Evolving Soft Robotic Locomotion in PhysX

John Rieffel
Biomimetic Devices Lab
john.rieffel@tufts.edu

Frank Saunders
Mechanical Engineering Department
frank.saunders@tufts.edu

Shilpa Nadimpalli
Computer Science Department
shilpa.nadimpalli@tufts.edu

Harvey Zhou
Computer Science Department
harvey.zhou@tufts.edu

Soha Hassoun
Computer Science Department
soha.hassoun@tufts.edu

Jason Rife
Mechanical Engineering Department
jason.rife@tufts.edu

Barry Trimmer
Biology Department
Tufts University
Medford, MA
barry.trimmer@tufts.edu

ABSTRACT

Given the complexity of the problem, genetic algorithms are one of the more promising methods of discovering control schemes for soft robotics. Since physically embodied evolution is time consuming and expensive, an outstanding challenge lies in developing fast and suitably realistic simulations in which to evolve soft robot gaits. We describe two parallel methods of using NVidia's PhysX, a hardware-accelerated (GPGPU) physics engine, in order to evolve and optimize soft bodied gaits. The first method involves the evolution of open-loop gaits using a reduced-order lumped parameter model. The second method involves harnessing PhysX's soft-bodied material simulation capabilites. In each case we discuss the the challenges and possibilities involved in using the PhysX for evolutionary soft robotics.

Categories and Subject Descriptors

I.2.9 [**Robotics**]:

General Terms

Design, Standardization, Verification

Keywords

PhysX, soft robot, caterpillar

1. INTRODUCTION

Imagine a robot that can squeeze through holes, climb up walls, and flow around obstacles. Thanks to modern advances in materials such as polymers [2], and nanocomposites [1] such a "soft robot" is becoming an increasing possibility. This ability to significantly deform and alter shape, at a much higher level of detail than discrete "modular" robots (such as Yim's Polybot [6] and Rus's Molecubes [3]) makes accessable new and increasingly important environments such as mine fields and collapsed buildings.

However, this incredible flexibility and deformability brings with it considerable complexity when it comes to control. Soft bodied robots can posess near-infinite degrees of freedom, and as a result conventional methods of robotic control, used with considerable success in rigid, jointed mechanical systems, no longer apply. Furthermore, the dynamics of these systems are complex enough that efforts to intuitively hand-design control schemes also come up short. The complexity of the problem, coupled with the non-intuitive nature of solutions, makes the control of soft robotics well suited to search via genetic algorithms.

Evolutionary optimization of robotic gaits by nature requires high-fidelity simulations, which can be quite computationally intensive. Given that thousands, if not millions, of evaluations are required to evolve highly-fit solutions, simulation is often the bottleneck on the road to progress. A conflict therefore exists between the fidelity of simulation and the speed at which at which that simulation runs.

In this paper we describe approaches to soft-bodied robotic simulation which harness general purpose computing on graphics processing units (GPGPU) in order to produce hardware-accellerated simulations without significant losses in fidelity. In particular, we use NVidia's PhysX physics engine, which runs on top of the CUDA GPGPU architecture, in order to arrive at simulations which are orders-of-magnitude faster than software-only simulation. In our first approach, we evolve fixed gait cycles using a lumped element reduced order model of a soft robot. In our second approach we employ PhysX's soft-bodied simulation features in order to evolve gaits in a truly soft system.

2. LUMPED DYNAMIC MODELING

Lumped Dynamic Modeling is a method which reduces

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

a soft-bodied robot to a set of rigid bodies connected by linear and torsional springs. This allows for the simulation of highly deformable robotic structures such soft robots,in a much more computationally efficient manner than continuum models such as Finite Element Analysis (FEA). Although not as accurate as continuum methods in resolving a soft robot's internal forces, lumped dynamic simulation are sufficient to capture the underlying physical phenomena of soft-bodied robots.

