# Calculated Moves: Generating Air Combat Behaviour

Toubman, A.

# 7 Validation of generated behaviour models in training simulations

In this chapter, 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?*

To answer this question, we implement the steps of the validation procedure that we presented in Chapter 6. Since we designed the validation procedure with five steps, we cover all steps in five sections (from Section 7.1 to Section 7.5) as follows. First, we describe the four 4P-models that together act as the baseline of the validation (Section 7.1). Second, we describe the 4M-models that together act as the subject of the validation (Section 7.2). Next, we discuss the human-in-the-loop simulations that are performed with the help of human F-16 pilots (Section 7.3). These pilots engage four-ships of CGFs that use the 4-models, in a manner that resembles the operation of training simulations. The behaviour of the CGFs in these simulations are assessed by a group of expert assessors (Section 7.4). We present the results of the assessments, including the equivalence tests that are performed (Section 7.5). Additionally, we discuss the results and our interpretation of the validity of the generated 4M-models (Section 7.6). Finally, we conclude this chapter by answering research question 5 (Section 7.7).

---

This chapter is based on the following publications.

- A. Toubman, J. J. Roessingh, P. Spronck, A. Plaat and H. J. Van den Herik (2016b). Rapid Adaptation of Air Combat Behaviour. In: *ECAI 2016 - 22nd European Conference on Artificial Intelligence*. Ed. by G. A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum and F. Van Harmelen. Vol. 285. Frontiers in Artificial Intelligence and Applications. The Hague, The Netherlands: IOS Press, pp. 1791–1796. DOI: 10.3233/978-1-61499-672-9-1791
- A. Toubman (2019). Validating Air Combat Behaviour Models for Adaptive Training of Teams. In: *Adaptive Instructional Systems*. Ed. by R. A. Sottilare and J. Schwarz. Springer International Publishing, pp. 557–571. DOI: 10.1007/978-3-030-22341-0_44

## 7.1 Defining the baseline: The 4P-models

We had to obtain four 4P-models that were written by professionals for use in training simulations. For this task we could rely on the work performed previously by a group of professionals (see Netherlands Aerospace Centre, 2017a) who had designed and worked out four 4P-models. These four 4P-models were inspected by us and considered to be fit for the task to be a sample that forms the baseline in the validation process (see Step 1, *Defining the baseline*, Section 6.6).

The differentiating factor between the four 4P-models was the starting formation of the involved CGFs. Each starting formation defines (1) the spatial configuration of the CGFs, and (2) their initial speeds. Therefore, the starting formation is an important factor in the interactions between the CGFs and the human participants in human-in-the-loop simulations. We refer to the four starting formations as $F_1$, $F_2$, $F_3$, and $F_4$. We return to the four starting formations in Section 7.2, where they are used in the 4M-models that are generated.

The 4P-models were modelled by the professionals in the SMART BANDITS[1] behaviour modelling program (Netherlands Aerospace Centre, 2017a). As such, the 4P-models were in the form of finite-state machines (FSMs). As a modelling technique, FSMs allow behaviour modellers to organise behaviour into different states, only one of which can be active at the same time (cf. Adam, Taillandier and Dugdale, 2017; Yildiz, Akcal, Hostas, Ure and Inalhan, 2018). Based on observations by the CGF that uses the behaviour model, the CGF enters a certain state in the model, and then only executes the behaviour belonging to that state. Models in the form of FSMs are easily displayed in a graphical manner, in contrast to, e.g., scripts. Especially as the number of rules in a script grows, it becomes difficult for the behaviour modeller to keep track of the possible interactions between the conditions and consequences of all rules in the script. Instead, FSMs clearly indicate which transitions between states are possible, and when these transitions are made.

## 7.2 Generating behaviour models: The 4M-models

Next, we had to generate four new 4M-models. We did so by means of machine learning, in the form of the dynamic scripting technique. This sample of 4M-models is the subject of the validation (see Step 2, *Generating models by means of machine learning*, Section 6.6).

