# Calculated Moves: Generating Air Combat Behaviour

Toubman, A.

# 5 Transfer of knowledge between scenarios

In this chapter we investigate research question 3: *To what extent can knowledge built with dynamic scripting be transferred successfully between CGFs in different scenarios?*

As the complexity of air combat scenarios increases, so does the time to learn good behaviour in these scenarios. Decreasing the learning time in complex scenarios may be possible by reusing the knowledge about air combat that has been stored in previously generated behaviour models. The reuse of previously gained knowledge in order to learn and solve new problems is known as *transfer learning* (see Definition 5.1). Transfer learning has been shown to have the potential of shortening the learning time between domains that are sufficiently similar (cf. Lu, Behbood, Hao, Zuo, Xue et al., 2015). In this chapter, we examine a particular use case, namely the transfer of knowledge from air combat CGFs that have learned to win two-versus-*one* scenarios, to CGFs that have to learn how to win two-versus-*two* scenarios. Next, we compare the performance of the CGFs with the transferred knowledge to that of CGFs that learn to behave from scratch in the two-versus-two scenarios. By doing so, we aim to determine to what extent the previously built knowledge aids in the development of behaviour models in the two-versus-two scenarios.

This chapter is organised as follows. First, we introduce the concept of transfer learning (Section 5.1). Next, we present the use case that we use as the foundation of our study of transfer learning in this chapter (Section 5.2). Guided by the use case, we conduct an experiment in which we transfer knowledge between two distinct two-ships of CGFs and then determine the success of the transfer (Section 5.3). We present the results of the experiment (Section 5.4) and discuss them (Section 5.5). Finally, we summarise the chapter and answer research question 3 (Section 5.6).

---

This chapter is based on the following publication.

- A. Toubman, J. J. Roessingh, P. Spronck, A. Plaat and H. J. Van den Herik (2015b). Transfer Learning of Air Combat Behavior. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. Miami, Florida: IEEE Press, pp. 226–231. DOI: 10.1109/ICMLA.2015.61

## 5.1 The concept of transfer learning

Transfer learning is a range of methods for reusing knowledge and skills that were gained when performing one task, and that are now applied on another task (cf. Taylor and Stone, 2009; Pan and Yang, 2010; Lu et al., 2015). The idea behind transfer learning is that it is easier for an agent (e.g., a CGF, robot, or software program) to perform a new task when the agent has already learned to perform a similar task. We refer to the task which the agent has already learned to perform as the *source task*. The new task is referred to as the *target task*. Below, we define the concepts of (1) transfer learning, (2) source task, and (3) target task.

**Definition 5.1** (Transfer learning)**.** Transfer learning is defined as learning to perform a target task by reusing knowledge that was previously gained on a source task.

**Definition 5.2** (Source task)**.** In the context of transfer learning, a source task is an intermediate task that a CGF has to learn to perform well, in order for the CGF to gain knowledge that may be useful for performing a different (but still similar) task.

**Definition 5.3** (Target task)**.** In the context of transfer learning, the target task is the task of interest, viz. the task that has similarities with the source task and that we desire to be performed by a CGF.

In the remainder of this section, we continue our introduction of transfer learning by briefly discussing three related topics: transfer learning methods (Subsection 5.1.1), transfer learning in reinforcement learning (Subsection 5.1.2), and transfer learning in dynamic scripting (Subsection 5.1.3). Finally, we conclude the section with a note on the burden of human knowledge (Subsection 5.1.4).

### 5.1.1 Transfer learning methods

Transfer learning methods have been successfully applied in classification, regression and clustering tasks (cf. Lu et al., 2015; Shao, Zhu and Li, 2015; Day and Khoshgoftaar, 2017). In these tasks, due to model availability and the time it takes to train new models, it can be desirable to reuse old models on new data. However, when the new data has different features or a different distribution, the models will have to be adapted. In these cases, the knowledge stored in the old models should be reused as efficiently as possible. The research on transfer learning methods concerns itself with studying effective ways for the reuse of this knowledge.

An example of transfer learning in practice is the work by Ferrucci, Brown, Chu-Carroll, Fan, Gondek et al. (2010). In 2011, an artificial intelligence system called Watson competed with human contestants in the open-domain question-answering television program *Jeopardy!* (Ferrucci et al., 2010). As part of Watson's preparation, Ferrucci et al. tested Watson's question-answering capabilities on various question-answer databases. In one instance, Watson's capabilities improved

significantly on a new, closed-domain database, by first allowing it to learn to answer *Jeopardy!*-style questions.

