



Universiteit
Leiden
The Netherlands

Calculated Moves: Generating Air Combat Behaviour

Toubman, A.

Citation

Toubman, A. (2020, February 5). *Calculated Moves: Generating Air Combat Behaviour*. *SIKS Dissertation Series*. Retrieved from <https://hdl.handle.net/1887/84692>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/84692>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/84692> holds various files of this Leiden University dissertation.

Author: Toubman, A.

Title: Calculated Moves: Generating Air Combat Behaviour

Issue Date: 2020-02-05

Calculated Moves

Generating Air Combat Behaviour

Calculated Moves

Generating Air Combat Behaviour

PROEFSCHRIFT

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 5 februari 2020
klokke 13.45 uur

door

Armon Toubman
geboren te Amsterdam in 1988

Promotores: Prof. dr. H.J. van den Herik Leiden University
Prof. dr. ir. P.H.M. Spronck Tilburg University
Copromotor: Dr. ir. J.J.M. Roessingh Royal Netherlands Aerospace Centre NLR

Promotiecommissie: Prof. dr. M.E.H. van Reisen Leiden University
Prof. dr. S. Manegold Leiden University / Centrum Wiskunde & Informatica
Prof. dr. H.X. Lin Leiden University / Delft University of Technology
Prof. dr. J.C. Scholtes Maastricht University
Dr. G.C.H.E. de Croon Delft University of Technology



The research reported in this thesis has been funded by and performed at the Royal Netherlands Aerospace Centre NLR, Amsterdam, the Netherlands.



In the first year, the research reported in this thesis has been performed at Tilburg center for Cognition and Communication (TiCC) at Tilburg University, the Netherlands.



The research reported in this thesis has been completed at the Leiden Centre of Data Science (LCDS) hosted by the Leiden Institute of Advanced Computer Science (LIACS) at the Faculty of Science, Leiden University, the Netherlands.



SIKS Dissertation Series No. 2020-01
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by: Gildeprint, Enschede
Cover/invitation design: Armon Toubman, derivative of “Dutch F-16 5” by Tony Hisgett.
Used under the CC BY 2.0 license.
Original photo available at: <https://flic.kr/p/8a6hs4>
Additional credits: Includes Vista Style People Demo Icons by: <http://www.icons-land.com>

©2019 Armon Toubman
ISBN 978-94-6402-000-7

An electronic version of this dissertation is available at:
<http://openaccess.leidenuniv.nl/>

Preface

Fighter jets are incredible machines. Equally incredible is the skill required to operate them. The pilots of these jets have to excel at observing, communicating, calculating, and making life-or-death decisions while zooming through the sky at inhuman speeds. Excelling at these tasks is only possible by rigorous training. However, defence budget cuts have resulted in fewer aircraft for air forces, and thus fewer aircraft available for real-world training. For instance, the Royal Netherlands Air Force had 213 F-16s available in 1992. Today, in 2019, there are 68 F-16s left. These are planned to be replaced by 37 F-35s in 2023, followed by additional F-35s at a later time. Innovative training methods are thus required to keep fighter pilots ready for future operations.

Another type of incredible machines are modern computers. Over the last decades, computers have become so powerful that they are able to simulate complex virtual worlds, in which humans can interact with life-like virtual entities. The computing power that is available today has also enabled the computers to reinvent their own programming, by means of machine learning algorithms. Since I started my PhD candidacy in 2013, the interest in machine learning has grown exponentially. From credit card fraud detection to self-driving cars, machine learning is now *everywhere*; so much so, that even the latest smartphones have separate processors dedicated solely to machine learning calculations.

One of the most important training tools in the arsenals of air forces is the flight simulator. A simulator relies on virtual entities called *computer generated forces* to create interesting situations that resemble the situations that fighter pilots may encounter in the real world. However, modelling and programming the behaviour of these entities remains challenging. As a result, only few behaviour models are created for the entities, and thus the simulators are left underused. In our research, we put one and one together by applying machine learning to fighter pilot training.

