conflict geometry Ownship (CGO) (climb, cruise or descent), Conflict angle (CA) at CPA and Turn Angle (TA) for the ownship, before the two aircraft reaches their CPA.

A real-valued representation with a linear chromosome structure is chosen to represent an air traffic scenario. Every gene of the chromosome encodes the characteristics of a conflict-pair, representing a conflict between a pair of aircraft.

As illustrated in Figure 3, every chromosome represents an air traffic scenario, where each pair of conflicting aircraft in the scenario is represented as a gene of the chromosome.

IV. SIMULATION ENVIRONMENT AND EVALUATION FRAMEWORK

A. Air Traffic Operations & Management Simulator (ATOMS)

To simulate air traffic scenarios and evaluate the performance of the TCT, we use the ATOMS air traffic simulator [20] developed by the authors. ATOMS is a medium-fidelity air traffic simulation system that enables us to test a large number of scenarios in a reasonable time. TCT state and vector conflict detection were programmed into ATOMS and every flight pair is checked for conflict at 5 second time interval. ATOMS is thus used as the evaluation objective function for air traffic scenarios: every time it is called with a scenario, it evaluates the performance of the TCT in a given scenario and returns performance measure.

B. Search Space

The search space for fine tuning the look-ahead time is illustrated in figure 4.

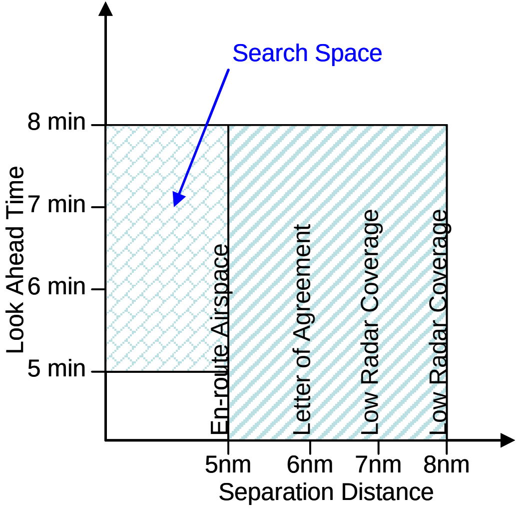

Figure 4. The tuning search area for TCT for 5 nm enroute seperation.

We fix the separation distance as 5 nm, as the experiments are conducted by simulating en-route airspace with look-ahead time of 5, 6 7 and 8 minutes respectively.

C. Evolutionary Computation framework

Figure 5 illustrates our methodology. The initial population (initial scenarios) is used to further generate complex conflict scenarios, which are then evaluated using ATOMS. A state of the art evolutionary algorithm (NSGA-II) [21] is used to evolve increasingly complex air traffic scenarios. Scenarios with higher fitness (i.e. higher Missed Detects or False Alarms) survive the evolutionary mechanism of the genetic algorithm and breed further to come up with more complex conflict scenarios.

This evolutionary mechanism helps to evolve complex conflict scenarios that cause TCT to fail; as the evolution proceeds, it will find scenarios in which the TCT fails even more. If the TCT performs well (detects all the conflicts) in a scenario, the scenario fitness is low; if it performs poorly (fails to detect the conflicts), the fitness of the scenario is high.

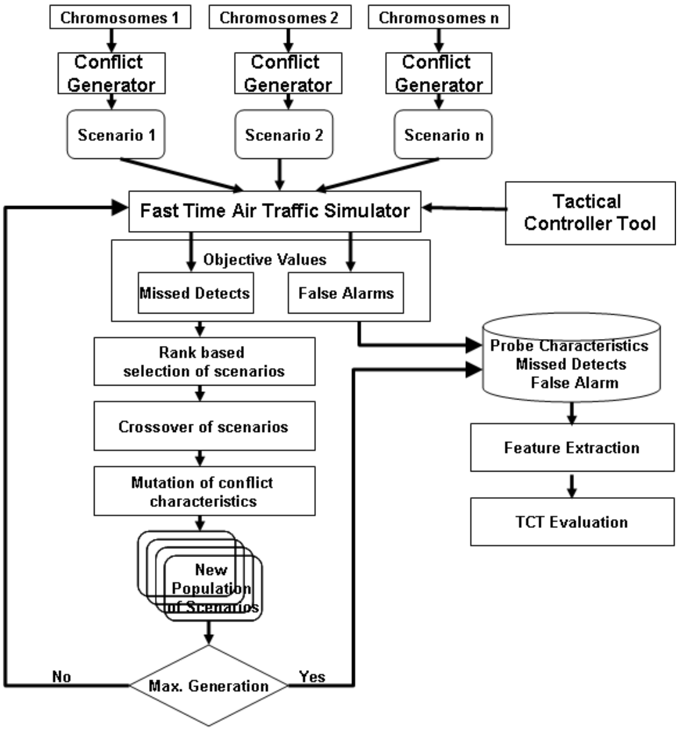

Figure 5. TCT process evaluation framework based on Red Teaming Concept[9]