### 5.1.2 Transfer learning in reinforcement learning

Transfer learning has been identified as a useful tool in reinforcement learning (see, e.g., Lazaric, 2012; Bianchi, Celiberto Jr, Santos, Matsuura and De Mantaras, 2015; Hou, Ong, Feng and Zurada, 2017a; Spector and Belongie, 2018). In reinforcement learning, transfer learning enables the reuse of previously learned behaviours (Bou Ammar, Chen, Tuyls and Weiss, 2014). For example, we consider an air combat scenario as the source task, and a more difficult air combat scenario as the target task. In both tasks, the learning CGF needs to discover that it should (1) fire missiles in order to win the scenario, and (2) evade missiles that were fired by the opponent(s) in order not to lose the scenario.

If the source task is relatively easy to complete, we may expect that the learning CGF will quickly discover relevant behaviours for which the CGF will be rewarded. When the CGF (including its knowledge) is transferred from the source task to the target task, the CGF may find that the behaviours for which it was rewarded in the source task are also applicable in the target task. Applying that knowledge then may speed up learning the remaining behaviours that are needed to perform the target task. If the CGF is not transferred, it starts with a *tabula rasa* instead, and then needs to begin to discover the behaviours that are necessary to win the scenario.

### 5.1.3 Transfer learning in dynamic scripting

The dynamic scripting algorithm, as we have used it in our research thus far, requires predefined knowledge (in the form of rules in a rulebase) to function. While learning to perform a task, a CGF using dynamic scripting builds up new knowledge about which rules are required to perform the task. This knowledge is stored as the weights that are associated to the rules. In the context of air combat simulations, the tasks (viz. the scenarios) that we use in our experiments are quite similar. For instance, in all scenarios, the opponent has to be hit by a missile, before the learning CGF is hit by one of the opponent's missiles. Therefore, rules that, e.g., enable missile-firing behaviour are required to have high weights in order to win each scenario.

An interesting question that now arises is the following: in the air combat domain, to what extent does the knowledge that was built up in one scenario (i.e., the source task) affect the performance and any further learning in a second scenario (i.e., the target task)? On the one hand, if a transfer of knowledge in this domain leads to higher performance on the target task at a faster rate than CGFs that have to learn to perform the target task starting with zero knowledge, such a transfer may speed up the development of challenging CGFs for real-world training simulations. On the other hand, if we find that the knowledge brought along from an earlier air combat scenario hampers the performance on the next scenario (see Subsection 5.1.4), measures should be taken to erase this knowledge between scenarios.

### 5.1.4 The burden of human knowledge

In contrast to the expectation that learning to perform the source task aids in learning to perform a related target task, the literature has also shown that it is possible that too much (human) knowledge becomes a burden for an agent such as a CGF. A high-profile example of the "burden of knowledge" is the difference between the ALPHAGO program (Silver et al., 2016), and the related ALPHAGO ZERO program (Silver et al., 2017b). Both ALPHAGO and ALPHAGO ZERO are machine learning programs that are designed to play the game of Go by means of a combination of deep neural networks and Monte-Carlo tree search. The ALPHAGO program learned successfully to play Go by using some 24 million recorded games that were played by humans as training data, and then using some 16 million games from self-play. This was sufficient to defeat the reigning world Go champion Ke Jie (see, e.g., Chao, Kou, Li and Peng, 2018). However, a superior ALPHAGO ZERO was developed thereafter. The new program learned to play Go by self-play only, i.e., by starting *tabula rasa* and having only the rules of the game at its disposal.

Furthermore, a more generalised version of ALPHAGO ZERO called ALPHAZERO was able to play two games in addition to Go, namely the games of chess and shogi (Silver, Hubert, Schrittwieser, Antonoglou, Lai et al., 2017a). This was achieved in part by taking away assumptions about the game of Go, such as symmetry of the board caused by certain reflections and rotations. In summary, the ALPHAGO family of programs is an example of how injected knowledge and biases can hamper performance. It remains to be seen whether this is also the case in the learning of air combat behaviour. In any case, we should be aware of the fundamental difference between human knowledge and machine-generated knowledge from scratch.

## 5.2 Use case

In this section, we present our use case for transfer learning in air combat simulations. Below, we first describe the use case (Subsection 5.2.1). Then, we explain our implementation of the use case using dynamic scripting (Subsection 5.2.2).

### 5.2.1 Description

The use case entails the transfer of knowledge built up by a two-ship of CGFs to a second two-ship. The first two-ship (henceforth: the reds′) builds up its knowledge by learning to defeat an opponent in a two-versus-one air combat scenario. The second two-ship (henceforth: the reds″) uses this knowledge to learn to defeat *two* opponents in four different two-versus-two scenarios. Thus, in our use case the two-versus-one scenario is the source task. By extension, the two-versus-two scenarios are the target tasks.

In order to determine to what extent the transferred knowledge benefits the performance of the reds″ in the two-versus-two scenarios, we introduce a third two-ship (henceforth: the

reds$_0$). Like the reds$''$, the reds$_0$ learn to defeat two opponents in the two-versus-two scenarios. However, unlike the reds$''$, the reds$_0$ do so without any transferred knowledge (hence the zero in the name).