Personally, combining fighter jets and machine learning has felt like turning a piece of science fiction into reality. I hope that the research in this thesis will lead to safer skies. To all fighter pilots training in simulators I extend the greeting used by the Klingon warriors in the *Star Trek* television shows: *Qapla'!*

Armon Toubman
Almere, November 24, 2019

Contents



Preface	ix
Contents	xi
Nomenclature	xvii
List of Acronyms	xix
List of Definitions	xxi
List of Figures	xxiii
List of Tables	xxv
List of Listings	xxvii

CHAPTERS

1 Introduction	1
1.1 The behaviour modelling process	2
1.1.1 Obstacles in the process	2
1.1.2 Consequences for training effectiveness	4
1.2 Generating air combat behaviour models	4
1.2.1 Challenges	5
1.2.2 Scope of the thesis	7
1.3 Problem statement and research questions	8
1.4 Research methodology	9
1.5 Structure of the thesis	11

2 Foundations	13
2.1 The steps in the behaviour modelling process	13
2.2 Machine learning in training simulations	15
2.2.1 Potential benefits	15
2.2.2 Potential drawbacks	16
2.3 Machine learning	17
2.3.1 The three categories of machine learning tasks	17
2.3.2 Reinforcement learning	18
2.3.3 Dynamic scripting	20
2.4 Past approaches to generating air combat behaviour	25
2.4.1 Neural networks	25
2.4.2 Evolutionary algorithms	27
2.5 Chapter summary	29
3 Team coordination	31
3.1 Two perspectives on team coordination	32
3.1.1 The air combat perspective	32
3.1.2 The multi-agent system perspective	33
3.1.3 Combining the perspectives into coordination methods	34
3.2 Team coordination in dynamic scripting	37
3.2.1 Implementing TACIT	38
3.2.2 Implementing CENT	42
3.2.3 Implementing DECENT	45
3.3 Experimental setup	46
3.3.1 The Lightweight Air Combat Simulator	47
3.3.2 Red team	47
3.3.3 Blue team	48
3.3.4 Scenarios	48
3.3.5 Independent and dependent variables	49
3.3.6 Method of analysis	49
3.4 Experimental results	50
3.5 Discussion	52
3.5.1 Key finding	52
3.5.2 The effect of centralised coordination on performance	52
3.5.3 The learning process of coordinating CGFs	53
3.5.4 The way forward	56

3.6 Answering research question 1	56
4 Improving the reward function	59
4.1 Reward functions in reinforcement learning	60
4.1.1 A formal description of reinforcement learning	60
4.1.2 The role of rewards in dynamic scripting	62
4.2 Designing reward functions	63
4.3 Sparse rewards	66
4.3.1 Problem description	66
4.3.2 Reward shaping	66
4.3.3 Sparse rewards in the literature	67
4.3.4 Proposed solution: DOMAIN-REWARD	70
4.4 Unstable rewards	72
4.4.1 Problem description	72
4.4.2 Unstable rewards in the literature	74
4.4.3 Proposed solution: AA-REWARD	76
4.5 Overview of the three reward functions	77
4.5.1 BIN-REWARD	78
4.5.2 DOMAIN-REWARD	78
4.5.3 AA-REWARD	79
4.6 Experimental setup	80
4.6.1 Red team	80
4.6.2 Blue team	81
4.6.3 Scenarios	81
4.6.4 Independent and dependent variables	81
4.6.5 Method of analysis	81
4.7 Results	81
4.8 Discussion	84
4.8.1 Using DOMAIN-REWARD	84
4.8.2 Using AA-REWARD	85
4.8.3 Sparsity and stability	85
4.9 Answering research question 2	85
5 Transfer of knowledge between scenarios	87
5.1 The concept of transfer learning	88
5.1.1 Transfer learning methods	88

