EXPLORING JOINT-LEVEL CONTROL IN EVOLUTIONARY ROBOTICS

By

Jared M. Moore

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

Computer Science - Doctor Of Philosophy

2015

ABSTRACT

EXPLORING JOINT-LEVEL CONTROL IN EVOLUTIONARY ROBOTICS

By

Jared M. Moore

Animals exhibit a remarkable variety of behaviors and morphologies. Evolving together over millions of years, brain and body are tightly coupled. By borrowing characteristics from nature, robotic systems can be produced that emulate the capabilities of natural organisms. In this dissertation, we use computational evolution and physics simulations to explore both control and morphology in robotic systems. Specifically we investigate joint-level control strategies and their interaction with morphological elements.

Our results demonstrate that evolutionary approaches are effective at producing controllers that are highly integrated with their morphology. Controllers are able to exploit passive properties of materials, such as flexibility, to effectively locomote in various environments. Moreover, the joint-level control strategy proposed in this dissertation, which abstracts the functionality of muscular systems, is used to study both biological principles and robotic controllers.

This dissertation explores a bio-inspired strategy that more closely resembles the cascading series of control observed in natural organisms. We demonstrate that evolved joint-level controllers can produce effective gaits in a variety of robotic systems, even when governed by a simple high-level control signal. Results support employing hierarchical control in robotic systems, and constructing control and morphology together during the design phase. In addition, we show that digital simulation can be an effective tool to study biomechanics, opening the door to further investigations of biological phenomena.

TABLE OF CONTENTS

LIST C	OF TABLES	vi		
LIST C	LIST OF FIGURES			
Chapte	$er 1 Introduction \dots \dots$	1		
1.1	Motivations	1		
1.2	This Research	4		
1.3	Outline	5		
Chapte	er 2 Background	8		
2.1	Controller Evolution	9		
2.2	Evolution of Morphology and Control	12		
2.3	Open Questions	14		
Chapte	er 3 Evolution of Control: Station Keeping	5		
3.1	Simulation Environment	17		
	3.1.1 Robot Model	17		
	3.1.2 Simulated Environment	18		
	3.1.3 Neural Controller	19		
3.2	Experiments	20		
3.3	Results	22		
	3.3.1 Evolved Behaviors	22		
	3.3.2 Fitness Measurements	23		
	3.3.3 Behavior Comparison	24		
	3.3.4 Discussion	25		
3.4	Toward Generalized Station Keeping	27		
3.5	Summary	27		
Chapte	er 4 Exploiting Passive Joints in an Amphibious Robot	60		
4.1	Introduction	30		
4.2	Methods	31		
	4.2.1 Robot Overview	32		
	4.2.2 Treatments and Evaluation	33		
	4.2.3 Evolutionary Process	34		
	4.2.4 Prototype Fabrication	35		
4.3	Experiments and Results	36		
	4.3.1 Treatment 1 - Terrestrial Environment Only	36		
	4.3.2 Treatment 2 - Aquatic Environment Only	37		
	4.3.3 Treatment 3 - Amphibious Environments	37		
	4.3.4 Fitness Evaluation	38		
	4.3.5 Treatment Comparisons	39		
	4.3.6 Physical Validation	12		
4.4	Summary	42		

Chapte	er 5 Evolution of Whole Body Morphology: The Role of the Tail in	
	Bipedal Hopping	45
5.1	Background and Related Work	47
5.2	Methods	48
	5.2.1 Virtual Animat	49
	5.2.2 Modeling of Muscles and Joints	49
	5.2.3 Evolutionary Setup	50
5.3	Experiments and Results	52
	5.3.1 Treatment 1: No Tail	52
	5.3.2 Treatment 2: Fixed, Rigid Tail	52
	5.3.3 Treatment 3: Actuated Tail	53
	5.3.4 Treatment 4: Tail Collision Removal	54
	5.3.5 Treatment 5: Evolvable Tail Morphology	54
	5.3.6 Performance Comparison	55
	5.3.7 Analysis	55
5.4	Conclusions	59
Chapte	er 6 The Digital Muscle Model	61
6.1	Background and Related Work	62
6.2	Digital Muscle Model	63
	6.2.1 Control	64
	6.2.2 Morphology	65
	6.2.3 Motor Control Signal Generation	65
6.3	Experiments	67
	6.3.1 Treatment 1 - Digital Muscle Model	68
	6.3.2 Treatment 2 - ANN Controller	69
6.4	Results	69
	6.4.1 Sample of Evolved Muscle Model Gaits	69
	6.4.2 Evolution of Symmetric Movements	70
	6.4.3 Evolution of a Functional Knee	71
	6.4.4 ANN Evolved Controllers	72
	6.4.5 Performance Comparison	75
6.5	Possible Applications	76
6.6	Conclusions	78
Chapte	er 7 Combining the Digital Muscle Model and High-Level Control .	79
7.1	Related Work	80
7.2	ANN/Muscle Model Integration	81
	7.2.1 Evolutionary Setup	82
7.3	Quadruped Locomotion	82
	7.3.1 Quadruped Results	84
7.4	Hexapod Locomotion	89
	7.4.1 Hexapod Results	90
75	Conclusions	95

Chapte	er 8 ANN/DMM Interactions 9	7	
8.1	Robot Platform)8	
8.2	Evolutionary Setup)8	
8.3	Results)8	
	8.3.1 Gaits)8	
	8.3.2 Analysis)()	
	8.3.3 Singly- versus Individually-Connected)5	
8.4	Conclusions)6	
Chapte	er 9 Conclusion and Future Work)8	
DIDTIC	OGILAI II I	IJ	

LIST OF TABLES

Table 3.1:	Description of Inputs to the ANN	20
Table 4.1:	Individual Gene Limits	34
Table 5.1:	Individual Gene Limits	51
Table 7.1:	NEAT Parameters for Gait Evolution	83
Table 8.1:	Worm-Like Robot NEAT Parameters	99
Table 8.2:	P-values of pairwise comparison using a Wilcoxon Rank Sum Test for the farthest traveling individual per replicate from the three treat- ments. The three metrics are listed on the left: fitness, number of con- nections and number of hidden nodes in the evolved networks. Treat- ments are abbreviated as follows: (S) singly-connected, (I) individually- connected, and (A) ANN-only.	107

LIST OF FIGURES

- Figure 2.1: Artificial neural networks consist of neurons connected by weighted synapses. Each neuron contains an activation function specifying how the inputs obtained from the weighted synapses map to an output. The activation function is typically a Sigmoid, see above, but can be other mathematical functions. Output of each neuron is propagated through the network to ultimately define commands to be sent to motors.
 10

17

23

- Figure 3.3: Behavior of an evolved solution in Trial 2. The first 60s, which is the transient phase, is used to reorient against a laminar flow pushing on the robot from the rear (left to right in the figure). The robot executes a 180 degree flip bringing the caudal fin into a position where it can provide the greatest thrust. Here, the flippers roll the robot over as well as make minor adjustments once the robot is in an effective position.
- Figure 3.4: An evolved solution in Trial 4. In this trial, an individual faces a laminar flow at a 45° angle to the robot's front. The robot spends the first 50s reorienting itself against the flow. After 50 seconds, the robot has achieved a stable station and begins to accumulate high levels of fitness by using the flippers and fin in a coordinated effort to maintain its center over the station point.

Figure 3.5:	(a) Fitness of the best evolved results from the trials. Trial 1 is able to achieve a near perfect fitness score as it does not have to reorient itself prior to holding station. The other trials have some success, although their fitness scores are lower than Trial 1. This is likely due to the movements required to reorient to flows. (b) Average fitness of the population of evolved results from the trials. Trials 2, 3, and 4 have low average fitnesses due to the difficulty of station keeping in these environments. Many individuals are able to swim, but leave the station area, accruing no fitness during the evaluation period	25
Figure 3.6:	Box plot of the best fitness values for each replicate run in the four trials. Box indicates the upper and lower quartiles, median is represented as the center line in the boxes. Ends of the whiskers represent the maximum and minimum values, excluding outliers	26
Figure 3.7:	Two-dimensional, top-down trajectory plot for an evolved solution and an early generation candidate solution from Trial 4. The grey circle represents the area of fitness reward. The evolved solution is initially displaced before identifying the direction of flow, reacting to it, and then achieving station.	26
Figure 3.8:	Depiction of the sweeping side-to-side flow facing a robot over the course of an evaluation. At time t=0s, the flow comes from the front of the robot. As the simulation progresses, a second component is added, altering the direction of flow up to a $\pm 63.4^{\circ}$. The magnitude of the force also varies, as the x component of the flow remains constant throughout the simulation.	28
Figure 4.1:	Simulation model of the robot used in this study. The arms move in a sweeping motion pivoting at the top of the fins through a passive hinge joint.	32
Figure 4.2:	CAD model of the passive hinge joint used in the fin structure. The arm extends from the body and is waterproofed by a flexible gasket. The arm attaches to the fin by a passive joint allowing the fin to move according to the environmental forces acting upon it.	33
Figure 4.3:	3D printed robot of the best performing solution in this study. The passive hinge joint can be seen between the arm and fin. The fins collapse backwards as the arms move forward allowing the robot to move in both terrestrial and aquatic environments	36

Figure 4.4:	Three different morphologies evolved in different treatments. Subfigure (a) shows the dominant morphology that emerged in Treatment 1; note the tall pectoral fins. The dominant morphology for Treatment 2 is presented in (b) and is characterized by shorter pectoral fins. The morphology that emerged in Treatment 3 can be seen in subfigure (c). This adaptive morphology exhibits a compromise in the pectoral fin height between the terrestrial and aquatic morphologies, enabling the robot to perform well in both environments.	37
Figure 4.5:	Fitness landscape of all individuals over the evolutionary trials for so- lutions evolved in Treatment 3. Fitness values have been normalized using the best values from Treatments 1 and 2 to allow comparison between single environment evolved individuals and amphibious indi- viduals. Darker shaded areas indicate a large number of solutions with similar performance. The box in the top right contains solutions that perform well in both environments	39
Figure 4.6:	Average performance of the best individual solutions per trial from Treatment 3 over evolutionary time. (a) Aquatic environment fitness and (b) terrestrial environment fitness. Fitness was based on the per- formance in both environments, as such, the performance does not nec- essarily increase over each generation. Larger variations in the aquatic environment could potentially indicate that solutions are more suscep- tible to small changes than those in the terrestrial environment	40
Figure 4.7:	Distribution of the oscillation frequency of the controllers in the aquatic environment in regard to fin area. The oscillation frequencies are dependent upon the fin area for a morphology with 90% of the variance in oscillation frequency being accounted for by the fin area with a p-value $< 2 \ge 10^{-16}$.	41
Figure 4.8:	Initial testing of the printed model in an aquatic environment verifies that the passively hinged fin performs similarly to the characteristics seen in the simulation. During the recovery stroke, the fin flexes up- wards, pivoting on the passive joint, to reduce drag and enable forward movement of the robot	43
Figure 4.9:	(a) A hand-designed prototype of the robot produced with multiple materials. The feet are composed of both soft and rigid materials. This initial prototype has a completely rigid arm which results in a small contact area between the foot and surface. During movement, the feet often slip. (b) Virtual robot used in this experiment modeled after the physical prototype. A flexible joint is located between the arm and foot of the robot allowing the foot to flatten on the surface, increasing its contact area with the ground based on evolved parameters.	44

Figure 5.1:	The kangaroo rat was selected as the base morphology for studying the evolution of bipedal hopping, due to its representative morphology and the availability of information on both the mechanics and dynamics of its behavior.	46
Figure 5.2:	X-ray video progression of a kangaroo rat hopping across a force plate to quantify hopping dynamics.	48
Figure 5.3:	(a) Initial simulated animat used in this study, with morphological dimensions and mass based on kangaroo rat. (b) Two-dimensional representation of the animat joints, with range of motion indicated.	49
Figure 5.4:	Behavior of evolved tailless and fixed-tail individuals. The fixed tail individual is able to hop more effectively by using its tail as a stabilizer to prevent flipping over backwards.	53
Figure 5.5:	An evolved hopping individual from Treatment 3 with an actuated tail. Note the coordination between tail and legs to maintain body pitch throughout the hopping motion. In the evolved individual from Treatment 5, the tail evolves to be shorter than those of the previous treatments, enabling faster hopping	54
Figure 5.6:	Fitness of 5 treatments over evolutionary time: (a) Best performing individual, averaged across 25 runs for each treatment; (b) Average performance in each evolved population, averaged across 25 runs for each treatment. Shaded bands indicate 95% confidence intervals	56
Figure 5.7:	(a) Relationship between the leg oscillation frequency and tail oscilla- tion frequency in Treatment 5. The straight lines indicate harmonics between the two frequencies. Evolved solutions tend to either fall near these lines or in the passively flexible region. (b) Relationship between the leg oscillation frequency and tail mass as a percentage of total body mass. Lighter tails are favored, although the evolved tail length remains relatively constant even for different masses	58
Figure 5.8:	Relationship between the leg oscillation frequency and moment of in- ertia for an individual. A low moment of inertia generally means the animat is likely to change body pitch during hopping	59

Figure 6.1:	A Digital Muscle Group is composed of nodes, radially distributed around a joint on a 2D plane. Conceptually, nodes exert a pulling force, which draws the limb segment towards the node's position on the plane. Antagonistic relationships emerge between nodes leading to coordinated movement of a joint. The outputs of a digital muscle group dictate the movement of a joint in a physics simulation engine.	64
Figure 6.2:	A top down view of an individual muscle group consisting of four nodes placed radially around a joint. Each node has both an activation function and a spatial component. Together, these determine the strength and direction of pull placed on a joint by the individual node	66
Figure 6.3:	Activation functions for four nodes in a muscle group. An input signal determines the response of nodes according to the Gaussian activation functions for each. The values of the activation functions for an input signal of -0.5 are highlighted	66
Figure 6.4:	Activations for the 2 DOF controlled by a muscle group. The nodes depicted in Figure 6.3 map to the two commands seen in this figure. The humps in both curves near 0.4 are the result of Node 3, which is a Gaussian function with $\mu = 0.4$. Activations take into account both the activation function for a node as well as its spatial location. Nodes for this muscle group are radially distributed at 45°, 135°, 225°, and 315° as in Figure 6.2.	67
Figure 6.5:	Example of an input signal being converted to joint commands in the rear hip of a quadrupedal animat. (a) A signal (in this case a simple sinusoid) from a higher level controller is distributed to each muscle group (b) which is then passed to all nodes in the group (c). Each node takes the input value and determines its output by finding the point on the Gaussian indicated by the input. This output is then combined with the spatial position of the node to determine the output for each DOF. The outputs of all nodes in the joint are aggregated (d) to derive the two motor movement commands for a joint in a robot (e). These commands are then sent to the motors (f) associated with the joint.	68
Figure 6.6:	Examples of evolved gaits in digital muscle based animats. (a) Rear leg driven bounding gait with left/right symmetric motion. (b) Three legged pace gait, where the left legs move in unison, out of phase with the right rear leg. (c) A three legged bounding gait with rear legs	

moving in near unison.

Figure 6.7:	Evolution of forward and back movement of the rear hips in a bounding individual. Positive angles indicate forward movement. Initially, the joint movements are not synchronized and differ in amplitude. As evolution progresses, movement of the hips becomes synchronized with the joint angles moving toward a common phase, amplitude and period.	72
Figure 6.8:	Evolution of movement away from the body in the rear hips of a bound- ing individual. Positive angles indicate movement away from the body. Movement of the rear hips is initially quite different, with the gap clos- ing over the course of evolution to ultimately converge towards similar phase, period and amplitudes.	73
Figure 6.9:	Evolution of the rear hip joint trajectories. The x-axis represents move- ment toward and away from the body. Values near 0 represent move- ments near the robot, while larger values indicate movements away from the robot. For the right hip, negative values on the x-axis indi- cate movements under the robot, while positive values are associated with movements away from the body. In the left hip, the opposite is true. The right hip initially crosses over the 0 boundary, resulting in the leg being under the robot. Later generations exhibit symmetric movements mirrored about 0 on the x axis	73
Figure 6.10:	The evolved muscle node positions for the two rear hips do not directly mirror each other as only three of the four nodes have similar spatial positions indicated by the red rectangles. Instead, symmetry in the expressed joint movements is a combination of both node position and activation functions relating to each muscle node. Actual movements are symmetric and coordinated as seen in Figures 6.7, 6.8, and 6.9	74
Figure 6.11:	Joint movement for the left rear knee over evolutionary time. The joint initially moves somewhat erratically in both degrees of freedom, with a noticeable hitch. At generation 50, the joint has a balanced movement between both axes but still has jitter. This results in jerky movement of the lower limb. As evolution continues, the movement becomes planar, using a combination of both degrees of freedom. A functional knee joint then evolves with the lower limb moving steadily back and forth without much side-to-side movement	74
Figure 6.12:	Evolved three legged bounding gait using an ANN-based controller. The main body remains low to the ground throughout the evaluation period, emphasizing stable locomotion. Limbs exhibit symmetric move- ments, likely due to the complexification of the ANN over evolutionary time.	75

Figure 6.13	Evolutionary fitness progressions for both treatments. Shaded areas indicate the 95% confidence intervals across 20 replicate runs per treatment. Both the maximum fitness distribution and average fitness distribution are significantly different (p <0.001)	76
Figure 6.14:	The average body height above the ground for all replicate runs be- tween the two treatments. Shaded areas indicate 95% confidence inter- vals with the two distributions being significantly different (p <0.001). As a whole, gaits from Digital Muscle Model controllers evolve higher postures as legs are typically held closer to vertical. Whereas, ANN controllers evolve gaits that tend to remain closer to the ground, splay- ing the legs outward	77
Figure 7.1:	Examples illustrating the two connection strategies tested in this study: (top) singly-connected and (bottom) individually-connected. Interac- tion between the ANN and DMM-based joint proceeds as follows: (a) ANN receives input from sensors and produces output(s), 1 for a singly- connected joint and 4 for an individually-connected joint. (b) For a singly-connected joint, the same ANN output signal is distributed to each of 4 muscle nodes. For an individually connected joint, each mus- cle node receives its own signal directly from the ANN. (c) The position and activation function of each muscle node determines its response to the incoming signal. (d) Responses of the muscle nodes are combined and (e) passed to the platform.	81
Figure 7.2:	The quadruped robot has eight 2-DOF joints, a hip and a knee for each leg.	83
Figure 7.3:	Average maximum fitness across 20 replicate runs per treatment in the quadruped platform. Shaded areas represent the 95% confidence intervals.	84
Figure 7.4:	The average number of connections in the evolved ANNs of the highest performing individual per replicate versus fitness. Each point repre- sents the average number of connections and average fitness across twenty replicates per treatment for a generation. Dashed vertical lines indicate the average number of connections in the highest performing generation per treatment	85
Figure 7.5:	The number of connections versus fitness for the highest performing individuals from each replicate run for the quadruped platform. Dashed vertical lines indicate the farthest traveling individual per each treatment.	86

Figure 7.6:	Distribution of fitnesses for the best individual per replicate across the three controllers for quadruped locomotion. Results are significantly different between the hybrid and ANN-only controllers. There is no significant difference between the singly- and individually-connected controllers	87
Figure 7.7:	Number of connections for the best individual from each repli- cate across the three treatments in quadruped locomotion. The individually-connected strategy is significantly different from the other two, while there is no significant difference between singly-connected and ANN-only controllers	88
Figure 7.8:	Number of hidden nodes versus fitness for the farthest traveling individ- ual from each replicate in the quadruped platform. In contrast to the number of connections versus fitness, there is not a clear relationship between the number of hidden nodes and fitness.	88
Figure 7.9:	Number of hidden nodes across treatments for the quadruped platform.	89
Figure 7.10	The hexapod robot has twelve 2-DOF joints, a hip and a knee for each leg. Movement of the legs can be away from the torso, or along the long axis of the torso.	90
Figure 7.11	Average maximum fitness per generation across 20 replicate runs per treatment in the hexapod platform. Performance between the two hybrid controllers is similar while the ANN-only controller lags behind.	91
Figure 7.12	Average number of connections of the farthest traveling individual across replicates per generation versus fitness for the hexapod platform.	91
Figure 7.13	Number of connections versus fitness for the farthest traveling individ- ual from each replicate in the hexapod robot. Dashed vertical lines indicate the farthest traveling individual per each treatment	92
Figure 7.14	The fitness distributions for the three controllers in the hexapod platform. There is no significant difference between singly- and individually-connected ANN/DMM controllers while both are signif- icantly different than the ANN-only controller	93
Figure 7.15	Number of connections for the farthest traveling individual from each replicate across the three treatments in the hexapod platform. Sim- ilar to the quadruped, the individually-connected strategy has a sig- nificantly higher number of connections while there is no significant difference between singly-connected and ANN-only controllers	94

Figure 7.16:	Number of hidden nodes versus fitness for the farthest traveling indi- vidual from each replicate for the hexapod platform. There is not a clear relationship between hidden nodes and fitness.	94
Figure 7.17:	Number of hidden nodes for the best individual from each replicate for the hexapod robot. There is no significant difference between singly- and individually-connected controllers while both are signifi- cantly larger than the ANN-only controllers	95
Figure 8.1:	Three different worm-like robots. The overall shape and mass of the robot remains constant throughout the different trials. (a) Three-joint, (b) five-joint, and (c) ten-joint robot.	97
Figure 8.2:	A sample of three gaits, one from each treatment. (Top) ANN-only evolved controller that exhibits a rolling gait, curling and unfolding to produce movement. (Middle) Singly-connected controller with a hopping gait. The rear of the worm acts as a primitive leg. (Bot- tom) Individually-connected controller with a walking gait. The ends of the robot act as legs, moving the robot sideways with step-like move- ments	99
Figure 8.3:	Mean maximum fitness across 20 replicate runs per treatment in the worm platform. Each plot represents the three treatments for the given number of joints.	101
Figure 8.4:	Boxplot showing the fitness of the farthest traveling individual per replicate for the three treatments across the different DOF. The hybrid ANN/DMM controllers tend to have higher fitnesses than the best ANN controllers.	102
Figure 8.5:	Number of connections versus fitness in the farthest traveling individ- uals from each replicate run for the worm platform across the twelve joints	103
Figure 8.6:	Number of connections for the farthest traveling individuals from 20 replicate runs per each DOF across the three treatments. Differences are statistically significant for all except singly- versus individually-connected one joint ($p = 0.9042$) and singly-connected versus ANN-only six ($p = 0.4017$) and eight joints ($p = 0.1404$)	104

Figure 8.7:	Number of hidden nodes for the farthest traveling individuals from 20		
	replicate runs per each DOF across the three treatments. Differences		
	are statistically significant for all ANN/DMM versus ANN-only con-		
	trollers. There are no significant differences in the number of hidden		
	nodes for singly- and individually-connected controllers except for 7		
	joints $(p = 0.0047)$	105	

Chapter 1 Introduction

1.1 Motivations

The growth of computing power in the past 60 years has coincided with a shift from large physical installations to increasingly mobile computing platforms. In particular, the application of robots and other embedded systems to industrial manufacturing [53], environmental monitoring [70, 124], and biological research [71, 80] has fundamentally changed the nature of computing, in that such systems need to interact with the physical world. An intriguing class of such systems includes autonomous robots, which operate independently of direct human control. As computational footprints continue to decrease, autonomous robots have the potential to service an expanding range of applications. However, outstanding issues relating to behavioral complexity and the physical robot design require continued research. With added functionality comes increased difficulty to effectively use onboard computing resources. Designing effective control strategies for autonomous robots and integrating morphological function into controllers remains an ongoing challenge.