Our goal was to generate a "counterpart" 4M-model to each of the four 4P-models. To generate these counterparts, we formulated two requirements for the 4M-models. First, we required that each 4M-model should use the same starting formation (either $F_1$, $F_2$, $F_3$, or $F_4$) as its counterpart 4P-model. The reasoning for using the same starting formations is that we viewed these starting formations as an essential part of the training simulations. Furthermore, reusing the same starting formations was a manner of forcing dynamic scripting to work within the same constraints as the

[1]The SMART BANDITS program is introduced in Appendix D as part of the Fighter 4-Ship simulator.

professionals do, when modelling the behaviour of the CGFs. Therefore, pairing each 4P-model with a counterpart 4M-model allows for a fair comparison between the modelling capabilities of the professionals and dynamic scripting.

As the second requirement for the counterparts, we required the 4M-model counterparts to use the same modelling technique as the 4P-models. In pretests when we trained ourselves with the problem at hand, we detected an important difference between (a) the behaviour that is produced by scripts such as generated by dynamic scripting, and (b) the behaviour that is produced by FSMs such as created in SMART BANDITS. In a direct comparison, the behaviour produced by the scripts appeared to be erratic and indecisive at certain moments during the simulations. We attribute this indecisiveness to unforeseen interactions between some rules and specific observations made by the CGFs. As an example, consider the case where a red CGF is simultaneously (a) attacking a blue CGF, as well as (b) being attacked by another blue CGF. Due to small changes in the movements of the two blue CGFs affecting the firing of red's offensive and defensive rules, the red CGF would appear to oscillate between (a) continuing to attack the first blue CGF, and (b) defending against the attack of the second blue CGF. This behaviour, while possibly (and unexpectedly) quite effective in automated simulations, is very *unhumanlike* and therefore unacceptable in a real-world human-in-the-loop simulation.

Rather than attempting to augment the rules to prevent this behaviour, we decided to modify the dynamic scripting algorithm, enabling it to generate FSMs. We will elaborate on the reason for generating FSMs by means of dynamic scripting in Appendix E. In this appendix, we also describe the modifications that we made to the dynamic scripting algorithm. In brief, we divided FSMs into their constituent states and transitions, and then treated these states and transitions as rules for use in dynamic scripting's rulebase. For simplicity, we continue to use the term *rules* and *rulebase* in the remainder of this chapter.

Below, we further discuss how the 4M-models were generated. First, we briefly discuss the origin of the rules that were used in the rulebases of the CGFs (Subsection 7.2.1). Thereafter, we describe the automated simulations by means of which we generated the 4M-models that later acted as the subjects of the validation (Subsection 7.2.2).

### 7.2.1 The rules in the rulebases

Because dynamic scripting requires a rulebase with rules in order to generate a behaviour model, we had to consider an appropriate source for the rules. We chose to derive the rules from the four 4P-models that we had available (see Section 7.1). To do so, we divided the 4P-models into their constituent states and transitions. From these states and transitions, we extracted any states and transitions related to the starting formations of the CGFs. Then, we removed any duplicates. The remaining rules formed the rulebase to be used by dynamic scripting.

Because dynamic scripting only recombines rules, and does not synthesise any new rules, the algorithm could only generate FSMs that closely resembled the 4P-model. Therefore, we

augmented the rulebase with rules that we call variant rules. Each variant rule was based on one of the rules that already existed in the rulebase. In each variant rule, we made one or more small changes compared to the rule on which the variant was based, in terms of altered values such as (but not limited to) headings, time-outs, and sensor readings. For example, if a state directed the CGF to turn 90 degrees, we added also a variant of that state which directed the CGF to turn $-90$ degrees. The rationale behind the altered values was to choose values that were (1) sensible (e.g., not firing all missiles at once) and (2) meaningful (viz. rather adding a few variants with large changes in values, than adding many variants with small changes in values). The variant rules were added to the rulebase alongside the existing rules.

Additionally, during the translation of the states and transitions to the rules, we discovered that transitions leading from one state to another were tightly coupled to the states from which the transitions originated. Therefore, rather than implementing each transition as a separate rule, we embedded each transition into the rule that defined the state from which the transition originated.

So, we created sixteen copies of the rulebase. These copies formed four groups of four rulebases. Each group of four rulebases served as a starting point for one of the four 4M-models that were generated (see Subsection 7.2.2). Finally, we assigned one of the four starting formations to each of the four groups, and then added this starting formation as a rule to each rulebase in that group.