2.1 Implementation Details

The lumped approximation of a soft robot consists of two steps: (1) qualitative identification of the modes of deformation that contribute to robot motion and (2) decomposition of the robot structure into a set of rigid bodies, joints, and internal forces that reproduce those modes. In practice, the two steps are closely related. Parts identified as deformable are decomposed into multiple segments, which are constrained by joints to move in the direction of deformation (translational or rotational).

A natural method for decomposing a soft structure into rigid segments is to begin by identifying flexures in a robot body. Since bending occurs most readily at these flexures, flexure locations are naturally represented as joints in the lumped model (an example seen in Figure 1). Additional joints can be introduced to represent internal deformations within segments of a soft body. Here segments are defined as sections of the robot body between flexures. In a qualitative sense, these deformations can be modeled by subdividing each body segment into two rigid sub-segments connected by a prismatic joint (to enable compression and extension) coincident with a spherical joint (to enable bending and twisting). Relative motion of the two rigid sub-segments provides a first-order representation for deformation of the soft segment. To enable compression, adjacent segments are allowed to interpenetrate if connected by a joint. Otherwise, when segments contact each other or the ground, collisions are resolved using a coefficient of restitution model.

Internal stiffness is implemented for the lumped dynamic model by applying linear visco-elastic forcing at the joints. Linear and rotational dampers can also placed at each joint, and damping coefficients are generally estimated assuming critical damping. Stiffness parameters can be estimated by applying finite element processing to a 3-D CAD model of actual hardware, or evolved via a genetic algorithm.

External forces acting on the lumped dynamic model include gravity, friction, and actuator forces. Gravity is generally defined to act perpendicular to the ground plane. Friction is modeled with Coulomb friction coefficients. A number of actuators can be implemented onto a soft robot lumped dynamic model. For example, pneumatic actuators can be modeled with a periodic control signal for actuation. Control commands can be allowed to turn the pneumatic actuators on and off at particular times during a gait period. Actuator attachment points can be defined for each pneumatic cylinder, and the actuator forces can be simulated to act along the line between these points. A restorative spring can also be applied between the actuator attachment points to mimic return springs in the pneumatic piston hardware. The minimum and maximum distance constraints between actuator attachment points can also be implemented with prismatic joints. When an actuator is turned on it can produce a constant force and extend until it hits the maximum displacement length. Upon the actuator turning off the restorative spring can take the actuator back to its initial minimum length. Other actuators can be implemented in lumped dynamic models in a similar manner.

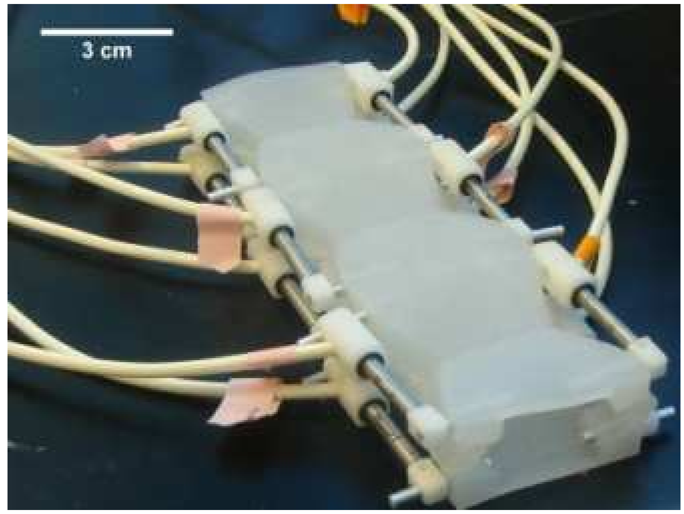

Figure 2: The pneumatic robot used for Lumped Parameter Modeling

2.2 Evolution of Morphology and Control

Genetic algorithms and PhysXl' are the two primary software tools leveraging the lumped dynamic modeling method to optimize soft robot morphologies and control patterns. PhysXl' is used to model the response of a system to actuation and evaluate the performance of a designated soft robot in an efficient manner. In order to simulate robot motion using PhysXl', lumped parameters describing the robotÕs design (actors, joints, and materials) must be provided. Accordingly, inputs are specified in the form of a genotype describing all physical and control parameters necessary to create and control a soft robot in simulation. These genotypes are evolved by a genetic algorithm to obtain high levels of simulated robot performance (fitness).

