

Acknowledgement

In memory of **Dr. Kevin P. Granata**, my graduate advisor, who was killed protecting others on the morning of April 16, 2007.

There are many others without whom I could not have completed my thesis. I wish to thank:

My family, for their love and support.

Kelli Sparks, for being with me through the hardest of times, giving me a place to stay, and helping me keep my sanity.

My graduate committee, especially Dr. Mary Kasarda who took the time to edit this thesis.

Dr. Jeffery Holland and Dr. Naira Hovakimyan for externally reviewing my thesis.

Shimon Whiteson for his helpful correspondence on the NEAT+Q algorithm.

The Kevin P. Granata Musculoskeletal Biomechanics Lab, especially Brad Davidson and Martin Tanaka for their helpful discussions spanning many of the topics covered in this thesis.

Dr. Jim Shine for taking the time to help me with neural networks and back-propagation.

Dr. Ramu Krishna for providing information on neural networks.

My friend Ross Alameddine (who was also lost on April 16), for volunteering to edit my thesis.

Table of Contents

1	Introduction	1
1.1	Motivation & Significance	1
1.2	Hypotheses & Specific Aims	2
1.3	Project Scope.....	3
1.4	Overview	4
2	Literature Review & Interpretation, Mathematical basis	6
2.1	Literature Review & Interpretation	6
2.1.1	Bipedal Robots	6
2.1.1.1	Active Static Walking	6
2.1.1.2	Passive Dynamic Walking.....	6
2.1.1.3	Active Dynamic Walking	7
2.1.2	Machine Learning.....	8
2.1.2.1	Markov Decision Process	9
2.1.2.2	Reinforcement Learning	9
2.1.2.3	Q-Learning.....	11
2.1.2.4	Neural Networks.....	13
2.1.2.5	Neuro-Evolution of Augmenting Topologies.....	18
2.2	Mixed Differential-Algebraic Equations of Motion.....	22
3	Pendulum Swing-up/Balance Task using Q-Learning	26
3.1	Introduction	26
3.2	Methods.....	27
3.2.1	Pendulum Dynamics.....	27
3.2.1.1	Q-Learning.....	29

3.3	Results and Discussion	30
3.4	Conclusion	34
4	Pendulum Swing-up/Balance Task using NEAT+Q.....	36
4.1	Introduction	36
4.2	Methods	38
4.2.1	Pendulum Dynamics	38
4.2.2	Evolutionary Function Approximation for Reinforcement Learning	39
4.3	Results and Discussion	40
4.4	Conclusion	45
5	Active Dynamic Bipedal Walking with NEAT+Q.....	46
5.1	Introduction	46
5.2	Methods	47
5.2.1	Walker Dynamics.....	47
5.2.2	Evolutionary Function Approximation for Reinforcement Learning	50
5.3	Results and Discussion	51
5.4	Conclusion	56
6	Summary, Interpretation, Future work.....	57
6.1	Summary	57
6.2	Interpretation	58
6.3	Future Work	58
7	Appendices:.....	60
7.1.1	Complete set of data tables	60
7.1.2	Computer code.....	61
7.1.2.1	C++ Table-Based Q-Learning Program Code:	61
7.1.2.2	MATLAB NEAT+Q Main File:.....	67

