

Abstract

In this work, an interface between Matlab, a high level matrix oriented programming language, and the Khepera, miniature, modular experimental robot platform was developed. This interface allowed learning control algorithms to be prototyped and tested quickly and easily on a real robot with minimal effort being invested in the development of the robot or learning the robot programming language. This Matlab/Khepera interface was then used in conjunction with a Behaviour Analysis and Training (BAT) methodology to develop a learning algorithm for a light approaching/obstacle avoidance task. This learning task provided a validation of the methodology and of the interface.

The learning architecture of the control algorithm used for the light approaching/obstacle avoidance task was based on Elman Simple Recurrent Networks trained using Complementary Reinforcement Back-Propagation (CRBP). It was determined that the Matlab-Khepera interface was very effective for developing and simulating the control algorithms within the Matlab environment as well as implementing the algorithms on the Khepera robot in real time. However, the limited sensing capabilities of the Khepera base unit (without additional hubs) caused the full application of the Behaviour Analysis and Training Methodology to become difficult and restricted the application of hierarchical control structures in building more intelligent behaviour. Finally, it was determined that training with CRBP can be speeded up by the application of an exploration rule based on a Gaussian distribution of random numbers instead of a uniform distribution.

Dedication

You know, almost everyone dedicates their work to their spouse. I used to wonder why that was.

Then one day, I found out.

We went Christmas shopping. I spent five hours in a mall with my wife (and six billion other people) and I spent the whole time staring inward at an imaginary black-board trying to figure out the current week's problem with my robot.

I had a hard time even remembering what mall we went to on cross-examination (and I drove).

I would like to dedicate this work to my wife Sharon who received an undue foretaste of life with me when I'm 90. I now know that she'll drag the "mumbling vegetable" to the mall every now and then for some fresh air and a walk.

Acknowledgements

I would like to acknowledge Dr. J. Wilson for his assistance in this endeavor.

I would also like to recognize the following people/entities:

- HAL 9000: who germinated the seed that grew into this work. Compared to HAL, my robot is as dumb as a rock but at least it won't try to kill me. Happy birthday HAL.
- My appendix. Enya. PEZ. Pookie, Lexus, and Tish. The fish. The Simpsons. COPS. Tim Hortons.
- The Internet: it provided me with an abundance of information, feedback, *research papers*, and games. I would also like to recognize the people on the internet with whom I have conversed but never actually met.
- The techs, Pierre Adam, and Kim. Not for the technical support, but rather for listening to my ramblings about how to micromanage a farm.
- Last, but certainly not least, I would like to thank my friend Eric Deoliviera who rounded my post-graduate experience in all aspects non-academic. He shamed me into going to the gym ("the House Of Pain") and forced me get a yellow belt in Karate. He would certainly make a good demon lord of hell.

Vita

Name: Richard Gary Goyette

Place and Date of Birth: Timmins, Ontario, 08 February 1968

Education: Royal Military College of Canada 1995-1997
Royal Military College of Canada 1987-1991. Awarded Bachelor of Engineering (Electrical) May 1991.

Experience: Post Graduate Student, Royal Military College of Canada, 1995-1997
Secure Systems Integration and Management, Directorate of Intelligence and Security Operations Automation, 1992-1995
Trainee, Canadian Forces School of Communications and Electronics, 1991-1992