The remainder of this section will describe the evolution of controls for a pneumatic robot and control implementation on prototype hardware, as shown in Figure 2 . The pneumatic robot body was divided into segments and the relationships between those segments were defined by a combination of constraints and visco-elastic forces, and pneumatic actuators were modeled as described in the previous section. The resulting reduced order model deformed in a manner characteristic of the soft-robot.

For the evolutionary optimization, the fitness metric was distance traveled per gait period. The duration of the periodic gait as well as the activation and deactivation times for each pneumatic actuator were the units of genetic variation. Each GA run consisted of 150 generations with a population size of 200 members. For each generation an elitist hall of fame of the top 10 genotypes was preserved without crossing or mutation. The remaining 190 genotypes in each new generation were randomly created using a ÒrouletteÓ method of selection, crossing, and mutating. After 150 generations the top performers included a variety of different gait periods with similar fitness values.

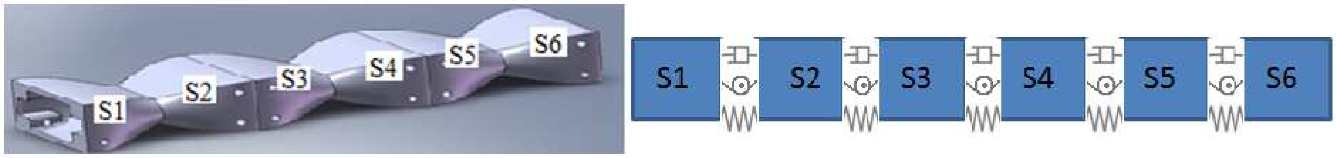

Figure 1: Illustration of the Lumped Parameter Model of a soft robot

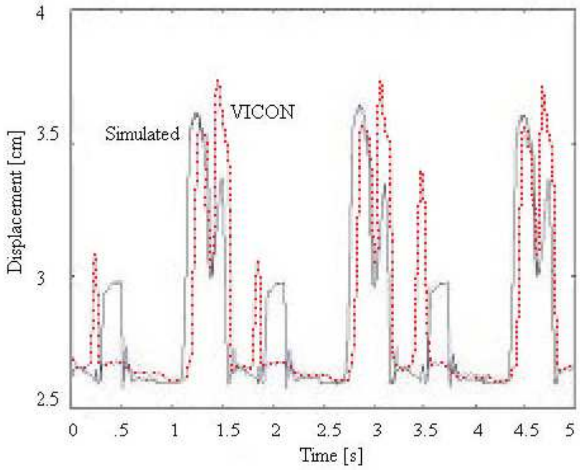

Figure 3: Simulated (predicted) vs Physical (actual) horizontal displacement for the evolved gait.

2.3 Validation of Evolved Results

Evolved control schemes were then validated on the physical pneumatic robot. Ten of the best genotypes in the final generation were chosen for evaluation. As a preliminary check, a qualitative visual comparison was performed. As predicted in simulation, all the tested gait patterns moved forward on the physical hardware as well

Three quantitative studies were then performed to assess the accuracy of the lumped dynamic model. The first study considered all ten gaits and compared travel distances achieved by the prototype hardware to those predicted by simulation. A second study compared motion capture measurements to the simulated gait kinematics for a trio of gaits. The final quantitative study considered the effect of gait duration on hardware performance for the same trio of representative gaits.

For the first quantitative study, which compared predicted to actual travel distances, the range of actual distance traveled was between 11 and 74 percent of the simulated distance, considering all outliers. The mean distance ratio for fifty tests was 35 percent. Mean values for individual gaits were close to this 35 percent level, demonstrating consistency in the error of the simulated travel distance.