D. Evaluation Metrics

We use False Alarms as our primary metrics for evaluating TCT performance in terms of state vector and tactical conflicts. False Alarms (FA) represent the number of conflict alerts that didn't actually materialize into a separation violation, but the TCT labelled them as potential conflicts. Thus the objective functions can be defined as a maximization problem in which the objective of the evaluation process is to maximize the events of False Alarms in an air traffic scenario on which TCT is applied (equation 15).

$$MAX\left(f1\right) = FA \qquad (15)$$

V. EXPERIMENT DESIGN & PARAMETER SETTINGS

A generic sector in the Australian National Airspace region [S32.0 E142.0 S38 E 150] is selected. Minimum flight altitude is set to 15,000 ft and maximum flight altitude is set to 38,000 ft. Speed of the aircraft is within the band of 300 knots to 550 knots. All flights are activated within the sector and deactivated at the sector boundary.

We use a population size of 50 which implies that there are 50 scenarios. In each scenario we have 100 flights, with 50 paired conflicts with different conflict characteristics. The number of generations is set to 30 and the crossover probability is set to 0.1; the mutation probability is set to 0.01. This gives us 2.5 million conflicts to evaluate. More conflicts may result from overlap of aircraft trajectories in a scenario. These parameter settings are not claimed to be optimal but our previous work suggests that they are reasonable for this problem.

Flights continue on their flight paths unless they reach the sector boundary/deactivation point, where they are removed from the scenario.

The experiments are repeated for four look-ahead time settings of 5, 6, 7 and 8 minutes. Instances of False Alarms are obtained by comparing the conflict alerts raised by TCT and actual loss of separation in simulation. Conflict characteristics at CPA are recorded for different look-ahead-time intervals.

VI. RESULTS & ANALYSIS

A. What is the best look-ahead time?

We first report results, in terms of False Alarms for each look-ahead time interval of TCT. We begin by presenting the overall False Alarms over 30 generations. There were 26740 False Alarms for the look-ahead time of 5 min, 19660 for 6 min, 19730 for 7 min, and 28100 for 8 min respectively. In the initial generations (see Figure 6) the numbers of False Alarms for all four look-ahead times are low, followed by a phase transition around the 17th generation. The evolutionary algorithm converges around the 26th generation for all four look-ahead timings.

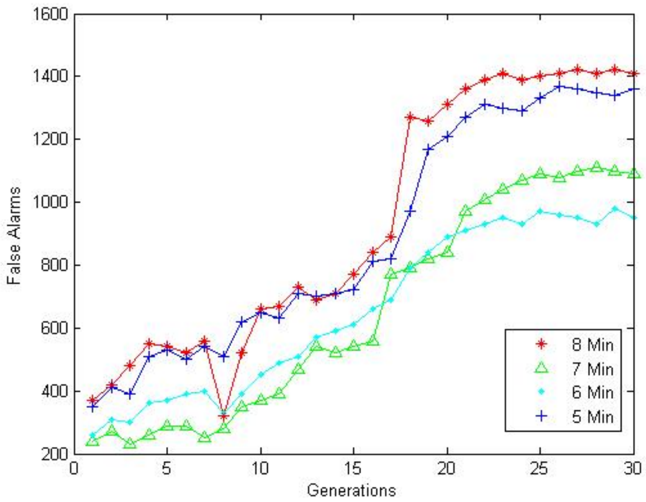

Figure 6: False Alarms generated by the TCT over 30 generations for look-ahead time intervals of 5, 6, 7, 8 minutes

It can be seen from Figure 6 that six minutes look ahead time results in the lowest number of false alarms. 8 minutes generates the most False alarms, due to increase in trajectory prediction errors, and because the straightforward extrapolation of state trajectories can result in turning manoeuvres being missed.

