Author: Toubman, A.
Title: Calculated Moves: Generating Air Combat Behaviour
Issue Date: 2020-02-05

Summary

By training with virtual opponents known as computer generated forces (CGFs), trainee fighter pilots can build the experience necessary for air combat operations, at a fraction of the cost of training with real aircraft. In practice however, the variety of CGFs is not as wide as it can be. This is largely due to a lack of behaviour models for the CGFs. The lack motivated me to design and improve air combat simulations. In this thesis we investigate to what extent behaviour models for the CGFs in air combat training simulations can be *automatically* generated, by the use of machine learning.

The domain of air combat is complex, and machine learning methods that operate within this domain must be suited to the challenges posed by the domain. In Chapter 1, we identify five challenges that must be met before newly generated behaviour models can effectively be applied in training simulations. These are: (A) producing team coordination, (B) computationally evaluating CGF behaviour, (C) efficient reuse of acquired knowledge, (D) validating generated behaviour models, and (E) generating accessible behaviour models.

From the above motivation for the research, together with the five challenges, we derive the following problem statement: *To what extent can we use dynamic scripting to generate air combat behaviour models for use in training simulations, in such a way that the five challenges of generating air combat behaviour models are met?* The problem statement mentions the use of the dynamic scripting algorithm. This algorithm produces human-readable behaviour models, and thus enables us to meet challenge E. Based on the remaining four challenges, we formulate five research questions that we investigate in the remainder of the thesis.

In Chapter 2, we present background information on the process by which behaviour models are created today. Furthermore, we introduce (a) machine learning, and (b) the dynamic scripting algorithm in particular. Additionally, we review earlier work on the subject of generating air combat behaviour models by means of machine learning.

In Chapter 3, we investigate research question 1: *To what extent can we generate air combat behaviour models that produce team coordination?* Today, the smallest unit that performs air combat missions is the *two-ship*, consisting of a *lead* and a *wingman* aircraft. To succeed in their missions, the lead and the wingman in a two-ship need to carefully coordinate their actions. Therefore, such coordination should be reflected in the behaviour models of a two-ship

of CGFs. We define three coordination methods within the rule-based framework of dynamic scripting: (1) a decentralised coordination method without communication called TACIT, (2) a centralised coordination method with communication called CENT, and (3) a decentralised coordination method with communication called DECENT. Next, we perform three series of automated simulations. In each series, we use dynamic scripting to generate behaviour models for a two-ship that engages a pre-programmed opponent while coordinating by one of the coordination methods. We find that each of the three methods leads to a flexible division of roles between the CGFs. Out of the three methods, the coordination produced by the CENT method resulted in the most effective behaviour that reached the highest win rates. Based on our research, we may conclude that by means of dynamic scripting, we are able to (a) generate multiple forms of team behaviour, and (b) easily inspect the roles assumed by the team members.

In Chapter 4, we investigate research question 2: *To what extent can we improve the reward function for air combat CGFs?* The reward function is an essential part of dynamic scripting. It evaluates the desirability of the behaviour produced by the behaviour models that are generated, and then produces a reward signal that stimulates the dynamic scripting algorithm to improve the models in a next iteration. A commonly used reward function is the *binary* reward function: a reward signal of 1 is provided if the CGFs win a simulated encounter (i.e., show desirable behaviour) using the generated behaviour model, otherwise a reward signal of 0 is provided. However, because this reward signal is both *sparse* (i.e., a CGF has to display exactly the right behaviour before a reward is obtained) and *unstable* (i.e., non-determinism in the CGF's environment may cause the same behaviour to lead to different results), it is possible that more desirable behaviour can be achieved by using a more suitable reward function. We develop two new reward functions for use in the air combat domain: DOMAIN-REWARD which is aimed at making the rewards less sparse, and AA-REWARD which is aimed at making the rewards stable. Both are tested in automated simulations. From the results we may conclude that while DOMAIN-REWARD fails to improve the behaviour of the CGFs over the use of a binary reward function, the use of AA-REWARD leads to a 12.6% increase in win rates.

In Chapter 5, we investigate research question 3: *To what extent can knowledge built with dynamic scripting be transferred successfully between CGFs in different scenarios?* The behaviour models generated by the dynamic scripting algorithm contain knowledge about air combat situations. For instance, in Chapter 3 and Chapter 4, we used dynamic scripting to generate behaviour models for use by a two-ship in a two-versus-one scenario. We hypothesise that the knowledge contained in these models is to some extent reusable between different scenarios. We place a two-ship of CGFs in scenarios in which they have to learn to defeat two opponents, and then generate behaviour models for the two-ship. We do so *twice*: once, the two-ship has to learn to defeat the two opponents with a *tabula rasa*; the next time, the algorithm that generates the behaviour models for the two-ship is initialised with the behaviour models (in the form of weighted rules) that were generated in earlier two-versus-*one* scenarios. In each of the two-versus-two scenarios, we find that the two-ship using the transferred knowledge learns more effective behaviour than the other two-ship. Furthermore, they take less time to reach their highest level of performance.

In Chapter 6, we investigate research question 4: *How should we validate machine-generated air combat behaviour models for use in training simulations?* Validation is an important step in the development of behaviour models, since it provides a structured way to determine whether the models are useful with regards to their intended purpose (in our case, training simulations). However, there is no *one-size-fits-all* to the validation of behaviour models. Therefore, in Chapter 6, we develop a new validation procedure specifically for machine-generated air combat behaviour models. In brief, our procedure consists of three steps. The first step is recording human-in-the-loop simulations, in which human participants engage CGFs that are controlled by a sample of either (a) behaviour models that have been manually designed by human professionals, or (b) newly generated behaviour models. The second step is a structured assessment of the behaviour displayed by the CGFs in the recordings. The assessment is performed by expert assessors, by means of the newly developed Assessment Tool for Air Combat CGFs (ATACC). The third step is the use of the TOST method (two one-sided t-tests) to determine whether the assessments of the behaviour produced by the manually designed behaviour models are statistically equivalent to the assessments of the behaviour produced by the machine-generated behaviour models. If so, we consider the generated behaviour models to be valid for application within training simulations.

In Chapter 7, we investigate research question 5: *To what extent are air combat behaviour models generated by means of dynamic scripting valid for use in training simulations?* We apply the validation procedure that is developed in Chapter 6 to a set of newly generated behaviour models. As a baseline, we use a set of manually designed behaviour models that has been used in real-world training simulations. We perform human-in-the-loop simulations in which Royal Netherlands Air Force (RNLAF) fighter pilots engage CGFs controlled by the behaviour models. The behaviour displayed by the CGFs in the simulations is assessed by instructor pilots, by means of the ATACC. On the ATACC, the assessors rate the occurrence of nine examples of behaviour. Between the CGFs using the manually designed behaviour models and the generated behaviour models, six out of the nine examples of behaviour are rated as occurring in an equivalent manner. Based on this result, we can neither conclude to a complete validity of the generated behaviour models, nor to a non-validity. However, since the literature advises us to recognise degrees of success, we may conclude that our behaviour models are valid to a moderate extent.

In Chapter 8, we conclude the thesis by summarising the answers given earlier to the five research questions and the problem statement. Our research shows that dynamic scripting greatly facilitates the automatic generation of air combat behaviour models, while being sufficiently flexible to be moulded into answers to the challenges. However, ensuring the validity of the newly generated behaviour models remains to be a point of attention for future research.