Nature provides a great deal of inspiration for robotic systems. *Bio-inspired* designs [4, 45, 75, 119] draw upon structures and behaviors from the natural world and apply them to their robotic counterparts. Body configurations, joint control patterns and weight distribution are a few of the characteristics of natural organisms that inform the design of hardware components and control systems. However, animals demonstrate a fluidity and grace to movement not currently matched in artificial systems. Musculoskeletal systems have high energy capacities, while at the same time exhibiting flexibility and elasticity allowing for smooth movements. Unlike motors and mechanical actuators, muscles are flexible, or *compliant* [129, 113, 24], responding to external forces as well as commands from the brain. Realizing this functionality in robots is desirable not only to enhance mobility, but also to facilitate safe interaction with humans [48].

One approach to bio-inspired design is *biomimetics*, where characteristics of natural systems are directly emulated in engineered systems. However, purely biomimetic approaches do not account for the many differences between biological organisms, composed of bones and soft tissue, and engineered systems comprising motors and (usually) rigid components. Instead, a biomimetic approach is often extended with additional optimizations to account for these disparities. For example, robotic fish have been engineered with a flexible caudal fin inspired by nature, but optimized to account for properties of the constituent materials [31]. Bio-inspiration has also been employed to integrate a hopping gait for a mobile robot in order to extend communication range in a sensor network [33].

Rather than mimic characteristics of a biological organism, an alternative approach is to harness the process that has led to complex natural organisms: evolution. Evolutionary Computation (EC) [69] utilizes concepts from natural evolution to solve optimization problems and, when applied to robots, can exploit the intrinsic properties of materials and actuators available in robotic systems [27]. Genetic Algorithms (GAs) [59], a subset of EC, address optimization or search problems through an iterative process. A population of genomes, each representing a candidate solution, is evaluated and perturbed with evolutionary operators [12]. Over the course of many generations, candidate solutions are altered by mutation, crossover with other individuals in the population, and selection for reproduction based on performance metrics. GAs are well suited to tasks where the steps toward a "good" solution are not known *a priori*, such as in the training of artificial neural networks (ANNs) [94] for controlling mechanical systems or performing other complex tasks. Computational evolution can aid in the integration of bio-inspired features by optimizing a design over generations, slowly incorporating and tweaking specific parameters to generate effective individuals. In the case of robots, evolutionary methods provide a means to explore many designs, often in a simulated environment, without requiring a supervisor to explicitly compose each solution. This area of study is referred to as Evolutionary Robotics (ER), where EC is applied to produce effective control systems and morphologies. Evolutionary approaches have produced effective gaits for quadrupeds [35, 49, 47], hexapods [6], bipeds [109], and robotic salamanders [63]. Furthermore, bio-inspired studies have yielded insight into function of the human brain [44], hopping as a form of robotic locomotion [134, 1], vertical climbing robots [119] and fins for aquatic robots [76]. In addition, evolutionary methods have been used to aid in the optimization of morphology [3, 106, 95, 34]. However, in many evolutionary robotic studies, the control strategy is monolithic. Specifically, a high-level controller of the system dictates not only overall behavior, but the movements of all (low-level) actuators. In this dissertation, we apply bio-inspiration to control models that more closely emulate the cascading system of control found in natural organisms.

Beyond optimizing robotic designs, ER approaches can also inform biological study. Behaviors exhibited by animals often involve a high degree of coordination as well as robustness to uncertain and changing environmental conditions. Understanding the fundamental drivers for these behaviors [92], or specific function of individual body parts [101, 91], depends on how the organisms evolved. However, observations of natural organisms are limited only to extant species and the fossil record. While such studies generate a wealth of knowledge, variations in body components might yield information regarding the specific role of morphological traits and expressed behaviors of animals. Digital simulation provides an effective method of exploring these phenomena and has previously been applied to the study of fish fins [31, 123, 131], salamanders [63, 65], social dynamics in populations [38, 13, 73] and bipedal hoppers [96]. Even with the tremendous amount of computing power available, however, fine-grained simulations of musculoskeletal systems is currently impractical when large numbers of individual simulations are required [91]. The application of evolutionary computation techniques to such studies further increases the demand for computational power, motivating the development of efficient models to approximate the function of natural organisms.

1.2 This Research

Musculoskeletal systems allow for precise movements with dexterity and finesse, as well as explosive movements requiring raw power. At the joint-level, multiple muscles work together moving limbs, enabling an organism to interact with its environment. These lowlevel behaviors are coordinated by the nervous system, which acts as a high-level controller. Low-level control includes reflexes that do not necessarily fall under the domain of conscious control. Moreover, preflexes, the intrinsic properties of muscles themselves, provide a zerodelay mechanism to stabilize an individual without any neural control. Together, these highand low-level systems, along with morphological traits, define the scope of behaviors available to an individual.

Several recent studies have addressed the evolution of controllers and morphological characteristics for robots. However, controllers typically generate signals to move joints through single degree of freedom (DOF) motors, in contrast to natural organisms that move by coordinating the action of multiple muscles. This research addresses the challenge of incorporating high- and low-level control and morphology in the robotic design process.

Thesis Statement Computational evolution can be applied to develop effective locomotion in robotic systems, where control exploits aspects of the system's morphology, including whole-body characteristics, passive joints, and flexible materials. Further, it is possible to evolve robotic systems that emulate the cascading network of control found in natural organisms, where the the actions of individual joints, in response to signals from a high-level controller, depends on both evolved joint-level control and joint morphology.

Our overall approach is to explore the optimization of control systems and morphology

together through computational evolution. In the following chapters, we present individual studies demonstrating the coupling between control and morphology that emerges from the evolutionary process. In addition, we show that effective robotic systems arise when some aspects of control are relegated to the joint-level. The following main contributions have been made in support of this thesis:

- 1. We demonstrate the relationships between control and morphology that emerge in aquatic and terrestrial robots.
- 2. We produce a joint-level control mechanism, modeled after biological muscles, that can generate basic locomotion with a simple oscillating input signal.
- 3. We integrate joint-level and high-level control, showing that the combination can outperform a monolithic controller in various locomotion tasks while having a smaller number of connections in the evolved high-level ANNs.

1.3 Outline

We begin our investigations by examining the evolution of control and morphology independently. Chapter 3 describes our work in evolving neural controllers for aquatic stationkeeping robots. Here, we apply evolutionary computation to the optimization of ANNs in a robot whose morphology is fixed. Evolved ANNs exhibit novel behaviors that exploit the morphology of the robot to hold station effectively in various flow conditions. Next, in Chapter 4, we investigate the integration of passive joints in simulated robots. Robotic joints are typically directly controlled by a motor, however, natural organisms have passive properties with automatic responses dictated by the physical structure of the joint. The dynamics of passive joints complicate their integration into robots, requiring new techniques to account for the lack of control. We demonstrate that passive characteristics of the joint can be accounted for in the control strategy as an integral component of the robotic system. In our evolutionary experiments, the highest-performing individuals exhibit clear relationships between their brain and body.

Chapter 5 addresses the evolution of whole body morphology and related control parameters for a simulated kangaroo rat. The study focuses specifically on the tail of the animat and how it affects bipedal hopping. Here, aspects of both control and morphology are evolved. Animats exhibit a variety of effective gaits as the form and function of the tail are changed across treatments. Gaits range from bounding to full bipedal hopping while being driven only by a periodic oscillating signal, demonstrating the effectiveness of low-level control for locomotion. Even though the initial animats are based on the morphological dimensions of a kangaroo rat, the most effective individuals evolve various tail morphologies, highlighting the difference between natural and engineered systems.

We examine joint-level control in greater detail in Chapter 6. Here, we propose the *Dig-ital Muscle Model* (DMM), a joint-level control strategy that emulates properties of natural muscles, specifically spatial positioning and activation, to apply to robotic systems. Yet, the model remains abstract, capable of generating commands for servo-driven robots even though they are single DOF actuators. The DMM also facilitates the study of biological questions regarding musculoskeletal systems. Such a computationally efficient model is essential for evolutionary approaches, as the high number of evaluations makes fine-grained simulations intractable. Our results indicate that the DMM is capable of evolving effective gaits for a quadruped animat even when driven by a simple sinusoidal signal. Furthermore, biological parallels arise during evolution, with symmetry and functional specialization of joints becoming evident.

Chapters 7 and 8 explore the integration of the DMM with a high-level controller. Alone, the DMM lacks sensory information about the environment and the internal state of the robot. A high-level controller, in our case an ANN, is required to integrate external feedback. This hierarchical model is similar to that of natural organisms, where some control is relegated to the joint-level. Together, the ANN/DMM controllers are able to effectively control the robotic systems. Our analysis focuses on both performance and the properties of the evolved ANNs. Results show that the hybrid ANN/DMM controllers outperform monolithic ANN-only controllers as the complexity (in terms of degrees of freedom) increases. Hybrid controllers exhibit a larger number of hidden nodes compared to ANN-only controllers, potentially indicating that ANNs must compensate for a concurrently evolving low-level control strategy. However, the ANNs of hybrid controllers also have fewer connections when compared to the ANN-only controllers. Moreover, performance in the hybrid controllers indicates that the combined ANN/DMM controllers are effective in producing high-performing gaits.

Next, we review background information on topics relevant to this research, followed by a presentation of the work itself.

Chapter 2

Background

Evolutionary Robotics (ER) [103] borrows concepts from natural evolution to develop the brain and body of both simulated and physical robotic systems. Since the seminal work of Brooks [29] and Sims [115, 116], the field of ER has grown extensively, reporting considerable success in evolving robots to perform specific tasks. Indeed, most studies address subproblems such as gaits [27, 35, 47, 114], single environment exploration [23], or instances of learning itself [7, 9]. While effective in performing prescribed functions, however, computationally evolved systems have yet to attain the robustness observed in natural organisms.

A major challenge in ER is the so-called "Reality Gap" [29, 104], namely, that solutions effective in simulation do not perform similarly in the real world. One potential solution is to evolve robot controllers on-board a physical robot [25, 104], however, doing so often requires significant real-world time and limits the effective population sizes in the experiment. Simulation therefore remains an important resource in ER but requires methods to address the reality gap. Approaches include: simulating only the basic properties of a problem [67, 68], adding noise to sensor data to emulate the imperfect nature of physical environments [93], introducing safety margins to component specifications [50], and even devising transferability metrics to include as an estimate of performance [74]. In addition, a technique known as self-modeling [18, 17, 83], wherein an online algorithm develops a model of the robot, can account for damage or changes in hardware operation over time. Clearly, such methods involve both control and morphology. Let us consider each in turn.

2.1 Controller Evolution

The majority of ER studies have focused on evolving controllers for robots whose morphologies are fixed. Designing robust control strategies is a difficult task, especially considering the myriad of possible constraints imposed by environmental and physical factors. Early work in cognition and robotics focused on simple robots, known as Braitenberg vehicles [23], driven by neural controllers that respond to external stimuli. These robots respond to environmental cues solely from the interaction between sensors and motors. Applying EC to the development of Braitenberg vehicles has been used to explore behaviors like foraging [14]. Further research has investigated evolving not only the control strategy of robots, but also the placement of sensors and effectors, demonstrating the effectiveness of co-evolutionary strategies in which both aspects of control and morphology are optimized [88]. While informative, however, these Braitenberg vehicles typically exhibit only very simple behaviors, mapping sensors directly to effectors.

Bio-inspired approaches such as artificial neural networks (ANNs) provide a possible means to address this problem. Modeled loosely on a biological brain, the basic ANN structure consists of neurons connected by weighted synapses, as shown in Figure 2.1. Each neuron contains an activation function, which may be any mathematical function, but is often a sigmoid. The inclusion of hidden neurons between input and output neurons adds complexity to a network, potentially enabling more robust behaviors. Synapses define the structure of the network, dictating how information flows from input, through hidden neurons and ultimately to outputs. Values transmitted to a synapse are multiplied by the weight of the connection. Basic memory can be implemented through recurrent connections in the network, which serve to "store" information between activations of the ANN. ANNs are particularly well-suited to evolution with GAs [132, 133], as neurons and edges can be mutated throughout the course of evolution. In addition to perturbations of weights and activation functions, the structure of a network itself may evolve. Neurons and synapses can be created, removed and rearranged, producing more complex neural structures. ANNs have proven effective in many domains, demonstrating control strategies for wheeled robot navigation [103], legged locomotion [128, 14], object manipulation [30] and feature detection in images [58].



Figure 2.1: Artificial neural networks consist of neurons connected by weighted synapses. Each neuron contains an activation function specifying how the inputs obtained from the weighted synapses map to an output. The activation function is typically a Sigmoid, see above, but can be other mathematical functions. Output of each neuron is propagated through the network to ultimately define commands to be sent to motors.

Central pattern generators (CPGs) are a special case of ANN and consist of nodes that generate an activation pattern based on an internal timer function influenced by external inputs [86, 87, 64]. These networks are well suited to generating the periodic oscillating signals necessary for cyclic gaits. Inputs can include both sensory inputs and connections with other nodes in a network. A number of CPGs have been evolved, especially in the development of gaits [11, 47, 78, 111, 130]. For example, Ijspeert [65] constructed a CPG network for a robot emulating the neural structure and morphology of a salamander. Evolved gaits were amphibious, expressing different behaviors depending on whether the robot was in an aquatic or terrestrial environment; the CPG adjusted automatically based on sensory input [63]. CPGs are also robust to changing environmental conditions such as friction, as demonstrated by Inoue, Sumi and Ma [66] in a snake robot moving on crawling substrates.

In the context of ANN evolution, the Neural Evolution of Augmenting Topologies (NEAT) algorithm [122] has emerged as an effective means to produce robust robotic controllers. NEAT notably addresses the crossover issue that faces ANN development, allowing the evolution of not only connection weights and neurons, but also the topology of the network. By employing a strategy of historical markers, speciation and the protection of innovation, NEAT enables increasingly complex networks to evolve. This process of "complexification" is the hallmark of NEAT, which begins with minimally structured networks, adding neurons and connections as dictated by the increases in fitness over generations.

The systems described above are examples of *direct* encodings. In contrast, *generative* encoding systems attempt to replicate the functionality of DNA by generating growth rules, rather than an explicitly encoded genotype-phenotype mapping. For example, HyperNEAT [121] extends the NEAT algorithm, introducing a level of indirection through the use of central pattern producing networks (CPPN) [120] to create ANNs. Using this indirect encoding, HyperNEAT attempts to emulate patterns seen in nature such as linearity, symmetry and repetition to produce networks that are modular and symmetric. The algorithm has been employed successfully in a variety of tasks, including the development of quadrupedal gaits [35], neural network fitting [36], and vision [121]. In other work, Mouret and Tonelli [99] demonstrated the effectiveness of generative encodings in generating regularity in ANNs, which led to an increased ability of the network to learn in a plastic ANN. Many generative encoding schemes address both aspects of morphology and control together in a single representation. We address these further in the next section.

Despite these advances, incorporating multiple behaviors in evolved robot controllers remains difficult, apparently requiring the specification of fitness functions that account for multiple tasks. In the *subsumption architecture*, as originally proposed by Brooks [26], a unified system of stratified controllers balance high-level reasoning and low-level control demands, with a focus on task-achieving behaviors rather than functional unit decompositions. Multiple controllers function in parallel, in contrast to the serial approach applied in many control strategies. Recently, Lessin et al. [77] demonstrated the power of subsumption architecture to evolve robots capable of locomotion, object discrimination and fight-or-flight response. The expression of multiple behaviors in one control strategy begins to address the gap between robot controllers and the robustness of natural organisms. Through its (engineered) control architecture, the subsumption architecture provides a strategy of controller development rooted in concepts such as modularity and encapsulation. A complementary approach is to consider the evolution of control and morphology together.

2.2 Evolution of Morphology and Control

Although it is possible to evolve morphology alone, as in the case of the evolved unactuated cranes demonstrated by Funes and Pollack [50], in robotics morphology and control are typically evolved together. The grounding of a controller in a morphology, termed *embodiment* [28, 108, 32], describes the coupled dynamics between these two facets. Even the early work of Sims was co-evolutionary, with both the control strategies and morphology of individuals undergoing evolution [115]. Co-evolution of control and morphology is especially effective in the domains of gait development [89, 128]. The boundary between control and morphology is not always clear, however. For example, Mautner and Belew [88] investigated the placement of sensors on Braitenberg vehicles through an evolutionary process, concluding that the evolved sensor placements outperformed those with fixed sensors. Although sensors contribute directly to the control strategy, they are also a physical element in a robot with defined position on the body.

In addition to investigating the placement of sensors, Mautner and Belew [88] evolved their robots with a rule-based growth model. These generative encodings create control/morphology pairings in a manner similar to biological DNA, generating complex structures from a relatively simple encoding. L-Systems, introduced by Lindenmayer [81], are an interpreted symbolic grammar that specify a set of growth rules. First used to describe the growth of plant systems, Hornby and Pollack [60] demonstrated the effectiveness of L-Systems in a quadruped, where aspects of control and morphology were co-evolved with this encoding strategy. Artificial ontogeny [21] is another approach to evolving morphology and control in digital systems, based on observations of genetic operations observed in DNA. Mazzapioda et al. [89] demonstrated the application of artificial ontogenetic processes to evolve the morphology and control of digital creatures capable of locomotion in multiple simulated environments. Additional examples of generative encodings include simulated multi-cellular organisms [112] which demonstrate effective swimming behaviors in a 2D environment. In addition to their use in understanding the growth process of morphology in biologically motivated studies, generative encodings provide insight into the design of robots. Generative encodings often produce modular structures [61, 82] which can aid in transferring individuals evolved in simulation into robots. Modularity, symmetry and repeated structures all emerge through these encodings in successful individuals.

Morphology also plays an important role in effective controller design; changes to a morphology during evolution lead to the emergence of more robust behaviors. Bongard has shown that ontogenetic changes to an individual's morphology increase behavioral robustness across different environments [19]. Furthermore, when evolved in different environments, individuals evolve effective behaviors more rapidly [20]. This *scaffolding* approach, introducing different morphologies and/or environments throughout the evolutionary process, is an effective means to evolve behaviors that tolerate different environments.

Clearly, the physical characteristics of the bones, tendons, and muscles that comprise a biological organism affect the resultant behaviors of an individual. In fact, there is evidence to suggest that natural organisms offload some aspects of control to the body [127] as a sort of "morphological computation" [106, 108]. While not always under active control, the passive properties of morphological traits are an essential component of a robotic system. Rieffel et al. [110] evolved tensegrity robots that harness emergent properties between different components to assist controllers in locomotion. Here, the spring and damper characteristics of the constituent parts lead to expressed locomotion, even with a simple vibrating motor. Individual components, such as the flexible joints described by Seo and Sitti [113], can also provide essential functionality for a robotic system. In some cases, it is even possible to offload all or some control to a morphology, as in the case of passive walkers [90, 37], where the mechanical design of a robot allows it to walk with little or no control input. By harnessing morphology as a means to assist in control, higher-level controllers may then be free to pursue other tasks.

2.3 Open Questions

Despite increasing interest of the research community on these areas of study, there remain many open questions regarding the integration of robot control and morphology, particularly with respect to evolutionary algorithms. How can controllers evolve to take advantage of morphological characteristics? What are the most effective methods to evolve generalized behaviors while avoiding over-training for specific conditions? Passive structures and the properties of the constituent materials can enhance functionality, but only if they are integrated with the control strategy. How can these passive properties be modeled efficiently, so they are amenable to evolutionary algorithms? How should the evolutionary process be structured in co-evolving morphology and control? Should morphology and control be evolved concurrently, or should evolution of control and morphology alternate? What relationships emerge between whole-body morphological characteristics, such as limb dimensions, and control parameters such as the period and phase offsets of joint movements? Do limits need to be placed on morphological parameters to prevent systems which are impractical in the physical world? How do these limits affect performance and evolvability? Does offloading some aspects of control to the joint level improve the overall capabilities of the system? Furthermore, how do power and energy concerns affect the performance, and evolution, of robotic systems? The following chapters describe our work to date on addressing these issues, and our plans for future studies.

Chapter 3 Evolution of Control: Station Keeping

We begin our investigations by exploring the evolution of controllers for aquatic robots. Mobile aquatic sensors are likely to play a critical role in ecosystem management, tracking of hazardous wastes, and surveillance of harbors and coastal waterways. While some of these applications can employ propeller-based robots, in others fin-based locomotion potentially offers better maneuverability, less noise, and less disruption of the environment. In such devices, often termed *robotic fish* [124], fin movements are typically achieved with either small motors [123, 34, 131, 45] or deformation of electroactive polymers [31]. For instance, Chen et al. [31] demonstrated that a carangiform (that is, propulsion primarily generated by a caudal fin) robotic fish can successfully navigate the surface of water with a single actuator. Other studies of biomimetic aquatic robots have yielded insight into the dynamics of fish locomotion [3, 62, 76] and collective behaviors [46, 85].

Despite these advances, however, aquatic robots still do not approach their biological counterparts in terms of maneuverability or autonomy; the materials, sensors and actuators that make up a robotic fish are simply not as effective as organic tissue. Unlike meter-sized autonomous underwater vehicles (AUVs) that can house sophisticated hardware for sensing, actuation and data processing, robotic fish are usually required to be small (8-30 cm in length) and relatively inexpensive. To that end, they are typically equipped with low-precision sensors for navigation (accelerometers, gyroscopes, GPS, and digital compasses) and relatively small batteries, making energy management a critical issue. Yet, these robots

are required to negotiate aquatic environments characterized by uncertainties resulting from waves, currents and turbulence, as well as plant growth and other obstacles. While mathematical models of the hydrodynamic interactions help to evaluate structures and mechanisms prior to development, the design process remains a challenge due to the large number of parameters involved in producing effective locomotion under different conditions. Each combination of materials and electromechanical constraints produces different performance and requires detailed knowledge of material properties. These factors directly affect not only low-level control, but also higher-level decisions on how to maneuver the robot to carry out complex tasks. Evolutionary computation methods are well suited to such high-dimensional problems. By broadly sampling the solution space, evolutionary algorithms are able to test for, and blend, the beneficial aspects of individual solutions in order to produce effective results.

