

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.

Genetic Network Programming with Fuzzy Reinforcement Learning Nodes for Multi-Behaviour Robot Control

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

**Masters of Science
In
Computer Science**

**Massey University – Albany Campus
New Zealand.**

**Wenhan Wang
2014**

Abstract

This research explores a new approach for building a complex intelligent robot multi-behaviour comprising of a variety of intelligent subsystems that are fused together into one hybrid system. The work mainly focuses on integrating reinforcement learning and fuzzy logic with genetic network programming, examining the different architectures, and aims to achieve multi-objective behaviours and alleviate the problem of learning and calibration by repeated interaction with the environment. Different components of the learning algorithm are studied separately and also in combination. They are developed systematically using an increasing level of complexity for robot behaviours. As a test bed, the work investigates how to achieve ball pursuit and wall avoidance behaviours simultaneously, in the realm of the robot soccer game. The training procedure and test environment is designed, as well as a variety of fitness functions are experimented for the multi-behaviour objectives. Furthermore, the novel evolutionary architecture is combined with hill-climbing to accelerate the search for the best individual.

Keywords—robot soccer; multi-behaviour; multi-objectives; genetic network programming; fuzzy logic; reinforcement learning;

Acknowledgments

I would like to express my gratitude to all those who helped me during my research. My deepest gratitude goes first and foremost to Dr. Napoleon Reyes (Supervisor) and Dr. Andre Barczak (Co-Supervisor), for their constant encouragement and guidance. They have walked me through all the stages of this research. Without their consistent and illuminating instruction, this thesis could not have reached its present form. Thanks for the time you spent with me.

My thanks would also go to my beloved parents for their loving considerations and great confidence in me through all these years. I also owe my sincere gratitude to my uncle's family who gave me their help and time in supporting and helping me living in New Zealand.

Table of Contents

| | |
|---|-----|
| Abstract..... | I |
| Acknowledgments..... | II |
| List of Figures | V |
| List of Tables..... | VII |
| List of Pseudo Codes | VII |
| Chapter 1 Introduction | 1 |
| 1.1. Overview of the Current State of Technology | 1 |
| 1.2. Research Objectives..... | 2 |
| 1.3. Scope and Limitations of Research | 2 |
| 1.4. Overview of the Problem Domain | 3 |
| 1.5. Significance of the Research | 3 |
| 1.6. Research Methodology..... | 4 |
| 1.7. Structure of the Thesis Documentation | 5 |
| Chapter 2 Review of Related Literature | 7 |
| 2.1. Fuzzy Logic Control | 7 |
| 2.1.1. Fuzzy Sets and Membership | 7 |
| 2.1.2. Algorithm Description..... | 8 |
| 2.2. Reinforcement Learning | 9 |
| 2.2.1. Markov Decision Process | 9 |
| 2.2.2. General Description | 10 |
| 2.2.3. Temporal Difference Learning | 12 |
| 2.3. Genetic Network Programming | 18 |
| 2.3.1 The Basics of GNP | 18 |
| 2.3.2 Initialization the GNP | 22 |
| 2.3.3 Running a GNP Individual | 22 |
| 2.3.4 Genetic Operators..... | 22 |
| 2.4. GNP with Reinforcement Learning | 23 |
| 2.4.1. Basic Structure of GNP-RL..... | 23 |
| 2.4.2. Running a GNP-RL Individual | 25 |
| 2.5. Summary..... | 26 |
| Chapter 3 Adaptations of the Algorithms for Robot Control: Single Behaviour | 27 |
| 3.1. Problem Domain Specifications..... | 27 |
| 3.2. Algorithm 1: Fuzzy Logic Controller..... | 28 |
| 3.2.1. General Architecture | 28 |
| 3.2.2. Problem-Specific Parameter Settings | 29 |