### 5.2.2 Implementation in dynamic scripting

In this subsection, we describe how we implement the use case by means of CGFs that learn by dynamic scripting. In brief, we implement a transfer of knowledge between two CGFs that learn by means of dynamic scripting by copying the rulebase, including the weights, from one CGF to the other CGF. The complete implementation of the use case consists of three steps. Below, we describe each of the three steps.

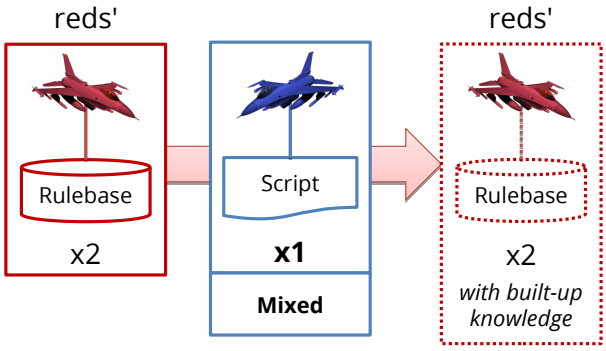


**Figure 5.1** Step 1 of the implementation of the use case. A two-ship of red CGFs (the reds′) learns to defeat a blue CGF in the two-versus-one mixed scenario. The learning process of the reds′ leads to new knowledge in the rulebases (in the form of weights).

**Step 1. The reds′ build up knowledge in a two-versus-one scenario.** The reds′ learn to defeat a blue CGF in the mixed two-versus-one scenario. The mixed scenario is combination of three two-versus-one scenarios: (1) the basic scenario, (2) the close range scenario, and (3) the evasive scenario. The mixed scenario and its constituent scenarios are described in Appendix A.4.1. The reds′ learn which rules are useful for defeating the opponent. This knowledge is stored in the form of the weights that are attached to the rules in the rulebases of the reds′. Figure 5.1 shows Step 1 graphically.

**Step 2. The reds$''$ use transferred knowledge in two-versus-two scenarios.** The rulebases of the reds′ (see Step 1) are copied to the reds$''$. The reds$''$ use the copied rulebases as their initial knowledge for learning to defeat two blue CGFs in four distinct two-versus-two scenarios. The two-versus-two scenarios are described in Appendix A.4.2. Three of the scenarios are based on the two-versus-one scenarios that the reds encountered as part of

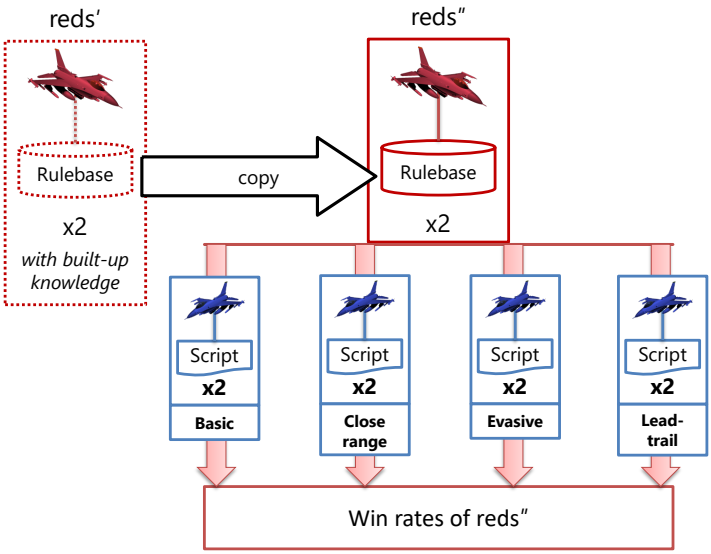


**Figure 5.2** Step 2 of the implementation of the use case. The rulebase of the reds′ (resulting from Step 1) is copied to the reds″. The reds″ use this rulebase to learn to defeat two opponents, in each of four two-versus-two scenarios: the basic scenario, the close range scenario, the evasive scenario, and the lead-trail scenario. We record and store the win rates of the reds″ in each of the four scenarios.