### 7.2.2 Automated simulations

In the end, we generated four 4M-models by means of automated simulations. In this subsection, we describe the strategy by which we did so. The strategy, which we refer to as the generation strategy, consists of three steps. Figure 7.1 shows the three steps graphically. Below, we discuss the three steps of our generation strategy.

**Step 1.** Four red CGFs engaged four blue CGFs in simulated air-to-air combat encounters. The reds learned by means of dynamic scripting, making use of a group of four rulebases (see Subsection 7.2.1). The reds approached the blues in the starting formation as was programmed in their rulebases. The blues were scripted to approach the reds as described in their own starting formation, which we call starting formation A. Once the blues detected the reds, the blues were scripted to attack the reds, only interrupting their attack to perform defensive manoeuvres if the blues were under attack themselves. The reds were allowed to learn over the course of 40 encounters. They coordinated their actions by means of the DECENT coordination method. Each encounter ended when either (a) each CGF in a team had been hit by a missile from the opposing team, or (b) a time limit of ten minutes was reached. We applied BIN-REWARD[2] as the reward function for the red team: the red team

[2]We preferred the use of AA-REWARD here. However, at the time, we were unable to correctly implement it in the STAGE simulation environment.

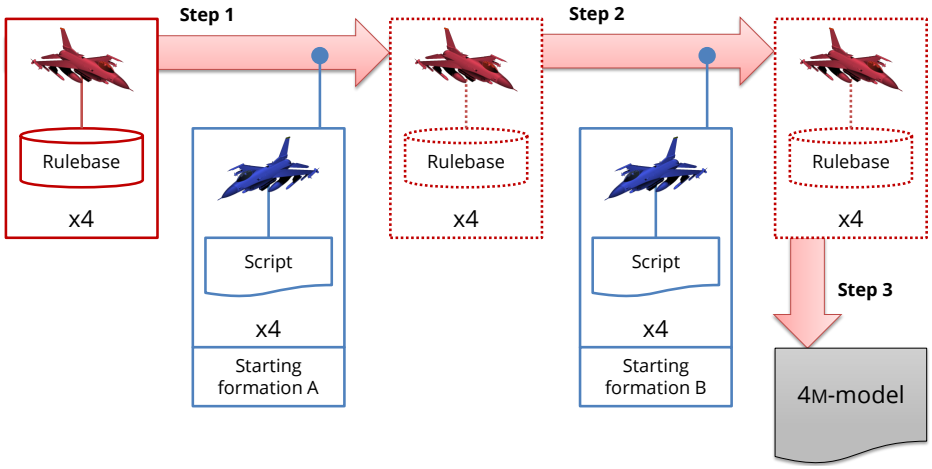


**Figure 7.1** The three steps of the generation strategy. Step 1: the reds learn to defeat a four-ship of blues (which use starting formation A) over the course of 40 encounters. After these encounters, the rulebases (shown with dotted lines) are optimised towards defeating blues that use starting formation A. Step 2: the same reds (viz. using the same rulebase and the weights therein) learn to defeat a different four-ship of blues (which use starting formation B) over the course of 40 encounters. After these encounters, the rulebases (again shown with dotted lines) are optimised towards defeating blues that use either starting formation A or B. Step 3: rules are extracted from the rulebases to create a 4M-model.

was awarded a reward value of 1 if the blue team was defeated, and a reward value of 0 otherwise.

**Step 2.** We took the simulation from Step 1, and replaced the starting formation of the blues by a new starting formation, called starting formation B. The remainder of the simulation was left unchanged, including the (weights in) the rulebases of the reds. Thereby, we essentially transferred the reds and their knowledge (see Chapter 5) from the simulation in Step 1, to the simulation in Step 2. Next, the reds were allowed to learn to defeat the blues with starting formation B over the course of 40 encounters.

**Step 3.** We formed a script out of the rules of each rulebase. We did so in the following manner. First, we divided the rulebase into groups. Each group contained one of the original rules, plus its variants (see Subsection 7.2.1). Next, out of each of these groups of rules, we selected the rule with the highest weight for inclusion in the script. In case of a tie between the weights of two rules, a rule was selected at random. This way, we ensured that the rules in the resulting script together formed a complete and functional FSM.

