

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# FUZZY MOTION CONTROLLERS AND HYBRIDS

A thesis presented in partial  
fulfilment of the requirements  
for the degree of

DOCTOR OF PHILOSOPHY  
IN COMPUTER SCIENCE

Massey University,  
Albany, New Zealand.

Anton Gerdelan

2011

# Contents

<b>Abstract</b>	<b>xv</b>
<b>1 Overview and Aim of This Thesis</b>	<b>1</b>
1.1 Method of Experimentation . . . . .	2
<b>2 A Brief History of Motion Control in Animation</b>	<b>3</b>
2.1 Introduction . . . . .	3
2.2 Overview of Key Algorithms . . . . .	5
2.2.1 Boids and Flocking . . . . .	6
2.2.2 Helbing's Crowds . . . . .	7
2.3 Evolving Motion Controllers . . . . .	8
2.3.1 rtNEAT . . . . .	9
<b>3 Fuzzy Logic Controllers</b>	<b>11</b>
3.1 Overview . . . . .	11
3.2 Standard Pattern 2-Input Fuzzy Controller . . . . .	12
3.3 As Part of a Hybrid Controller . . . . .	15
<b>4 Assembling a Toolkit for Doing Science with Simulations</b>	<b>17</b>
4.1 Overview . . . . .	17
4.2 Constructing a 3D Simulation . . . . .	18
4.3 A Delta-State Video Capture Tool . . . . .	23
4.4 Real-Time Data Plots and Time Tools . . . . .	24
4.5 Measurements of Uncertainty . . . . .	25
<b>5 A Modular Agent Middleware</b>	<b>27</b>
5.1 Introduction . . . . .	27
5.2 Agent Society Model . . . . .	30
5.3 Modular Architecture . . . . .	32
5.4 Modelling the Environment . . . . .	35
5.5 Agent Behaviour . . . . .	39
5.6 Summary . . . . .	42
5.7 Possible Extensions to this Architecture . . . . .	43
<b>6 On Design of Automatic Calibration Systems</b>	<b>45</b>
6.1 Introduction . . . . .	45
6.2 Hybrid Algorithm . . . . .	46
6.3 Visualisation . . . . .	50
6.4 Proposed Self-Training Architecture . . . . .	50
6.5 Conclusions . . . . .	53

<b>7</b>	<b>Adding Agent-Based Road Networks To Simulations</b>	<b>55</b>
7.1	Introduction . . . . .	55
7.2	Representing Complex Road Networks in 3D Simulations . . . . .	57
7.3	Architecture for Non-Intrusive Data Structures . . . . .	61
7.4	Rapid Construction of Virtual Road Networks . . . . .	61
7.5	Future Works . . . . .	67
7.6	Discussion and Conclusions . . . . .	67
<b>8</b>	<b>Agents and Motion Controllers for Road Vehicles</b>	<b>69</b>
8.1	Introduction . . . . .	69
8.2	Agent Paradigm . . . . .	71
8.3	Path Planning . . . . .	73
8.4	Reactive Vehicle Control . . . . .	74
8.5	Discussion and Conclusions . . . . .	81
<b>9</b>	<b>On Simulation Frameworks for Automatic Calibration Systems</b>	<b>87</b>
9.1	Introduction . . . . .	87
9.2	Works of Note . . . . .	88
9.3	Initial Approach . . . . .	88
9.4	Preliminary Experiment . . . . .	90
9.5	Discussion and Conclusions . . . . .	93
<b>10</b>	<b>A Genetic-Fuzzy System for Optimising Motion</b>	<b>97</b>
10.1	Introduction . . . . .	97
10.2	Related Work . . . . .	98
10.3	Background: Fuzzy Controllers in Agent Steering . . . . .	99
10.4	Architecture of the GFS . . . . .	101
10.5	Benchmarking the Genetic Algorithm Component . . . . .	104
10.6	Experiments and Results . . . . .	105
10.7	Conclusions and Future Works . . . . .	112
<b>11</b>	<b>Mechanix: Vehicle Mechanical Simulation</b>	<b>113</b>
11.1	Overview of Architecture . . . . .	113
11.2	Visualisation . . . . .	115
11.3	Drive-train Simulation . . . . .	118
11.4	Resistance Forces . . . . .	125
11.5	Effective Torque . . . . .	126
11.6	Suspension Simulation . . . . .	126
11.7	Trailers, Articulated Vehicles, Trains & Trams . . . . .	131
11.8	Limitations . . . . .	131
<b>12</b>	<b>Gremlin: A System for Benchmarking Mechanical Motion</b>	<b>135</b>
12.1	Introduction . . . . .	135
12.2	Definition of Test-Course Environment . . . . .	136
12.3	Evaluation Method . . . . .	137
12.4	Control System Design . . . . .	140
12.5	Experiments and Results . . . . .	141
12.6	Future Works . . . . .	143