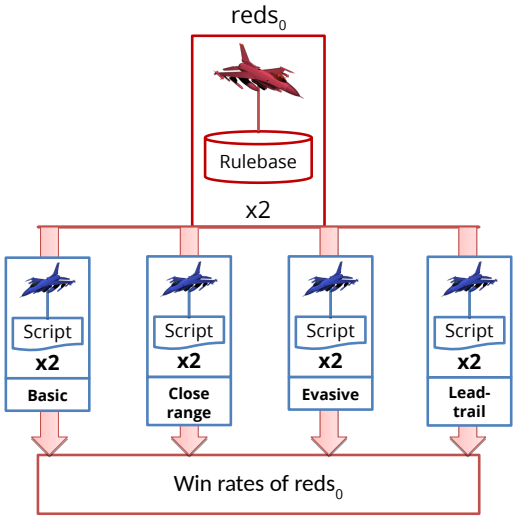


**Figure 5.3** Step 3 of the implementation of the use case. The $reds_0$ learn to defeat two opponents, in each of four two-versus-two scenarios: the basic scenario, the close range scenario, the evasive scenario, and the lead-trail scenario. In contrast to Step 2, the knowledge of the reds' is *not* transferred to the $reds_0$. We record and store the win rates of the $reds_0$ in each of the four scenarios.

the mixed scenario: (a) the basic scenario, (b) the close range scenario, and (c) the evasive scenario. The fourth scenario is (d) the lead-trail scenario. This is a new scenario, in which the blue lead approaches the reds″, while the blue wingman follows closely behind. In each of the four scenarios, we record and store the win rates of the reds″, i.e., how often the reds″ defeat their opponents throughout the learning process (see Subsection 3.3.5). Figure 5.2 shows Step 2 graphically.

**Step 3. The $\text{reds}_0$ perform the two-versus-two scenarios without transferred knowledge.** Step 3 is similar to Step 2. A two-ship of reds, in this case the $\text{reds}_0$, are placed in four two-versus-two scenarios. The $\text{reds}_0$ have to learn to defeat the two opponents in each of the scenarios. However, the $\text{reds}_0$ have to do so from scratch, viz. with a newly initialised rulebase that does not contain any previously built-up knowledge. We collect the win rates of the $\text{reds}_0$ so that they may be compared to the win rates of the reds″ (obtained in Step 2). Figure 5.3 shows Step 3 graphically.

After Step 3, we have (a) the win rates of the reds″ in the two-versus-two scenarios, and (b) the win rates of the $\text{reds}_0$ in the same scenarios. By comparing the win rates, we should be able to determine the *success of the transfer*, i.e., the extent to which the behaviour of the reds″ has improved over the behaviour of the $\text{reds}_0$ because of the transferred knowledge. In the next section, we treat determining the success of the transfer as an experiment. There, we also elaborate on the specific comparison that we will perform on the win rates (see Subsection 5.3.4).

## 5.3 Experimental setup

To determine the success of the transfer in our use case as it is outlined in Section 5.2 we designed an experiment. The experiment consists of automated simulations in LWACS. The capabilities of LWACS are presented in Appendix A. Below, we present the setup of the experiment in detail. The setup is divided into four parts: the red teams (i.e., the reds′, the reds″, and the $\text{reds}_0$) (Subsection 5.3.1), the blue team (Subsection 5.3.2), the independent and dependent variables (Subsection 5.3.3), and a description of our method of analysis, by which we determine the success of the transfer (Subsection 5.3.4).

### 5.3.1 Red teams

In the use case, there are three red teams: the reds′, the reds″, and the $\text{reds}_0$. Apart from the scenarios in which they operate (and thus build up and/or use their knowledge), the red teams are equal. Each of the red teams consists of two fighter jet CGFS, a lead and a wingman. The capabilities of the CGFS are described in Appendix A.2. The goal of each red team is to learn how to defeat the blue team in a selection of different scenarios (see Section 5.2).

Each of the red teams uses the DECENT method for team coordination. This method has been explained in Subsection 3.2.3. The lead and the wingman both learn by means of dynamic scripting. Each uses their own rulebase. The reward function used during learning is AA-REWARD. This function has been described in Chapter 4.

### 5.3.2 Blue team

Depending on the task, the blue team consists of either one CGF (see Section 5.2, *Step 1*) or two CGFS, i.e., a lead and a wingman (see Section 5.2, *Step 2* and *Step 3*). The capabilities of the CGFS are described in Appendix A.2. The goal of the blue team is not to be defeated by red. The behaviour of blue is governed by scripts (see Appendix A.3).

### 5.3.3 Independent and dependent variables

Based on the use case, we define two independent variables in the experiment. The first independent variable is whether knowledge is transferred to the red teams that operate in the two-versus-two scenarios. This is the case for the reds$''$ (see Section 5.2, *Step 2*), but not for the reds$_0$ (see Section 5.2, *Step 3*). The second independent variable consists of the four two-versus-two scenarios (i.e., basic, close range, evasive, and lead-trail) for which we gather the win rates. Rather than averaging the win rates over the four scenarios, we are interested to see if any changes in performance caused by the transfer of knowledge to the reds$''$ vary between the four scenarios. The combination of these independent variables results in a $2 \times 4$ fully factorial design with eight conditions. The win rates are the dependent variable in the experiment.