By applying the generation strategy, we obtained four scripts. These scripts together form a

single 4M-model. We repeated the generation strategy for each of the four 4M-models that we wished to generate. This resulted in the four 4M-models that were the counterparts to the four 4P-models.

The simulations described in this section were performed in the STAGE simulation environment, which is part of the Fighter 4-Ship simulator (see Appendix D). To allow CGFs in STAGE to learn by means of dynamic scripting, we implemented the dynamic scripting algorithm in the form of a new program. We call this program STAGEDS. STAGEDS used the application programming interface (API) of STAGE to control (a) the CGFs in the simulations, as well as (b) the simulations themselves (viz. starting, stopping, and restarting the simulations) in order to automate the simulations as required for the learning process of the red CGFs.

## 7.3 Human-in-the-loop simulations

We use human-in-the-loop simulations to determine how a four-ship of red CGFs behaves when the CGFs interact with human participants (see Step 3, *Human-in-the-loop simulations*, Section 6.6). The simulations were performed in the Fighter 4-Ship simulator.

The behaviour of the reds was controlled by means of eight 4-models: the four 4P-models (see Section 7.1) plus the four 4M-models (see Section 7.2). Using these eight 4-models, we defined eight *scenarios*. Each scenario was a simulation configuration in which a four-ship of red CGFs approached the human participants from the simulated north. In each scenario, the red four-ship used either (a) one of the four 4P-models or (b) one of the four 4M-models, so that each of the 4-models was used in one of the scenarios.

The human participants in the simulations were active-duty Royal Netherlands Air Force (RNLAF) F-16 pilots from Volkel Airbase (all male, n = 16, age $\mu = 32.0$, $\sigma = 5.35$), and one former RNLAF F-16 pilot (age = 60).[3] No selection criteria were applied. The active-duty pilots were assigned to the human-in-the-loop simulations based on availability. Experience levels ranged from *wingman* to *weapons instructor pilot*.

Over the course of three days, five teams of four participants controlled the blue CGFs in the Fighter 4-Ship. Before the simulations took place, the participants received a "mission briefing" document that described (1) the capabilities of the blue CGFs that they would control, and (2) the capabilities of the red CGFs that the participants were to expect in the simulator. The eight scenarios were presented sequentially in a random order. The participants were unaware of the origin of the 4-models controlling the red CGFs (i.e., the simulations were performed in a

[3] One of the active-duty participants had to leave after four scenarios. This situation presented us with three options: (1) continue without this participant (viz. with a three-ship), (2) cancel the remaining simulations, or (3) substitute the participant with a former F-16 pilot who was available. Since the participant had a non-commanding role in the four-ship, we deemed his influence in the decision-making of the human participants to be minimal. Still, by controlling the fourth blue CGF, he provided valuable input that allowed the red CGFs to function. Furthermore, participants were scarce. We decided that the collection of data was paramount, and let the former F-16 pilot (mentioned above) substitute the participant in the remaining simulations.

single-blinded fashion). Each scenario ended when either all four red CGFs, or all four human participants were defeated.

The human-in-the-loop simulations were recorded using Personal Computer Debriefing System (PCDS). These recordings included (1) the voice communication that took place among the human participants, and (2) video recordings of the multi-functional displays (MFDs) of the ships occupied by the human participants. In total, 33 recordings[4] were stored.

## 7.4 Behaviour assessments

The behaviour that the reds displayed in the human-in-the-loop simulations were assessed by human experts (see Step 4, *Assessments*, Section 6.6). Active-duty RNLAF F-16 pilots from Leeuwarden Airbase acted as assessors (all male, $n = 5$, age $\mu = 35.2$, $\sigma = 5.17$). Assessors were selected on having *tactical instructor pilot* or *weapons instructor pilot* qualification. We considered either of these qualifications to be sufficient in order to function as the *training specialist* that the validation criterion calls for. All five assessors had the weapons instructor pilot qualification.