Table of Contents

Abstract	i
Vita	ii
Acknowledgements	iii
Dedication	iv
Table of Contents	v
List of Figures	ix
List of Tables	x
Chapter 1 Introduction	
1.1 General	1-1
1.1.1 Integrated Tools for Experimentation	1-2
1.1.2 Structured Methodologies for Control System Design	1-4
1.1.3 Reactive Versus Dynamic Behaviour	1-5
1.1.4 Simple Recurrent Networks	1-7
1.1.5 Speeding Up Learning With SRN's	1-8
1.2 Thesis Objectives	1-8
1.3 Thesis Outline	1-9
Chapter 2 Background	
2.1 Introduction	2-1
2.2 The Behaviour Analysis and Training Methodology	2-1
2.3 Reinforcement Learning Model	2-3
2.4 Learning from Delayed Reinforcements	2-4
2.4.1 Mathematical Framework for Delayed Reinforcement	2-5
2.4.2 AHC and Q-Learning	2-7
2.4.3 Advantages and Disadvantages of Delayed Reinforcement	2-9
2.5 Learning from Immediate Reinforcements	2-9
2.6 Feed-Forward and Simple Recurrent Neural Networks	2-11
2.6.1 Feed Forward Neural Networks	2-11
2.6.1.1 Neural Network Training: Forward Propagation	2-12
2.6.1.2 Neural Network Training: Back-Propagation	2-14
2.6.2 Elman Simple Recurrent Networks	2-16
2.6.3 Advantages of Neural Networks for Knowledge Representation	2-18
2.7 Network Training Algorithm	2-19
2.7.1 The Structural Credit Assignment Problem	2-19
2.7.2 Complementary Reinforcement Back-Propagation	2-20
2.7.3 Increasing the Learning Speed of CRBP	2-22

2.7.3.1	Exploration VS Exploitation and Network Confidence	2-22
2.7.3.2	Gaussian Distributed Random Exploration	2-24
Chapter 3	Application Description	
3.1	Introduction	3-1
3.2	Description of the Robot Shell	3-1
3.3	Description of the Initial Environment	3-4
3.4	Requirements on the Target Behaviour	3-4
3.5	The Programming Environment	3-4
3.5.1	Introduction	3-4
3.5.2	Robot/PC Interface	3-5
3.5.3	Khepera Toolbox Core Functionality	3-6
3.5.4	Additional Functionality and Graphical User Interfaces's	3-7
3.5.5	The Khepera Simulator	3-7
Chapter 4	Behaviour Analysis	
4.1	Introduction	4-1
4.2	Behaviour Decomposition	4-1
Chapter 5	Specification	
5.1	Introduction	5-1
5.2	Modifications to the Environment	5-1
5.3	The Controller Architecture	5-2
5.4	The Training Strategy	5-3
5.4.1	Shaping Policy	5-3
5.4.2	Reinforcement Program	5-3
5.5	Learning System Design and Implementation	5-5
5.5.1	Introduction	5-5
5.5.2	Behaviour Module Design and Implementation	5-5
5.5.2.1	Introduction	5-5
5.5.2.2	Module Block Diagram Overview	5-6
5.5.2.3	Obstacle Avoidance Module	5-11
5.5.2.3.1	General	5-11
5.5.2.3.2	Sensor Input Representation: prepoo.m	5-11
5.5.2.3.3	Reinforcement Program: rprobst.m	5-13
5.5.2.4	Light Seeking/Approaching Module	5-15
5.5.2.4.1	General	5-15
5.5.2.4.2	Sensor Input Representation: prepol.m	5-15
5.5.2.4.3	Reinforcement Program: rplight.m	5-16
5.5.2.5	Controller Architecture Design and Implementation	5-18
5.5.2.5.1	Introduction	5-18
5.5.2.5.2	Modifications to the Behaviour Module	5-18
5.5.2.5.3	Module Output Integration Function	5-19

Chapter 6	Training	
6.1	General	6-1
6.2	Network Training Variables	6-1
	6.2.1 Summary of Variables	6-1
	6.2.2 Selection of Variable Default Values	6-2
6.3	Behaviour Module Training: Obstacle Avoidance	6-8
	6.3.1 Real World Training Environment	6-9
	6.3.2 Simulation Training Environment	6-9
	6.3.2.1 Sensor Calibration	6-9
	6.3.2.2 Simulation Environment	6-10
6.4	Behaviour Module Training: Light Seeking	6-11
	6.4.1 Real World Training Environment	6-11
	6.4.2 Simulation Training Environment	6-12
	6.4.2.1 Sensor Calibration	6-12
	6.4.2.2 Simulation Environment	6-15
6.5	Integration Function Training	6-16
Chapter 7	Behaviour Assessment	
7.1	Introduction	7-1
7.2	Behaviour Metrics	7-1
7.3	Obstacle Avoidance Module Performance	7-5
	7.3.1 Uniform versus Gaussian Random Exploration Rule	7-5
	7.3.2 Threshold Exploration	7-9
	7.3.3 Simulator Trained and Transplanted Modules	7-10
7.4	Light Seeking Module Performance	7-13
	7.4.1 Real World Trained Module	7-13
	7.4.2 Simulator Trained and Transplanted Module	7-16
7.5	Global Behaviour Analysis	7-17
	7.5.1 Initial Test Environment	7-17
	7.5.2 Global Performance in the Initial Environment	7-18
	7.5.3 Extended Environments	7-21
	7.5.4 Global Performance in the Extended Environments	7-22
Chapter 8	Discussion and Conclusions	
8.1	Discussion	8-1
	8.1.1 Effectiveness of Khepera Toolbox: Matlab-Khepera Interface	8-1
	8.1.2 Effectiveness of Khepera Toolbox: Khepera Simulator	8-2
	8.1.3 Integration of BAT Methodology and Khepera Toolbox	8-3
	8.1.2.1 Complexity of Reinforcement Programs	8-3
	8.1.2.2 Limitation on Intelligent Control	8-7
	8.1.4 Exploration vs Exploitation in CRBP	8-10
8.2	Conclusions	8-11
8.3	Future Work	8-13