The second quantitative study used motion capture data to enable a more resolved comparison of the kinematics for the actual and simulated robots. For this experiment, only three gait patterns were considered. A VICONl’ system was used to track six markers on the hardware. The markers were attached to the top of the robot prototype, with matching locations also tracked in the simulation. The point tracking experiments showed a close correlation between the

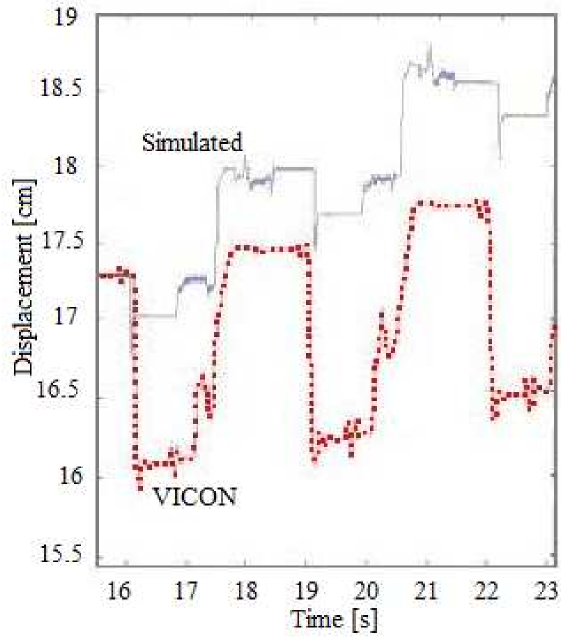

Figure 4: Simulated (predicted) vs Physical (actual) vertical displacement for an evolved gait.

timing of specific movements for the experimental and simulated pneumatic robot. Results for one gait pattern are illustrated in Figures 3 and 4. Similar results were obtained for all three gait patterns tested. Figure 4 plots vertical (y) displacement, and Figure 4 plots horizontal (x) displacement as a function of time. In the figures, the simulated motion is solid black while the hardwareÕs motion is a dashed line.

The final qualitative analysis evaluated the impact of gait period on robot speed for each of the evolved actuation patterns. These tests were performed to assess how closely the evolved actuation patterns were matched to the specific morphology of the hardware prototype. For these tests, only the overall duration of the gait period was altered. Actuators were always activated or deactivated at the same percentage gait period. In these tests, it was observed that the distance traveled by the robot hardware was adversely affected as the gait period deviated from the optimal evolved location. Figure 5 shows distances achieved per gait cycle as a function of gait duration. It is clear that the evolved gait period is optimal for each case tested.

The experimental results indicated that there is close coupling between evolved gait periods and the performance of the soft-robot hardware. The fact that robot distance traveled per gait period is maximized at a particular gait duration indicates that the motion is not quasi-static and that

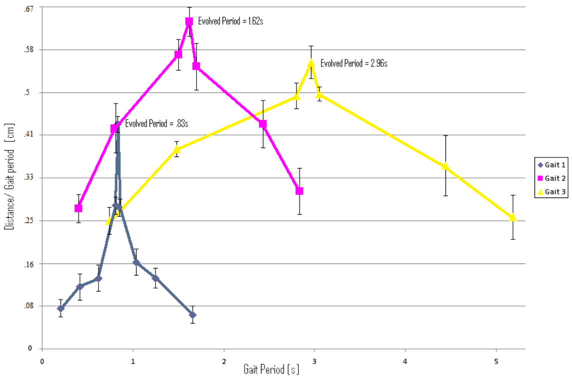

Figure 5: As the speed of the gait is sped up or slowed down relative to the evolved speed, the distance covered by the robot is significantly diminished. This suggests a tight coupling between the dynamics of the system and the evolved solution.

dynamics play an important role in gait performance. Moreover, the dramatic decrease in performance away from the evolved gait period, as shown in Figure 5, provides clear evidence that the lumped dynamic model simulation in PhysXl' captured the system dynamics well enough to enable effective gait optimization with a genetic algorithm. In this sense, the lumped dynamic model appears to be an effective method for modeling certain soft-robot gaits. The effectiveness of the lumped dynamic model in simulating the robot dynamics is also supported by the motion capture data, since the motion capture trends closely match the trends predicted by the simulation.