| | | |
|-------------|---|----|
| 3.2.3. | Experiment Results and Analysis | 31 |
| 3.2.4. | Limitations of the Algorithm..... | 32 |
| 3.3. | Algorithm 2: Reinforcement Learning with Fuzzy Logic | 32 |
| 3.3.1 | General Architecture | 33 |
| 3.3.2 | Problem-Specific Parameter Settings | 35 |
| 3.3.3 | Results and Analysis..... | 36 |
| 3.3.4 | Limitations of the Algorithm..... | 40 |
| 3.4. | Algorithm 3: Genetic Network Programming with Reinforcement Learning..... | 41 |
| 3.4.1. | General Architecture | 41 |
| 3.4.2. | Problem-Specific Parameter Settings | 43 |
| 3.4.3. | Results and Analysis..... | 45 |
| 3.4.4. | Limitations of the Algorithm..... | 47 |
| 3.5. | Summary..... | 47 |
| Chapter 4 | Fuzzy-Reinforcement Learning for Robot Multi-behaviour | 49 |
| 4.1 | Introduction | 49 |
| 4.2 | General Architecture | 50 |
| 4.3 | Problem-Specific Parameter Settings | 51 |
| 4.3.1 | Fuzzy Logic Parameters..... | 51 |
| 4.3.2 | Reinforcement Learning Parameters | 52 |
| 4.4. | Results and Analysis..... | 54 |
| 4.4 | Limitations of the Algorithm..... | 61 |
| Chapter 5 | GNP with Trained Fuzzy-RL Nodes for Learning Multi-Behaviours | 63 |
| 5.1 | General Architecture | 63 |
| 5.2 | GNP with Trained Fuzzy-RL Pseudo Code: Training Phase..... | 66 |
| 5.3 | Problem-Specific Settings | 66 |
| 5.3.1. | Judgment Nodes | 66 |
| 5.3.2. | Processing Nodes..... | 67 |
| 5.3.3. | GNP Fitness Function for Integrated Target Pursuit and Wall | 69 |
| 5.3.4. | Parameters for GNP | 72 |
| 5.3.5. | Hill-climbing Algorithm | 72 |
| 5.4 | Results and Analysis..... | 73 |
| 5.5 | Limitations of the Algorithm..... | 78 |
| Chapter 6 | Summary and Future Work | 79 |
| Appendix A. | Codes for the implementation of Fuzzy-RL | 84 |
| Appendix B. | Codes for the implementation of GNP with trained Fuzzy-RL nodes...88 | |
| Reference | | 96 |

List of Figures

| | |
|---|----|
| Figure 2.1 Classic sets and fuzzy sets | 8 |
| Figure 2.2 Sample fuzzy sets for distance | 8 |
| Figure 2.3 Structure of genetic network programming | 18 |
| Figure 2.4 Structure of the gene of a node | 19 |
| Figure 2.5 Schematic diagram of the genetic network programming algorithm (training phase) | 20 |
| Figure 2.6 Schematic diagram of genetic network programming (testing phase) | 21 |
| Figure 2.7 Judgment and Processing Node in GNP | 22 |
| Figure 2.8 Processing node and judgment node with sub-nodes | 24 |
| Figure 2.9 Schematic diagram of GNP- RL running in the testing phase | 25 |
| Figure 3.1 2D simulation environment | 28 |
| Figure 3.2 Flowchart of fuzzy logic control system | 28 |
| Figure 3.3 Fuzzy logic system design (NL-Negatively Large, NM-Negatively Medium, NS-Negatively Small, ZE-Zero, PS-Positively Small, PM-Positively Medium and PL-Positively Large) | 29 |
| Figure 3.4 Angle fuzzy sets | 29 |
| Figure 3.5 Distance Fuzzy Sets | 30 |
| Figure 3.6 Trace of ball and robot (fuzzy logic controller) | 31 |
| Figure 3.7 Schematic diagram of RL with fuzzified input algorithm | 33 |
| Figure 3.8 Schematic diagram of the Fuzzy-RL algorithm | 34 |
| Figure 3.9 Angle Fuzzy Sets | 35 |
| Figure 3.10 Initial part of RL with fuzzified input algorithm | 37 |
| Figure 3.11 Results of Algorithm 2a: Average angle from ball every 50 time steps (y-axis = ave. angle; x-axis: 1 unit = 50 time steps) | 37 |
| Figure 3.12 Results of Algorithm 2b: Learning phase of the Fuzzy-RL algorithm | 38 |
| Figure 3.13 Results of Algorithm 2b: Average angle from ball every 50 time steps (y-axis = ave. angle; x-axis: 1 unit = 50 time steps) | 38 |
| Figure 3.14 The performance after running for a while (RL with fuzzified input algorithm) | 39 |
| Figure 3.15 The performance after running for a while (Fuzzy-RL algorithm) | 40 |
| Figure 3.16 Schematic diagram of GNP with RL for training phase | 41 |
| Figure 3.17 Schematic diagram of GNP with RL for testing phase | 43 |
| Figure 3.18 Judgment node settings | 43 |
| Figure 3.19 Processing node settings | 44 |
| Figure 3.20 Fitness of the best individual (y-axis = fitness; x-axis = generation count) | |