In this chapter, we describe a study on the evolution of controllers for *station keeping*, whereby an aquatic robot is required to maintain a specified position despite surrounding water flow. A behavior exhibited by many species of fish, station keeping is important to robotic tasks such as identification of stationary objects and collection of water quality data at a specified location. Here, we use the NEAT algorithm [122] to evolve controllers for an aquatic robotic platform, shown in Figure 3.1, that includes two actuated lateral "flippers," an actuated caudal fin, and an inertial measurement unit (IMU). This morphology does not include passive, flexible components; that topic is addressed in later chapters. In this study, we address station keeping in the presence of external forces produced by laminar water flow. As opposed to a turbulent flow, which is characterized by eddies, a *laminar* flow occurs when the water flows at a constant rate in parallel layers, with no mixing between layers. To achieve station, a robotic controller must coordinate the actuation of all motors in an effort to locomote against external forces by interpreting inertial (i.e., linear and angular acceleration) data. We first evolve controllers for station keeping in single-flow environments, then consider generalized control strategies that accommodate changing flows.



Figure 3.1: Modeling and fabrication of an aquatic robot. From left to right: (a) evolutionary experiment based on a simulation model; (b) corresponding SolidWorks model for prototype; (c) integration of electronic components and battery into the prototype; (d) assembled, painted and waterproofed prototype in the flow tank. The physical prototype's main body is 13cm long and 8cm in diameter with fins that are 8cm long and 2cm wide.

3.1 Simulation Environment

In this section, we describe the models and concepts relevant to evolving neural controllers for the aquatic robot.

3.1.1 Robot Model

The robotic model developed for this study emulates the form and function of a physical device, seen in Figure 3.1(d). The model consists of a static body, a caudal fin and two continuous rotation lateral flippers. This design has some resemblance to a biological fish, however, the functionality of the flippers is significantly different in both range of motion and possible behaviors. Specifically, the flippers are not limited to a defined range, instead exhibiting a 360° range of motion in both directions, while the caudal fin is limited to a \pm 30° symmetric range of motion. The fins used in this study are assumed to be rigid; other studies address flexible components [97, 34]. With these three actuators, a wide range of three-dimensional maneuvers is possible, providing evolution with a broad slate to discover unique gait patterns for aquatic environments.

An important requirement in this study is that the simulated model mimic the physical prototype in form and sensing capability. Despite the availability of servo encoders, which provide feedback on the state of the motors, the capabilities of the robot are kept minimal in order to examine how the evolved solutions perform with limited sensory information. Specifically, this disconnect between mechanical positioning and the control signal sent to a motor creates a situation in which the neural controller is dependent only upon its perception of the surrounding environment, rather than feedback from a motor position. Hence, the virtual robot does not have exact position or speed information for its servo motors. Instead, inertial data is provided by a simulated inertial measurement unit (IMU), matching the hardware of the physical prototype. Thus, the robot's controller must interpret sensory data provided by the simulated IMU to determine how actuators change the body's state. In this study, we do not consider error in the IMU data.

3.1.2 Simulated Environment

The simulation environment is based on the Open Dynamics Engine (ODE) [117], a rigid body physics simulation engine. While ODE provides a method for resolving forces and torques into motion, it does not include fluid dynamics. Therefore, we implemented a model based on hydrodynamic drag [131]. This model evaluates hydrodynamic forces by examining each component (i.e., simulated rigid body) of the robot independently. Algorithm 1 outlines how hydrodynamic drag is calculated for all robot components. A drag force is applied to each face of a component. Drag acts in opposition to linear velocity and is scaled by the area of a given face and a constant hydrodynamic drag coefficient. In this method, only the faces that oppose the direction of travel experience drag. Propulsion is the result of a net force, the summation of each individual face's force.

This simulation environment provides efficient computation of robot-fluid interactions while minimizing CPU time required to evaluate solutions. A more accurate fluid dynamics simulator might provide higher fidelity, but the overhead incurred by such a system would significantly limit the number of individuals and generations that we could evaluate in the same amount of time. Furthermore, in this initial phase of our investigation, we are most interested in the *general* behaviors that evolve.

Algorithm I Hydrodynamic model. Adapted from 1	116	ή.
--	-----	----

```
for all body do

lin\_vel \leftarrow getLinearVelocity(body)

body\_rot \leftarrow getBodyRotation(body)

for all face do

area \leftarrow face\_area

norm \leftarrow (face\_normal * body\_rot)

force \leftarrow norm * lin\_vel * area * drag\_coeff

if force > 0 then

addForce(force)

end if

end for

end for
```

3.1.3 Neural Controller

Neural controllers are produced with NEAT, which evolves recurrent artificial neural networks (ANNs) with a modified genetic algorithm [122]. In NEAT, only the number of inputs and outputs must be specified, while the hidden layers and connectivity are modified through the evolutionary process. Relevant NEAT parameters in this study are: a dropoff age of 200, survival threshold of 0.2, mutation only probability of 0.25 and a mate only probability of 0.2. ANNs have nine inputs, three outputs, and are activated every 5ms of simulation time. Two outputs control speed of the flipper servos, and the third governs oscillation of the caudal fin servo. Table 3.1 the inputs to the ANN. Three of the inputs are the previous ANN outputs, another three are the robot's current three-dimensional position (obtained through the simulated IMU data), and the final three inputs describe the difference between the current position and the target position. These inputs were chosen because they represent the values available to the physical robot. It is important to note that servo motor inputs for the ANN are based upon the previous outputs, and do not directly reflect the mechanical position of the simulated motors.

Input	Description
Left Pectoral Rate	Servo Velocity
Right Pectoral Rate	Servo Velocity
Caudal Fin Position	Servo Angle
X Position	Limit of ± 1
Y Position	Limit of ± 1
Z Position	Limit of ± 1
X Distance to Station	Limit of ± 1
Y Distance to Station	Limit of ± 1
Z Distance to Station	Limit of ± 1

Table 3.1: Description of Inputs to the ANN

3.2 Experiments

We address the evolution of station keeping in the presence of a constant surrounding flow, as depicted in Figure 3.2a. Four separate trials, illustrated in Figure 3.2b, were conducted to evolve station keeping for flows originating from different directions relative to the robot's initial orientation. Trial 1 simulates a flow from the front; Trial 2 from the rear; Trial 3 from the left; and Trial 4 from the right-front. Each trial consists of 25 replicate runs evolved for 2000 generations with a population of 100 individuals. The four trials are independent, with solutions being evolved to handle a specific flow situation. A time step of 5ms was used in the simulation environment, giving each individual solution a total of 24,000 neural controller updates during a run.

A major component of this initial work involved deriving the fitness function to capture the elements of station keeping and to account for the dynamics of the aquatic environment. Accordingly, each simulation is divided into a transient phase and an evaluation phase. In preliminary results without the transient period, the evolved individuals were incapable of station keeping. Instead, solutions would attempt to hold station immediately, but ultimately were forced out of the station keeping area by the flow. The transient phase allows the robot to reorient itself with movements that might otherwise cause a decrease in fitness (i.e., temporarily moving away from the station point); such behavior is similar to rheotropism in


Figure 3.2: (a) A demonstration of the station keeping task. The sphere and crossed white lines indicate the desired station keeping point for the robot. Maximum fitness is accrued when the robot brings its center of mass, denoted by the green lines, to intersect with the station point. (b) Direction of flow in each of the four trials. Trial 1 involves a flow coming from straight ahead. In Trial 2, the flow comes directly from behind. For Trial 3, the flow is from the side of the body. Finally, in Trial 4 the flow comes from 45 degrees straight ahead.

biological fish [5]. During the evaluation phase, fitness is accumulated periodically at 250ms intervals. Fitness is awarded in a spherical zone surrounding the station point. Individuals holding station near the point accrue high fitness, with the reward dropping off with distance. No fitness is awarded to an individual when it is outside the sphere, but neither is this behavior explicitly penalized. Specifically, fitness is calculated as follows:

$$distance = \left(\sqrt{(x_T - x_t)^2 + (y_T - y_t)^2 + (z_T - z_t)^2}\right)^3$$
(3.1)

$$fitness = \begin{cases} 10 - (distance) & \text{if } > 0 \\ 0 & \text{otherwise,} \end{cases}$$
(3.2)

where (x_T, y_T, z_T) represents the desired station position and (x_t, y_t, z_t) is the current position.

3.3 Results

In this section, we present results of the evolutionary experiments. Videos of selected behaviors are available in the supplementary materials.

3.3.1 Evolved Behaviors

Depending on the direction of the flow, evolved behaviors varied from simple forward swimming to complex acrobatic maneuvers reorienting the robot towards a flow. In Trial 1, a simulated laminar flow from *upstream* (i.e., from the front to the back of the robot) is applied. This configuration serves as a benchmark to measure performance of the subsequent trials, as the robot does not have to reorient itself prior to maintaining station. Evolved gaits are reminiscent of natural fish locomotion, with both the caudal fin and flippers working in a coordinated oscillating motion to swim against the flow. Trial 2 simulates a flow pushing on the rear of the robot. Initially, we had expected to see solutions employ the flippers to maintain station while keeping the orientation relatively stable. However, evolved solutions instead exhibit a more effective flipping maneuver in order to bring the caudal fin into an effective position to counteract the flow. As depicted in Figure 3.3, the robot flips itself over, and then executes a forward motion similar to that seen in the first trial. Reorienting the body into an effective position for *forward* propulsion demonstrates the controller's ability to identify and counteract the flore generated by a laminar flow.

In Trial 3, a simulated flow exerts force against the side of the robot. Of the four trials conducted, this turned out to be the most challenging, apparently due to the difficulty of turning 90 degrees to the left in the allotted time. For this trial, the expected behavior was to turn 90 degrees and face the flow without any need to rotate along another axis. However, a more complex maneuver evolved, apparently because a 90 degree turn is time-intensive. The evolved behavior combines the flipping motion seen in Trial 2 with a roll to bring the body into an effective position for swimming against the flow. Trial 3 individuals have difficulty achieving station within the time allowed, however, as the initial reorientation requires a

significant amount of the evaluation period.



Figure 3.3: Behavior of an evolved solution in Trial 2. The first 60s, which is the transient phase, is used to reorient against a laminar flow pushing on the robot from the rear (left to right in the figure). The robot executes a 180 degree flip bringing the caudal fin into a position where it can provide the greatest thrust. Here, the flippers roll the robot over as well as make minor adjustments once the robot is in an effective position.

For Trial 4, a simulated flow is applied at a 45 degree angle to the robot's initial rightfront. As depicted in Figure 3.4, evolved individuals demonstrate the ability of the controller to respond to the direction of flow and attain station keeping during the course of an individual evaluation. Images in Figure 3.4 are taken at 10 second intervals over the first 70 seconds of simulation time. Initially, the robot is displaced from its station. The robot begins to react at approximately 10s when it starts to orient itself to the flow by using its flippers to rotate the body while the fin provides forward propulsion. Fitness evaluation begins at 60s. By this time the robot has achieved, and can maintain, station by working to correct its position relative to the given station point.

3.3.2 Fitness Measurements

Fitness results from the trials are shown in Figure 3.5a and 3.5b. These results provide insight into the relative difficulty that each flow presents to the evolutionary process. Specifically, in Trial 1, where the robot directly faces the flow, solutions achieve near perfect results, where a fitness of 1 correlates to solutions that maintain station for the entire evaluation phase. Apparently, the lack of need to reorient the body helps to produce such high fitness



Figure 3.4: An evolved solution in Trial 4. In this trial, an individual faces a laminar flow at a 45° angle to the robot's front. The robot spends the first 50s reorienting itself against the flow. After 50 seconds, the robot has achieved a stable station and begins to accumulate high levels of fitness by using the flippers and fin in a coordinated effort to maintain its center over the station point.

values. Figure 3.6 shows the final distribution of the best evolved individuals for each of the replicate runs. This plot also illustrates the relative difficulty associated with the different flows. As can be seen in the results from Trials 2, 3 and 4, only a few solutions are able to gain high fitness.

3.3.3 Behavior Comparison

Figure 3.7 plots trajectory information for the final solution and an early generation solution in Trial 4 (Figure 3.4 presents the gait that the final evolved solution takes to achieve station). For the final evolved solution, the robot initially moves outside of the fitness zone. However, this maneuver occurs during the transient phase, allowing the evolved solution to move without losing fitness. The earlier controller fails to identify the flow and can be seen drifting away over time. Even though the robot does manage to swim with some coordination, it lacks the ability to identify the flow direction and coordinate its swimming to act against the force. Consequently, this individual ultimately is pushed out of the fitness reward area.



Figure 3.5: (a) Fitness of the best evolved results from the trials. Trial 1 is able to achieve a near perfect fitness score as it does not have to reorient itself prior to holding station. The other trials have some success, although their fitness scores are lower than Trial 1. This is likely due to the movements required to reorient to flows. (b) Average fitness of the population of evolved results from the trials. Trials 2, 3, and 4 have low average fitnesses due to the difficulty of station keeping in these environments. Many individuals are able to swim, but leave the station area, accruing no fitness during the evaluation period.

Many evolved solutions exhibit swimming behaviors but are not able to coordinate those movements with the task of holding station against the flow. For example, one individual can effectively swim directly forward, however, the direction of flow in the trial (from the side) causes it to gradually drift out of the fitness reward zone, resulting in the individual accruing less than 1% of the available fitness.

3.3.4 Discussion

In this initial investigation, we have shown that neuroevolution is capable of generating control strategies to address station keeping. Solutions exhibit unexpected locomotion strategies that involve both simple swimming gaits along with complex maneuvers to reorient the robot in different laminar flows. However, the evolved results do not extend beyond environments encountered during evolution. Although perhaps unsurprising due to the single



Figure 3.6: Box plot of the best fitness values for each replicate run in the four trials. Box indicates the upper and lower quartiles, median is represented as the center line in the boxes. Ends of the whiskers represent the maximum and minimum values, excluding outliers.



Figure 3.7: Two-dimensional, top-down trajectory plot for an evolved solution and an early generation candidate solution from Trial 4. The grey circle represents the area of fitness reward. The evolved solution is initially displaced before identifying the direction of flow, reacting to it, and then achieving station.

environment per trial employed during evolution, it remains a challenge to achieve more generalized control strategies. Moreover, characterizing the difficulty of specific environments is not straightforward, as a 90-degree flow proved more difficult than one coming from the rear of the robot. The next section presents results of our initial approach to addressing this challenge.

3.4 Toward Generalized Station Keeping

Evolved controllers in the above experiments cannot handle flows other than the single one encountered during evolution. However, it is desirable not only to hold station in one environment, but to address the general problem of holding station against a variety of flows. Over-training, that is optimizing a solution for a small set of conditions, is a risk for evolutionary algorithms. We have conducted a preliminary set of experiments to evolve generalized station keeping, whereby the robot can adapt dynamically to changes in flow. We initially attempted to evolve generalized controllers by randomly selecting a flow, per generation, from any combination of x and y components on the horizontal plane. However, controllers were unable to evolve station keeping, or even coordinated swimming behaviors. Presumably, the introduction of multiple flows in such a random manner prevented evolution from gaining a foothold on the essential elements of the task. Instead, we found that a more structured strategy of introducing environmental variation is required, where the direction of flow changes gradually. An example is shown in Figure 3.8. Results are promising, but incomplete and will be presented in the dissertation.

3.5 Summary

In our initial investigation of station keeping, we focused on the process required to evolve individuals capable of holding station against a single flow. One of the challenges facing any evolutionary robotics process is how to assign fitness. Station keeping requires



Top Down View of Flow

Figure 3.8: Depiction of the sweeping side-to-side flow facing a robot over the course of an evaluation. At time t=0s, the flow comes from the front of the robot. As the simulation progresses, a second component is added, altering the direction of flow up to a $\pm 63.4^{\circ}$. The magnitude of the force also varies, as the x component of the flow remains constant throughout the simulation.

rewarding solutions for maintaining station at a desired point, while not penalizing solutions that move outside of the fitness area to reach more effective orientations. In preliminary tests, fitness was allowed to accumulate from the beginning of the run, creating an unnecessary pressure to perform well from the start. Such a fitness metric makes it difficult for an evolutionary algorithm to find strategies that sacrifice initial fitness for an overall better strategy. Given the robot's morphology in this study, the caudal fin produces the greatest propulsion. Therefore, an effective strategy is simple swimming, combined with more complex acrobatic maneuvers in challenging flows to reorient the robot to different laminar flows. The results show that neuroevolution is capable of generating control strategies to address station keeping against a variety of different flow situations.

While these individuals are capable of station keeping in evolved environments, they fail to generalize to new conditions. Autonomous robotic systems need to be capable of operating in novel environments, which are not seen during off-line evolution. Techniques previously studied include scaffolding [20, 135], that is, introducing different elements of a task over time. However, such approaches require expert knowledge to assess the difficulty of a task in a specific environment, which can be difficult for aquatic environments. In our ongoing studies, we are exploring techniques to evolve individuals capable of station keeping in multiple environments.

The robotic platform discussed in this chapter is based on a physical prototype with 3Dprinted rigid materials and traditional robotic actuators. As such, we modeled the simulated robot after the physical device for both dimensions and mechanical capabilities. Evolution is thus able to optimize only the control strategy for this robot, neglecting a second important aspect, an individual's morphology. Natural organisms have tightly intertwined control strategies and bodies, working together to produce robust behaviors. In the following chapters, we investigate the role of morphology, and its integration with control strategies through co-evolution.

Chapter 4

Exploiting Passive Joints in an Amphibious Robot

4.1 Introduction

Natural organisms exhibit an astounding array of functionality via complex interactions among muscles, bones and nerves. Movement is typically produced through muscles, guided by the nervous system, with passive parts of morphology (such as fins, feathers or webbed feet) enhancing these actions. Passive components can also increase performance in robots, by emulating capabilities found in biological organisms. For example, a passively flexible caudal fin has been shown to produce more thrust than a rigid fin [34]. Moreover, integrating passive components into design reduces the number of actuators, and correspondingly the mechanical complexity needed by a system, which is particularly important in small robots.

Despite these advantages, the introduction of passive components into a robot poses numerous challenges in the development of control strategies. Because the joints are not directly actuated, any control strategy must account for the characteristics of passive structures in determining the overall response of the system. Our approach to this problem combines evolutionary computation with efficient methods for modeling materials and their interaction with the environment. Whereas evolutionary computation guides the search process, computationally efficient models determine how constituent materials behave when acted upon by forces, enabling accurate evaluation of the robot in simulation. This approach, coupled with 3D printing for rapid prototyping, provides an opportunity to bridge the gap between artificial and natural systems in terms of agility and maneuverability.

In this chapter, we present an evolutionary approach to discovering effective combinations of morphology and control for an amphibious robot with passive arm joints. Candidate solutions are evaluated using a rigid-body physics simulation environment, with successive generations created through a process of selection, mutation and crossover. Each evolved solution comprises a body plan and two controllers, one for crawling on a flat surface and another for swimming in water. This two-controller approach exploits the ability of the robot to change control patterns between different environments, whereas the morphology of a robot is fixed after fabrication.

Results of this study demonstrate that evolved solutions harness the properties of passive joints to move effectively in both terrestrial and aquatic environments. The passive joints become an integral part of locomotion. Moreover, the evolutionary process finds solutions whose control and morphology are highly intertwined, demonstrating the importance of exploring both facets together. We fabricated the best evolved solution using a multi-material 3D printer, and have conducted preliminary experiments with a prototype on an evolutionary robotics test bed.

4.2 Methods

This study focuses on developing a robot whose passive joints not only reduce motor requirements, but also enhance performance. Many aspects of the robot (dimensions, materials, controller parameters) are evolved, while others (for example, mechanical components), are designed. The following sections describe specifics of the robot, simulation model, evolutionary approach used, and the fabrication of the physical model.

4.2.1 Robot Overview

The robot used in this study features a main body and two arms on each side near the front; a simulation model is shown in Figure 4.1. Battery, controller, and motors are assumed to be contained within the main body, with the arms connected directly to the motors.



Figure 4.1: Simulation model of the robot used in this study. The arms move in a sweeping motion pivoting at the top of the fins through a passive hinge joint.

Each arm has a passive hinge joint between the arm and flipper, as illustrated in Figure 4.2, enabling locomotion in both terrestrial and aquatic environments. The arms function similarly to the pectoral fins of fish, and move in a sweeping motion from the front to back. A passive joint between arm and fin locks at vertical on the power stroke, providing a larger surface area for swimming or acting as a leg to lift the body off of the ground for crawling. As the arms move forward on the recovery stroke, the fin collapses backward, reducing drag. Since the passive joint does not require a dedicated motor, it reduces both the power requirements imposed on the robot, as well as the mechanical complexity of the physical design. Instead, the passive joint moves with a combination of gravity or hydrodynamic drag, in concert with the arms, driven by a motor at the base of each arm. Hence, complexity is shifted to the controller, which along with the dimensions and characteristics of the arms and fins, is the focus of optimization during the evolutionary process.

Control of the arms is governed by a sinusoidal input with parameters related to joint



Figure 4.2: CAD model of the passive hinge joint used in the fin structure. The arm extends from the body and is waterproofed by a flexible gasket. The arm attaches to the fin by a passive joint allowing the fin to move according to the environmental forces acting upon it.

limits and the speed of oscillation. Parameters are optimized through the evolutionary process and are executed by a controller that moves both limbs synchronously. An individual robot also has two controllers, one for each environment. In related work, we study the behaviors produced by more complex artificial neural network controllers in an aquatic environment [95] and the use of flexible materials in robots [97, 34].

4.2.2 Treatments and Evaluation

Three evolutionary treatments were conducted. In Treatments 1 and 2, simulated robots evolved in a terrestrial and an aquatic environment, respectively. These single-environment treatments served as benchmarks for comparison with results of Treatment 3, wherein evolved robots were evaluated in both environments. Evaluation of each candidate solution was based on the total forward distance traveled in 10 seconds of simulation time.

The two evaluation environments were built atop the ODE. The terrestrial environment emphasizes parameters relevant to locomotion across a flat surface, such as a sidewalk or tabletop. In the aquatic environment, forces are required to account for the propulsion component of the fins and the hydrodynamic drag on the robot components during forward movement. Hydrodynamic forces were simulated using the methods described in Chapter 3.

4.2.3 Evolutionary Process

Populations comprised 250 individuals and were evolved for a total of 400 generations. Each treatment was conducted using 20 replicate runs in order to produce statistically significant results. The initial population was seeded with individuals containing randomly generated genomes. At each generation, the individual solutions were evaluated in the simulator, and the next generation was formed based on a two-phase selection process. Elitism was used to maintain the best performing solution across generations. Remaining parent solutions were selected using tournament selection with a tournament size of 3. New individuals were generated using single-point crossover with a probability of 25% and through mutation with a 30% chance to alter a genome.

Each genome consists of 11 real-valued parameters, listed in Table 4.1. Practical constraints were also placed on certain genes such that the simulated robot is more readily transferable into physical designs. For example, fin width is allowed to range between 1 and 3 centimeters, while also being constrained to a maximum value of the overall arm width. Other intergene constraints impose restrictions on the total width of the robot body and arms, as well as to keep joint oscillation from crossing the high and low limits of a servo motor.