The assessments were performed by means of the ATACC. We implemented the rating items of the ATACC as a single-page paper form. In our implementation, we made three additions to the rating items: (1) we added a field for the *tactical* (i.e., the code name) of the assessors for later reference, (2) we added a field for the *operational status* of the assessors to gain insight into their experience level, and (3) we added two fields for indicating the specific recorded encounter that was viewed by the assessor. The form is presented as it was used in the behaviour assessments in Appendix F.

Originally, we had planned to let each assessor assess all of the 33 recordings within a three hour time span. However, a pilot study with two weapons instructor pilots (not counted above) revealed that this was infeasible. We subsequently reduced the pool of recordings available for rating to 16 recordings. These 16 recordings came from two teams that completed all eight scenarios (i.e., simulations with the four 4P-models and the four 4M-models) in human-in-the-loop simulations. From this reduced pool of recordings, we assigned ten recordings to each rater, consisting of (1) eight recordings from one of the two teams in random order, and (2) two recordings from the other team. Furthermore, the weapons instructor pilots in the pilot study expressed that they were unable to adequately assess the intelligence of the red CGFs (rating item 8) and the extent to which the red CGFs tested the skills of the pilots in the simulator (rating item 9) without knowing the experience levels of these pilots. Based on this feedback, we made the decision to disclose the experience levels to the assessors during the assessments.

For the assessments, the assessors were provided with (1) a laptop computer with mouse and headphones, (2) a stack of ten ATACCs, and (3) an instruction sheet. The PCDS recordings were

[4]Two teams were not available to complete all eight scenarios. Together, these two teams completed nine scenarios: the eight scenarios, plus one duplicate.

opened on the computer. Each ATACC was marked with a unique code that referred to a specific recording in PCDS. The assessors were instructed to view the recordings in the order as indicated by their ATACCS.

We planned two analyses on the responses to the ATACC: (1) *equivalence testing* on the responses to the ATACC, and (2) calculating of the *inter-rater reliability*. We briefly describe them below.

**Equivalence testing.** We apply a method known as TOST (cf. Meyners, 2012; Anderson-Cook and Borror, 2016; Lakens, 2017) on the responses to the ATACC to determine the extent of the validity of the 4M-models. Equivalence testing is part of the validation procedure (see Step 5, *Equivalence testing*, Section 6.6).

**Inter-rater reliability.** We calculate the intraclass correlation (ICC) as a measure of inter-rater reliability, viz. how consistently recordings are rated between assessors. We did not include the calculation of the ICC in the validation procedure. However, since the number of assessors in our validation is limited, the ICC serves as an indication of the trustworthiness of the assessments.

## 7.5 Results of the behaviour assessments

A summary of the responses to the ATACC is given in Table 7.1. The responses to the Likert scale rating items were coded as integer values ranging from 1 (Never/Strongly disagree) to 5 (Always/Strongly agree). The coding for rating item four (*Blue air was able to fire without threat from red air*) was inverted so that the values reflected the occurrence of red behaviour (i.e., red influencing blue's ability to fire). Below, we present the results of the equivalence tests (Subsection 7.5.1) and the inter-rater reliability analysis (Subsection 7.5.2). Furthermore, we include a brief review of feedback that was received from the assessors during the assessments (Subsection 7.5.3).

### 7.5.1 Equivalence testing

We applied Schuirmann's (1987) TOST method to determine the equivalence of (1) the responses given on the ATACC for 4P-models, and (2) the responses given on the ATACC for 4M-models. We calculated $\delta$ (as Juzek's $\delta$) for the responses to each rating item of the ATACC, and then performed the TOST on the responses to each rating item. The TOST was performed using the `TOSTtwo.raw` function from R's `TOSTER` package, with Welch's $t$-test as the underlying one-sided test. We chose to use Welch's $t$-test here because of the unequal sample sizes.[5] The $\delta$ and the results of the TOST

[5]There is an ongoing discussion on the topic of whether parametric tests such as the $t$-test are suitable for use on ordinal Likert-scale data. Parametric tests have on multiple occasions been shown to be robust against violated assumptions (such as non-normal, ordinal data) (cf. Norman, 2010; De Winter, 2013; Derrick and White, 2017). Using parametric tests in our TOST allows us to use well-tested, publicly available tools such as the mentioned R package.