7.1.2.3	MATLAB NEAT+Q Walker Experiment File:.....	72
7.1.3	References	80

List of Figures

Figure 2.1. Q-learning: An off-policy TD control algorithm. From Sutton et al.[47].	11
Figure 2.2. Visual representation of how the SxA Q-value matrix (a) composes the policy (b). The states shown $(\theta, \dot{\theta})$ represent position and velocity respectively. Each matrix of (a) corresponds to one of three control actions: top = +20 Nm, middle = 0 Nm, bottom = -20 Nm. The policy, represented as a combined matrix (b) shows which actions are favored in which states.	12
Figure 2.3. Diagram of an artificial node y . Inputs are denoted by x_i , and the bias unit is represented by input b_j .	14
Figure 2.4. Plot of the sigmoid function, and the derivative of the sigmoid function over the interval $[-8, 8]$.	15
Figure 2.5. Simple example of how NEAT evolves more complicated structure. A node is added by splitting a connection in two, a link is added by adding a connection between nodes (bold line) that did not previously exist. From Whiteson et al. [59].	18
Figure 2.6. Genetic cross-over involves matching up the genomes of both parents. Innovation numbers are shown at the top of each gene; connections are represented in the middle. Non-matching genes are either disjoint or excess, depending on whether their innovation number is larger or smaller than the other parent's largest innovation number. If a gene is disabled in either parent, there is a chance it will be disabled in the offspring as well. From Stanley et al. [44].	19
Figure 2.7. NEAT+Q Algorithm designed to evolve an ANN function approximator for Q-learning. From Whiteson et al. [59].	21
Figure 3.1. Diagram of single pendulum (mass, $m = 10$ kg, length, $L = 2$ m, length scalar $d = 0.5$).	27
Figure 3.2. Q-learning: An off-policy TD control algorithm. From Sutton et al. [47].	29
Figure 3.3. (Color Plot) Optimized Value Function: maximum Q-value of each action over state-space.	31
Figure 3.4. (Color Plot) The policy determined by selecting the best action in each state. The policy appears to be converging on a radially-symmetric solution. The state-space behavior of the pendulum (white) as it swings-up and balances.	32
Figure 3.5. Periodic behavior of the pendulum as it oscillates near the vertical.	33
Figure 3.6. (Color Plot) De-parameterized pendulum behavior. Where ω is the angular frequency of the pendulum defined by equation 3.5.	34
Figure 4.1. Diagram of single pendulum (mass, $m = 10$ kg, length, $L = 2$ m, length scalar $d = 0.5$).	38

Figure 4.2. (color plot) Evolved neural network structure for the pendulum swing-up/balance task. The connection weights ranged from red excitory (positive) connections to blue inhibitory (negative) connections with values ranging from -8 to $+8$ respectively. Node 1 represents position, node 2 represents velocity, node 3 is the bias node, and nodes 4-6 are the Q-values corresponding to $+20$, 0 , and -20 Nm. 41

Figure 4.3. (Color Plot) Evolved Value Function: maximum Q-value of each action over state-space. 42

Figure 4.4. (Color Plot) The policy determined by selecting the best action in each state. The white line represents the state-space behavior of the pendulum. The policy with NEAT+Q appears to only consider one of the two solutions that the look-up table method found. 43

Figure 4.5. Periodic behavior of the pendulum as it oscillates about $(0, \pi)$ 44

Figure 4.6. (Color Plot) De-parameterized pendulum behavior. Where ω is the angular frequency of the pendulum defined by equation 4.5. 45

Figure 5.1. Diagram of the three-segment bipedal walker. The square denotes the revolute ankle joint of the stance leg fixed to the coordinate plane. The joint at the hip is also revolute. 48

Figure 5.2. (color plot) Evolved ANN topology for the active dynamic walking task. The network is composed of input nodes (1-8), a bias node (9), output nodes (10-15), and a hidden node (110). 1-3 represent stance, swing, and torso angle; 4-6 represent stance, swing, and torso angular velocity; 7-8 represent torso and swing leg torque; 10-15 represent a Q-value corresponding to each action respectively (Table 5.1). The connection weights ranged from blue inhibitory (negative) connections to red excitory (positive) connections with values of -8 to $+8$ respectively. 52

Figure 5.3. Alternative walking solutions. Both networks are more complex with worse behavior than the solution presented previously. 53

Figure 5.4. Filtered control torques, normalized by the maximum available torque. Maximum torque values were 200 Nm and 30 Nm , respectively. (a) Torso torque (blue) spikes during foot-strike and decreases throughout the swing phase. Swing leg torque (green) is largest right after foot-strike and tends to decrease by the end of the swing phase. (b) More detail can be seen by focusing in on a shorter time period. 53

Figure 5.5. (Color Plot) (a) Stance leg (red) , swing leg (green) , and torso (blue) angles over time. Discontinuities in leg angles exist at foot-strike when the stance leg becomes the swing leg, and vice-versa. (b) Focusing on the first few seconds reveals how chaotic the gait was in the first few steps. (c) Several seconds later the gait appears to become more uniform, and certain patterns emerge. 54

Figure 5.6. Results of a passive downhill compass-gait walker with a constrained torso. From Wisse [64]. 55

Figure 5.7. Comparison between the mid-point of the stance and swing leg angles (maroon) and the torso angle (blue). Foot-strike events are indicated with orange diamonds..... 56

List of Tables

Table 4.1. An approximate number of elements composing a value function with 3 actions. White rows utilize the multi-dimensional matrix representation, while gray* rows utilize the equivalent neural net function approximator (note: table assumes a fully connected network with no hidden nodes).	37
Table 5.1. Each action represents two binary inputs (torque recruitment) to two first-order torque filters.	51
Table 7.1. Anthropomorphic data used for three-segment active dynamic bipedal walker. From de Leva [10].	60