Parameter	Min. Value	Max Value
Lower Fin Height	1cm	5cm
Lower Fin Width	1cm	$3 \mathrm{cm}$
Body Width	$4.5 \mathrm{cm}$	$7.5\mathrm{cm}$
Arm Width	1cm	$5 \mathrm{cm}$
Fin Friction	0.7	1.0
Osc. Freq. Land	.25 Hz	1 Hz
High Limit Land	-70 °	70 °
Low Limit Land	-70 °	70 °
Osc. Freq. Water	.25 Hz	1Hz
High Limit Water	-70 °	70 °
Low Limit Water	-70 °	70°

Table 4.1: Individual Gene Limits

Under this encoding scheme, the morphological parameters, except for fin friction, which is primarily used in the terrestrial environment, are subject to competing environmental pressures. Over the course of evolution, parameters relating to the robot morphology reach values that allow the robot to move in both environments. The control parameters, oscillation frequency and joint limits, are specific to each environment. High and low joint limits were incorporated into the controller to evaluate different ranges of motion. In this way, a controller can define the range of movement, within the physical constraints of the motor, to use in locomotion. Range of motion and oscillation frequency are the driving factors governing the behavior of the passive joints, producing a robot whose morphology is adapted for movement in both environments, but with controllers specifically adapted to each environment and its morphology.

In Treatments 1 and 2, fitness is defined simply as the forward movement in 10 seconds of simulation time. For Treatment 3, fitness is a function of performance in the two environments, as defined by Equation 4.1:

$$Fitness = [4 - (2 - aqua_dist) * (2 - terr_dist)]^2,$$

$$(4.1)$$

where *aqua_dist* represents the normalized distance traveled in the aquatic environment and *terr_dist* represents the normalized distance traveled in the terrestrial environment. This fitness function rewards solutions that perform well in both environments.

4.2.4 Prototype Fabrication

Taking the best evolved solution from Treatment 3 as a model, a prototype robot was fabricated using an Objet Connex 350 multi-material 3D printer. The prototype is shown in Figure 4.3. The controller was implemented using an Arduino microcontroller with servo motors actuating the arms. This physical model was used to validate the results of evolution and identify any differences in movement characteristics and performance when transferring from simulation to reality.



Figure 4.3: 3D printed robot of the best performing solution in this study. The passive hinge joint can be seen between the arm and fin. The fins collapse backwards as the arms move forward allowing the robot to move in both terrestrial and aquatic environments.

4.3 Experiments and Results

The following sections describe the results of the study. We first present details of the three individual treatments, followed by a discussion of the relationships between control and morphology that emerged during evolution.

4.3.1 Treatment 1 - Terrestrial Environment Only

A primary challenge faced by any robot is how to move its body efficiently. In the terrestrial environment, strategies generally minimized the contact area between the body and ground. Specifically, Treatment 1 arrived at solutions where fins that were significantly taller than the main body in 19 of the 20 replicate runs. An evolved individual, shown in Figure 4.4(a), lifted the body off the surface and rocked forwards as the arms moved towards the rear of the body. This gait allowed the individual to move forward at a relatively rapid pace. Furthermore, the main body evolved to its narrowest allowable value, while the fin friction evolved to be near its maximum value. Evolved controllers tended to move the arms near the fastest allowable frequency while also favoring the largest range of motion. A large range of motion allows the robot to keep its body off the ground for a longer period during the rocking gait, increasing distance traveled during a simulation.



Figure 4.4: Three different morphologies evolved in different treatments. Subfigure (a) shows the dominant morphology that emerged in Treatment 1; note the tall pectoral fins. The dominant morphology for Treatment 2 is presented in (b) and is characterized by shorter pectoral fins. The morphology that emerged in Treatment 3 can be seen in subfigure (c). This adaptive morphology exhibits a compromise in the pectoral fin height between the terrestrial and aquatic morphologies, enabling the robot to perform well in both environments.

4.3.2 Treatment 2 - Aquatic Environment Only

In contrast to robots in the terrestrial environment, individuals evolved in an aquatic environment tended to reduce the surface area exposed to the direction of travel. As a result, fins of robots in the aquatic environment evolved to be on average 29% the height of those in the terrestrial environment, although fin widths were similar. An example is shown in Figure 4.4(b). Shorter fins produce less drag during the recovery stroke and also reach a fully vertical position faster during the power stroke . Body widths of individuals in this treatment evolved toward the smallest allowed value, even more so than Treatment 1, in order to minimize drag. Unlike Treatment 1, the oscillation frequency of the evolved controllers exhibited more variation, apparently reflecting the different fin height values observed in this treatment. The dynamics of this relationship are discussed later.

4.3.3 Treatment 3 - Amphibious Environments

Treatment 3 produced morphologies and controllers that performed well in both environments. Specifically, evolved solutions arrived at fins tall enough to propel the robot forward in the terrestrial environment while also being effective swimmers. However, the gaits in the terrestrial environment were characteristically different than those from Treatment 1. Due to shorter fins, which were better for swimming, the terrestrial gait resembled a sliding motion, where the fins were able to lift the body slightly off the ground, dragging the body instead of lifting it. In terms of swimming, the evolved solutions solutions reached approximately 75% of the average distance traveled by solutions in Treatment 2. Figure 4.4(c) shows the dominant morphology that emerged in this treatment. In comparison to the first two treatments, the evolved fin widths were approximately 66% as wide, further indicating a likely compromise between fin height and width.

4.3.4 Fitness Evaluation

The fitness landscape of solutions found in Treatment 3 is illustrated in Figure 4.5, where each of 2,000,000 individual solutions is represented by a small circle. The darker shaded circles indicate areas where multiple solutions perform similarly. This plot includes all individuals found during the evolutionary runs, illustrating the space explored by evolution. The axes represent the fitness for the two respective environments. Solutions that fall along the axes performed well in one environment at the expense of performance in the other environment. Many individual solutions did not perform well in either environment, as indicated by the dark marks in the lower left of the plot. These solutions likely occurred in early generations of evolutionary runs. The area of particular interest in this plot is surrounded by the dashed box in the upper right corner. Here we see solutions that performed relatively well in both environments. In analyzing the results, we found that the best solutions from Treatment 3 performed about 95% as well as those evolved only from Treatment 1 in the terrestrial environment, and about 75% as well as those from Treatment 2 in the aquatic environment. In this region, we also see clusters of solutions, as indicated by the darker shaded circles. The clear definition of circles indicates many solutions that performed similarly. This area also demonstrates that there are multiple possible solutions, each of which arrives at slightly different fitness while being effective in both environments.

Figure 4.6 plots the performance of solutions from Treatment 3 in each environment over evolutionary time. Fitness values in this figure have been normalized to the best results from



Figure 4.5: Fitness landscape of all individuals over the evolutionary trials for solutions evolved in Treatment 3. Fitness values have been normalized using the best values from Treatments 1 and 2 to allow comparison between single environment evolved individuals and amphibious individuals. Darker shaded areas indicate a large number of solutions with similar performance. The box in the top right contains solutions that perform well in both environments.

Treatments 1 and 2 respectively. This plot demonstrates that the fitness progression in the terrestrial environment increases at a relatively stable rate, whereas the aquatic environment experiences more variation over the course of evolution.

4.3.5 Treatment Comparisons

In two validation tests, we evaluated the controllers evolved in one environment, matched with the morphology evolved for the other environment. These tests demonstrate the shortcomings of only evolving morphology for one environment when compared to Treatment 3. Specifically, inserting the evolved morphology for Treatment 1 into an aquatic environment, yielded a fitness that was 38% of the maximum distance traveled by solutions for Treatment 2. Moreover, as is apparent in Figure 4.4, the morphology from Treatment 2 is not even capable of moving in the terrestrial environment, since its fins do not touch the ground.



Figure 4.6: Average performance of the best individual solutions per trial from Treatment 3 over evolutionary time. (a) Aquatic environment fitness and (b) terrestrial environment fitness. Fitness was based on the performance in both environments, as such, the performance does not necessarily increase over each generation. Larger variations in the aquatic environment could potentially indicate that solutions are more susceptible to small changes than those in the terrestrial environment.

These two tests demonstrate the coupled dynamics that form between a virtual robot's brain and body and the importance of the fin dimensions for effective swimming. Altering these dynamics by swapping morphologies proved to be extremely disruptive to their function.

Treatment 3 addresses these shortcomings by evolving solutions in both environments. Each environment was allowed to have its own controller, as switching between controllers is considered to be a trivial process. In both environments, controllers evolved joint limits near their maximum values, indicating that a shortened stroke is not beneficial in either environment. Although the range of motion was similar, controllers for the two environments differed significantly (paired t-test: p < 0.001) in the speed of moving the arms. Average values of 0.6 Hz evolved in the aquatic environment, while values in the terrestrial environment were closer to 1 Hz.

Effects of brain/body evolution become apparent when comparing results of Treatments 2 and 3. Specifically, the best controller from Treatment 2 exhibited an oscillation frequency of 0.89 Hz, while that of the best aquatic controller in Treatment 3 had an oscillation frequency of 0.6 Hz. The final distributions of evolved oscillation frequencies were significantly different (p < 0.001) between the two treatments. However, a relationship between oscillation frequency and fin area can be seen in Figure 4.7. The fin area distributions for the two treatments were significantly different (p < 0.001) with these differences evident in Figures 4.4(b) and 4.4(c). A quadratic regression model demonstrates a relationship between the fin morphology and oscillation frequency in the aquatic environment. In the regression model, fin



Figure 4.7: Distribution of the oscillation frequency of the controllers in the aquatic environment in regard to fin area. The oscillation frequencies are dependent upon the fin area for a morphology with 90% of the variance in oscillation frequency being accounted for by the fin area with a p-value $< 2 \ge 10^{-16}$.

area accounts for 90% of the variation in the oscillation frequency ($p < 2 \ge 10^{-16}$), indicating a strong relationship between the two parameters. Furthermore, this model provides insight into the relationships that form between a robot's controller and morphology. Specifically, as the fin area increased, a slower oscillation frequency was used to reach effective locomotion strategies. This result demonstrates that brain and body evolutionary approaches can discover these relationships during the evolutionary process to optimize an overall design. This feature is especially beneficial when integrating structures such as passive joints, as these dynamics are not always known *a priori*.

4.3.6 Physical Validation

Upon conclusion of the simulation trials, we selected the best performing solution from Treatment 3 and fabricated it using a 3D printer. Figure 4.3 shows the printed model, including the passively hinged fins. Initial experiments with this physical model found movement similar to that of the simulation results. The passive joint moved as expected, with the fin collapsing backwards during the recovery stroke, before returning to a vertical position on the power stroke. Aquatic testing conducted with the physical robot was especially promising, as the robot was able to swim well (Figure 4.8), indicating that we had simulated the passive joint dynamic correctly. However, extensive testing was not possible due to design issues unrelated to the evolutionary process. Specifically, the servo to arm connection proved to be too fragile when operating in a terrestrial environment.

4.4 Summary

Although the morphology of a robot is often fixed following fabrication, the controller need not be static. Considering the capabilities of current microcontrollers, maintaining multiple controllers is a feasible approach that has been demonstrated in amphibious robots [63]. Working under this assumption, we evolved both the morphology and basic controller scheme for our simulated robot. We demonstrated controllers that were uniquely tied to their re-



Figure 4.8: Initial testing of the printed model in an aquatic environment verifies that the passively hinged fin performs similarly to the characteristics seen in the simulation. During the recovery stroke, the fin flexes upwards, pivoting on the passive joint, to reduce drag and enable forward movement of the robot.

spective morphologies. As such, performance of a robot with either an altered morphology or controller would not be expected to perform as well. Simulation results indicate that a unique set of control parameters exists for each environment, given a static morphology. Accordingly, the two-controller approach used in this study is a way to generate effective locomotion across different environments.

Passive characteristics of materials and joints have the potential to enhance robotic designs. In a prior study, we investigated the integration of flexible materials in a robotic design [97]. The robotic platform and physical prototype can be seen in Figure 4.9. Evolved individuals in simulation demonstrated improved locomotion on low-friction surfaces with flexible ankle joints, whereas those with stiff legs did not move as far. The flexible joint provided evolved individuals with the increased traction necessary to move effectively. Similar to natural organisms, flexibility can provide enhanced functionality provided the morphology allows for it.

The study described here addressed the use of a passive hinge joint, which allows each fin to be controlled by one servo motor, leading to a robot with a simple drive-train design and less dependence upon gearboxes and other mechanical parts that are subject to failure. Reducing the number of motors can produce more efficient mobile sensor platforms, as well as potentially smaller robots. Additionally, the incorporation of passive joints in this robot



Figure 4.9: (a) A hand-designed prototype of the robot produced with multiple materials. The feet are composed of both soft and rigid materials. This initial prototype has a completely rigid arm which results in a small contact area between the foot and surface. During movement, the feet often slip. (b) Virtual robot used in this experiment modeled after the physical prototype. A flexible joint is located between the arm and foot of the robot allowing the foot to flatten on the surface, increasing its contact area with the ground based on evolved parameters.

design allows the motors, controller and power supply to be housed inside the main body, reducing the waterproofing requirements of the design and limiting the potential failure areas. Solutions in this study tended to have a strong relationship between their controllers and morphologies, as shown through regression analysis. In the next chapter, we continue our exploration of control/morphology co-evolution, shifting our focus from individual components to relationships among major morphological characteristics and their role in evolved gaits.

Chapter 5

Evolution of Whole Body Morphology: The Role of the Tail in Bipedal Hopping

Having explored the integration of flexible materials and passive joints into the evolutionary process, in this chapter we turn to the evolution of "whole body" morphological characteristics, such as limb dimensions and masses. Our approach is to evolve a particular mode of locomotion, bipedal hopping, where such characteristics, along with passive and flexible joints, play a critical role.

Bipedal hopping is defined as a cyclic bouncing gait in which only the hind limbs contact the ground and swing in phase, or nearly in phase [56, 57]. Thus, contrary to common perception, the gaits used by animals such as rabbits and toads are not truly hopping. Within mammals, hopping has evolved independently in only a few species, but apparently for different reasons. In small animals such as kangaroo rats (Figure 5.1), spring hares, and jerboas, hopping is primarily used as a predator escape mechanism [16]. In larger animals, such as kangaroos and wallabies, hopping offers an energy-efficient means of locomotion over long distances [41]. Despite size differences, however, the overall morphologies of these animals are quite similar. Specifically, bipedal hoppers tend to have long tails and powerful hind legs, which perform the majority of work during locomotion. Yet, the evolutionary origins of this behavior, as well as many related issues, remain obscure. A better understanding of the evolution and mechanics of bipedal hopping not only can inform biology, but has application in robotics, biomechanics, and the development of prosthetics.

In this study we investigate the evolution of bipedal hopping in a virtual animat, focusing specifically on the characteristics and role of the tail. The virtual animat model approximates muscles, joints, mass and torque, enabling us to evolve biologically plausible patterns of movement. Through a series of five evolutionary treatments we investigate the effect of different initial (and evolvable) tail configurations on the evolution of effective hopping gaits. We initially start with a fixed morphology, but restrictions on the morphology are loosened with each subsequent treatment. Results indicate that even using a simple highlevel control strategy, morphological characteristics evolve to be tightly coupled with control dynamics. Moreover, while many of the evolutionary results are consistent with behaviors and morphologies observed in natural organisms, in some cases effective hopping evolved despite key differences from nature, potentially inspiring new design approaches in robotic and biomechanical systems.



Figure 5.1: The kangaroo rat was selected as the base morphology for studying the evolution of bipedal hopping, due to its representative morphology and the availability of information on both the mechanics and dynamics of its behavior.

The contributions of this chapter are as follows. First, the proposed muscle model produces locomotion patterns similar to those of natural organisms and limits the output potential of each individual joint. This model is computationally less expensive than a musculoskeletal dynamics simulator, enabling the large number of evaluations necessary in evolutionary approaches. Second, the results demonstrate that a tail is essential to hopping, but that different configurations can lead to very different gaits, some closely resembling those of biological counterparts (namely kangaroo rats and wallabies), and others different from any known species. Third, while we observed a close coupling between tail movement and the oscillation frequency of leg joints, we discovered multiple combinations that produced effective bipedal hopping behavior. Finally, we were surprised that many evolved tails had relatively low mass, as it is hypothesized that a heavy tail helps maintain a high moment of inertia in animals, producing a more stable gait. This result might be due to our relatively simple model of the morphology (we plan to use more detailed musculoskeletal models in the future), but might also represent a combination of morphology and behavior that has application outside biology.

5.1 Background and Related Work

The role of the tail in locomotion is of considerable interest within biology. In their studies of geckos (which are not bipedal hoppers), Full and colleagues found that the tail is essential to both orientation control and gait stability [72, 80]. Alexander and Vernon [2] studied the musculoskeletal system of kangaroos and described the overall mechanical system and the forces exerted during hopping. They also first hypothesized that the tail was necessary to balance the angular momentum produced by the swinging legs during hopping. However, to our knowledge no one has yet tested this hypothesis, nor explored its significance in other hopping species.

A previous study into the evolution of hopping using a 2D musculoskeletal model found that both quadrupedal and bipedal hopping gaits are very sensitive to changes in morphology [55]. However, such a model does not take into account many aspects of hopping, such as maintaining balance, that are essential in the physical world. Our work explores the evolution of hopping in 3D physics-based simulation environments. While our early studies, described here, rely on rigid-body physics environments, more complex musculoskeletal models have been developed [54] and will be integrated into our investigations as computational capacity permits. In robotics, hopping is an intriguing locomotion strategy for its potential energy efficiency and the ability to rapidly change elevation. The latter is particularly important to radio communication, as signal propagation distance is greatly increased by moving transmitters above ground level [33]. Indeed, research in this area has led to the development of small robots capable of both self stabilization and hopping [134]. Prior studies on hopping have also addressed mechanics of simple, single-joint actuated robots that were able to achieve stable hopping gaits [15], and single-hop robots have been constructed using pneumatic muscle actuators [102]. It has also been shown that combining several hops was more energy efficient than a single, powerful hop, while producing the same jumping height [1]. This efficient hopping motion was discovered after analyzing thousands of results, lending support to harnessing the search capability of evolutionary computation in order to address similar problems. By applying evolutionary approaches to the study of bipedal hopping in 3D animats, we hope to gain insights into this behavior at a level not previously explored.

5.2 Methods

We began our study with an animat based roughly on the morphology of a kangaroo rat. The gaits of this animal have been analyzed extensively with the aid of high-speed, highresolution video cameras [54]; see Figure 5.2. We first evolved gaits for fixed morphologies, then allowed evolution of morphological parameters such as limb dimensions, joint output potential and mass distribution.



Figure 5.2: X-ray video progression of a kangaroo rat hopping across a force plate to quantify hopping dynamics.

5.2.1 Virtual Animat

Figure 5.3a shows the initial animat constructed in ODE, with body part dimensions corresponding to that of the kangaroo rat. The animat also features a controller that actuates all joints. Kinematic data of the kangaroo rat's hopping gait indicated that the individual joints move in a periodic motion similar to a sine wave. Hence, for this initial study where we focus on steady state hopping gaits, a relatively simple sinusoidal controller was implemented; our ongoing investigations use more complex neural-based controllers. In addition, left/right symmetry was enforced. This decision was made primarily due to the difficulty in evolving a controller for a predefined morphology (unlike nature, where they evolve together). In particular, preliminary experiments found that asymmetric controllers had difficulty achieving stable gaits due to large differences in the length of hind and fore limbs. Moreover, observation of kangaroo rats demonstrates left/right symmetry during hopping.



Figure 5.3: (a) Initial simulated animat used in this study, with morphological dimensions and mass based on kangaroo rat. (b) Two-dimensional representation of the animat joints, with range of motion indicated.

5.2.2 Modeling of Muscles and Joints

Animals exhibit fluid movements produced by muscles contracting and relaxing in a coordinated manner. To approximate such dynamics in a rigid-body simulator such as ODE, we modeled muscular connections using hinge joints with appropriate constraints. In particular, we devised a model that limits the energy an individual joint can expend during actuation. Doing so prevents situations in which a joint can move with an infinite amount of force, an impossibility in biological organisms. Moreover, animal joints do not always move throughout their entire range of motion during locomotion (for example, strides may be shortened to accommodate rough terrain, or the center of gravity may be lowered by crouching to improve balance). If the potential were unlimited, joints would always move throughout their full range of motion, irrespective of external forces.

Figure 5.3b shows the range of motion and relative power of each joint in the morphology. Limiting the maximum force exerted by an individual joint produces a system in which multiple joints must work together to move the animat. Specifically, the range of motion of one joint is indirectly determined by the evolved muscle output parameters of other joints. Moreover, limiting the overall output potential of each joint allowed the limbs to flex and react to the ground when landing, increasing stability and the "naturalness" of the gait. Here, this model is applied only to the rear legs, as the fore legs do not factor heavily into the locomotion pattern for evolved individuals.

5.2.3 Evolutionary Setup

For each of five treatments, described in the next section, we executed 25 replicate runs, each with a unique random number seed. In each run, a population of 150 individuals evolved for 4000 generations. Fitness was defined to be the distance traveled in 10 seconds of simulated time. No special selective pressure was applied to prefer hopping to other forms of locomotion. Successive generations were populated using 2-way tournament selection with mutation and crossover as defined below. The genome comprised 12, 14, or 16 values, depending on the treatment, as shown in Table 5.1. For treatments 1 and 2, the genome did not include parameters for an actuated tail.

The mutation rate was relatively high, 20%, but mutations were defined according to

Parameter	Min. Value	Max Value	
Actuation Freq.	0 Hz	2.5 Hz	
Hip Orientation	0°	337.5°	
Knee Orientation	0°	337.5°	
Ankle Orientation	0°	337.5°	
Toe Orientation	0°	337.5°	
Shoulder Orientation	0°	337.5°	
Elbow Orientation	0°	337.5°	
Center of Mass	body center - $0.25 \times \text{length}$	body center $+$ 0.25 \times length	
Hip Power	0 (passive)	1.0	
Knee Power	0 (passive)	1.0	
Ankle Power	0 (passive)	1.0	
Toe Power	0 (passive)	1.0	
Treatments 3, 4 and 5			
Tail Actuation Freq.	0 Hz	2.5 Hz	
Tail Orientation	0°	337.5°	
Treatment 5 Only			
Tail Length	$0.07 \times \text{body length}$	$2.2 \times \text{body length}$	
Tail Mass	$3.25 \times 10^{-4} \times \text{body mass}$	$0.6 \times \text{body mass}$	

Table 5.1: Individual Gene Limits

a Gaussian distribution, so an individual mutation was unlikely to produce a large change in value. We found this approach to be effective given the control strategy used, where a large change in a single key parameter, such as a phase offset, often produced an unstable solution. A more conservative mutation approach allowed for gradual change to gait patterns over generations.

Single-point crossover was applied with a probability of 25% per genome. Crossover exhibited spatial locality, in that parents of an individual solution were chosen within a defined range. Specifically, we applied a geographical approach [118], where the population is considered as a one-dimensional line with wrap-around. Individuals are produced from parents that are considered to be close to their offspring.