3. SOFT-BODIED MODELING

The lumped parameter modeling described above has achieved considerable results, but there is a hidden cost to the process, in that considerable human knowledge is required to hand-code the parameters of the reduced model. As designers develop new body shapes, new models must be derived from scratch. The alternative described in the section below is to directly model the entire soft-bodied robot, without an intermediate reduced-order stage.

PhysX is unique among off-the-shelf physics simulators for its ability to simulate cloth and soft bodies. This makes it particularly well suited for the direct simulation of soft bodies, without the intermediate step of hand-coding a reduced order lumped parameter model.

3.1 Quantification of Soft Bodied Characteristics

PhysX treats soft bodies as tetrahedral meshes with two principal variable characteristics: stiffness and damping. The details of these numbers are incredibly poorly documented, and so our first task was to attempt to characterize these numbers in real world terms. In order to to this we generated three soft bodied sheets with known dimensions, and performed stretching experiments by fixing the top edge of the sheet and attaching a weight to the bottom edge. By varying the stiffness and damping values we were able to a) affirm that simulated soft bodies behave quite similarly to their physical counterparts, and b) quantify the effect

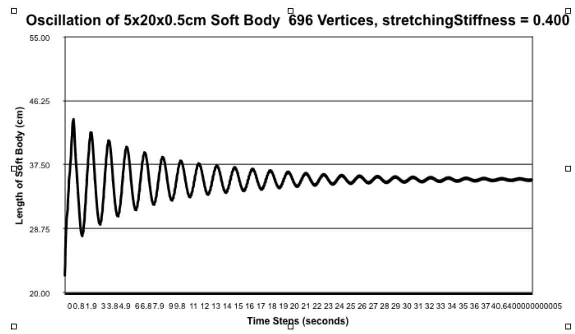

Figure 6: Change in length over time of a 0.5x5x20cm sample with a 10 gram weight hanging from it

of variable stiffness and damping ratios in order to compare them to known properties of actual materials. Figure 6 shows the change in length over time of a representative sample.

Figures 7 illustrates a linear response in the final resting length of the 0.5x5x20cm, 0.5x10x20cm and 0.5x5x30cm systems across a range of stiffness values between 0.0 and 1.0 and a range of vertex counts. As would be expected in a corresponding physical system, as the width of the strip doubles the change in length halves, and as the length of the strip increases by 50%, so does the change in length. Unlike the stretching results above, these show a slightly less coherent response.

The results above are comforting, in the sense that we have validated that the simulated system behaves as one expects a physical system to. To the extent that there are deviations of expected behavior, such as in the final subfigure of Figure 7, we suspect it is due to “boundary conditions” which occur at particularly low verex counts. We have yet to fully analyze the data for damping coefficients of the system, but are optimistic that similar behavior will be found. This latter will be especially important in the sense that we want our evolved gaits to be highly dynamic, and so fidelity not only to soft-bodied steady state behavior, but to dynamical responses will be highly critical.

3.2 Evolution of Soft-Bodied Gaits

Armed with these quantitative values, we can now with some confidence create a soft bodied simulation with physical characteristics similar to those of the silicone elastomer used to construct the physical robots.

To perform this evolution, a three-dimensional CAD design was created using SolidWorks and then converted into the OBJ/TET format required by PhysX. Stiffness values from the validion experiments above were then chosen to most closely model the characteristics of the silicone elastomer used to create the physical soft bodied robot.

The physical and simulated robots used for this experiment are shown in Figure 8. Unlike the pneumatic robot in the above section, this soft robot is actuated by a total of sixteen SMA “muslce wires” attached between adjecent segments - eight along the top and bottom of each side.