5.1.2	Transfer learning in reinforcement learning	89
5.1.3	Transfer learning in dynamic scripting	89
5.1.4	The burden of human knowledge	90
5.2	Use case	90
5.2.1	Description	90
5.2.2	Implementation in dynamic scripting	91
5.3	Experimental setup	94
5.3.1	Red teams	94
5.3.2	Blue team	95
5.3.3	Independent and dependent variables	95
5.3.4	Method of analysis	95
5.4	Experimental results	96
5.4.1	Win rates of the reds'	96
5.4.2	Win rates of the reds'' and the reds ₀	96
5.4.3	Application of the three measures	98
5.5	Discussion	100
5.5.1	Success of the transfer	100
5.5.2	Improved performance in the lead-trail scenario	101
5.5.3	Stationary win rates	101
5.6	Answering research question 3	102
6	A validation procedure for generated air combat behaviour models	103
6.1	Validating behaviour models	104
6.1.1	What does the validation process precisely entail?	104
6.1.2	How should we determine the accuracy of the models?	105
6.1.3	Section conclusion and outlook	106
6.2	Terminology	106
6.3	Designing a validation process	107
6.4	The Assessment Tool for Air Combat CGFs	111
6.5	Equivalence testing	114
6.5.1	Equivalence testing with TOST	114
6.5.2	Measuring an extent of validity	115
6.6	Implementing the validation process	116
6.7	Answering research question 4	116

7	Validation of generated behaviour models in training simulations	119
7.1	Defining the baseline: The 4P-models	120
7.2	Generating behaviour models: The 4M-models	120
7.2.1	The rules in the rulebases	121
7.2.2	Automated simulations	122
7.3	Human-in-the-loop simulations	124
7.4	Behaviour assessments	125
7.5	Results of the behaviour assessments	126
7.5.1	Equivalence testing	126
7.5.2	Inter-rater reliability	127
7.5.3	Feedback on the assessments	127
7.6	Discussion	129
7.6.1	Key finding	129
7.6.2	Placing our key finding in context	129
7.6.3	Implications	130
7.6.4	Limitations	131
7.7	Answering research question 5	131
8	Conclusions	133
8.1	Answers to the research questions	133
8.2	Answer to the problem statement	135
8.3	Recommendations for future research	136

References	139
-------------------	------------

APPENDICES

A	The Lightweight Air Combat Simulator	159
A.1	Description	159
A.2	Computer generated forces	160
A.3	Scripting language	161
A.4	Scenarios	162
A.4.1	Two-versus-one scenarios	162
A.4.2	Two-versus-two scenarios	164

B The LWACS scripting language	165
B.1 Grammar	165
B.2 Function descriptions	168
B.2.1 Boolean functions	168
B.2.2 Numerical functions	168
B.2.3 Action functions	169
C Rulebases and scripts	171
D The Fighter 4-Ship simulator	173
D.1 The ships	173
D.2 The instructor operating station	175
D.3 Software packages	176
D.4 Dynamic scripting in the Fighter 4-Ship	177
E Generating finite-state machines	179
E.1 Expressing finite-state machines as rules	180
E.2 The modified dynamic scripting algorithm	181
E.3 Summary	182
F The Assessment Tool for Air Combat CGFs	185
<hr/>	
Summary	189
Samenvatting	193
List of publications	197
Curriculum vitae	199
Acknowledgements	201
SIKS dissertation series	203



Nomenclature

AA-REWARD The reward function based on the probability-of-kill of missiles.

BIN-REWARD The binary reward function.

CENT The centralised coordination method with communication.

DECENT The decentralised coordination method with communication.

DOMAIN-REWARD The reward function based on domain knowledge.

P_K The probability-of-kill of a missile.

TACIT The decentralised coordination method without communication.

List of Acronyms

- AI** artificial intelligence.
- ANOVA** analysis of variance.
- API** application programming interface.
- ATACC** Assessment Tool for Air Combat CGFs.
- BARS** behaviourally anchored rating scale.
- BOS** behaviour observation scale.
- BVR** beyond-visual-range.
- CAP** combat air patrol.
- CGF** computer generated force.
- CI** confidence interval.
- CLI** command-line interface.
- DIS** distributed interactive simulation.
- FSM** finite-state machine.
- GUI** graphical user interface.
- HSD** honest significant difference.
- HUD** head-up display.
- ICC** intraclass correlation.
- ICP** integrated control panel.
- IOS** instructor operating station.