### 5.3.4 Method of analysis

In our analysis of the results, we aim to measure the success of the transfer by comparing the win rates of the reds$''$ to the win rates of the reds$_0$. We apply three measures to the win rates in order to perform a meaningful comparison. The measures are (1) the initial performance measure, (2) the final performance measure, and (3) the turning point measure.

The initial performance measure calculates the mean win rate at the first encounter in the learning process. This measure captures how well the knowledge that was built up by the reds$_0$ can be directly applied by the reds$''$ in the two-versus-two scenarios before any further learning is allowed to take place.

In Chapters 3 and 4, we used the final performance measure and the turning point measure to analyse the performance of the learning CGFS. The two measures are explained in Subsection 3.3.6. Earlier, we have defined the final performance as the mean performance over the last 50 encounters. For the remainder of this chapter we redefine the final performance measure to be the mean performance of the last 30 encounters. By taking into account fewer encounters, we expect the final performance measure to more accurately reflect the stabilised performance

after learning has taken place. The turning point in the learning process is the encounter at which point a moving window of 10 encounters contains more encounters that were won than encounters that were lost.

By use of the three measures, we are now able to compare the performance of the reds″ to the performance of the $reds_0$ in three areas. We perform the actual comparison by means of an ANOVA on the results of each of the three measures. The ANOVAs will show whether the transfer of knowledge leads to significantly better performance in the two-versus-two scenarios.

## 5.4 Experimental results

In this section, we present the results of the experiment. We begin by presenting the win rates of the reds′ in the two-versus-one scenario (Subsection 5.4.1). The win rates of the reds′ are an intermediary result, obtained by performing Step 1 in the use case (see Subsection 5.2.2). Next, we present the win rates of both the reds″ (Step 2) and the $reds_0$ (Step 3) in the two-versus-two scenarios (Subsection 5.4.2). Finally, we present the results of applying the three measures for the success of the transfer (Subsection 5.4.3).

### 5.4.1 Win rates of the reds′

In this subsection, we present and briefly discuss the win rates that are achieved by the reds′ in the two-versus-one scenario. Figure 5.4 shows the win rate of the reds′. The win rate starts at .407 at the first encounter, and then rises to an average win rate of .644 over the last 30 encounters. As can be seen in Figure 5.4, the win rate grows mildly but steadily over the course of the encounters. The win rate is the average of 150 runs of encounters, with each run consisting of 150 encounters in which the reds′ engaged the blue opponent. It is only after this number of encounters that the trend in the win rate became clearly visible, and that it became apparent that the win rate would not grow any further.

### 5.4.2 Win rates of the reds″ and the $reds_0$

In this subsection, we present the win rates achieved by both (a) the reds″ and (b) the $reds_0$ in the two-versus-two scenarios. Figure 5.5 shows the win rates. As in Subsection 5.4.1, the win rates are the average of 150 runs of encounters, where each run consists of 150 encounters (so, in total, 22,500 encounters). Here, we make two observations. The first observation is that the win rates of the reds″ and the $reds_0$ are clearly separated to some extent. The separation is prevalent in the basic and close range scenarios. Here, the win rate of the reds″ is clearly higher than that of the $reds_0$. The win rates converge after around 60 encounters. The second observation is that in the basic and close range scenarios, the win rate of the reds″ does not seem to improve over time. However, in the evasive and lead-trail scenarios, some improvement is visible in the win rates of both the reds″ and the $reds_0$.