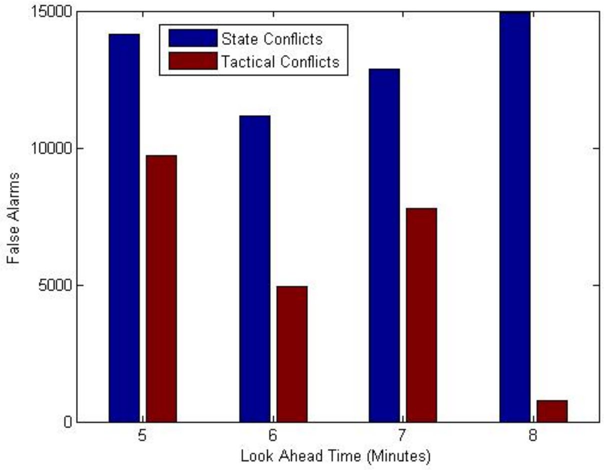

Figure 7: False Alarms generated by the State and Tactical conflicts for different look ahead time summed over 30 generations.

5 minutes look-ahead time also generates many False Alarms due to the small time interval, which may mean that level off segments for climbing or descending flights are missed.

We then analyzed State and Tactical conflicts in TCT separately, to identify the individual role of each conflict in the overall TCT performance.

Figure 7 shows that False Alarms from Tactical Conflicts in TCT are fewest, whereas State Conflicts are highest, with 8 min look-ahead time. Tactical Conflicts are based on updated flight plans that account for small deviations from the expected position, so are more accurate even for the longer look-ahead time, while State Conflicts are based on straightforward extrapolation of current aircraft state for a duration equal to the TCT look-ahead time, which reduces prediction accuracy as look-ahead time increases.

A 5 min look-ahead time is also poor for State conflicts: a large number of False Alarms are generated, mostly when the flight is in transition (climb/descent), due to use of a standard climb model instead of actual height-rate of aircraft.

We conclude from Figures 6 and 7 that a 6-minute look-ahead time provides a good tradeoff between State and Tactical conflicts while minimizing the False Alarms.

B. Conflict characteristics

We then investigated the conflict characteristics at CPA that were observed with a 6 min look-ahead time. Figure 8 shows the False Alarms (aggregated over 30 generations) generated for different conflict angles between two aircraft. From Figure 9 it can be seen that both the State and Tactical conflicts are susceptible of generating FA when the convergence angle between two flights is wide (90–180 degrees).

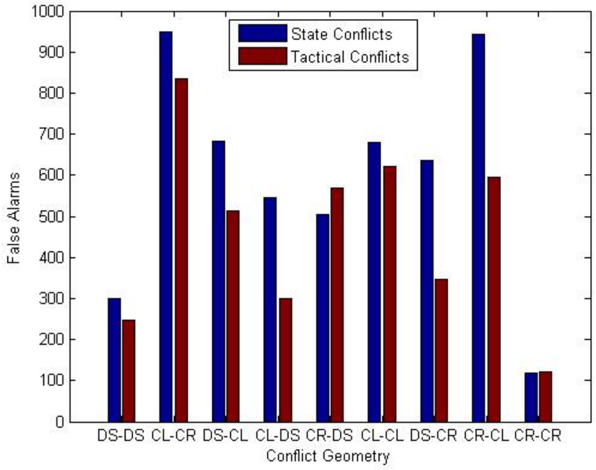

Figure 8: Aircraft-Aircraft conflict geometry for the False Alarms generated by the State and Tactical conflicts for 6 minutes look ahead time.

Next we investigated the conflict geometry of flights that lead to False Alarm under a 6 minutes look-ahead time, for both state and tactical conflicts. Figure 8 shows that State Conflicts have high False Alarms when either ownship or intruder or both are climbing. For state conflicts this may be because the state vector trajectory uses the BADA vertical profile to reach the CFL and not the height-rate of the aircraft. For Tactical conflicts it may be because when either ownship or intruder or both are climbing an intermediary CFL between the AFL and XFL is ignored, leading to False Alarms

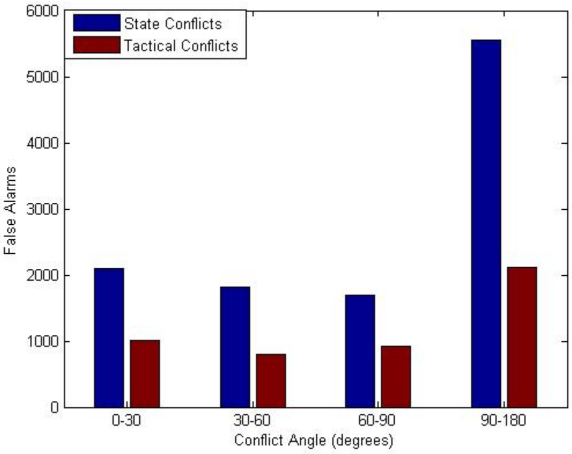

Figure 9: Aircraft-Aircaft conflict angle for the False Alarms generated by the State and Tactical conflicts for 6 min look-ahead time.