LCS learning classifier system.

LWACS Lightweight Air Combat Simulator.

MEC mission essential competency.

MFD multi-functional display.

NLR Netherlands Aerospace Centre.

PCDS Personal Computer Debriefing System.

PS problem statement.

RNLAF Royal Netherlands Air Force.

RQ research question.

RWR radar warning receiver.

SB Smart Bandits.

TOST two one-sided *t*-tests.

WVR within-visual-range.

XCS accuracy-based learning classifier system.

List of Definitions

- 1.1 Behaviour model 2
- 1.2 Accessible behaviour model 7

- 2.1 Behaviour rule 22
- 2.2 Script 22
- 2.3 Rulebase 22

- 3.1 Scenario 48

- 4.1 Desirable behaviour 60
- 4.2 Terminal state 63
- 4.3 Optimal reward function 64
- 4.4 Probability-of-kill 73

- 5.1 Transfer learning 88
- 5.2 Source task 88
- 5.3 Target task 88

- 6.1 Validation 104
- 6.2 4-model 107
- 6.3 4P-model 107
- 6.4 4M-model 107
- 6.5 Measure of validity of the 4M-models 111

List of Figures

- 1.1 The four steps in the behaviour modelling process. 3
- 2.1 The reinforcement learning loop. 19
- 2.2 The three steps of the dynamic scripting learning process. 23
- 3.1 The axes of team coordination. 36
- 3.2 TACIT, CENT, and DECENT implemented in dynamic scripting. 39
- 3.3 The win rates achieved by red using TACIT, CENT, and DECENT. 51
- 3.4 The weights of the rules over time, averaged over 50 runs (CENT, basic scenario, red lead). 54
- 3.5 The weights of the rules over time, averaged over 50 runs (CENT, mixed scenario, red lead). 55
- 4.1 The spectrum of desirable air combat behaviour. 71
- 4.2 The probability-of-kill curve of the missiles in LWACS. 73
- 4.3 The win rates achieved by red using BIN-REWARD, DOMAIN-REWARD, and AA-REWARD. 82
- 5.1 Step 1 of the implementation of the use case. 91
- 5.2 Step 2 of the implementation of the use case. 92
- 5.3 Step 3 of the implementation of the use case. 93
- 5.4 The win rates achieved by the reds'. 97
- 5.5 The win rates achieved by the reds'' and the reds₀. 97
- 6.1 The outline of the validation process. 109
- 6.2 Adding 4P-models as a baseline. 109
- 6.3 Executing the 4-models in human-in-the-loop simulations. 109
- 6.4 Assessments of the simulation results. 110
- 6.5 Equivalence testing of the assessment results. 110

7.1 The three steps of the generation strategy. 123

A.1 A screenshot of LWACS. 161

A.2 The initial positions of the CGFs in the four LWACS scenarios. 163

D.1 Schematic top view of a ship. 174

D.2 Photograph of a ship being operated by a participant. 175

E.1 An FSM as a behaviour model. 180

List of Tables

- 1.1 Research methods used to answer the research questions. 10
- 1.2 Answering the problem statement and the research questions. 11

- 2.1 A selection of dynamic scripting applications from the literature. 25

- 3.1 The final performance of red. 51
- 3.2 The turning points of red. 51

- 4.1 A comparison of BIN-REWARD, DOMAIN-REWARD, and AA-REWARD. 77
- 4.2 The final performance of red. 83
- 4.3 The turning points of red. 83

- 5.1 The initial performance of the reds'' and the reds₀. 98
- 5.2 The final performance of the reds'' and the reds₀. 99
- 5.3 The turning points of the reds'' and the reds₀. 100

- 7.1 Summary of the ATACC responses. 128
- 7.2 Results of the TOST method. 128
- 7.3 Results of the intraclass correlation analysis. 128

List of Listings

- 2.1 Example behaviour rule. 22
- B.1 Grammar of the LWACS scripting language. 165
- E.1 The FSM from Figure E.1 expressed in the form of rules. 181
- E.2 Modified script generation algorithm. 182