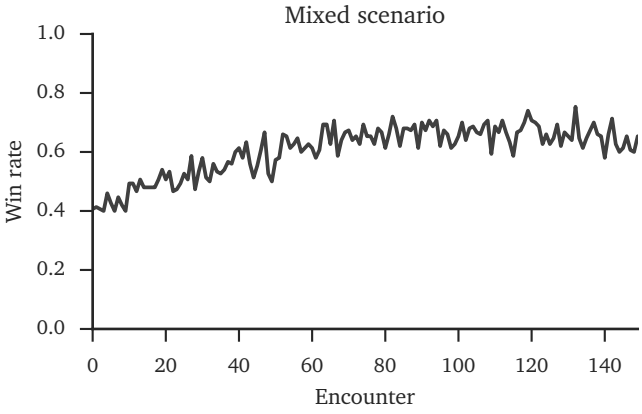


**Figure 5.4** The win rates achieved by the reds′ in the two-versus-one mixed scenario.

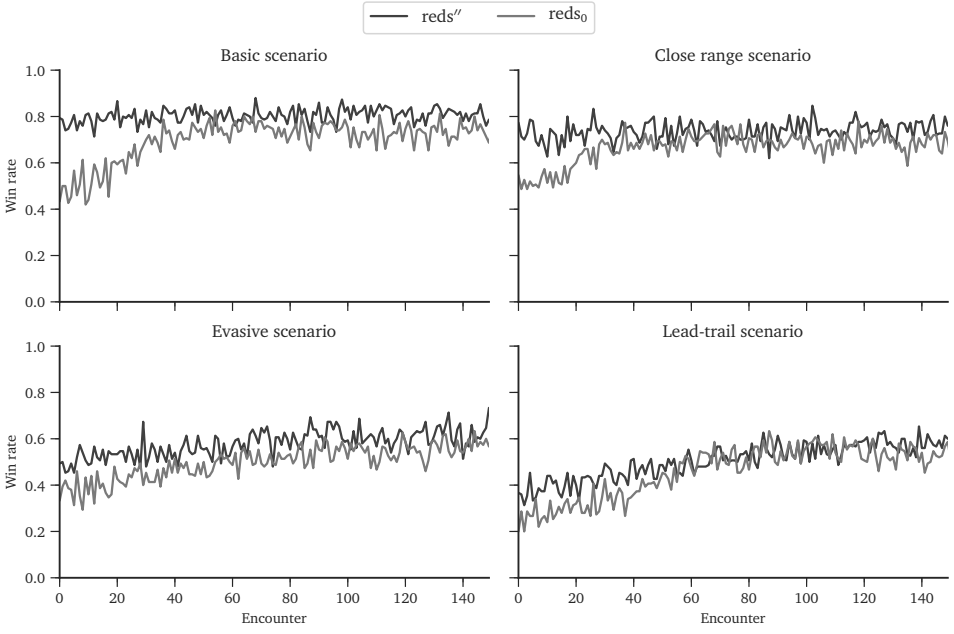


**Figure 5.5** The win rates achieved by the reds″ and the $reds_0$ in the two-versus-two scenarios: (1) the basic scenario, (2) the close range scenario, (3) the evasive scenario, and (4) the lead-trail scenario.

### 5.4.3 Application of the three measures

In this subsection, we present the results of applying the three measures for the success of the transfer to the win rates of the reds″ and the $reds_0$: (A) the initial performance, (B) the final performance, and (C) the turning points. Furthermore, we apply an ANOVA to test for significant differences in the results of each measure.

**A: Initial performance**

First, we applied the initial performance measure to the win rates of the reds″ and the $reds_0$. Table 5.1 shows the results of applying the measure. In Table 5.1, we see that the initial performance of the reds″ is higher than that of the $reds_0$ in each of the four scenarios. The results indicate that the knowledge that is transferred from the reds′ to the reds″ provides an immediately observable benefit to the reds″. However, in the lead-trail scenario the difference in initial performance (0.003) appears to be somewhat negligible.

**Table 5.1** The initial performance of the reds″ and the $reds_0$. A higher initial performance is better.

| | Basic scenario | | Close range scenario | | Evasive scenario | | Lead-trail scenario | | Grand mean | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| reds″ | .820 | .385 | .740 | .440 | .467 | .501 | .460 | .500 | .622 | .457 |
| $reds_0$ | .556 | .498 | .616 | .488 | .252 | .435 | .457 | .500 | .470 | .480 |

A two-way ANOVA was conducted on the influence of the two independent variables (transfer condition, scenario) on the initial performance of the reds″ and the $reds_0$. All effects were statistically significant at the $\alpha = .01$ level. The main effect of transfer condition yielded an $F$ ratio of $F(1, 1192) = 29.857$, $p < 0.001$, indicating a significant difference between the initial performances of the reds″ and the $reds_0$. The main effect of scenario yielded an $F$ ratio of $F(3, 1192) = 36.491$, $p < 0.001$, indicating a significant difference in the initial performance in each of the four scenarios. Furthermore, the interaction between the two independent variables was found to be statistically significant, $F(3, 1192) = 4.467$, $p < 0.01$. We performed a post hoc Tukey HSD test (see, e.g., Holmes et al., 2016) to determine the significant differences in initial performance between specific pairs of scenarios. The post hoc test revealed that the initial performance differed significantly between all scenarios, $p < 0.05$, except between the basic and close range scenarios.

**B: Final performance**

Second, we applied the final performance measure to the win rates of the reds″ and the $reds_0$. Table 5.2 shows the results of applying the measure. In each of the four scenarios, the reds″ reach

a higher final performance than the $reds_0$ do. This means that the reds″ learn more effective behaviour in the scenarios. We observed a similar pattern for the initial performance (see A). However, in the case of the lead-trail scenario, the difference in initial performance was relatively small (0.003). Now, for the final performance, the difference has grown somewhat to 0.044.