5.3 Experiments and Results

The 5 treatments, described below, investigate the role of the tail in bipedal hopping, including interaction with other aspects of the morphology and the effect on gaits. Videos of selected behaviors are available in the supplementary materials.

5.3.1 Treatment 1: No Tail

In Treatment 1, individuals lack a tail. Most (18) of the 25 replicate runs failed to produce bipedal hopping, instead evolving *bounding* gaits, where fore and hind limbs alternate contact with the ground. Such gaits were common throughout the study, apparently since they offer relatively stable locomotion, albeit slower than bipedal hopping. Six of the replicate runs were able to manage two or three hops before settling into a forward-leaning gait and then regressing to a bounding gait. However, in one run, the dominant individual, shown in Figure 5.4 and the Treatment 1 video, exhibited a fairly effective bipedal hopping gait, although it flipped over near the end of the simulation period. Examination of early generations found that many individuals attempting to hop tended to flip over backwards, resulting in low fitness scores. However, one encouraging trend that emerged in this and subsequent treatments was the effectiveness of the muscle model in simulating flexible joints. During locomotion, joints flexed to react to contact with the ground, resembling the function of biological musculoskeletal systems.

5.3.2 Treatment 2: Fixed, Rigid Tail

In the second treatment, individuals had a fixed, rigid tail, and were able to evolve hopping gaits with relatively high fitness values. However, we observed that the majority of successful hoppers used the tail as a "kickstand" to prevent flipping over, as had occurred in Treatment 1. The increased stability enabled individuals to hop farther. The best evolved individual for this treatment can be seen in Figure 5.4. Most of the replicate runs produced individuals that used their tail in this manner through the entire simulation period, however,



Figure 5.4: Behavior of evolved tailless and fixed-tail individuals. The fixed tail individual is able to hop more effectively by using its tail as a stabilizer to prevent flipping over backwards.

a few managed to execute two or three hops between tail taps. Although not ideal, this tailtapping motion turned out to be an important aspect in the emergence of hopping gaits.

5.3.3 Treatment 3: Actuated Tail

The fixed tail in Treatment 2 approximates the initial posture of a kangaroo rat at the start of a hopping motion. In Treatment 3, we expanded the genome to allow the tail to evolve a speed of oscillation value as well as a starting position. We expected to see hopping gaits that did not use the tail as a kickstand. Evolved solutions for this treatment did tend to favor oscillating tails that counteracted the angular momentum of the body. However, the kickstand effect was still present in many individuals, although not as predominant as in Treatment 2. In addition to the kickstand function of the tail, evolved individuals demonstrated a coupling between tail and leg oscillation, with the tail moving against the legs to limit the rotation of the body during the hop. An evolved individual for this treatment can be seen in Figure 5.5, which shows the use of the actuated tail to stabilize the body pitch.

5.3.4 Treatment 4: Tail Collision Removal

In a natural environment, hopping species tend not to drag their tails on the ground, or even allow the tail to contact the ground at high speeds, in order to avoid injury. In Treatment 4, we explicitly removed the kickstand effect by simply preventing the tail from interacting with the ground. (Effectively, the tail could contact and penetrate the ground with no effect on the animat.) We expected solutions to instead use the tail as a counterbalance to angular momentum, consistent with a prevailing hypothesis in biology [10, 2, 80]. Instead, the results from all replicate runs were similar to the bounding gaits in Treatment 1. We suspect that the additional mass associated with a tail made it more difficult for the individuals to maintain balance, resulting in the tendency to lean forward.



Figure 5.5: An evolved hopping individual from Treatment 3 with an actuated tail. Note the coordination between tail and legs to maintain body pitch throughout the hopping motion. In the evolved individual from Treatment 5, the tail evolves to be shorter than those of the previous treatments, enabling faster hopping.

5.3.5 Treatment 5: Evolvable Tail Morphology

In the first four treatments, tails appeared to be essential to maintaining stability. In biology, it is generally agreed that an important function of the tail is to counter the angular momentum of the body, discouraging body pitch changes over the hopping period [10, 80]. Since we had based the animat's morphology on the kangaroo rat, we were curious what solutions would be discovered if tail length and tail mass were allowed to evolve. Indeed, Treatment 5 runs produced bipedal hoppers with tails approximately half as long as those in the earlier treatments; an example is shown in Figure 5.5.

5.3.6 Performance Comparison

Figure 5.6 plots the best and average fitness for each of the 5 treatments. In Treatment 1, solutions were forced to focus on stable locomotion rather than maximizing the speed of movement, resulting in low fitness. Treatment 4 exhibited even worse performance in both plots, demonstrating that in these experiments tail tapping is an important part of the behavior, at least as the animat starts moving. Treatment 5 included the best performing individuals across all treatments, although the average performance was similar to that of Treatment 3. The latter is likely due to individuals that were unstable and attained low fitness scores. Individuals in Treatment 2 had the second best performance, presumably by using the tail to stabilize the animat during hopping. Treatment 2 also had the best average fitness, perhaps indicating that the static nature of the morphology made finding stable solutions easier.

5.3.7 Analysis

Considering the high performance achieved in Treatment 5, we sought to determine which factors and relationships gave rise to effective bipedal hopping. We discovered that in the top 10% of evolved solutions in this treatment, there existed a relatively tight coupling between tail and leg oscillation frequencies. Figure 5.7a presents these data for individuals in the final generation. In the figure, the tail oscillation frequencies are generally near either a harmonic of the leg oscillation frequency, or they act as a passively flexible joint (lower right). Results that fall on or near these harmonic values have tails that move directly opposite to the rotation of the body, apparently helping to maintain a more effective body orientation. In the solutions indicated as passively flexible, the tail oscillation frequencies are so low that they behave as a flexible joint that moves only in reaction to the hopping motion, thus



Figure 5.6: Fitness of 5 treatments over evolutionary time: (a) Best performing individual, averaged across 25 runs for each treatment; (b) Average performance in each evolved population, averaged across 25 runs for each treatment. Shaded bands indicate 95% confidence intervals.

countering rotational movement. The coordination in phase between tail and leg movement appears to be essential for successful individuals and is supported by biological observation.
In hopping species, tails tend to move in concert with the rest of the body, producing a unified gait pattern. In our observations of evolved animats, individuals lacking this coordination tend to produce extraneous or detracting movements that actually hinder performance.

A second area of interest is the evolved mass of the tails and the resulting moments of inertia. As seen in Figure 5.7b, the evolved results tended towards tail masses that were less than 15% of the total body mass. Indeed, tails in some of the best performing individuals accounted for less than 5% of total body mass. These lightweight tails resulted in relatively low moments of inertia, as seen in Figure 5.8. Lower moments of inertia in these individuals potentially allow the body to change pitch throughout the hopping motion rather than maintain a stable body orientation. This is in contrast to kangaroos, which have a high moment of inertia from tail movement that stabilizes motion.

This result is intriguing because stable orientation in hopping species benefits from a high moment of inertia in tails [126]. Moreover, Figure 5.8 indicates that there is no direct relationship between the tail moment of inertia and leg oscillation frequency. A possible explanation is related to our evaluation period. While the insight into high moments of inertia for the tails is well understood, the biological observations leading to this conclusion generally focus on *steady-state* hopping. However, in our treatments, fitness evaluation begins at the start of the simulation period which includes the startup phase. Hence, individuals begin from a stationary starting position and must begin to hop before reaching their final steady state. The inclusion of the startup period places an emphasis on stability during the transition from stationary pose to hopping to avoid falling over or becoming unstable. This pressure likely forces the solutions to evolve parameters that encourage stable startup gaits over those that are most efficient or fastest during the steady-state phase. One possible approach is to delay the evaluation until the animat has had an opportunity to start moving. Adding such a transient phase, which has proven successful in other recent studies [95], might encourage tail parameter evolution towards steady state hopping. However, we note that at the time of this writing, a preliminary set of experiments showed that a transient phase



Figure 5.7: (a) Relationship between the leg oscillation frequency and tail oscillation frequency in Treatment 5. The straight lines indicate harmonics between the two frequencies. Evolved solutions tend to either fall near these lines or in the passively flexible region. (b) Relationship between the leg oscillation frequency and tail mass as a percentage of total body mass. Lighter tails are favored, although the evolved tail length remains relatively constant even for different masses.



Figure 5.8: Relationship between the leg oscillation frequency and moment of inertia for an individual. A low moment of inertia generally means the animat is likely to change body pitch during hopping.

actually *reduced* fitness. This issue is a topic of our ongoing research.

5.4 Conclusions

Although relatively uncommon in the animal kingdom, bipedal hopping provides benefits both for energy efficiency and as a survival mechanism. A better understanding of this behavior, and how it evolved, not only informs biology but has implications for the design of robotic systems. In 5 treatments, we explored the role of the tail in hopping gaits. We found that a tail is essential to hopping, as tailless individuals resorted to bounding or shuffling gaits. Evolved gaits exhibit similarities to their biological counterparts in terms of tail movement and joint coordination. However, our results also show that bipedal hopping is not limited to the morphological configurations observed in nature, but can evolve in other morphologies (i.e., those with short, light tails). Indeed, the initial morphology based on the kangaroo rat dimensions proved not to be the most effective morphology. Finally, the inclusion of the startup phase in fitness evaluation led to use of the tail as a stabilizer, which to our knowledge has not been previously reported. For this study, we developed a computationally-efficient kinematic model that approximates the function of natural muscles and is suitable for integration into evolutionary algorithms. Combined with joint power limits, this model allows for the evolution of natural looking gaits. However, from a biological standpoint, the model does not incorporate the dynamics of individual muscles or their contributions to joint-level movement. In the next chapter, we introduce the digital muscle model, which addresses these issues while remaining feasible for computational evolution.

Chapter 6 The Digital Muscle Model

The previous chapters have demonstrated that like natural evolution, computational evolution when applied to robots can exploit material properties and passive responses, as well as couple morphological characteristics and aspects of control. However, the actuators and control strategies of robots are very different than those of animals. At the joint-level, for instance, outputs of robotic controllers typically dictate the actions of the joint motors directly. In contrast, animal joints are composed of muscles, bones, and tendons that, guided by a central nervous system, interact in complex ways to collectively define the behavior of the joint. Morphology and control are tightly intertwined, as multiple muscles work together to express different behaviors, from simple locomotion to fine motor skills.

In this chapter, we describe the *digital muscle model* (DMM), which emulates the function of biological muscles, yet is abstract enough to apply to both engineered systems and biological study. Receiving commands from a high-level controller, muscle *groups* define the movement repertoire of individual joints. The model integrates aspects of both morphology and control, which are evolved together. By delegating joint-level control to muscle groups, a high-level controller is free to address complex tasks and decision making.

The digital muscle model helps to bridge the gap between control and morphology and provides a foundation for our proposed research. Instead of being explicitly encoded in the genome, aspects such as joint ranges *emerge* from the interaction between components of the model and the environment. In this chapter, we apply the digital muscle model to the evolution of locomotion in quadruped animats. Evolved agents demonstrate effective gaits that exhibit symmetry and coordination among individual joints, even though they are driven by a single, shared sinusoidal control signal. Our ongoing work, described in the next chapter, includes the integration of the DMM with more complex controllers, such as ANNs, as well as integration of an energy model that constrains output forces at joints in a manner similar to that found in nature. We emphasize that this model is not intended to replicate the functionality of physical muscles, but rather to provide an abstraction of joint-level control that can be mapped into robotic systems or used to help understand the evolution of natural organisms.

6.1 Background and Related Work

As described earlier, controllers such as ANNs [133] and CPGs [86] are particularly amenable to evolutionary optimization, and have proven successful in many forms of robotic locomotion, including salamander gaits [65], bipedal walking [109], and crawling [66]. Typically, these controllers produce high-level signals to govern the movement of each joint, such as specifying the desired angle of an actuator. In addition, a single control strategy often directs multiple aspects of behavior, from high-level decision making down to the individual joint-level commands [95]. In contrast, the movement of natural organisms is a complex interaction between an individual's neuromuscular and musculoskeletal systems.

In nature, morphology and control evolve together to produce effective behaviors. In the case of a robot, the inclusion of morphology in the evolutionary process can greatly increase the robustness of an individual [19]. Bongard and Paul [107] demonstrated the importance of co-evolving control and morphology in a bipedal robot platform; small changes to the robot's mass distribution had large effects on the resultant gait. Other studies have shown success in modular self-organizing systems [84] and the development of locomotion strategies for robots with different morphologies [8]. Doncieux and Meyer [43] have shown that it may be difficult, if not impossible, to develop complex control strategies without structural modularity in neural networks. In the proposed digital muscle model, we focus on evolving parameters that affect aspects of both joint control and joint morphology. Whether modularity associated with joint-level control adds further benefits remains an open question.

Lessin et al. [77] recently developed models of physical muscles that connect rigid bodies, with movement determined by a variable spring constant. Actuation occurs when the constant is changed, resulting in contraction or expansion of the simulated muscle. The authors investigated the evolution of multiple behaviors in these virtual creatures, with natural looking movements produced by the muscle-based effectors. Geijtenbeek et al. [51] demonstrated virtual bipeds, controlled by simulated muscles, capable of walking on both flat and uneven surfaces. Muscles were simulated with physical properties governing attachment points, contraction paths and actuation parameters that are determined through optimization. Gaits exhibited realistic movements for virtual creatures with different morphologies. While promising, these systems emulate physical muscles directly, which may complicate their mapping into some types of robotic actuators. The proposed digital muscle model does not consider muscles to be effectors in the traditional sense. Instead, the model is abstract, providing commands that govern the movement of actuators, while enabling evolved solutions to be mapped into joints driven by various types of actuators.

6.2 Digital Muscle Model

Biological muscles provide the power necessary for organisms to move and interact with their environment [42]. Working in antagonistic pairs, muscles allow for flexion and extension of individual joints, coordinated by neural systems [39, 40]. Although in some cases movement may appear to occur within a single DOF (for example, a knee extending), multiple muscles work together to both move and stabilize a joint. The digital muscle model provides an abstract control layer that emulates the fundamental properties of biological muscles, while still being suitable for realization in terms of conventional robotic actuators. Aspects of both control and morphology are integrated in the model, allowing for both to



Figure 6.1: A Digital Muscle Group is composed of nodes, radially distributed around a joint on a 2D plane. Conceptually, nodes exert a pulling force, which draws the limb segment towards the node's position on the plane. Antagonistic relationships emerge between nodes leading to coordinated movement of a joint. The outputs of a digital muscle group dictate the movement of a joint in a physics simulation engine.

be evolved concurrently.

Figure 6.1 depicts a simple example of the digital muscle model. Movement of the lower limb segment is controlled by four muscle *nodes*. All nodes in a muscle group receive the same signal from a controller (in this paper a simple sinusoid), with the *activation function* of each individual node determining its behavior. The locations and activation functions of each node is evolved. The combined responses of the muscle nodes along with their positions determine the behavior of the joint. In Figure 6.1 the muscle nodes are equally spaced around the joint, but in general this would not be the case.

6.2.1 Control

The activation function of a digital muscle node governs when and how strongly it *contracts*, that is, pulls on a limb segment. This pulling force determines how far and

fast a limb segment moves toward the node's location relative to the joint. Activation functions can be any function that maps an input signal to output values. For this study, activation functions are Gaussians with evolvable parameters: μ (center), σ^2 (spread), and α (magnitude). Nodes are limited to positive exertion values, similar to the function of natural muscles, which are only capable of active contraction. Consequently, at least two nodes, aligned as an antagonistic pair, are necessary to have both flexion and extension in a joint.

6.2.2 Morphology

Figure 6.2 shows a top-down view of the spatial component of muscle nodes, namely, where they are located with respect to their associated joint. Each node has an evolvable parameter that defines its position on a unit circle around the joint. This position determines which direction the limb will be pulled when a node contracts. Evolution of relative positions may produce a joint with a wide range of motion, as in a human shoulder, or one with limitations, as in a knee joint.

6.2.3 Motor Control Signal Generation

The activation functions of the nodes in a muscle group collectively define the response of the joint to an input signal. Activation functions for a sample group with 4 nodes can be seen in Figure 6.3. In the figure, an input signal value of -0.5 results in nodes 0, 1, and 2 exerting themselves with activations of 0.77, 0.42, and 0.28, respectively. For example, if the nodes were aligned as shown in Figure 6.2 and the input value was -0.5, then movement of the limb would be away from node 3, which does not contract at inputs under -0.1.

Joint behavior is calculated by combining the activation outputs from all nodes in a group for a given input value. The outputs for each node are projected into two values, one for each axis of the 2-DOF joint, according to the activation output and (x,y) coordinate of each node in a group. Figure 6.4 shows the results of aggregating the muscle node activations



Figure 6.2: A top down view of an individual muscle group consisting of four nodes placed radially around a joint. Each node has both an activation function and a spatial component. Together, these determine the strength and direction of pull placed on a joint by the individual node.



Figure 6.3: Activation functions for four nodes in a muscle group. An input signal determines the response of nodes according to the Gaussian activation functions for each. The values of the activation functions for an input signal of -0.5 are highlighted.

plotted in Figure 6.3. In this manner, both the activation function of each node and its spatial location contribute to the response of the joint. That is to say, the response is an *emergent*

property of the model, rather than directly dictated by the specific activation function or a single evolved parameter. Joint movement speeds are calculated as the difference between current and desired joint positions for the next time step. (The speed of movement may be included as an output in the model in future extensions.)



Figure 6.4: Activations for the 2 DOF controlled by a muscle group. The nodes depicted in Figure 6.3 map to the two commands seen in this figure. The humps in both curves near 0.4 are the result of Node 3, which is a Gaussian function with $\mu = 0.4$. Activations take into account both the activation function for a node as well as its spatial location. Nodes for this muscle group are radially distributed at 45°, 135°, 225°, and 315° as in Figure 6.2.

Figure 6.5 shows the mapping of a high-level control signal (in this case a simple sinusoidal wave) to the response of a hip joint in a quadruped. A muscle group, composed of multiple nodes, is mapped to a single joint in an animat. Each muscle group receives a single control signal which is distributed to the nodes. A high-level controller then needs only to provide one signal per joint, rather than unique signals for each muscle node.

6.3 Experiments

We conducted experiments in evolving walking gaits for the quadruped animat shown in Figure 6.5f. This animat has two 2-DOF joints per leg. Evaluations were conducted using the ODE. In the proposed model, 2-DOF joints allow the connected limb to move anywhere in 3D space, within the physical constraints of the animat. This approach also allows for fabricating evolved individuals in a physical robot using two servo motors, rotated 90 degrees



Figure 6.5: Example of an input signal being converted to joint commands in the rear hip of a quadrupedal animat. (a) A signal (in this case a simple sinusoid) from a higher level controller is distributed to each muscle group (b) which is then passed to all nodes in the group (c). Each node takes the input value and determines its output by finding the point on the Gaussian indicated by the input. This output is then combined with the spatial position of the node to determine the output for each DOF. The outputs of all nodes in the joint are aggregated (d) to derive the two motor movement commands for a joint in a robot (e). These commands are then sent to the motors (f) associated with the joint.

to each other.

6.3.1 Treatment 1 - Digital Muscle Model

In Treatment 1, the high-level control signal for an individual is a sinusoid. A population of 100 individuals evolves for 12,000 generations, using a conventional genetic algorithm, with both crossover and mutation applied at each generation. For each treatment, we executed 20 replicate runs, each initialized with a unique starting seed. Individuals contain 8 muscle groups, in which the positions of nodes are initially evenly distributed around the joint with randomized parameters for the nodes. Muscle groups have four muscle nodes in this study. However, the structure of the model allows for a variable number of nodes, with three being the minimum to produce 3D movement. In preliminary investigations, we have tested 3, 4, 6, and 8 muscle nodes per group, with effective gaits evolving in a quadruped platform in each case. During a simulation controllers are activated every 20 milliseconds. Fitness is the Euclidean distance from the starting location after 5 seconds of simulation time. The next generation is populated using 2-way tournament selection. Elitism is not used in this study. Crossover and mutation are applied with 10% and 2.5% probabilities, respectively. For purposes of crossover, individuals are treated as a composition of muscle groups. During the operation, a child individual is created from two parents, with muscle groups assigned to the corresponding joints. Genomes in this treatment consist of 128 parameters (8 muscle groups * 4 nodes * (3 Gaussian + 1 position parameter)), resulting in an average of 3 mutations per genome. Individual parameters are mutated using a normal distribution around the current value with an approximate range of $\pm 10\%$ of the parameter value.

6.3.2 Treatment 2 - ANN Controller

In the second treatment, individuals are controlled by ANNs evolved with the NEAT algorithm [122]. We emphasize that the proposed digital muscle model is not intended to compete with ANNs, but rather to complement ANNs and other high-level controllers. Here, we provide this benchmark comparison to ground results to a known method. ANNs are configured with a single input for the sinusoidal signal and 16 outputs, one for each DOF in the quadruped. ANN input is intentionally limited to the same information as in Treatment 1. Population size, the number of generations and number of replicate runs are the same as those of Treatment 1. Parameters specific to NEAT used in this study include: overall mutation rate of 33%, with specific weight mutation rate of 90%, add neuron probability of 10% and add link probability of 80%.

6.4 Results

We first describe various gaits and behaviors that evolved in Treatment 1. Next, the evolution of symmetry between independent muscle groups, as well as functional specialization are discussed. Finally, comparisons to ANN based controllers are presented. Videos of selected behaviors are available in the supplementary materials.

6.4.1 Sample of Evolved Muscle Model Gaits

Several distinct gaits evolved using the digital muscle model; three examples are shown in Figure 6.6. The evolution of multiple different gaits across the replicates demonstrates the expressive capacity of the muscle model for a given morphology. The emergence of relatively complex gaits suggests that individual muscle groups evolve to coordinate with each other. As a whole, behaviors tend to balance speed (fore/aft movement of the limbs) with stability (splaying limbs outwards from the body).



Figure 6.6: Examples of evolved gaits in digital muscle based animats. (a) Rear leg driven bounding gait with left/right symmetric motion. (b) Three legged pace gait, where the left legs move in unison, out of phase with the right rear leg. (c) A three legged bounding gait with rear legs moving in near unison.

6.4.2 Evolution of Symmetric Movements

We observed the evolution of symmetric behavior among joints in the gait depicted in Figure 6.6a. In this gait, the rear legs provide forward propulsion, moving symmetrically, with the front acting to keep the body upright. Coordination among legs is evident, along with the evolution of left/right symmetry.

Figures 6.7 and 6.8 plot the two axes at different points in the evolutionary process, of movement for the rear hips from the individual in Figure 6.6a. Early in evolution, individuals do not demonstrate symmetric or in-phase movement, instead exhibiting a less coordinated, shuffling gait. The symmetric, in-phase coordination between the rear hip joints is evident in later generations as the two joints evolve similar phase, amplitude and period. Perfectly symmetric movement is difficult to evolve, due to the high number of parameters and interaction between multiple muscle groups required to express effective gaits. Evolution of symmetric movement is apparent in Figure 6.8, wherein, movement away from the body is initially quite different between the hips. Over the course of evolution, these two muscle groups exhibit like behaviors, ultimately demonstrating similar phase, period, and amplitudes. Coordination in both axes of movement of the hips results in an effective forward bounding gait.