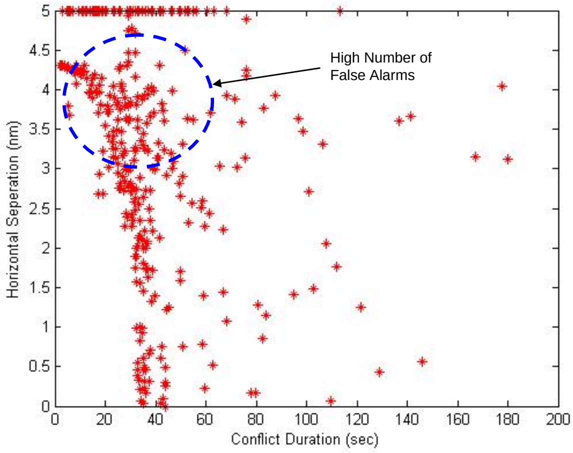

Figure 10: Horizontal Seperation at the given conflict duration which lead to False Alarms for 6 minutes look-ahead time in TCT.

.

Finally we looked at the effect of conflict duration and the TCT estimated horizontal and vertical separation on the False Alarms.

Figure 10 shows that as the predicted horizontal separation approaches close to 5 nm, conflicts with duration less than 45 seconds result in a higher number of False Alarms. Whereas from Figure 11 it can be seen that conflict durations of less than 45 seconds lead to high number of False Alarms regardless of the predicted vertical separation. This suggests that conflicts which are on the threshold of outer separation boundary with conflict duration less than 45 seconds should be monitored further before they are flagged as conflicts.

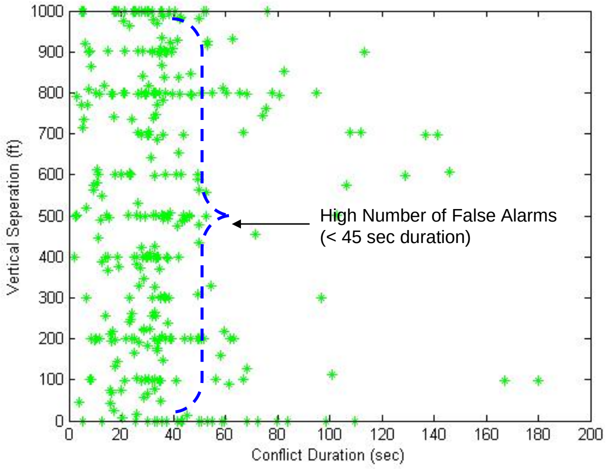

Figure 11: Vertcial Seperation at the given conflict duration which lead to False Alarms for 6 minutes look ahead time in TCT.

VII. CONCLUSIONS

In this paper, we investigated the performance of TCT for different look-ahead times, and attempted to identify patterns in conflicts that lead to False Alarms.

We found that 6 minutes look-ahead time provides the best trade-off between the two components (State and Tactical) of TCT.

Results also indicate that the climb model in TCT, which uses a standard climb model instead of actual height-rate of a climbing aircraft, leads to inaccuracies in predicting the level-off segments resulting in high number of False Alarms. Further conflicts with wider conflict angles, which mostly happen at waypoint crossings, may also lead to higher False Alarms.

Conflict durations also affect the performance of TCT, especially with conflicts that are on the boundary of horizontal separation (4-5 nm) especially if the duration is less than 45 seconds.

Overall results indicate that 6 minutes look-ahead time, coupled with delayed alerts for conflicts whose duration is less than 45 seconds, and using actual height rate of aircraft in transition instead of a standard climb model, may improve the performance of TCT system.

In future we will be extending our work by investigated Missed Detects in TCT. We will further our investigations by employing data mining techniques to identify intrinsic patterns that may exist in conflict characteristics that lead to False Alarms.

ACKNOWLEDGMENT

This work has been co-financed by the European Organisation for the Safety or Air Navigation (EUROCONTROL) under its University Research Grant programme. The content of the work does not necessarily reflect the official position of EUROCONTROL on the matter.