Bibliography

Bib-1

Attachments

*Matlab Functions for Obstacle Avoidance and Light Seeking
Khepera Toolbox for Matlab V4.1c (Tech Report RMC97-2)*

List of Figures

Figure 2-1	Generic Reinforcement Learning Model	2-4
Figure 2-2	Typical Feed-Forward Neural Network	2-11
Figure 2-3	Typical Neuron Activation Computation	2-13
Figure 2-4	Elman Simple Recurrent Network	2-17
Figure 3-1	Two Views of the Khepera	3-2
Figure 3-2	Block Diagram of the Khepera Robot	3-3
Figure 3-3	Typical Tethered Experimental Configuration	3-5
Figure 5-1	Flat Architecture with Integrated Outputs	5-2
Figure 5-2	The Learning System-Trainer Relationship	5-4
Figure 5-3	Block Diagram of Typical Behaviour Module	5-6
Figure 5-4	Elman Network Connectivity	5-7
Figure 5-5	Typical Proximity Sensor Response	5-12
Figure 5-6	Trained Module Architecture	5-18
Figure 6-1	Log Sigmoid Activation Function	6-3
Figure 6-2	Hyperbolic Tangent Activation Functions	6-6
Figure 6-3	Histogram of Gaussian Distribution	6-7
Figure 6-4	Playpen Arrangement for Training in the Real World (obstacle avoidance)	6-9
Figure 6-5	Simulator Training Environment (obstacle avoidance)	6-10
Figure 6-6	Playpen Arrangement for Training in the Real World (light seeking)	6-12
Figure 6-7	Calibration Results	6-14
Figure 6-8	Simulator Training Environment (light seeking)	6-15
Figure 7-1	Obstacle Avoidance Punishment Index Data for Real Khepera	7-6
Figure 7-2	Creating a “New Situation”	7-7
Figure 7-3	Individual Training Run of Gaussian Random Exploration Rule	7-8
Figure 7-4	Obstacle Avoidance Punishment Index Data for Simulated Khepera	7-11
Figure 7-5	Punishment Data - Real Khepera with Initial Knowledge	7-12
Figure 7-6	Light Seeking Punishment Index Data for Real Khepera	7-14
Figure 7-7	Approach Path due to Bad Sensor	7-15
Figure 7-8	Light Seeking Punishment Index Data for Simulated Khepera	7-16
Figure 7-9	Punishment Data - Real Khepera with Initial Knowledge	7-17
Figure 7-10	Initial Test Environment	7-18
Figure 7-11	The Eternal Loop	7-20
Figure 7-12	Extended Environments for Global Performance Evaluation	7-22
Figure 8-1	Original Controller Architecture	8-8
Figure 8-2	The Near Sighted Problem	8-9

List of Tables

Table 3-1	Khepera Serial Commands Executable from within Matlab	3-6
Table 5-1	Binary Vector Action Selection	5-9
Table 5-2	Determination of T and LR in rpx.m	5-10
Table 6-1	List of Module Variables	6-2
Table 6-2	Calibration Variable Settings for Simulated Light Sensors	6-13
Table 7-1	Global Performance Results	7-24