| | |
|--|----|
| | 45 |
| Figure 3.21 Performance of the GNP with RL | 46 |
| Figure 4.1 Calculation of difference between the heading angle of the ball, and the nearest wall..... | 49 |
| Figure 4.2 Schematic diagram of the Fuzzy-RL algorithm..... | 50 |
| Figure 4.3 Angle from ball Fuzzy Sets | 51 |
| Figure 4.4 Angle from wall Fuzzy Sets..... | 51 |
| Figure 4.5 Distance from wall Fuzzy Sets | 52 |
| Figure 4.6 Average angle from ball (measured every 500 time steps) during robot training..... | 54 |
| Figure 4.7 Pre-defined restricted area used in the experiments. The ball is initially placed within the black region depicted in the figure. The white region is the prohibited area. | 55 |
| Figure 4.8 Close to the wall counts every 500 time steps | 55 |
| Figure 4.9 Trained sample close to wall 1 performance 1 | 57 |
| Figure 4.10 Trained sample close to wall 1 performance 2 | 57 |
| Figure 4.11 Trained sample close to wall 2 performance 1 | 58 |
| Figure 4.12 Trained sample close to wall 2 performance 2 | 58 |
| Figure 4.13 Trained sample close to wall 3 performance 1 | 59 |
| Figure 4.14 Trained sample close to wall 3 performance 2 | 59 |
| Figure 4.15 Trained sample close to wall 4 performance 1 | 60 |
| Figure 4.16 Trained sample close to wall 4 performance 2 | 60 |
| Figure 5.1 Modified GNP Individual used in the new algorithm | 63 |
| Figure 5.2 Schematic diagram of GNP with trained Fuzzy-RL nodes algorithm | 64 |
| Figure 5.3 Judgment node settings..... | 67 |
| Figure 5.4 Absolute angle of the robot relative to the field | 67 |
| Figure 5.5 Angle from ball fuzzy sets | 70 |
| Figure 5.6 Sample GNP individual with the minimum number of nodes. Note that the algorithm may generate a variety of individuals with different nodes and connections..... | 73 |
| Figure 5.7 General performance of a good individual | 74 |
| Figure 5.8 The performance close to wall 1..... | 75 |
| Figure 5.9 The performance close to wall 2..... | 75 |
| Figure 5.10 The performance close to wall 3..... | 76 |
| Figure 5.11 The performance close to wall 4..... | 76 |
| Figure 5.12 Fitness of top 3 individuals with hill climbing..... | 77 |
| Figure 5.13 Fitness of top 3 individuals without hill climbing | 77 |

List of Tables

| | |
|--|----|
| Table 2.1 State-action space for RL in GNP-RL algorithm | 24 |
| Table 3.1 Fuzzy Associative Memory Matrix for Ball Pursuit: Steering Angle Adjustment..... | 30 |
| Table 3.2 Fuzzy Associative Memory Matrix for Ball Pursuit: Speed Control | 31 |
| Table 3.3 State-Action space (States are the truth value from Fuzzy system, Actions are steering angles for the robot) | 36 |
| Table 3.4 The state-action space of RL..... | 42 |
| Table 3.5 Performance data of top five individuals | 46 |
| Table 4.1 The ID of states for corresponding input combination (Distance from wall is near)..... | 52 |
| Table 4.2 State-Action space (y-axis: ID of RL States, x-axis: actions)..... | 53 |
| Table 5.1 Fuzzy rules for calculating the ball pursuit behaviour fitness. | 70 |
| Table 6.1 Comparison of different algorithms | 79 |

List of Pseudo Codes

| | |
|---|----|
| Pseudo code 1: TD(0) algorithm (Sutton, et al., 2012) | 12 |
| Pseudo code 2: Sarsa (On-Policy) algorithm (Sutton, et al., 2012) | 13 |
| Pseudo code 3: Q-Learning (On-Policy) algorithm (Sutton, et al., 2012)..... | 13 |
| Pseudo code 4: TD(λ) algorithm (Sutton, et al., 2012)..... | 16 |
| Pseudo code 5: SARSA(λ) algorithm (Sutton, et al., 2012)..... | 17 |
| Pseudo code 6: Q(λ) algorithm (Sutton, et al., 2012) | 17 |
| Pseudo code 7: Reward function for RL with FLS..... | 36 |
| Pseudo code 8: Reward function for GNP-RL..... | 44 |
| Pseudo code 9: Reward function for ball pursuit and wall avoidance..... | 54 |
| Pseudo code 10: GNP with Trained Fuzzy-RL..... | 66 |
| Pseudo code 11: Fitness function for speed control behaviour | 69 |
| Pseudo code 12: Fitness function for wall avoidance | 70 |
| Pseudo code 13: Final fitness function | 71 |
| Pseudo code 14: Hill-climbing algorithm | 72 |