<i>CONTENTS</i>	iii
<b>13 Conclusions and Discussion</b>	<b>145</b>
13.1 Review of Fuzzy Motion Controllers . . . . .	145
13.2 Conclusions on Genetic Hybrid Systems . . . . .	146
13.3 Overarching Conclusions . . . . .	147
13.4 Future Works . . . . .	148
<b>Appendices</b>	<b>150</b>
<b>A Summary of Publications</b>	<b>153</b>
A.1 Peer-Reviewed Articles . . . . .	153
A.2 Technical Reports . . . . .	153
<b>B Reference Cards: Simulation Models and Specifications</b>	<b>155</b>
<b>Glossary of Terms</b>	<b>166</b>



# List of Figures

1.1	Overview of subjects on which hybrid fuzzy controllers are based. . . . .	1
2.1	Levels of detail classified spatially from a camera. . . . .	4
2.2	The evolution of behavioural control in animation. . . . .	5
2.3	Boids used in animated film. . . . .	6
2.4	The boid “neighbourhood”. . . . .	7
3.1	Classifying frequency of light into fuzzy sets for colour. . . . .	12
3.2	2-input 1-output fuzzy controller design . . . . .	13
3.3	Example 5-set fuzzy value for speed. . . . .	14
3.4	Levels-of-detail behaviour. Fuzzy behaviour used for mid-distance pedestrians. . . .	16
4.1	A combination of simulation models displayed in a 3D simulation. . . . .	19
4.2	Production process for generating to-scale vehicle models. . . . .	20
4.3	Crowd simulation based on perceptual studies. . . . .	21
4.4	Motion capture acting and capture process. . . . .	22
4.5	User-assisted dynamic landscape generation. . . . .	23
4.6	Real-time fuzzy controller plot visualisation. . . . .	26
5.1	Agents with near-identical architectures in different scenarios. . . . .	28
5.2	A society of agents based on a hierarchical organisation. . . . .	30
5.3	An agent hierarchy used to control road traffic. . . . .	31
5.4	A typical “stack” architecture used for a robot soccer player agent. . . . .	32
5.5	A diagram of the rôle of middleware for agents. . . . .	33
5.6	A modular middleware architecture with exchangeable components. . . . .	34
5.7	A typical abstracted 2D environment map, and features. . . . .	36
5.8	Resolutions of planning used by a 3-level agent hierarchy in a 3D environment. . . .	38
5.9	Heuristic representation of the Fuzzy A* navigation algorithm . . . . .	39
5.10	Convergent cascades of fuzzy associative memory matrices for action-selection. . . .	40
5.11	Agents cooperating over a shared world model . . . . .	41
6.1	Hybrid fuzzy-A* soccer robots. . . . .	45
6.2	To-scale model of real vehicle with mechanical simulation. . . . .	47
6.3	Representing combined heuristic cell weights in environment graph. . . . .	48
6.4	Classifying environment elements into overlapping fuzzy sets. . . . .	49
6.5	Scalable fuzzy navigation cascade with feedback loop. . . . .	49
6.6	Graphing navigation data in real time. . . . .	51
6.7	Proposed algorithm for machine-learning agent. . . . .	52
7.1	Simulated city requiring traffic. . . . .	57
7.2	Street map of city model area. . . . .	58