**Table 5.2** The final performance of the reds″ and the $reds_0$. A higher final performance is better.

| | Basic scenario | | Close range scenario | | Evasive scenario | | Lead-trail scenario | | Grand mean | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| reds″ | .819 | .074 | .734 | .084 | .553 | .093 | .498 | .111 | .651 | .091 |
| $reds_0$ | .731 | .116 | .652 | .093 | .439 | .116 | .454 | .141 | .569 | .117 |

A two-way ANOVA was conducted on the influence of the two independent variables (transfer condition, scenario) on the final performance of the reds″ and the $reds_0$. All effects were statistically significant at the $\alpha = .01$ level. The main effect of transfer condition yielded an $F$ ratio of $F(1, 1192) = 182.092$, $p < 0.001$, indicating a significant difference between the final performances of the reds″ and the $reds_0$. The main effect of scenario yielded an $F$ ratio of $F(3, 1192) = 582.882$, $p < 0.001$, indicating a significant difference in the final performance in each of the four scenarios. Furthermore, the interaction between the two independent variables was found to be statistically significant, $F(3, 1192) = 5.685$, $p <= 0.001$. A post hoc Tukey HSD test revealed that the final performance differed significantly between all scenarios, $p < 0.001$, except between the evasive and lead-trail scenarios.

#### C: Turning points

Third, we applied the turning point measure to the win rates of the reds″ and the $reds_0$. Table 5.3 shows the results of applying the measure. In each of the four scenarios, the reds″ reach lower turning points than the $reds_0$ do. This means that the reds″ learn more efficiently (viz. learn more effective behaviour in less encounters). The turning points in the evasive and the lead-trail scenarios show the largest differences (19.5 and 20.8, respectively). Thus, the turning points indicate that the knowledge that is transferred from the reds′ to the reds″ help the reds″ to efficiently learn to defeat the blue two-ships in the evasive and the lead-trail scenarios.

A two-way ANOVA was conducted on the influence of the two independent variables (transfer condition, scenario) on the turning points of the reds″ and the $reds_0$. All effects were statistically significant at the $\alpha = .01$ level. The main effect of transfer condition yielded an $F$ ratio of $F(1, 1192) = 90.943$, $p < 0.001$, indicating a significant difference between the turning points of the reds″ and the $reds_0$. The main effect of scenario yielded an $F$ ratio of $F(3, 1192) = 80.719$, $p < 0.001$, indicating a significant difference in the turning points in each of the four scenarios. Furthermore, the interaction between the two independent variables was found to be statistically

**Table 5.3** The turning points of red. Lower turning points are better.

| | Basic scenario | | Close range scenario | | Evasive scenario | | Lead-trail scenario | | Grand mean | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| reds″ | 10.4 | 1.6 | 10.8 | 2.4 | 16.4 | 9.8 | 21.6 | 19.6 | 14.8 | 8.4 |
| $\text{reds}_0$ | 12.1 | 5.1 | 12.9 | 5.7 | 35.9 | 21.5 | 42.4 | 46.8 | 25.8 | 19.8 |

significant, $F(3, 1192) = 20.993$, $p <= 0.001$. A post hoc Tukey HSD test revealed that the turning points differed significantly between all scenarios, $p < 0.01$, except between the basic and close range scenarios.

## 5.5 Discussion

In this section, we discuss the results of the experiment. We cover three topics. First, we determine the success of the transfer (Subsection 5.5.1). Second, we discuss the performance of the reds″ in the new, unseen lead-trail scenario. Third, we briefly review the stationary win rates in the basic and close range scenarios (Subsection 5.5.3).

### 5.5.1 Success of the transfer

The results show clearly and consistently that the reds″ outperform the $\text{reds}_0$. As we defined in our use case (see Section 5.2), the common goal of the reds″ and the $\text{reds}_0$ was to defeat two blue opponents in two-versus-two scenarios. However, the reds″ received a *transfer of knowledge* of the knowledge built by the reds′ in a two-versus-one scenario.

In Subsection 5.3.4, we defined three measures for the success of the transfer: (a) the initial performance measure, (b) the final performance measure, and (c) the turning point measure. The application of these measures to the win rates of the reds″ and the $\text{reds}_0$ shows that:

(a) the transferred knowledge provides an immediate advantage to the reds″ in defeating the blue two-ship, before any learning by the reds″ takes place,

(b) the transferred knowledge enables the reds″ to learn more effective behaviour than the $\text{reds}_0$ throughout the encounters with the blue two-ship, and

(c) because of the transferred knowledge, the reds″ require less time than the $\text{reds}_0$ to start winning over 50% of the encounters with the blues.

Based on these results, we may conclude that the transfer of knowledge as outlined by our use case is to a large extent successful. Of course, for practical reasons our use case only includes a narrow selection of air combat scenarios. However, since our findings are consistent in each of

the four two-versus-two scenarios, we may expect that the success of the transfer will generalise to some extent to other scenarios as well.