We chose to use Spiking Neural Networks (SNNs) as controllers, since unlike more conventional ANNs they are able to alter their timing and as a result capture and exploit

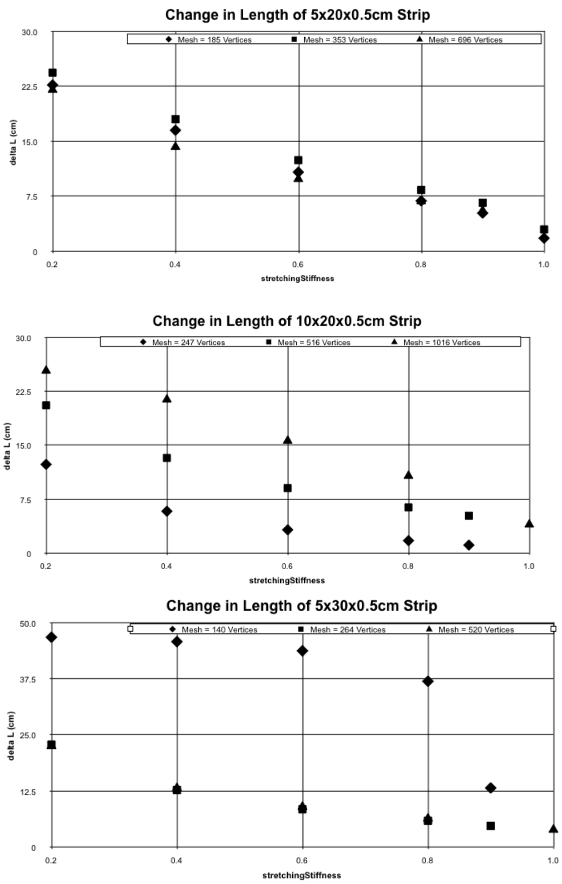

Figure 7: Change in length over varying soft body stiffness values for three different samples, all with a 10g weight.

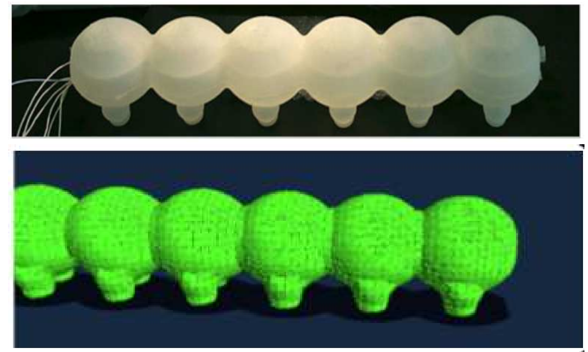

Figure 8: The physical robot (above), and the corresponding PhysX model. The model was produced by transforming a CAD file into the OBJ/TET format used by PhysX.

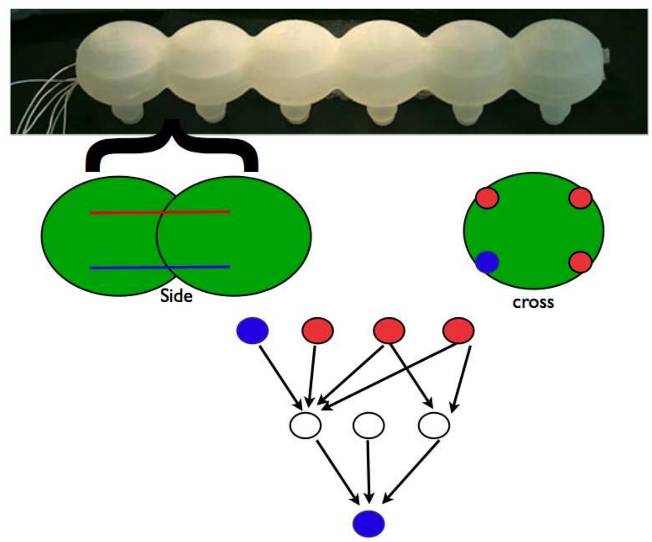

Figure 9: Schematic of Spiking Neural Network (SNN) organization in the simulated soft robot

dynamical properties of systems. Spiking neural networks (SNNs) were developed to model more continuous processes: input and outputs are both represented as single-value spikes (as opposed the sigmoid outputs of a conventional ANN) [4]. Instead of a sigmoid function, every SNN node contains a simple persistent counter, with adjustable offset and limit. At every time step, an SNN node sums its weighted inputs with the current counter value, and if the sum surpasses the limit the node fires a single "spike" to its output; otherwise the contents of the counter are decremented by a fixed decay rate, and persist until the next time step.