7.3	Nodular graph representation of road network. . . . .	59
7.4	Lane occupancy and following distance model. . . . .	60
7.5	Non-intrusive traffic module design. . . . .	62
7.6	Laying road segments in an existing simulation. . . . .	63
7.7	Laying road networks over a 3D landscape. . . . .	64
7.8	3D Dublin Streets with Abstract Road Included. Birds-Eye. . . . .	65
7.9	Simulated traffic follow lane-mapped roads in 3D city model. . . . .	66
8.1	Agent traffic system in operation. . . . .	70
8.2	Test vehicle model: Enviro400 Bus . . . . .	70
8.3	Agent architecture for road traffic. . . . .	72
8.4	Fuzzy input set functions; angle to obstacle. . . . .	75
8.5	Fuzzy input set functions; angle to destination. . . . .	76
8.6	Fuzzy input set functions; distance to destination. . . . .	76
8.7	Fuzzy input set functions; distance to obstacle. . . . .	77
8.8	Fuzzy set midpoints; desired speed. . . . .	78
8.9	Fuzzy set midpoints; to-destination steering. . . . .	79
8.10	Fuzzy set midpoints; avoidance steering. . . . .	79
8.11	Prototype in operation; congestion. . . . .	82
8.12	Prototype in operation; lane demarcations. . . . .	83
8.13	Final product; simulated traffic with lane map and fuzzy steering. . . . .	85
9.1	Test environment for agents to navigate through. . . . .	91
9.2	To-scale 3D model of the T-28E vehicle with continuous tracks. . . . .	92
9.3	Score obtained by an agent moving through test environment . . . . .	93
9.4	Architecture for a dynamic real-time training. . . . .	94
10.1	Diagram of fuzzy set functions; angles. . . . .	99
10.2	Overlapping fuzzy input sets in a spatial example. . . . .	100
10.3	Component architecture; fuzzy process. . . . .	101
10.4	Simulation plug-in architecture. . . . .	102
10.5	Operation of the breeding tool chain. . . . .	102
10.6	Mapping fuzzy rules to a chromosome representation. . . . .	104
10.7	Graph; controlled benchmark of genetic algorithm process. . . . .	106
10.8	Vehicle controlled by GFS move through test environment. . . . .	107
10.9	Graph; effect of mutation probability on fitness. . . . .	108
10.10	Graph; effect of mutation range on fitness. . . . .	109
10.11	Graph; effect of population size on fitness. . . . .	110
10.12	Graph; effect of number of parents on learning curve. . . . .	111
11.1	Loosely coupled mechanical simulation architecture. . . . .	113
11.2	Views of a simulated Willys MB jeep. . . . .	116
11.3	Before and after effect of car paint shader. . . . .	117
11.4	Several vehicles rendered with shaders and compositors . . . . .	119
11.5	Comparison of buses rendered with and without effects. . . . .	120
11.6	Comparison of DUKWs rendered with and without effects. . . . .	121
11.7	Diagram of forces acting in complete drive-train. . . . .	122
11.8	The RPM-torque curve for the Willys MB Jeep. . . . .	123
11.9	The RPM at final drive for gears in a jeep. . . . .	124
11.10	A mechanical blueprint overlays its simulated 3D shape. . . . .	128
11.11	An inexpensive method for simulating tracked vehicles. . . . .	129
11.12	Views of animated Caterpillar track deformation. . . . .	130



11.13	Articulation in <i>Mechanix</i> with an unpowered trailer. . . . .	131
11.14	Photo of a universal joint with axiis of rotation overlaid. . . . .	132
12.1	Close-up view of “forest” obstacle course. . . . .	138
12.2	Wide-angle view of “forest” obstacle course. . . . .	139
12.3	Control system used for destination-seeking behaviour. . . . .	141
12.4	A mixed function motion control system with algorithmic switches. . . . .	142
12.5	Accumulated mean fitness for pillars course. . . . .	144



# List of Tables

3.1	Complete table of example rules. . . . .	14
8.1	Fuzzy input term definitions; route following. . . . .	74
8.2	Fuzzy input term definitions; obstacle avoidance. . . . .	75
8.3	Fuzzy output term definitions; route following. . . . .	77
8.4	Fuzzy output term definitions; obstacle avoidance. . . . .	78
8.5	$3 \times 3$ FAMM for desired speed; obstacle avoidance . . . . .	80
8.6	$3 \times 3$ FAMM for desired steering; obstacle avoidance . . . . .	80
8.7	$3 \times 3$ FAMM for desired speed; route following . . . . .	81
8.8	$3 \times 3$ FAMM for desired steering; route following . . . . .	81
10.1	Comparison of features of agent control systems. . . . .	99
10.2	Rules for change to steering in the obstacle avoidance component of a simulated car. . . . .	100
10.3	The most-fit chromosomes from selected generations in an evolutionary run. The chromosomes are converging towards an optimal individual of all zeros. . . . .	105
10.4	Baseline genetic parameters for GFS . . . . .	112
B.1	Classical mechanics simulation model parameter values used. . . . .	156
B.2	Summary of mechanical simulation models used. . . . .	157
B.3	Specifications used for mechanical simulation of a Willys MB Jeep. . . . .	158
B.4	Specifications used for mechanical simulation of a Enviro 400 bus. . . . .	159