### 5.5.2 Improved performance in the lead-trail scenario

Out of the four two-versus-two scenarios in the use case, the lead-trail scenario is perhaps most interesting. In this scenario, the blues use a tactic that is not represented in the two-versus-one scenario. Therefore, this tactic is also not represented in the knowledge that is transferred from the reds′ to the reds″. Thus, the lead-trail scenario allows us to determine how well the transferred knowledge generalises to scenarios in which the opponents use new, unseen tactics.

In the results, we see that the initial and final performance of the reds″ are only slightly improved by the transferred knowledge (see Table 5.1 and Table 5.2, respectively). However, compared to the $\text{reds}_0$, the turning points of the reds″ are reduced by nearly 50% (from 42.4 to 21.6). So, in the case of the new, unseen tactic, the transferred knowledge does not appear to increase the effectiveness of the behaviour of the reds″, while it does increase the speed by which they find effective behaviour against the tactic.

In a survey of the rulebases of the reds′ (i.e., the knowledge that was transferred to the reds″), we observed that (a) the red lead had assigned a high weight to a particular evasive rule (i.e., evade incoming missiles by turning 180 degrees), but also that (b) the lead and the wingman had not converged to a role division for firing missiles. Therefore, we suspect that the combination of (a) the lead's preference for this particular evasive rule and (b) any firing rules in the rulebases of the two-ship with weights higher than the starting weights were sufficient to kick-start the learning process of the reds″.

Our results are in line with other works studying transfer learning in reinforcement learning applications. For instance, Spector and Belongie (2018) studied transfer learning in an application involving the automated playing of a simple Atari-like game. They report an increase in learning speed of 50 times in the case with transfer over the case without transfer.

### 5.5.3 Stationary win rates

In Figure 5.5, we see upward trends in most of the win rates. However, the win rates of the reds″ in (a) the basic scenario and (b) the close range scenario do not show an upward or downward trend. Instead, they appear to remain stationary around 0.8. This indicates that some form of optimum has been found in the weights in the rulebases of the reds″. Here, one of two situations is possible. On the one hand, the reds″ may be just successful enough to maintain the weights in the rulebase, without being forced to try new combinations of rules in the scripts. On the other hand, the stationary win rates might be the highest possible win rates that can be achieved in the scenarios using the rules that the reds″ (and thus also the $\text{reds}_0$) have available in their rulebases. An exhaustive search of the win rates that are achieved by all possible scripts in the basic and close range scenarios might indicate which of the two situations is currently at hand.

Still, if the reds″ no longer improve their behaviour, is there any benefit of the transfer of knowledge for the basic and close range scenarios? Despite the win rates of the reds″ remaining stationary, they also remain above the win rates of the $\text{reds}_0$. It takes nearly forty encounters for the win rates of the $\text{reds}_0$ to approach that of the reds″. This shows that overall, the transfer of knowledge leads to a better performance. However, the stationary win rates of the reds″ may also indicate that given blue's behaviour in these scenarios, the learning problem becomes easier when a second opponent is added. In essence, the reds may have been able to collect more reward (i.e., fire missiles with a higher $P_k$) *because* of the addition of a second, possibly easy-to-hit target. Further research should point out whether this causal relation actually exists.

## 5.6 Answering research question 3

In this chapter, we investigated the transfer of knowledge between CGFs. Specifically, we addressed research question 3.

Research question 3 reads: *To what extent can knowledge built with dynamic scripting be transferred successfully between CGFs in different scenarios?* To answer this question, we designed and implemented a use case for transfer learning in air combat simulations (Section 5.2). The use case consists of three steps. In Step 1 of the use case, a two-ship of red CGFs (which we call the reds′) engages a blue opponent in a two-versus-one scenario. In Step 2, the knowledge built up by this two-ship is transferred to a second two-ship (the reds″), who then use the knowledge to learn how to defeat two blue opponents in four different two-versus-two scenarios. In Step 3, a third two-ship (the $\text{reds}_0$) also learns how to defeat the blue two-ship in the four two-versus-two scenarios. However, they do so tabula rasa, viz. without a transfer of knowledge.

We used three measures to determine the success of the transfer: (a) the initial performance measure, (b) the final performance measure, and (c) the turning point measure (Section 5.3). The reds″, using the transferred knowledge, reach significantly higher performance than the $\text{reds}_0$ did, on each of the three measures. Even in the lead-trail two-versus-two scenario (which the reds′ had not seen, and thus could not transfer any knowledge of to the reds″), the transferred knowledge allowed the reds″ to learn more efficiently than the $\text{reds}_0$.

In conclusion, we answer research question 3 as follows. Based on the results of the simulations as outlined in the use case, we may conclude that we have to a large extent successfully transferred knowledge between air combat CGFs in different scenarios. Because air combat simulations often share common elements, we expect that the success of the transfer may extend beyond the scope of our use case as well.