Figure 6.9 shows the movement paths for the rear hips at different points during evolution. Here, the early generation individual exhibits a shuffling gait where the right rear hip pulls the leg under the robot. Early generation individuals also exhibit random movement trajectories, with little observable coordination between the two axes of movement. This is illustrated by the erratic paths for both joints. Over the course of evolution, however, these movements smooth out, ultimately producing roughly ellipsoidal trajectories. In addition to the smooth movements within a joint group, the evolution of left/right symmetry can also be seen between the two hip joints.

Figure 6.10 shows the evolved configuration of the muscle nodes for both muscle groups in the rear hips. Here, three of the four nodes in each muscle group are relatively similar in spatial position. Even though the fourth nodes are not close to each other, the expressed behaviors, as indicated by the previously discussed figures, are quite similar. In the muscle model, similar behaviors can emerge, despite completely different muscle node configurations, as *both* activation and spatial positioning determine the contraction of each node.

6.4.3 Evolution of a Functional Knee

Although joints in the animat have 2 DOF, the muscle model allows for functional specialization to 1 DOF joints. For example, in one of the replicates, the second joint of the rear left leg evolved to a functional knee joint. Figure 6.11 shows the evolution of this joint-level control. In the first few generations, the joint flexes in response to the movement



Figure 6.7: Evolution of forward and back movement of the rear hips in a bounding individual. Positive angles indicate forward movement. Initially, the joint movements are not synchronized and differ in amplitude. As evolution progresses, movement of the hips becomes synchronized with the joint angles moving toward a common phase, amplitude and period.

of other joints in the animat, as opposed to providing direct thrust for movement. This behavior serves to keep the animat stable while the other limbs provide thrust for movement. In later generations, however, the joint assumes an active role, as different muscle groups start exhibiting coordination across the animat. The expressed behavior in the muscle group evolves to an ellipsoid, elongating and narrowing. By generation 300, most of the reactive and jerky movements observed in earlier generations disappear. Planar movement and knee-like functionality are observable by generation 1,000, with this behavior becoming more refined in the final individual at generation 11,999.

6.4.4 ANN Evolved Controllers

Evolved gaits with ANN controllers were also found to be effective, although the individuals tended to remain low to the ground, apparently to maintain stability. Figure 6.12 shows a representative gait evolved using NEAT. Many ANN-based controllers exhibited



Figure 6.8: Evolution of movement away from the body in the rear hips of a bounding individual. Positive angles indicate movement away from the body. Movement of the rear hips is initially quite different, with the gap closing over the course of evolution to ultimately converge towards similar phase, period and amplitudes.



Figure 6.9: Evolution of the rear hip joint trajectories. The x-axis represents movement toward and away from the body. Values near 0 represent movements near the robot, while larger values indicate movements away from the robot. For the right hip, negative values on the x-axis indicate movements under the robot, while positive values are associated with movements away from the body. In the left hip, the opposite is true. The right hip initially crosses over the 0 boundary, resulting in the leg being under the robot. Later generations exhibit symmetric movements mirrored about 0 on the x axis.



Figure 6.10: The evolved muscle node positions for the two rear hips do not directly mirror each other as only three of the four nodes have similar spatial positions indicated by the red rectangles. Instead, symmetry in the expressed joint movements is a combination of both node position and activation functions relating to each muscle node. Actual movements are symmetric and coordinated as seen in Figures 6.7, 6.8, and 6.9.



Figure 6.11: Joint movement for the left rear knee over evolutionary time. The joint initially moves somewhat erratically in both degrees of freedom, with a noticeable hitch. At generation 50, the joint has a balanced movement between both axes but still has jitter. This results in jerky movement of the lower limb. As evolution continues, the movement becomes planar, using a combination of both degrees of freedom. A functional knee joint then evolves with the lower limb moving steadily back and forth without much side-to-side movement.

similar movement among all legs, rather than anti-phase movements, as in walking or pace gaits. This pattern emerged in the majority of the replicates. This behavior led to lower fitness than the gaits evolved in Treatment 1. The synchronous movement is likely due to the evolutionary method of complexification in NEAT, which results in the growth of networks from an initially fully connected state. In addition, the lack of environmental inputs forces the ANN to work with only a single stimulus. As a result, multiple legs receive the same or similar control signals from the ANN.



Figure 6.12: Evolved three legged bounding gait using an ANN-based controller. The main body remains low to the ground throughout the evaluation period, emphasizing stable locomotion. Limbs exhibit symmetric movements, likely due to the complexification of the ANN over evolutionary time.

6.4.5 Performance Comparison

Figure 6.13 plots the evolutionary trajectories and fitness performance for the two treatments. In this study, evolved muscle model controllers outperform ANN controllers, with the maximum and average fitnesses being significantly different (p < 0.001 t-test). Quantifying the behaviors across treatments is difficult due to the variety of gaits that evolve. However, one indicator of general behavior is the average height above the ground for the main body of the animat, plotted in Figure 6.14. Individuals with a muscle model controller tend to maintain higher main body positions than those with ANN controllers. Higher postures are indicative of walking, as opposed to a low crawl or shuffling gait. Observations of the sample gaits from each treatment support this interpretation, the individuals from Treatment 1 exhibit more vertical leg standing postures, whereas ANN based controllers limbs are splayed outward, with the main body often contacting the ground. Although it provides stability, such contact results in drag, reducing velocity. Individuals that avoid this behavior, as is often the case in Treatment 1, are able to move more effectively, resulting in higher fitness values.



Figure 6.13: Evolutionary fitness progressions for both treatments. Shaded areas indicate the 95% confidence intervals across 20 replicate runs per treatment. Both the maximum fitness distribution and average fitness distribution are significantly different (p < 0.001).

6.5 Possible Applications

As noted earlier, although the digital muscle model presented in this paper is compared to an ANN based control strategy, the two are not meant to be competitive. Instead, the muscle model is intended to provide a means to co-evolve joint-level control and joint morphology. High-level control strategies can be governed by an ANN or rule-based control strategy. The muscle model can then produce basic gaits without requiring much input from



Figure 6.14: The average body height above the ground for all replicate runs between the two treatments. Shaded areas indicate 95% confidence intervals with the two distributions being significantly different (p < 0.001). As a whole, gaits from Digital Muscle Model controllers evolve higher postures as legs are typically held closer to vertical. Whereas, ANN controllers evolve gaits that tend to remain closer to the ground, splaying the legs outward.

a higher level controller, freeing those resources to be applied to more complex maneuvers and decision making. Gaits evolved in Treatment 1 are driven using only a sinusoid control signal, in order to help us understand how the muscle model functions under controlled conditions. In our proposed research, we plan to investigate multi-tiered control strategies.

In addition to serving as a robotic control strategy, information about evolved orientations of muscle nodes and activation functions can inform biology. Given the incomplete fossil record, evolutionary algorithms and simulation provide a means to test different hypotheses regarding joint-level control and its role in locomotion. Understanding the mechanics of specific motions and muscle configurations is of interest to the study of biological organisms [92, 101, 96]. Evolutionary experiments can yield insight into the biomechanics underlying basic movements [91]. However, finely detailed musculoskeletal simulations are impractical for use in computational evolution. In such musculoskeletal simulators, single experiments can require multiple days of computation time for a single analysis, with the process often requiring multiple iterations. Moreover, while these simulations provide insight into the neuromuscular control of living animals, they are limited in their ability to explore alternative morphologies or behaviors. The digital muscle model provides a way to simulate the basic mechanics underlying natural muscles without the high overhead cost of more detailed simulations. Individual runs (1.2 million evaluations) conducted for this study took approximately 24 hours using a server with 24 cores. Information obtained from this model can potentially be used as a basis for creating models in more detailed simulators.

6.6 Conclusions

The digital muscle model presented in this chapter is meant to provide a bridge between aspects of control and morphology at the joint level. Each muscle group provides a lowlevel control strategy that defines the behavior of an individual joint. In contrast to a comprehensive control strategy that supports high-level cognitive tasks (waypoint following, path planning), mid-level (walking, turning), and low-level (basic movements, extension, flexion), the muscle model is intended to function as a low-level controller only. Additionally, the proposed model serves as a computationally efficient tool to assist biological study when using computational evolution. These features, along with the abstract nature of the control strategy, allow the digital muscle model to map into robotic systems while also informing biological study. Results show that effective gaits can evolve using the digital muscle model, with instances of functional specialization, coordination, and symmetry all appearing in evolved individuals. In the next chapter, we combine the low-level DMM with a high-level ANN to realize a comprehensive controller for legged locomotion.

Chapter 7

Combining the Digital Muscle Model and High-Level Control

In the previous chapter, we introduced the Digital Muscle Model and demonstrated the evolution of effective gaits, even when driven by a simple oscillating signal. Alone, the responses produced by a DMM controller strictly map an input value to joint commands, without integrating sensory information from the environment. In natural systems, touch, body posture, and environmental information aid control decisions, providing information necessary for robust responses. The brain accounts for these sensory modalities, allowing responses to adapt to different environmental conditions. In this chapter, we investigate the integration of DMM-based joints and neuroevolution, employing ANNs as a high-level controller. Specifically, we explore different types of connectivity between ANN and DMM controllers in legged robots. First, we evolve locomotion in an eight-joint quadruped animat, then evolve gaits for a twelve-joint hexapod.

Results indicate that the ANN/DMM controller achieves higher performance than ANNonly controllers in locomotion tasks while maintaining a comparable number of connections in the evolved networks. This is consistent with theories of control in biological organisms where movement "primitives" in the spinal cord are thought to govern the coordination of multiple muscles, simplifying the high-level commands dictating locomotion [52]. The main contributions of this chapter are as follows. First, we evolve a control architecture that more closely resembles that of natural organisms, where some control is relegated to the joint level. Second, we demonstrate that these controllers are capable of achieving higher performance than ANN-only controllers in legged locomotion. Finally, we present an analysis of the ANNs in the hybrid controllers, comparing them to monolithic ANN controllers in terms of the network structure, specifically the number of connections and hidden nodes.

7.1 Related Work

This work is motivated by the idea that modularity in the controller may lead to more robust behaviors. Doncieux and Meyer [43] have demonstrated that it can be difficult to develop complex control strategies without structural modularity in ANNs. Pasemann et al. [105] found that by focusing on evolving modular neurocontrollers, small networks can be constructed to address specific tasks. These subnetwork behaviors can then be combined to realize complex behaviors. Li and Miikkulainen [79] introduced the idea of a switching ANN, wherein small subunits of computation are employed to address a specific task, with the highlevel ANN deciding which subunit to employ given a set of environmental conditions. Perhaps the most similar control architecture to the ANN/DMM control proposed in this chapter is the Subsumption Architecture introduced by Brooks [26]. There, hierarchical control is employed so that multiple tasks can be expressed by a single controller, with behaviors preempting each other through a predefined ranking system. Subsumption architectures have since been employed to control exploratory behaviors in individual robots [125] and at the swarm level to govern collective behaviors [100]. However, the previous strategies investigate modularity and hierarchy at the control level only. In this chapter, we investigate a related, but novel area, addressing the integration of high-level, low-level control and the morphology of a system as in natural organisms.

7.2 ANN/Muscle Model Integration.

We investigated two strategies for connecting ANN to DMM, illustrated in Figure 7.1. The top figure shows the singly-connected strategy, where the ANN has a single output for each muscle group. Similar to the single input signal in the previous chapter, each node in a muscle group shares the same ANN output. Movement of the joint is then determined by combining the individual responses of each node to the signal. The bottom figure shows the individually-connected strategy, where each node of a muscle group is connected to a unique ANN output. The individually-connected strategy potentially allows for more fine-grained control of the individual muscle nodes by the ANN, but at a cost of an increased number of ANN outputs. In the example shown in Figure 7.1, individually-connected ANNs require 4 times as many outputs as the singly-connected ANNs as we consider muscle groups with 4 nodes each.



Figure 7.1: Examples illustrating the two connection strategies tested in this study: (top) singly-connected and (bottom) individually-connected. Interaction between the ANN and DMM-based joint proceeds as follows: (a) ANN receives input from sensors and produces output(s), 1 for a singly-connected joint and 4 for an individually-connected joint. (b) For a singly-connected joint, the same ANN output signal is distributed to each of 4 muscle nodes. For an individually connected joint, each muscle node receives its own signal directly from the ANN. (c) The position and activation function of each muscle node determines its response to the incoming signal. (d) Responses of the muscle nodes are combined and (e) passed to the platform.

In this chapter, we are specifically interested in the following questions. How do the two proposed ANN/DMM controllers perform for robotic platforms with differing degreesof-freedom (DOF)? Second, how do the ANN/DMM controllers perform against ANN-only controllers in legged locomotion? Finally, what differences, if any, arise between the two connection strategies such as the overall performance or number of connections in the ANNs?

7.2.1 Evolutionary Setup

Populations comprise 100 individuals and are evolved for 2000 generations. We conduct 20 replicate runs per treatment, each with a unique starting seed. Evolution is conducted with the NEAT algorithm [122] which handles the ANN component of the controller. NEAT parameters used in these experiments are listed in Table 7.1. These parameters were derived from preliminary experiments and found to evolve high-performing individuals in various locomotion tasks. Here, we extend the NEAT algorithm by pairing DMM controllers with an ANN through a genome identifier. Specifically, the genome identifier is used to match two components of the genome through the evolutionary process as NEAT handles ANN evolution. Crossover is applied to the DMMs by tracking when it is performed on the ANN component. The two parent genomes are selected by the genome IDs and the DMM component is then crossedover to form the full ANN/DMM child genome. Mutation is applied per generation to individual DMMs with a 5% chance per parameter; each joint has four nodes with four parameters each.

7.3 Quadruped Locomotion

We begin our investigation of the the ANN/DMM controllers by evolving gaits for quadruped robots. The simulated 8-DOF quadruped robot can be seen in Figure 7.2. Movement of the legs can be away from the torso, or along the long axis of the torso. Each joint has a $\pm 120^{\circ}$ range of motion. Evolved individuals need not utilize this entire range as the DMM controller is intended to evolve a range of movement within this hard limit.

Parameter	Value
Compatibility Threshold	5.0
Young Age Threshold	15
Species Stagnation	1000
Old Age Threshold	35
Min Species	1
Max Species	25
Recurrent Prob	0.25
Crossover Rate	0.75
Mutation Rate	0.33
Mutate Weights Prob	0.90
Weight Mutation Rate	0.75
Max Weight	20
Add Neuron Prob	0.4
Add Link Prob	0.4
Rem Link Prob	0.05

Table 7.1: NEAT Parameters for Gait Evolution



Figure 7.2: The quadruped robot has eight 2-DOF joints, a hip and a knee for each leg.

We conduct three treatments: singly-connected ANN/DMM controller (SC), individually-connected ANN/DMM controller (IC), and an ANN-only controller (ANN). ANNs in the ANN/DMM controllers have 8 outputs in the singly-connected controllers and 32 outputs in the individually-connected controllers. The ANN-only controller has 16 outputs, one for each DOF in the quadruped. Inputs to the ANN for all three treatments include a sinusoidal signal, a bias, one touch sensor for each foot, and 2 angle sensors per joint. Individuals are evaluated based on the distance traveled by the torso in 10 seconds of simulation time on a high friction surface.

7.3.1 Quadruped Results

Figure 7.3 plots the average maximum fitness (distance traveled) across replicates for the three control strategies. Here, the hybrid ANN/DMM controllers outperform the ANNonly controller across replicate runs. Individually-connected controllers perform only slightly better than singly-connected controllers with the confidence intervals often overlapping.



Figure 7.3: Average maximum fitness across 20 replicate runs per treatment in the quadruped platform. Shaded areas represent the 95% confidence intervals.

In addition to performance, we are also interested in characteristics of the evolved ANNs, specifically with respect to the number of connections and hidden nodes. Differences in the structure of the ANNs could indicate that the low-level DMM controller is offloading control functionality from the ANN or perhaps changing the interaction between ANN and the underlying joint-level control. Figure 7.4 plots the average network size, by number of connections, versus fitness across the three treatments. Each point in the figure represents the average number of connections in the highest fitness individuals from the twenty

replicates at each generation. Although the singly- and individually-connected controllers achieve similar fitness (Figure 7.3), the singly-connected controllers have a lower number of connections, reaching their peak fitness near an average ANN size of 1293.25 connections. Individually-connected controllers achieve similar performance but have an increased number of connections in the evolved networks (1395.35 connections). The ANN-only controllers, while not as high-performing, have a smaller network size on average (713.25 connections) when compared to the ANN/DMM controllers.



Figure 7.4: The average number of connections in the evolved ANNs of the highest performing individual per replicate versus fitness. Each point represents the average number of connections and average fitness across twenty replicates per treatment for a generation. Dashed vertical lines indicate the average number of connections in the highest performing generation per treatment.

Although difficult to see in the plot, further analysis shows that the individuallyconnected treatment reaches a peak average fitness of 25.32 (8.44 body lengths), the singlyconnected treatment 24.79 (8.26 body lengths) and the ANN-only treatment 21.76 (7.25 body lengths). From these results, it would appear that the singly-connected control strategy is the better of the two ANN/DMM controllers, having similar performance with a reduced number of connections in the evolved ANNs. However, the ANN-only controllers are less complex than both ANN/DMM controllers, but performance is not as good.

The previous plot shows the relationship between network complexity and performance, but the data points are averages across replicates of the farthest traveling individual at each generation. Averaging statistics across replicates reveals general trends in the evolutionary performance of the various treatments. However, these statistics do not provide insight into the performance of the best individuals. Figure 7.5 plots the highest performing individual from each replicate run across the three treatments. This plot shows that the farthest traveler has an individually-connected controller (992 connections, Fitness = 29.987, 9.99 Body Lengths) with the best singly-connected controller a close second (1051 connections, Fitness = 29.883, 9.96 Body Lengths). Between the two ANN/DMM controllers, the fitnesses are not significantly different (p = 0.1207). For this, and subsequent pairwise comparisons conducted in this chapter we employ the Wilcoxon Rank-Sum Test as the populations cannot be assumed to be normally distributed.



Figure 7.5: The number of connections versus fitness for the highest performing individuals from each replicate run for the quadruped platform. Dashed vertical lines indicate the farthest traveling individual per each treatment.

Figures 7.6 and 7.7 plot the distribution of the farthest traveling individuals from each

replicate in terms of fitness and number of connections, respectively. The ANN/DMM controllers are both significantly better than the ANN-only controllers (p < 0.001). However, the number of connections in the singly-connected controllers is significantly less than the individually-connected controllers (p < 0.001). Moreover, there is no significant difference in the number of connections between singly-connected and ANN-only controllers (p = 0.3408). To summarize, the singly-connected strategy provides good performance with a relatively compact ANN.



Figure 7.6: Distribution of fitnesses for the best individual per replicate across the three controllers for quadruped locomotion. Results are significantly different between the hybrid and ANN-only controllers. There is no significant difference between the singly- and individually-connected controllers.

While the number of ANN connections in the two types of hybrid controllers is significantly different, the number of hidden nodes is similar (p = 0.718). These results are shown in Figure 7.8. The number of hidden nodes in the ANN-only controllers is significantly less than those of the hybrid controllers (p < 0.001). Figure 7.9 plots the distribution of the number of hidden nodes across replicates. Connections between nodes in an ANN facilitate information transfer while the hidden nodes act as computational units [133]. Here, we speculate that the DMMs may cause the evolved ANNs to have more hidden nodes to



Figure 7.7: Number of connections for the best individual from each replicate across the three treatments in quadruped locomotion. The individually-connected strategy is significantly different from the other two, while there is no significant difference between singly-connected and ANN-only controllers.



Figure 7.8: Number of hidden nodes versus fitness for the farthest traveling individual from each replicate in the quadruped platform. In contrast to the number of connections versus fitness, there is not a clear relationship between the number of hidden nodes and fitness.

compensate for the hybrid controller as the ANN does not have direct control of the joints themselves.



Figure 7.9: Number of hidden nodes across treatments for the quadruped platform.

7.4 Hexapod Locomotion

Our second experiment applies the ANN/DMM controller to locomotion in hexapods, see Figure 7.10 for the 12-DOF robot. Again, movement of the legs can be away from the torso, or along the long axis of the torso. Each joint has a $\pm 90^{\circ}$ range of motion.

As in quadruped locomotion, we conduct three treatments with the singly-connected, individually-connected, and ANN-only controller configurations. ANNs have 12 outputs in the singly-connected case, 48 outputs in the individually-connected case, and 24 outputs in the ANN-only case. ANN inputs are the same as those in the quadruped robot, including a sinusoidal signal, a bias, one touch sensor per foot, and 2 angle sensors per joint. Populations comprise 120 individuals evolved for 2000 generations. 20 replicates are conducted per treatment. Individuals are evaluated based on the distance traveled by their torso in 10 seconds of simulation time on a high friction surface.



Figure 7.10: The hexapod robot has twelve 2-DOF joints, a hip and a knee for each leg. Movement of the legs can be away from the torso, or along the long axis of the torso.

7.4.1 Hexapod Results

Figure 7.11 plots the average maximum fitness over generations, while Figure 7.12 plots the average number of connections of the farthest traveling individual versus fitness. The results are similar to those of the quadruped platform, but with a larger difference in the number of connections between the singly-connected and individually-connected treatments.

Figure 7.13 plots the number of connections versus fitness for the farthest traveling individual from each replicate. For the highest performing individuals per replicate the two ANN/DMM control strategies have similar performance (p = 0.3408), but the number of connections in the ANNs diverge further than in the quadruped (p < 0.001). The singlyconnected strategy's highest performing controller has 798 connections (Fitness = 38.185, 12.73 Body Lengths) while the individually-connected strategy's best performer has 1941 connections (Fitness = 39.948, 13.32 Body Lengths). Again, the ANN-only controllers have a significantly lower fitness (p < 0.001 for both) with 1169 connections (Fitness = 30.256, 10.09 Body Lengths) in the highest performing individual. The ANN-only controller still has a significantly smaller number of connections compared to the individually-connected controller (p < 0.001). Here, the increased DOF appears to induce an increased number of connections across all three controllers. However, the singly-connected strategy has the



Figure 7.11: Average maximum fitness per generation across 20 replicate runs per treatment in the hexapod platform. Performance between the two hybrid controllers is similar while the ANN-only controller lags behind.



Figure 7.12: Average number of connections of the farthest traveling individual across replicates per generation versus fitness for the hexapod platform.

highest performance while maintaining lower connectivity in the evolved ANNs.