($t$-value, degrees of freedom [$df$], $p$-value, and the 90% confidence interval (CI) of the difference of the means) are shown in Table 7.2. In Table 7.2, the bold $p$-values indicate a significant result of the TOST. Based on the results of the TOST, we may conclude that the responses to rating items 1, 2, 5, 7, 8, and 9 are equivalent between the 4P-models and the 4M-models.

The TOST did not find equivalence for rating items 3, 4, and 6. For these rating items, we conducted a follow-up test to determine if the responses to these rating items significantly differed between the 4P-models and the 4M-models. This follow-up test was a standard two-sided Welch's $t$-test. A significant difference was found for rating items 3 and 6. These two rating items read as *Red air was within factor range* (rating item 3), and *Red air acted on blue air's weapons engagement zone* (rating item 6). For both rating items, the responses indicated a higher frequency of the behaviour that was rated for the 4M-models (see Table 7.1). The remaining rating item read as *Blue air was able to fire without threat from red air* (rating item 4). The responses to rating item 4 were neither significantly equivalent, nor significantly different. Therefore, we may conclude that their relationship is undecided.

### 7.5.2 Inter-rater reliability

An inter-rater reliability analysis was carried out on the nine rating items of the ATACC. The ICC estimate and its 95% CI were calculated using the `icc` function from R's `irr` package, based on a two-way random effects model (consistency, multiple raters/measurements) (cf. Koo and Li, 2016). The ICC estimate and its 95% CI are shown in Table 7.3. The reported values indicate moderate agreement between the assessors (Koo and Li, 2016).

### 7.5.3 Feedback on the assessments

The assessors that took part in the validation provided direct verbal feedback during and after the assessments. The feedback concerned both (a) the ATACC questionnaire, and (b) the simulations that were shown. Below, we briefly review the feedback that we received.

A general topic of feedback on the ATACC questionnaire was its reliance on the insight (or "gut feeling") of the assessors over quantifiable measures. Assessors noted that they were trained to deal with quantifiable measures, and as such on occasion they found it difficult to assess the behaviour of the CGFs along the rating items of the ATACC. However, the assessors also understood that if the behaviour could be defined completely in quantifiable terms, their insight would not have been required.

Rating item 7 (*Red air flew with kinematic realism*) was a frequent subject of comments by the assessors. This rating item was either called unclear, or the assessor stated that he did not have the means to assess the flying performance of the red CGFs. Table 7.1 shows that out of the nine rating items, the fewest responses were collected for this rating item.

One assessor commented that several manoeuvres that the CGFs (using 4M-models, unknown to the assessor) displayed were interesting from a training perspective, but also unrealistic

**Table 7.1** Summary of the ATACC responses: the number of responses ($n$), mean response ($\mu$), and standard deviation ($\sigma$) of the responses to the ATACC rating items for the 4P-models and the 4M-models. The highest means (viz. behaviours that were observed the most) and the lowest standard deviations (viz. the most agreement between the raters) are highlighted.

| | 4P-models | | | 4M-models | | |
|---|---|---|---|---|---|---|
| Rating item | $n$ | $\mu$ | $\sigma$ | $n$ | $\mu$ | $\sigma$ |
| 1 | 28 | 3.04 | 0.79 | 24 | 3.25 | 0.99 |
| 2 | 28 | 2.07 | 0.98 | 24 | 2.33 | 1.13 |
| 3 | 28 | 3.18 | 1.19 | 24 | **3.92** | 1.02 |
| 4 | 27 | 2.26 | 0.86 | 24 | 2.71 | 0.91 |
| 5 | 28 | 3.29 | 0.71 | 24 | 3.42 | **0.58** |
| 6 | 28 | 2.75 | 0.89 | 24 | 3.33 | 0.70 |
| 7 | 22 | **3.82** | **0.66** | 20 | 3.70 | 0.73 |
| 8 | 28 | 2.86 | 0.80 | 24 | 2.96 | 0.69 |
| 9 | 27 | 3.81 | 0.68 | 24 | 3.63 | 0.65 |

**Table 7.2** Results of the TOST method per rating item (r.i.). The TOST was based on Welch's $t$-test. For rating items where the TOST method did not find equivalence, an additional standard (Welch's) $t$-test was performed. Significant $p$-values at the $\alpha = 0.05$ level are indicated in bold. The relevance (rel.) of the outcome of the tests is indicated in the rightmost column.