# List of Algorithms

- 1 Mechanical simulation component update loop. . . . . 115
- 2 End run condition detection in “forest” scenario. . . . . 140
- 3 Logic for mixed motion control system output switch. . . . . 142



# Acknowledgements

This work would not have been possible without my supervisors at Massey University; Professor Ken Hawick, who insisted upon the highest scientific standards, and Doctor Napoleon Reyes, who inspired me to start moving things with fuzzy logic. I am in a great debt of gratitude to Professor Carol O'Sullivan, who provided support and guidance for 2 years of study in Ireland with the Graphics, Vision, and Visualisation (GV2) group at Trinity College Dublin.

I would like to acknowledge my talented peers, from whom I have gained a great deal of knowledge and experience during the last 3 years; Prof. Henry Rice, Dr Martin Johnson, Dr Chris Messom, Dr Sébastien Paris, Dr Simon Dobbyn, Dr Rachel McDonnell, Dr Yann Morvin, Dr Veronica Sundstedt, Dr Guy Kloss, Dr Ljiljana Skrba, Dr Sophie Jörg, Dr Darren Caulfield, Dr Cathy Ennis, Dr Steve Collins, Dr Rozenn Dahyot, Cormac O'Brien, Nithin Tharakan, Andrew Corcoran, Dr Michéal Larkin, Dr Arno Leist, Dr Daniel Playne, Fintan McGee, Martin Pražák, Teo Susnjak, Brian Cullen, Eric Risser, Jonathan Ruttle, Robert Smyth, Paul McDonald, and Tom Van Eyck.

I would also like to thank Craig Reynolds for his invaluable advice regarding motion control.





# Abstract

This thesis describes implementations of motion control systems that are based on fuzzy logic; fuzzy motion controllers. The controllers are used by to drive a variety of simulated vehicles and computer-animated characters. The problem of heading towards a destination whilst simultaneously avoiding static and dynamic obstacles is addressed with fuzzy motion controllers. For situations where a level above reactive motion control is required, such as path-planning behaviour or traffic rule following, then hybrid algorithms are proposed; combining fuzzy motion controllers with other navigation algorithms. Consideration is given to behavioural level of detail models, with transition between behavioural models of different complexity based on the proximity, or visual importance of characters to the camera.

Fuzzy controllers have a set of fuzzy rules, or a “rule base” that defines the inference of the controller. There is no assurance that hand-calibrated rule bases are optimal, and indeed that calibration based on fixed test environment will apply well to a dynamic environment. Special consideration is given to genetic-fuzzy systems, which use a genetic algorithm to automatically calibrate a rule base. Various architectures for genetic-fuzzy system are proposed and evaluated including dynamic systems, which have the ability to learn “on the fly”, rather than in fixed experiment scenarios. A relationship between genetic algorithm parameters and time-efficient fitness improvement is found. The time requirements of training more complex “cascading” fuzzy systems are discussed. Distributed and parallel training models are also considered.

A new, modular agent middleware is proposed, which is the underpinning software that perceives the complex environment, feeds inputs into the fuzzy motion controllers, and effects output actions for each character and vehicle. The middleware model is successfully used to drive a range of vehicles and characters used in experiments.

The problem of evaluating motion controllers within a scientific framework is discussed. Several candidate solutions are used, and a system for objectively evaluating mechanically simulated vehicle motion is defined and evaluated. A complete tool-chain for designing complex simulations and doing scientific experiments with them is developed and discussed in detail, including simulation software design methods, libraries, visualisation tools, and useful algorithms, a well-defined mechanical simulation system, and practices for collecting data from simulations, and quantifying uncertainty.