Figures 7.14 and 7.15 show the distributions of the fitnesses and number of connections,



Figure 7.13: Number of connections versus fitness for the farthest traveling individual from each replicate in the hexapod robot. Dashed vertical lines indicate the farthest traveling individual per each treatment.

respectively, of the highest fitness individuals per replicate for the three treatments. The hybrid ANN/DMM controllers both outperform the ANN-only controller, with the highest performing individual across all treatments arising in the individually-connected treatment. We do note however that there is no significant difference in performance between the singlyconnected and individually-connected strategies. In fact, the next three highest performers come from the singly-connected treatment with less than half the number of connections in the ANNs. Overall, the number of connections for the singly-connected controllers is half that, on average, of the individually-connected controller. This difference is larger than observed in the quadruped and suggests that there is a benefit to having a singly-connected control strategy for increasing DOF, even though the coupling between ANN and DMM is reduced. In this platform, the singly-connected controller has 12 outputs connecting the ANN to the DMM while the individually-connected controllers require 48 outputs between ANN and DMM. Furthermore, the number of connections suggests that there may be an issue of scalability for individually-connected controllers as the number of joints increases.
The connections in the evolved networks are almost double those of the quadruped for the individually-connected strategy while a similar increase is not observed in the singlyconnected or ANN-only controllers.



Figure 7.14: The fitness distributions for the three controllers in the hexapod platform. There is no significant difference between singly- and individually-connected ANN/DMM controllers while both are significantly different than the ANN-only controller.

Figure 7.16 plots the number of hidden nodes versus fitness in the farthest traveling individual per replicate. Similar to the results for the quadruped, there is not a clear relationship between the number of hidden nodes and fitness in the evolved networks. Figure 7.17 plots the number of hidden nodes across the three treatments. Unlike the increase seen in the number of connections between the quadruped and hexapod platforms, the number of hidden nodes does not increase as rapidly. This perhaps suggests a different relationship between the hidden nodes in an ANN and the pairing to a DMM. Still, with only two different DOF, it is difficult to determine if this is a trend or simply limited to these two configurations.



Figure 7.15: Number of connections for the farthest traveling individual from each replicate across the three treatments in the hexapod platform. Similar to the quadruped, the individually-connected strategy has a significantly higher number of connections while there is no significant difference between singly-connected and ANN-only controllers.



Figure 7.16: Number of hidden nodes versus fitness for the farthest traveling individual from each replicate for the hexapod platform. There is not a clear relationship between hidden nodes and fitness.



Figure 7.17: Number of hidden nodes for the best individual from each replicate for the hexapod robot. There is no significant difference between singly- and individually-connected controllers while both are significantly larger than the ANN-only controllers.

7.5 Conclusions

In biological organisms, multiple muscles coordinate to collectively realize movement of joints. The Digital Muscle Model provides a computationally efficient means to evolve joint-level control in 3D animats. In this chapter, we have examined the integration of a high-level ANN controller with low-level DMM-based joints to realize effective gaits in legged robots. The quadruped and hexapod platforms provide two different DOF (8 and 12) to assess characteristics of the control configurations. Singly-connected controllers perform comparably to the individually-connected controllers, exhibiting a reduced number of connections and similar number of hidden nodes in the evolved ANNs. Surprisingly, this property holds even with a reduced level of coupling between ANN and DMM.

Both hybrid controllers exhibit superior performance to their ANN-only counterparts. This result is consistent with theories of control in biological organisms, where movement primitives in the spinal cord are thought to govern the coordination of multiple muscles, simplifying the high-level commands dictating locomotion [52]. Our results suggest that hybrid ANN/DMM controllers may be preferable to ANN-only controllers in evolving gaits. However, the two platforms studied here have a large gap in the total number of joints. In the next chapter, we consider an animat where the number of joints can be gradually increased, providing a set of experiments where the DOF ranges from 1 to 12.

Chapter 8 ANN/DMM Interactions

In the previous chapter, we examined the interaction between ANN and DMM for a quadruped and hexapod. However, it is difficult to determine what impact the number of joints in a robot might have on the evolved controllers due to the disparity between the two platforms. The limitations of legged platforms prevent evaluating the hybrid controllers across a range of joints. Specifically, adding legs and joints requires an increase of multiple joints to maintain symmetry and balance. In order to explore the effect of joint number, we shift our platform to a worm-like robot, shown in Figure 8.1. The worm-like design allows us to increase the number of joints incrementally, examining changes in evolved ANNs as individual joints are added.



Figure 8.1: Three different worm-like robots. The overall shape and mass of the robot remains constant throughout the different trials. (a) Three-joint, (b) five-joint, and (c) ten-joint robot.

8.1 Robot Platform

Throughout these experiments, the overall length, width, height, and mass of the robot remain fixed. The body is divided into increasingly shorter segments. Each joint is a 2 DOF hinge allowing for movements perpendicular to the radial axis of the worm with a range of motion of $\pm 90^{\circ}$ in each axis. The maximum force of each joint is set so that an individual joint alone is able to move the robot. This prevents undue bias towards coordinated movements across multiple joints, were individual joints too weak to move the platform by themselves. The different segments of the robot are allowed to intersect with each other to approximate the dexterity of biological organisms. Inputs to the ANN controllers comprise two angle sensors per joint, a touch sensor for each body segment, and a bias input. The touch sensor triggers when the associated body segment contacts the ground, irrespective of orientation.

8.2 Evolutionary Setup

We again conduct three separate treatments in this chapter: singly-connected AN-N/DMM (SC), individually-connected ANN/DMM (IC), and ANN-only controller (ANN). Each muscle group contains four muscle nodes. Populations comprise 120 individuals and are evolved for 1000 generations. We conduct 20 replicate runs per treatment, each with a unique starting seed. Evolution is conducted with the NEAT algorithm [122]. NEAT handles the ANN component of evolution with DMM controllers paired through a genome identifier as explained in Chapter 7. Parameters used in the NEAT algorithm are presented in Table 8.1.

8.3 Results

8.3.1 Gaits

As with the legged robots, a variety of gaits evolved. Although the focus of this investigation is not on the characteristics of the gaits, we briefly review the types of locomotion

Parameter	Value	Parameter	Value
Compatibility Threshold	5.0	Mutation Rate	0.33
Young Age Threshold	15	Mutate Weights Prob	0.90
Species Stagnation	1000	Weight Mutation Rate	0.75
Old Age Threshold	35	Max Weight	20
Min Species	1	Add Neuron Prob	0.4
Max Species	25	Add Link Prob	0.4
Recurrent Prob	0.25	Rem Link Prob	0.05
Crossover Rate	0.75		

Table 8.1: Worm-Like Robot NEAT Parameters

observed for this platform. Figure 8.2 shows a sample of three gaits, one from each treatment. Videos of selected behaviors are available in the supplementary materials. We note that all three treatments evolve effective gaits. Many different behaviors evolved including folding (middle of the worm hinges while ends act as feet), hopping (one end curls and acts as a primitive leg), and rolling (robot curls into a wheel), among others.



Figure 8.2: A sample of three gaits, one from each treatment. (Top) ANN-only evolved controller that exhibits a rolling gait, curling and unfolding to produce movement. (Middle) Singly-connected controller with a hopping gait. The rear of the worm acts as a primitive leg. (Bottom) Individually-connected controller with a walking gait. The ends of the robot act as legs, moving the robot sideways with step-like movements.

8.3.2 Analysis

Figure 8.3 plots, one for each number of joints, the mean of the maximum fitnesses across 20 replicate runs. The three treatments exhibit similar performance for lower joint animats (joints < 6), while ANN-only controllers outperform the ANN/DMM controllers for higher joint robots. Note that for a 12-joint robot, the ANN-only treatment only exceeds the ANN/DMM controllers near the final generation.

Evaluating the farthest traveling individual per replicate, we find that the ANN/DMM controllers attain similar or better performance than ANN-only controllers. Figure 8.4 plots the distribution of fitnesses of the farthest traveling individuals, one per each replicate, across the three treatments. In contrast to the mean results, here we find that the highest performers arise out of the hybrid ANN/DMM controllers in the low (< 5) and high (> 8) joint robots. Table 8.2, at the end of this chapter, provides all pairwise comparisons using a Wilcoxon Rank Sum Test between treatments. In robots with 8 joints and greater, we hypothesize that the hybrid ANN/DMM controllers are able to establish basic movements through the DMM as shown earlier in Chapter 6. The ANN component then only needs to provide control signals for these movements, whereas, ANN-only controllers instead need to evolve a control strategy for each joint, potentially making the problem more difficult.

The number of connections in the evolved ANNs for the farthest traveling individuals per replicate varies largely across the three treatments. Figure 8.5 plots the number of connections versus fitness, grouped by the number of joints in the robot, for the farthest traveling individual per replicate in each of the three treatments. In general, the individually-connected controllers grow the fastest in number of connections as the number of joints increase. Both singly-connected and ANN-only controllers increase at reduced rates. By eleven and twelve joints, the singly-connected controllers have the fewest number of connections across the three treatments.

Figure 8.6 presents boxplot distributions for the number of connections in the farthest traveling individuals for each of the number of joints examined in this study. For one to seven



Figure 8.3: Mean maximum fitness across 20 replicate runs per treatment in the worm platform. Each plot represents the three treatments for the given number of joints.



Figure 8.4: Boxplot showing the fitness of the farthest traveling individual per replicate for the three treatments across the different DOF. The hybrid ANN/DMM controllers tend to have higher fitnesses than the best ANN controllers.

joints, the ANN-only controllers have the lowest number of connections. However, for nine joints and higher, the singly-connected controllers have significantly fewer connections in their evolved networks, see Table 8.2. As shown in Figure 8.4, above eight joints the singlyconnected controller achieves similar or higher fitnesses than the ANN-only controllers, while having fewer connections. This result could suggest that the ANN "offloads" some control functionality to the DMM while maintaining similar performance. Furthermore, this result could indicate a point where an ANN/DMM controller is effective for locomotion (8 joints and higher). These results are similar to those observed in the quadruped and hexapod platforms examined in the previous chapter, where the number of connections in the evolved networks increase along with the number of joints.

All three types of controllers exhibit a relatively constant number of hidden nodes. This result contrasts with that for the number of connections, which steadily increases with the number of joints. Figure 8.7 plots the number of hidden nodes across the range of joints.



Figure 8.5: Number of connections versus fitness in the farthest traveling individuals from each replicate run for the worm platform across the twelve joints.



Figure 8.6: Number of connections for the farthest traveling individuals from 20 replicate runs per each DOF across the three treatments. Differences are statistically significant for all except singly- versus individually-connected one joint (p = 0.9042) and singly-connected versus ANN-only six (p = 0.4017) and eight joints (p = 0.1404).

ANN-only controllers have the lowest number of hidden nodes across all joints. We speculate that this result is related to the fact that hidden nodes typically act as computational units while connections facilitate information transfer in ANNs [133]. These results suggest that there is a certain threshold of hidden nodes required in any evolved ANN, regardless of the number of joints. In the case of the hybrid controllers, the hidden nodes may provide additional computation to make up for the limited communication capacity between the ANN and the low-level DMM control. Furthermore, the decreased number of connections in combination with the hidden nodes may indicate that independent computations within the ANN are more prevalent in hybrid controllers when compared to ANN-only controllers.



Figure 8.7: Number of hidden nodes for the farthest traveling individuals from 20 replicate runs per each DOF across the three treatments. Differences are statistically significant for all ANN/DMM versus ANN-only controllers. There are no significant differences in the number of hidden nodes for singly- and individually-connected controllers except for 7 joints (p = 0.0047).

8.3.3 Singly- versus Individually-Connected

In the quadruped and hexapod platforms, we observed that the singly-connected controllers exhibited fitnesses comparable to that of the individually-connected controllers while having significantly smaller ANNs. The singly-connected controllers offer similar performance to the individually-connected strategy, while requiring fewer ANN outputs and, therefore, less connectivity between ANN and DMM. From an efficiency perspective, smaller ANNs will require fewer computational resources to calculate command outputs. In the case of the worm, the fitnesses between singly- and individually-connected controllers are significantly different only for 3, 7, and 11 joints (p < 0.001, p = 0.0143, p = 0.01217). However, the number of connections in these evolved networks is significantly different for all robots except those with 1 joint (p = 0.9042). As shown in Figure 8.6, the number of connections in the singly-connected controllers grows at a lower rate than the individually-connected controller.

8.4 Conclusions

In this chapter, we expanded our exploration into a model of control where a high-level ANN provides signals to a joint-level system that integrates control and morphology. As with our studies of quadruped and hexapod locomotion, hybrid ANN/DMM controllers exhibit similar performance to ANN-only controllers for the majority of test cases. However, as the number of joints in a robot increases, the farthest traveling hybrid controllers outperform their ANN-only counterparts, while exhibiting fewer connections in evolved ANNs. This result suggests that the ANN is offloading some control functionality to the DMM, similar to theories of biological control [52].

These results indicate that a single connection between the ANN and each muscle group is sufficient. This configuration requires fewer connections in the ANN while maintaining performance similar to that of individually-connected controllers. Such modularization in control might free the high-level controller to focus on tasks other than governing low-level movement of joints.

	Comp:	1	2	3	4	5	6	7	8	9	10	11	12
Fitness:	S v I	0.265	0.862	0.001	1	0.862	0.547	0.014	0.602	0.232	0.698	0.012	0.091
	S v A	0.030	0.043	0.002	0.001	0.004	0.314	0.001	0.165	0.127	0.091	0.001	0.076
	A v I	0.121	0.072	0.001	0.001	0.002	0.495	0.211	0.429	0.621	0.211	0.068	0.002
Num Con:	S v I	0.904	0.028	0.003	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	S v A	0.006	0.001	0.020	0.010	0.001	0.402	0.001	0.140	0.007	0.003	0.001	0.001
Num Hid:	A v I	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	S v I	0.465	0.449	0.083	0.695	0.978	0.310	0.005	0.137	0.850	0.756	0.120	0.473
	S v A	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	A v I	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

Table 8.2: P-values of pairwise comparison using a Wilcoxon Rank Sum Test for the farthest traveling individual per replicate from the three treatments. The three metrics are listed on the left: fitness, number of connections and number of hidden nodes in the evolved networks. Treatments are abbreviated as follows: (S) singly-connected, (I) individually-connected, and (A) ANN-only.

Chapter 9 Conclusion and Future Work

In this dissertation we applied computational evolution to the control and morphology of robotic systems. Specifically, we leveraged bio-inspired control models and the intrinsic properties of materials to produce robots capable of aquatic and terrestrial locomotion. The results highlight the importance of optimizing morphology and control together by demonstrating the coupling that evolves between the two. This research informs the design of future robotic systems as well as providing insight into characteristics of biological organisms.

In our first study, we investigated the evolution of ANN controllers capable of station keeping, where the system experiences various flow situations while keeping the robot close to the target station point. To accomplish this task, we constructed a fitness function that rewarded individuals for being close to the station point, while at the same time granting them time to reorient themselves in order to most effectively swim against the simulated laminar flow. Some evolved individuals exhibited unexpected behaviors, such as flipping over, to swim against the various flows. Furthermore, we observed that the swimming behaviors exploit characteristics of the underlying morphology. The emergence of novel behaviors highlights the ability of evolutionary approaches to discover solutions not considered *a priori*.

We next explored the effect of passive material properties on the performance of an amphibious crawling robot. Robotic systems involve complex interactions between control strategy and morphological characteristics. Here, we focused on passive joints, that is, those without an active motor to control their behavior. Evolved controllers were tasked with indirectly controlling these joints to produce effective locomotion in both surface and aquatic environments. The highest performing individuals exhibit a strong coupling between control and morphology, demonstrating that the EA exploits relationships between the two.

Digital simulations allow us to explore biological phenomena by observing evolution in action, rather than examining fossils and extant species. We studied the evolution of bipedal hopping by simulating various morphological configurations based on a kangaroo rat, specifically focusing on the role of the tail. Effective individuals demonstrated coupling between control parameters and tail configurations. The behaviors exhibited by the farthest traveling individuals were similar to those seen in nature. However, we also observed effective solutions different from natural organisms. Conducting these simulation-driven experiments can produce insight into biology and yield improvements in robotic systems. For example, hopping is an effective strategy to increase communication range in robotic systems, yet it requires the ability to model the dynamics of movement to balance performance and efficiency.

Building on the earlier chapters, we introduced the Digital Muscle Model to investigate the role of low-level control in locomotion. Here, we consider low-level control to be the basic movement primitives available to a system. High-level control combines these basic movements to accomplish locomotion and other behaviors. We first studied joint-level control as driven by a simple sinusoid control signal in quadruped animats. Even with a basic highlevel oscillating signal, evolved individuals locomoted effectively and exhibited biological parallels, including symmetry and functional specialization of joints. The emergence of these traits demonstrate that joint-level mechanisms, alone, can govern basic locomotion.

We next introduced a high-level ANN to provide input to the DMM, addressing locomotion in quadruped and hexapod animats. In these experiments, evolved ANN/DMM controllers achieved comparable performance to ANN-based controllers. However, we observed that in higher DOF robots the hybrid controllers had smaller ANNs in terms of the number of connections. This result suggests that a low-level control strategy, such as the DMM, can assume some computation from a high-level controller, potentially freeing the latter to focus on other behaviors. The results further demonstrated that, as the number of joints increases, the hybrid controllers evolved higher performance than ANNs alone.

The studies described in this dissertation draw upon biology for inspiration in the morphological configuration, control strategies, and tasks examined. Throughout the investigations, biological parallels emerged in the evolved behaviors, resembling those of natural organisms. While EAs have been shown to produce effective behaviors in robotic systems, we hope that this dissertation will lead to more effective control systems capable of leveraging their morphology to address increasingly complex tasks. In addition, the studies presented here support integrating evolutionary approaches during the design phase to produce systems whose control and morphology are optimized together.

9.1 Future Work

Results from this work lay a foundation for several future research directions. We briefly discuss four possibilities:

Digital Muscle Model. We introduced the Digital Muscle Model (DMM) in this dissertation to explore aspects of low-level control in robotic systems. While we were able to elicit high-performing gaits, there remain a number of investigations to conduct both related to, and beyond the DMM. First, we employed muscle groups with four nodes. Increasing the number of nodes may allow for different interactions between nodes, or more specialized behaviors. Second, the muscle nodes themselves did not change the maximum output force of a joint, instead dictating angular movements. Expanding the model to include modifications to the maximum force a joint can exert could have implications for efficiency and behavior. Third, the encoding strategy we employed mapped one muscle group to one joint in an animat. Symmetry and repetition are pervasive in natural organisms. Exploring methods to incorporate symmetry could simplify control, and also produce insights into how best to incorporate it in artificial systems. Finally, the activation function employed in the muscle nodes was a Gaussian throughout our investigations, but the activation function need not be limited in this way. Other functions such as a square, sigmoid, or step might elicit new behaviors or unique interactions among nodes in a muscle group.

Increased Morphological Resolution. The robotic systems in this dissertation were modeled as a collection of rigid components connected by one or two DOF joints. Typically, the torsos were modeled as a single rigid component. Biological organisms are not constrained to such simple models. For example, flexible spines increase the range of motion and introduce new dynamics to movements. Future studies may benefit from providing similar features and resolution of morphology in simulation. Indeed, we have conducted a preliminary study showing that the addition of a passively flexible spine can increase locomotion performance in quadruped animats. As computing capabilities increase, the issue of computation overhead will diminish, allowing for simulating more complex morphologies. Furthermore, if morphological computation scales with body complexity, an increased number of body segments should allow the controller to offload more aspects of behavior to the morphology.

Increased Task Complexity. In this dissertation we investigated tasks related to locomotion. Robotic systems will be required to move effectively through various environments as the range of tasks increases. However, the complexity of these tasks will also expand, requiring new methods to not only evolve locomotion behaviors, but also incorporate object manipulation, environmental surveying, and obstacle avoidance, among others. Evolutionary approaches will likely need to be expanded and augmented to incorporate multiple behaviors allowing for more robust control systems.

Energy Efficiency. Energy efficiency, a fact of life for biological organisms, remains a secondary consideration in evolutionary robotics. Typically, we focus on evaluating individuals strictly on their performance. Neglecting energy efficiency affects the range and level of autonomy in robotic systems. However, incorporating energy efficiency in the robot design process may not be as straightforward as adding a second objective to an evolutionary experiment. Instead, efficiency might play a complimentary role, desirable, but not at the same level of importance as completing the task at hand. New evolutionary techniques need to be examined to determine how best to incorporate energy conservation strategies while minimizing performance degradation and considering how morphological configurations may impact energy consumption. Such approaches might lead to robotic systems that, like natural organisms, are not only more energetically efficient, but also more robust to their environment.

BIBLIOGRAPHY

BIBLIOGRAPHY

- J. Aguilar, A. Lesov, K. Wiesenfeld, and D.I. Goldman. Lift-off dynamics in a simple jumping robot. *Physical Review Letters*, 109(174301), 2012.
- [2] R. Alexander and Alexandra Vernon. The mechanics of hopping by kangaroos (macropodidae). Journal of Zoology, 177(2):265–303, 1975.
- [3] Jamie M. Anderson and Narender K. Chhabra. Maneuvering and stability performance of a robotic tuna. *Integrative and Comparative Biology*, 42(1):118–126, 2002.
- [4] P. Arena, L. Fortuna, M. Frasca, and G. Sicurella. An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. *IEEE Transactions* on Systems, Man, and Cybernetics, Part B: Cybernetics, 34(4):1823–1837, 2004.
- [5] G. P. Arnold. Rheotropism in fishes. *Biological Reviews*, 49(4):515–576, 1974.
- [6] Joshua E. Auerbach and Josh C. Bongard. Evolution of functional specialization in a morphologically homogeneous robot. In *Proceedings of the 11th Annual Conference* on Genetic and Evolutionary Computation, pages 89–96, Montreal, Quebec, Canada, 2009. ACM.
- [7] Joshua E. Auerbach and Josh C. Bongard. How robot morphology and training order affect the learning of multiple behaviors. In *Proceedings of the 2009 IEEE Congress* on Evolutionary Computation, pages 39–46, Trondheim, Norway, May 2009. IEEE.
- [8] Joshua E. Auerbach and Josh C. Bongard. Dynamic resolution in the co-evolution of morphology and control. In *Proceedings of the Twelfth International Conference on Artificial Life*, pages 451–458, Odense, Denmark, 2010.
- [9] Joshua E. Auerbach and Josh C Bongard. Evolving monolithic robot controllers through incremental shaping. In New Horizons in Evolutionary Robotics, volume 341 of Studies in Computational Intelligence, pages 55–65. Springer Berlin / Heidelberg, 2011.
- [10] G. A. Bartholomew and H. H. Caswell. Locomotion in kangaroo rats and its adaptive significance. J. Mamm., 32:155–169, 1951.
- [11] Atilim Günes Baydin. Evolution of central pattern generators for the control of a five-link planar bipedal walking mechanism. *Paladyn. Journal of Behavioral Robotics*, 3(1):45–53, 2012.
- [12] D. Beasley, D. R. Bull, and R. R. Martin. An Overview of Genetic Algorithms: Part 1, Fundamentals. University Computing, 15:58–69, 1993.
- [13] Benjamin E. Beckmann and Philip K. McKinley. Evolving quorum sensing in digital organisms. In Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pages 97–104, Montreal, Quebec, Canada, 2009. ACM.