In an effort to achieve a non-centralized and highly dynamic gait akin to earlier SNN-based tensegrity robots [5], we also chose a distributed modular approach in which each muscle is controlled by a single independant SNN with only four inputs - corresponding to the length of the affected muscle as well as well as the three adjecent muscles in the intersegmental group, as illustrated by Figure 9. At every simulation time step, each muscle controller measures its inputs and feeds them through the SNN. Output spikes are converted into string actuations by measuring the duty cycle of network spikes. Any spike rate above 30% over a 100 step period is considered "active", and the corresponding muscle is ativated. Our choice of relatively simple binary actuation in this regard is an effort to simplify overall control, and to reduce the difficulty in translating simulated results into actuation values for the SMA wires.

The weights of all 16 SNNs on the robot where then encoded into single linear genotype for evolutionary manipulation. Genotypes were then evaluated by placing them within the simulated robot and measuring positive x-axis displacement of the leading edge of the robot in the over 3000 time steps. This is by some measures a challenging fitness function, since any solution which begins by contracting the leading segment muscles will pull the leading edge inwards, resulting in negative fitness.

A representative evolved gait is best seen on the author's web page (www.tufts.edu/ jrieff01/). It is worth noting that the evolved gait consists of both retrograde vertical waves, much like the caterpillar which we used as inspiration for this robot, as well as much less expected horizontal waves which cause bipedal motion in the leading segment. This bipedal motion emerges because we have placed to axis-symmetric constraints on the muscle controllers.

4. DISCUSSION

PhysX is a frustratingly cryptic system. The documentation for the product is sparse, and the online support is practically nonexistent. Nonetheless, its ability to simulate soft materials, combined with its support for hardware accelleration, make in a compelling platform for the simulation and evolution of soft bodied robots.

As mentioned earlier, the primary advantage of PhysX, aside from its ability to model soft materiels, lies in its ability to be accellerated thru GPGPU on high-end NVidia graphics cards. This accelleration has borne out in our experiments, and this has certainly been a boon to our evolutionary experiments. On a 2.8GHz Intel Xeon platform with 8GB of RAM and dual SLI NVidia 9800GT video cards, hardware acceleration provides a 10-50 fold increase in speed, depending upon the number of vertexes in the soft body. This translates into evolutionary runs which unfold over the course of a day or two rather than a week or two, leading to significantly faster turnaround times.

An incredible amount of work remains before we can be certain that PhysX is a viable tool for soft bodied evolutionary robotics, but the results are quite promising. We hope that in the work described above we have provided fellow researchers with some insight into the challenges and possibilities which can emerge from using GPGPU-enabled physics engines to perform evolutionary soft robotics research.

5. ADDITIONAL AUTHORS

6. REFERENCES

[1] J. Capadona, K. Shanmuganathan, D. J. Tyler, and S. Rowan. Stimuli-responsive polymer nanocomposites inspired by the sea cucumber dermis. *Science*, 319(7), 2008.

[2] J. Huang, C. W. P. Foo, and D. Kaplan. Biosynthesis and applications of silk-like and collagen-like proteins. *Polymer Reviews*, 2007.

[3] K. Kotay, D. Rus, M. Vona, and C. McGray. The self-reconfiguring robotic molecule. In *IEEE International Conference on Robotics and Automation*, 1998.

[4] W. Maass and C. M. Bishop. *Pulsed Neural Networks*. MIT Press, 1999.

[5] J. Rieffel, B. Trimmer, and H. Lipson. Mechanism as mind: what tensegrities and caterpillars can teach us about soft robots. In *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALIFE XI)*, 2008.

[6] M. Yim, D. Duff, and K. Roufas. Polybot: a modular reconfigurable robot. In *IEEE International Conference on Robotics*, 2000.