REFERENCES

[1] Eurocontrol Air Traffic Management Strategy for the Years 2000+ Vol. 2, EATMP Information Centre, Brussels, 2003 Edition

[2] U. Metzger, R. Parasuraman, “Automation in future air traffic management: Effects of decision aid reliability on controller”, Human Factors, vol 47, no. 1 pp 37. 2005

[3] J.W. Andrews, H. Erzberger, J.D. Welch and others, “Safety analysis for advanced separation concepts” Air Traffic Control Quarterly, vol. 14, no. 1, pp 5-24, 2006

[4] H. Erzberger, “Decision support system in tactical air traffic flow management for air traffic flow controllers”, In proc. 25th International Congress of the Aeronautical Sciences, Hamburg, Germany, 2007

[5] L. Weigang, B.B. de Souza, A.M.F. Crespo, and D..P.Alves, “Decision support system in tactical air traffic flow management for air traffic flow controllers”, Journal of Air Transport Management vol. 14, no. 6 pp 329-326, Elsevier, 2008

[6] R.A. Paielli, H. Erzberger, D. Chiu, and K.R. Heere, “Tactical conflict alerting aid for air traffic controllers”, Journal of Guidance, Control, and Dynamics vol. 32, no. 1, AIAA, 2009

[7] P. Brooker, “STCA, TCAS, airproxes and collision risk”, The Journal of Navigation, vol. 58, no. 3, pp. 389-404, Cambridge Univ Press, 2005

[8] C.W. Johnson, “Final Report: review of the BFU Uberlingen Accident Report”, C/1.369/HQ/SS/04, Eurocontrol, 2004

[9] S. Alam, H.A. Abbass, C.J. Lokan, “Computational Red Teaming to Investigate Failure Patterns in Medium Term Conflict Detection”, 8th Eurocontrol Innovation Research Workshop, Eurocontrol Experimental Center, Brtigny-sur-Orge, France, 2009

[10] Eurocontrol, “eDEP development and evaluation platform: TCT (Tactical Controller Tools) Concept of Operations”, GL/eDEP/CONCEPT/TCT/1/1.0, Eurocontrol Experimental Centre, Brtigny-sur-Orge, Farnce, 2009

[11] Eurocontrol, “TCT Real Time Simulation Evaluation Report,” Technical Report 09-110142-C, Eurocontrol Experimental Centre, Brtigny-sur-Orge, France, 2009

[12] R. Gingell, C. Strachan, A. Taylor, S. Kinnersly, and S. Fox, “INTEGRA Metrics & Methodologies Execution Phase--Final Report”, Eurocontrol Experimental Centre, Brtigny-sur-Orge, Farnce, 2005

[13] S. Alam, K. Shafi, H. A. Abbass, and M. Barlow, “Evolving air traffic scenarios for the evaluation of conflict detection models,” in Proc. 6th CARE Eurocontrol Eurocontrol Innovative Research Workshop, Eurocontrol Experimental Centre, Brtigny-sur-Orge, Farnce, 2007

[14] A. Yang, H. Abbass, and R. Sarker, “Characterizing warfare in red teaming,” IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 36, no. 2, pp. 268–285, 2006.

[15] B. Bennett, S. Gardiner, and D. Fox, “Not Merely Planning for the Last War,” New Challenges for Defense Planning: Rethinking How Much Is Enough, RAND Corp. p. 477, 1994.

[16] A. Warren, “Medium term conflict detection for free routing: operationalconcepts and requirements analysis,” in Digital Avionics Systems Conference, 1997. 16th DASC., AIAA/IEEE, vol. 2, 1997

[17] Eurocontrol, “Eurocontrol development and evaluation platform, edep atc layer detailed design document,” Technical Report GL/DEP/DDD/1/2.5.3, Eurocontrol Experimental Centre, Brtigny-sur-Orge, Farance, 2009.

[18] D. Goldberg, Genetic algorithms in search, optimization and machine learning, 1st ed. Boston, MA: Addison-Wesley Longman Pub. Co, 1989.

[19] K. Bilimoria, “A methodology for the performance evaluation of a conflict probe,” AIAA Journal of Guidance, Control, and Dynamics, vol. 24, no. 3, pp. 444–451(8), 2001.

[20] S. Alam, H. Abbass, and M. Barlow, “Air traffic operations and management simulator ATOMS,” IEEE Transactions on Intelligent Transportation System, vol. 9, no. 2, pp. 209–225, 2008.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, pp. 182–197, 2002.