- [14] Randall D. Beer and John C. Gallagher. Evolving dynamical neural networks for adaptive behavior. Adaptive Behavior, 1(1):91–122, 1992.
- [15] M.D. Berkemeier and R.S. Fearing. Sliding and hopping gaits for the underactuated acrobot. *IEEE Transactions on Robotics and Automation*, 14(4):629-634, 1998.
- [16] A. A. Biewener and R. Blickhan. Kangaroo rat locomotion: Design for elastic energy storage or acceleration? *Journal of Experimental Biology*, 140:243–255, 1988.
- [17] J.C. Bongard and H. Lipson. Automated damage diagnosis and recovery for remote robotics. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 4, pages 3545–3550, New Orleans, Louisiana, USA, April 2004.
- [18] J.C. Bongard and H. Lipson. Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials. In *Proceedings* of the 2004 NASA/DoD Conference on Evolvable Hardware, pages 169–176, Seattle, Washington, USA, June 2004.
- [19] Josh Bongard. Morphological change in machines accelerates the evolution of robust behavior. Proceedings of the National Academy of Sciences, 108(4):1234–1239, 2011.
- [20] Josh C. Bongard. Morphological and environmental scaffolding synergize when evolving robot controllers. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 179–186, Dublin, Ireland, 2011. ACM.
- [21] Josh C. Bongard and R. Pfeifer. Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In L. Spector, editor, *Proceedings of* the Genetic and Evolutionary Computation Conference, GECCO-2001, pages 829–836, San Francisco, CA, USA, 2001. Morgan Kaufmann.
- [22] Joshua C. Bongard. The utility of evolving simulated robot morphology increases with task complexity for object manipulation. *Artificial Life*, 16(3):201–223, 2010.
- [23] Valentino Braitenberg. Vehicles: Experiments in Synthetic Psychology. The MIT Press, February 1986.
- [24] D.J. Braun, F. Petit, F. Huber, S. Haddadin, P. van der Smagt, A. Albu-Schaffer, and S. Vijayakumar. Robots driven by compliant actuators: Optimal control under actuation constraints. *IEEE Transactions on Robotics*, PP(99):1–17, 2013.
- [25] N. Bredeche, E. Haasdijk, and A. Eiben. On-line, on-board evolution of robot controllers. In Pierre Collet, Nicolas Monmarché, Pierrick Legrand, Marc Schoenauer, and Evelyne Lutton, editors, Artifical Evolution, volume 5975 of Lecture Notes in Computer Science, pages 110–121. Springer Berlin / Heidelberg, 2010.
- [26] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal* of Robotics and Automation, 2(1):14–23, 1986.

- [27] Rodney A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. *Neural Computation*, 1(2):253–262, 1989.
- [28] Rodney A. Brooks. Intelligence without representation. Artificial Intelligence, 47(1– 3):139 – 159, 1991.
- [29] Rodney A. Brooks. Artificial life and real robots. In Proceedings of the First European Conference on Artificial Life, pages 3–10, Paris, France, 1992. MIT Press.
- [30] H. Chaoui, P. Sicard, and W. Gueaieb. ANN-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Transactions on Industrial Electronics*, 56(8):3174–3187, 2009.
- [31] Zheng Chen, S. Shatara, and Xiaobo Tan. Modeling of biomimetic robotic fish propelled by an ionic polymer metal composite caudal fin. *IEEE/ASME Transactions on Mechatronics*, 15(3):448-459, 2010.
- [32] Hillel J. Chiel and Randall D. Beer. The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neurosciences*, 20(12):553 – 557, 1997.
- [33] F.J. Cintrón and M.W. Mutka. Hopping enhanced sensors for efficient sensor network connectivity and coverage. In 2010 IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS), pages 119–126, San Francisco, CA, USA, 2010. IEEE.
- [34] Anthony J. Clark, Jared M. Moore, Jianxun Wang, Xiaobo Tan, and Philip K. McKinley. Evolutionary design and experimental validation of a flexible caudal fin for robotic fish. In Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems, pages 325–332, East Lansing, Michigan, USA, 2012.
- [35] Jeff Clune, Benjamin E. Beckmann, Charles Ofria, and Robert T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings* of the IEEE Congress on Evolutionary Computation, pages 2764–2771, Trondheim, Norway, 2009.
- [36] Jeff Clune, Benjamin E. Beckmann, Robert T. Pennock, and Charles Ofria. HybrID: a hybridization of indirect and direct encodings for evolutionary computation. In Proceedings of the 10th European Conference on Advances in Artificial Life: Darwin Meets von Neumann, pages 134–141, Budapest, Hungary, 2011. Springer-Verlag.
- [37] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.
- [38] Brian D. Connelly and Philip K. McKinley. Evolving social behavior in adverse environments. In Proceedings of the 10th European Conference on Advances in Artificial Life: Darwin Meets von Neumann, pages 490–498, Budapest, Hungary, 2009. Springer-Verlag.

- [39] Monica A. Daley, Gladys A. Felix, and Andrew A. Biewener. Running stability is enhanced by a proximo-distal gradient in joint neuromechanical control. *The Journal* of Experimental Biology, 210(3):383–394, 2007.
- [40] Monica A. Daley, Alexandra Voloshina, and Andrew A. Biewener. The role of intrinsic muscle mechanics in the neuromuscular control of stable running in the guinea fowl. *Journal of Physiology*, 587(11):2693–2707, Jun 2009.
- [41] T. J. Dawson and C. R. Taylor. Energetic cost of locomotion in kangaroos. Nature, 246(5431):313–314, 1973.
- [42] Michael H. Dickinson, Claire T. Farley, Robert J. Full, M. A. R. Koehl, Rodger Kram, and Steven Lehman. How animals move: An integrative view. *Science*, 288(5463):100– 106, 2000.
- [43] S. Doncieux and J.-A. Meyer. Evolving modular neural networks to solve challenging control problems. In Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems (EIS 2004), pages 1–7, Madeira, Portugal, 2004.
- [44] Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Charlie Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 2012.
- [45] M. Epstein, J.E. Colgate, and M.A. MacIver. Generating thrust with a biologicallyinspired robotic ribbon fin. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2412 –2417, Beijing, China, 2006.
- [46] Jolyon Faria, John Dyer, Romain Clément, Iain Couzin, Natalie Holt, Ashley Ward, Dean Waters, and Jens Krause. A novel method for investigating the collective behaviour of fish: introducing 'robofish'. *Behavioral Ecology and Sociobiology*, 64:1211– 1218, 2010.
- [47] Huashan Feng and Runxiao Wang. Construction of central pattern generator for quadruped locomotion control. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 979–984, Xi'an, China, July 2008. IEEE.
- [48] F. Ferland, A. Aumont, D. Letourneau, and F. Michaud. Taking your robot for a walk: Force-guiding a mobile robot using compliant arms. In *Proceedings of the 2013* 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 309–316, 2013.
- [49] L.A. Fuente, M.A. Lones, A.P. Turner, L.S. Caves, S. Stepney, and A.M. Tyrrell. Adaptive robotic gait control using coupled artificial signalling networks, hopf oscillators and inverse kinematics. In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pages 1435–1442, Cancun, Mexico, 2013.
- [50] Pablo Funes and Jordan Pollack. Evolutionary body building: Adaptive physical designs for robots. Artificial Life, 4(4):337–357, 1998.

- [51] Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. Flexible muscle-based locomotion for bipedal creatures. ACM Transactions on Graphics, 32(6):1–11, 2013.
- [52] Simon F. Giszter, A. Mussa-Ivaldi, and Emilio Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, 1993.
- [53] E. Guizzo and E. Ackerman. The rise of the robot worker. *IEEE Spectrum*, 49(10):34–41, 2012.
- [54] A. K. Gutmann, D. V. Lee, and C. P. McGowan. Collision dynamics of bipedal hopping. In Annual Meeting of the Society for Integrative and Comparative Biology, San Francisco, California, USA, 2013.
- [55] K. Hase, Gon Khang, and Gwang-Moon Eom. A simulation study on the evolution of hopping motions in animals. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):353–362, 2004.
- [56] G. Hayes and R. M. Alexander. The hopping gaits of crows (Corvidae) and other bipeds. *Journal of Zoology*, 200:205–213, June 1983.
- [57] M. Hildebrand. Analysis of asymmetrical gaits. *Journal of Mammalogy*, 58(2):131–156, 1977.
- [58] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.
- [59] John H. Holland. Genetic algorithms and the optimal allocation of trials. SIAM Journal of Computing, 2:88–105, 1973.
- [60] Gregory S. Hornby and Jordan B. Pollack. Body-brain co-evolution using L-systems as a generative encoding. In *Proceedings of the 2001 ACM Genetic and Evolution*ary Computation Conference, pages 868–875, San Francisco, California, USA, 2001. Morgan Kaufmann.
- [61] G.S. Hornby, H. Lipson, and J.B. Pollack. Generative representations for the automated design of modular physical robots. *IEEE Transactions on Robotics and Au*tomation, 19(4):703 – 719, 2003.
- [62] Qingsong Hu, D.R. Hedgepeth, Lihong Xu, and Xiaobo Tan. A framework for modeling steady turning of robotic fish. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2669 –2674, Kobe, Japan, 2009.
- [63] A. J. Ijspeert. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 84(5):331–348, 2001.
- [64] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. Neural Networks, 21(4):642–653, 2008.

- [65] Auke Jan Ijspeert, Alessandro Crespi, and Jean-Marie Cabelguen. Simulation and robotics studies of salamander locomotion. *Neuroinformatics*, 3(3):171–195, 2005.
- [66] Kousuke Inoue, Takaaki Sumi, and Shugen Ma. CPG-based control of a simulated snake-like robot adaptable to changing ground friction. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1957 -1962, San Diego, California, USA, 2007.
- [67] Nick Jakobi. Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In *Proceedings of the 4th European Conference on Artificial Life*, pages 348– 357, Brighton, UK, 1997. MIT Press.
- [68] Nick Jakobi. Running across the reality gap: Octopod locomotion evolved in a minimal simulation. In Proceedings of the First European Workshop on Evolutionary Robotics, pages 39–58, Paris, France, 1998. Springer-Verlag.
- [69] Kenneth A. De Jong. Evolutionary Computation: A Unified Approach. The MIT Press, 2006.
- [70] David S. Jung, Peter P. Pott, Taavi Salumae, and Maarja Kruusmaa. Flow-aided path following of an underwater robot. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4602–4607, Karlsruhe, Germany, 2013. IEEE.
- [71] Adrian Jusufi, DT Kawano, T Libby, and Robert J. Full. Righting and turning in midair using appendage inertia: reptile tails, analytical models and bio-inspired robots. *Bioinspiration & Biomimetics*, 5(4):1–12, 2010.
- [72] Ardian Jusufi, Daniel I. Goldman, Shai Revzen, and Robert J. Full. Active tails enhance arboreal acrobatics in geckos. *Proceedings of the National Academy of Sciences*, 105(11):4215–4219, 2008.
- [73] David B Knoester and Philip K. McKinley. Evolving virtual fireflies. In Proceedings of the 10th European Conference on Advances in Artificial Life: Darwin Meets von Neumann, volume 5777, pages 474–481. Springer Berlin / Heidelberg, Budapest, Hungary, 2011.
- [74] Sylvain Koos, Jean Baptiste Mouret, and Stéphane Doncieux. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of* the 2010 ACM Genetic and Evolutionary Computation Conference, pages 119–126, Portland, Oregon, USA, 2010. ACM.
- [75] P. Krishnamurthy, F. Khorrami, J. De Leeuw, M. E. Porter, K. Livingston, and J.H. Long. An electric ray inspired biomimetic autonomous underwater vehicle. In *Proceedings of the American Control Conference (ACC)*, pages 5224–5229, Baltimore, Maryland, USA, 2010.
- [76] G. V. Lauder and E. G. Drucker. Morphology and Experimental Hydrodynamics of Fish Fin Control Surfaces. *IEEE Journal of Oceanic Engineering*, 29(3):556–571, 2004.

- [77] Dan Lessin, Don Fussell, and Risto Miikkulainen. Open-ended behavioral complexity for evolved virtual creatures. In *Proceedings of the 2013 ACM Genetic and Evolutionary Computing Conference*, pages 335–342, Amsterdam, Netherlands, 2013. ACM.
- [78] M. Anthony Lewis and George A. Bekey. Gait adaptation in a quadruped robot. Autonomous Robots, 12(3):301–312, May 2002.
- [79] Xun Li and Risto Miikkulainen. Evolving multimodal behavior through subtask and switch neural networks. In Proceedings of The Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE 14), New York, NY, 2014.
- [80] Thomas Libby, Talia Y. Moore, Evan Chang-Siu, Deborah Li, Daniel J. Cohen, Ardian Jusufi, and Robert J. Full. Tail-assisted pitch control in lizards, robots and dinosaurs. *Nature*, 481(7380):181–184, January 2012.
- [81] Aristid Lindenmayer. Mathematical models for cellular interactions in development II. Journal of Theoretical Biology, 18(3):300–315, March 1968.
- [82] H. Lipson and J. B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, August 2000.
- [83] Siavash Haroun Mahdavi and Peter J. Bentley. An evolutionary approach to damage recovery of robot motion with muscles. In *Proceedings of the 7th European Conference* on Artificial Life, pages 248–255, Dortmund, Germany, 2003.
- [84] Daniel Marbach and Auke Jan Ijspeert. Co-evolution of configuration and control for homogenous modular robots. In *Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS8)*, pages 712–719, Amsterdam, Netherlands, 2004.
- [85] Stefano Marras and Maurizio Porfiri. Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion. *Journal of the Royal Society of Interface*, 2012.
- [86] Kiyotoshi Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56:345–353, 1987.
- [87] Kiyotoshi Matsuoka. Analysis of a neural oscillator. *Biological Cybernetics*, 104(4-5):297–304, 2011.
- [88] Craig Mautner and Richard Belew. Evolving robot morphology and control. Artificial Life and Robotics, 4:130–136, 2000.
- [89] M. Mazzapioda, A. Cangelosi, and S. Nolfi. Evolving morphology and control: A distributed approach. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, pages 2217–2224, Trondheim, Norway, 2009. IEEE.
- [90] Tad McGeer. Passive dynamic walking. International Journal Robotic Research, 9(2):62–82, March 1990.

- [91] Craig P. McGowan, Richard R. Neptune, and Walter Herzog. A phenomenological model and validation of shortening-induced force depression during muscle contractions. *Journal of Biomechanics*, 43(3):449–454, 02 2010.
- [92] Craig P. McGowan, Richard R. Neptune, and Walter Herzog. A phenomenological muscle model to assess history dependent effects in human movement. *Journal of Biomechanics*, 46(1):151 – 157, 2013.
- [93] Orazio Miglino, Henrik Hautop Lund, and Stefano Nolfi. Evolving mobile robots in simulated and real environments. Artificial Life, 2:417–434, 1996.
- [94] David J. Montana and Lawrence Davis. Training feedforward neural networks using genetic algorithms. In Proceedings of the 11th International Joint Conference on Artificial Intelligence, pages 762–767, Detroit, Michigan, USA, 1989.
- [95] Jared M. Moore, Anthony J. Clark, and Philip K. McKinley. Evolution of station keeping as a response to flows in an aquatic robot. In *Proceedings of the 2013 ACM Genetic and Evolutionary Computing Conference*, pages 239–246, Amsterdam, Netherlands, 2013. ACM.
- [96] Jared M. Moore, Anne K. Gutmann, Craig P. McGowan, and Philip K. McKinley. Exploring the role of the tail in bipedal hopping through computational evolution. In *Proceedings of the 12th European Conference on Artificial Life*, pages 11–18, Taormina, Italy, 2013.
- [97] Jared M. Moore and Philip K. McKinley. Evolving flexible joint morphologies. In Proceedings of the 2012 ACM Genetic and Evolutionary Computing Conference, pages 145–152, Philadelphia, Pennsylvania, USA, 2012. ACM.
- [98] Jared M. Moore and Philip K. McKinley. Evolving energy-efficient locomotion in legged robots. Technical Report MSU-CSE-15-6, Computer Science and Engineering, Michigan State University, East Lansing, Michigan, March 2015. submitted for publication.
- [99] Jean Baptiste Mouret and Paul Tonelli. On the relationships between generative encodings, regularity, and learning abilities when evolving plastic artificial neural networks. *PloS One*, 8(11):e79138, 2013.
- [100] Fusaomi Nagata, Akimasa Otsuka, Keigo Watanabe, and MakiK. Habib. Networkbased subsumption architecture for broadcast control of multiple mobile robots based on a poor hardware/software platform. In Yong-Tae Kim, Ichiro Kobayashi, and Euntai Kim, editors, Soft Computing in Advanced Robotics, volume 269, pages 1–17. Springer International Publishing, 2014.
- [101] Richard R. Neptune and Craig P. McGowan. Muscle contributions to whole-body sagittal plane angular momentum during walking. *Journal of Biomechanics*, 44(1):6– 12, 2011.

- [102] R. Niiyama, A. Nagakubo, and Y. Kuniyoshi. Mowgli: A bipedal jumping and landing robot with an artificial musculoskeletal system. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pages 2546–2551, Roma, Italy, 2007.
- [103] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to Evolve Autonomous Robots: Different Approaches in Evolutionary Robotics. In R. A. Brooks and P. Maes, editors, *Proceedings of the 4th International Workshop on Artificial Life*, pages 190– 197. MA: MIT Press, 1994. R. A. Brooks and P. Maes (eds.).
- [104] Stefano Nolfi and Dario Floreano. Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines. The MIT Press, 2000.
- [105] Frank Pasemann, Uli Steinmetz, Martin Hulse, and Bruno Lara. Robot control and the evolution of modular neurodynamics. *Theory in Biosciences*, 120(3-4):311–326, 2001.
- [106] Chandana Paul. Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54(8):619 – 630, 2006.
- [107] Chandana Paul and Joshua C. Bongard. The road less travelled: Morphology in the optimization of biped robot locomotion. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 226 – 232, Maui, Hawaii, USA, 2001.
- [108] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. Self-Organization, Embodiment, and Biologically Inspired Robotics. *Science*, 318(5853):1088–1093, 2007.
- [109] Torsten Reil and Phil Husbands. Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6(2):159–168, 2002.
- [110] John A. Rieffel, Francisco J. Valero-Cuevas, and Hod Lipson. Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of The Royal Society Interface*, 7(45):613–621, April 2010.
- [111] L. Righetti and A. J. Ijspeert. Design methodologies for central pattern generators: an application to crawling humanoids. In *Proceedings of Robotics: Science and Systems*, pages 191–198, Philadelphia, Pennsylvania, USA, 2006.
- [112] Lisa Schramm, Yaochu Jin, and Bernhard Sendhoff. Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In Proceedings of the 10th European Conference on Advances in Artificial Life: Darwin Meets von Neumann, pages 27–34, Budapest, Hungary, 2011. Springer-Verlag.
- [113] TaeWon Seo and M. Sitti. Tank-like module-based climbing robot using passive compliant joints. *IEEE/ASME Transactions on Mechatronics*, 18(1):397–408, 2013.

- [114] Robert F. Shepherd, Filip Ilievski, Wonjae Choi, Stephen A. Morin, Adam A. Stokes, Aaron D. Mazzeo, Xin Chen, Michael Wang, and George M. Whitesides. Multigait soft robot. *Proceedings of the National Academy of Sciences*, 108(51):20400–20403, 2011.
- [115] Karl Sims. Evolving 3D morphology and behavior by competition. Artificial Life, 1(4):353–372, 1994.
- [116] Karl Sims. Evolving virtual creatures. In Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pages 15–22, 1994.
- [117] Russell Smith. Open Dynamics Engine, http://www.ode.org/, 2013.
- [118] Lee Spector and Jon Klein. Trivial geography in genetic programming. In Tina Yu, Rick Riolo, and Bill Worzel, editors, *Genetic Programming Theory and Practice III*, volume 9 of *Genetic Programming*, pages 109–123. Springer US, 2006.
- [119] M. J. Spenko, G. C. Haynes, J. A. Saunders, M. R. Cutkosky, A. A. Rizzi, Robert J. Full, and D. E. Koditschek. Biologically inspired climbing with a hexapedal robot. *Journal of Field Robotics*, 25(4-5):223–242, April 2008.
- [120] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8:131–162, 2007.
- [121] Kenneth O. Stanley, David B. D'Ambrosio, and Jason Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.
- [122] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, June 2002.
- [123] Xiaobo Tan, M. Carpenter, J. Thon, and F. Alequin-Ramos. Analytical modeling and experimental studies of robotic fish turning. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 102 –108, Anchorage, Alaska, USA, 2010.
- [124] Xiaobo Tan, D. Kim, N. Usher, D. Laboy, J. Jackson, A. Kapetanovic, J. Rapai, B. Sabadus, and Xin Zhou. An autonomous robotic fish for mobile sensing. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5424-5429, Beijing, China, 2006.
- [125] J.T. Turner, S.N. Givigi, and A. Beaulieu. Implementation of a subsumption based architecture using model-driven development. In *Proceedings of the 2013 IEEE International Systems Conference*, pages 331–338, Orlando, Florida, USA, 2013.
- [126] James R. Usherwood and Tatjana Y. Hubel. Energetically optimal running requires torques about the centre of mass. *Journal of The Royal Society Interface*, 9(73):2011– 2015, 2012.

- [127] F.J. Valero-Cuevas, Jae-Woong Yi, D. Brown, R.V. McNamara, C. Paul, and H. Lipson. The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Transactions on Biomedical Engineering*, 54(6):1161–1166, June 2007.
- [128] Vinod K. Valsalam and Risto Miikkulainen. Modular neuroevolution for multilegged locomotion. In Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pages 265–272, Atlanta, GA, USA, 2008. ACM.
- [129] R. Van Ham, T. G. Sugar, B. Vanderborght, K. W. Hollander, and D. Lefeber. Compliant actuator designs. *IEEE Robotics & Automation Magazine*, pages 81–94, September 2009.
- [130] B. W. Verdaasdonk, H. F. J. M. Koopman, and F.C.T. Van Der Helm. Energy efficient and robust rhythmic limb movement by central pattern generators. *Neural Networks*, 19:388–400, 2006.
- [131] Jianxun Wang, Freddie Alequin-Ramos, and Xiaobo Tan. Dynamic modeling of robotic fish and its experimental validation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 588–594, San Francisco, California, USA, 2011.
- [132] Darrell Whitley, Stephen Dominic, Rajarshi Das, and Charles W. Anderson. Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13(2-3):259–284, 1993.
- [133] X. Yao. Evolving artificial neural networks. Proceedings of the IEEE, 87(9):1423–1447, 1999.
- [134] Jianguo Zhao, Ruiguo Yang, Ning Xi, Bingtuan Gao, Xinggang Fan, Matt W. Mutka, and Li Xiao. Development of a miniature self-stabilization jumping robot. In Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems, pages 2217–2222, St. Louis, MO, USA, 2009. IEEE Press.
- [135] Tom Ziemke, Nicklas Bergfeldt, Gunnar Buason, Tarja Susi, and Henrik Svensson. Evolving cognitive scaffolding and environment adaptation: A new research direction for evolutionary robotics. *Connection Science*, 16(4):339–350, 2004.