| | TOST | | | | | Standard $t$-test | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| R.i. | $\delta$ | $t$ | $df$ | $p$ | 90% CI | $t$ | $df$ | $p$ | 95% CI | Rel. |
| 1 | 0.798 | 2.322 | 43.9 | **.012** | [−0.637, 0.208] | | | | | eq. |
| 2 | 0.944 | 2.307 | 45.9 | **.013** | [−0.758, 0.234] | | | | | eq. |
| 3 | 1.000 | 0.855 | 50.0 | .198 | [−1.251, −0.225] | -2.41 | 50.0 | **.020** | [−1.353, −0.124] | diff. |
| 4 | 0.800 | 1.414 | 47.5 | .082 | [−0.866, −0.032] | -1.81 | 47.5 | .077 | [−0.949, 0.050] | und. |
| 5 | 0.590 | 2.551 | 49.9 | **.007** | [−0.432, 0.170] | | | | | eq. |
| 6 | 0.725 | 0.643 | 49.7 | .262 | [−0.953, −0.214] | -2.64 | 49.7 | **.011** | [−1.018, −0.149] | diff. |
| 7 | 0.697 | -2.674 | 38.5 | **.005** | [−0.247, 0.483] | | | | | eq. |
| 8 | 0.677 | 2.779 | 50.0 | **.004** | [−0.448, 0.246] | | | | | eq. |
| 9 | 0.604 | -2.223 | 48.8 | **.015** | [−0.122, 0.502] | | | | | eq. |

eq. = equivalent, diff. = different, und. = undecided

**Table 7.3** Results of the intraclass correlation analysis.

| | | F-test with true value 0 | | | |
|---|---|---|---|---|---|
| ICC | 95% CI | value | $df1$ | $df2$ | $p$ |
| 0.651 | [0.494, 0.770] | 2.86 | 63 | 252 | .000 |

as the manoeuvre did not seem to provide a direct tactical advantage. This is an example of the creativity offered by machine learning. However, in particular this case, the creativity was detrimental to the realism of the behaviour. In the future, it may be possible to detect and filter out such manoeuvres from generated behaviour models.

## 7.6 Discussion

In this section, we discuss the results from the behaviour assessments. These results are the foundation on which we base our perception of the validity of the 4M-models. In our validation procedure, we defined the extent of the validity of our 4M-models as the extent to which these models were assessed as equivalent to the 4P-models that were obtained. Below, we cover the following five topics: our key finding (Subsection 7.6.1) and the context in which it should be interpreted (Subsection 7.6.2), the implications of the finding (Subsection 7.6.3), and the limitations of the study (Subsection 7.6.4).

### 7.6.1 Key finding

Our key finding is that out of the nine rating items of the ATACC, six items are assessed as equivalent between the 4M-models and the 4P-models by expert human assessors. Of the remaining three rating items, the responses to two rating items (i.e., rating items 3 and 6) were found to be statistically different between the 4P-models and the 4M-models, whereas the responses to one rating item (i.e., rating item 4) were found to be inconclusive. Although the responses to these three rating items do not directly support the validity of the 4M-models, the responses to rating items 3 and 6 indicate that the behaviour produced by the 4M-models was perceived as *more* challenging (rather than *less* challenging) than the behaviour produced by the 4P-models. Therefore, we nonetheless consider these responses to be a positive signal (and to some extent in support of our key finding) for the use of machine-generated behaviour models in training simulations.

As mentioned in Chapter 6, degrees of success in a validation study must be "recognized and accepted" (Birta and Arbez, 2013), since it is practically impossible to "completely validate" behaviour models. Although the ATACC can certainly be improved, we have successfully used it to demonstrate that machine learning is capable of generating behaviour models that are perceived as equivalent, at least on six out of the nine rating items, to behaviour models that have been manually written by professionals. We interpret these results as a moderately strong indicator for validity of the generated models regarding the application of training simulations.

### 7.6.2 Placing our key finding in context

In contrast to Chapters 3 to 5, which together demonstrated the problem-solving power of machine learning in automated simulations, this chapter addressed the application of a machine

learning technique in a setting dominated by humans. We applied machine-generated behaviour models in human-in-the-loop simulations, and then worked with human assessors to assess the behaviour produced by the models. Furthermore, the simulations and the assessments that followed were only possible after consulting human subject matter experts on the best ways to assess the behaviour of opponent CGFs in training simulations, and then capturing this knowledge in the validation procedure (see Chapter 6). Clearly, applying machine learning in training simulations is as much a social challenge as it is a technical one.

In Chapter 6, we put a large amount of effort in critically considering each step of our validation procedure. We consider this to be a major strength of our validation study, aiding in both social and technical acceptance of our proposed use of machine learning. The inclusion of subject matter experts in the formulation of this procedure enables the procedure to focus on the envisioned usage of the behaviour models, viz. opponent behaviour in training simulations. For contrast, we briefly consider one of the few related studies that made use of human-in-the-loop simulations. Teng et al. (2013) applied both (a) adaptive and (b) non-adaptive machine-generated behaviour models in WVR air combat simulations involving human fighter pilots, with the goal of improving training simulations. The fighter pilots were presented with questionnaires on which they could assess six properties of the behaviour produced by the two kinds of models, i.e., to what extent the behaviour was perceived to be (1) predictable, (2) intelligent, (3) skillful, (4) challenging, (5) adaptive, and (6) aggressive. It is not mentioned why these specific properties were used, or what the desirable assessment scores would be for the intended application of the generated behaviour models. In our view, studies regarding the use of machine learning in training simulations would greatly benefit from involving training experts and other subject matter experts at an early stage, so that the research can be more focused on the potential added value to the training simulations, and therefore to the humans that depend on the simulations.

### 7.6.3 Implications

To the best of our knowledge, the validation study that we present in this chapter is the first of its kind in the context of BVR air combat training simulations, using behaviour models generated by means of machine learning. As such, an important step has been made in bringing a machine learning application to the area of military training simulations. Furthermore, the equivalence of the responses to six of the rating items show, if not complete validity, at least a large potential for the use of machine learning in this area. As a next step, the behaviour models that are currently available for use in training simulations could for instance be supplemented by machine-generated models, in order to simultaneously (1) provide more variation in the training, and (2) gather further experience with applying machine-generated models in a real-world training setting.

### 7.6.4 Limitations

While we have critically reviewed our validation procedure (see Chapter 6) and implemented it to the best of our abilities with the available resources (this chapter), two limitations affect our study. First, the ATACC questionnaire was only informally validated in preliminary simulations. Second, the number of assessors in the validation was limited. Both can be attributed to our attempt to optimise the use of limited resources. Disregarding these limitations somewhat, we hope that the results of our study may serve as an incentive for further research in this area, including, e.g., a refinement of the ATACC as a research instrument. Such a refinement should increase (a) the inter-rater reliability, and therefore also (b) the value of the assessments. One approach might be the inclusion of mission essential competencies (MECS) for human F-16 pilots (cf. Alliger, Beard, Bennett Jr, Symons and Colegrove, 2013; Tsifetakis and Kontogiannis, 2017) in the ATACC.

## 7.7 Answering research question 5

In this chapter, we investigated the validation of machine-generated behaviour models. Specifically, we addressed research question 5: *To what extent are air combat behaviour models generated by means of dynamic scripting valid for use in training simulations?*

To answer research question 5, we apply the validation procedure that we presented in Chapter 6. We generated behaviour models by means of dynamic scripting, and then used these behaviour models to control CGFS in human-in-the-loop simulations. Equivalence testing shows that on six of the nine rating items of the ATACC, the CGFS that are controlled by machine-generated behaviour models are rated equivalently to CGFS that are controlled by behaviour models written by professionals.

Answering the research question very precisely proves to be quite difficult. While we could, for instance, translate six out of nine rating items to 66.667 %, we consider such a percentage to be meaningless regarding validity. In our view, the results appear to *moderately* indicate validity, but the responses to the remaining three rating items do not support the notion of validity as we have defined it for ourselves. Therefore, we answer research question 5 as follows: the air combat behaviour models generated by means of dynamic scripting are *to a moderate extent* already valid for use in training simulations. In the future, this will certainly improve.