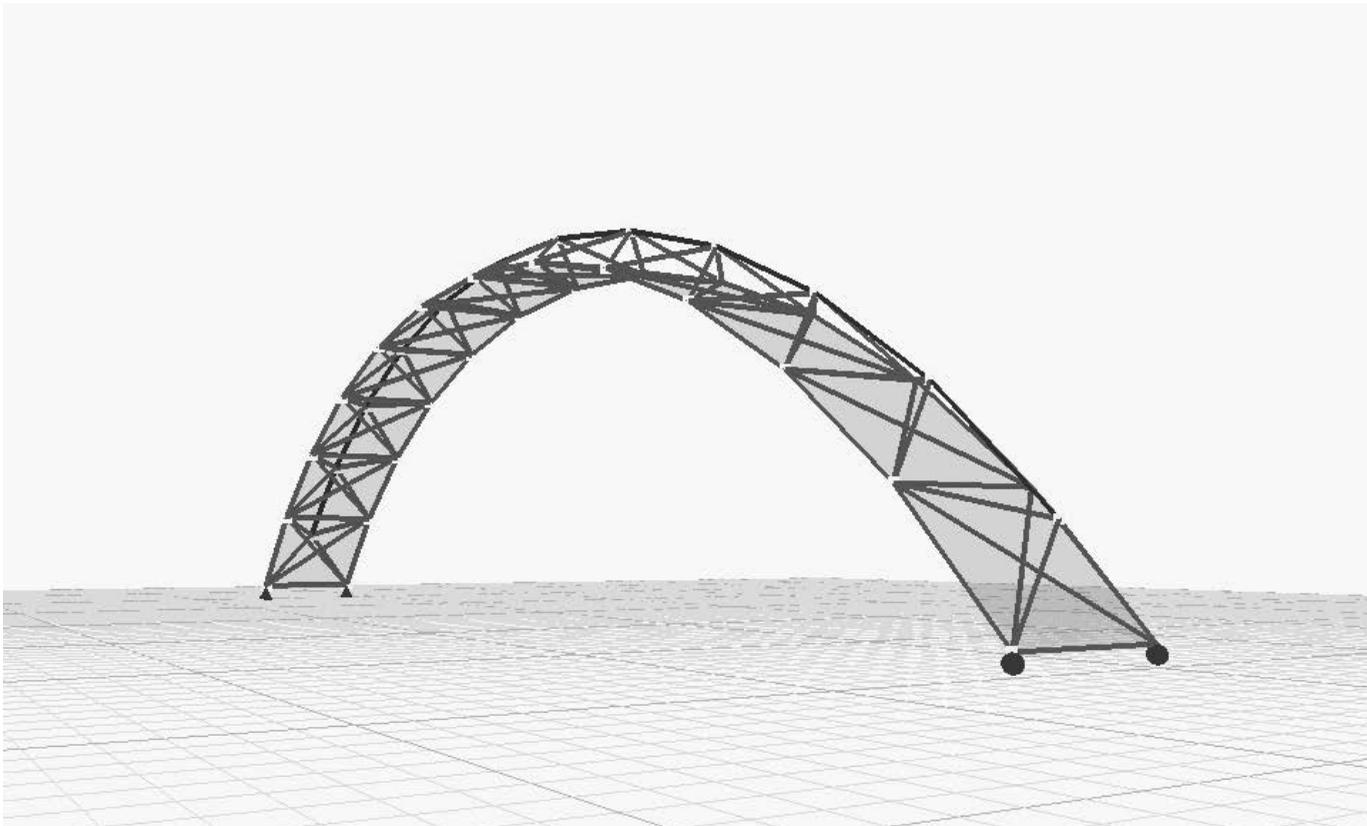


RESILIENT STRUCTURES THROUGH MACHINE LEARNING AND EVOLUTION

Ryan Mehanna Bartlett School of Graduate Studies, University College London



1 A series of actuated units combined as an adaptable truss

ABSTRACT

In the context of the growing usefulness of computation within architecture, structures face the potential for being conceived of as intelligent entities capable of resilient, adaptive behavior. Building on this idea, this work explores the use of machine learning for structures that may learn to autonomously “stand up”. The hypothesis is that a neural network with genetically optimized weights would be capable of teaching lightweight, flexible, and unanchored structures to self-rectify after falling, through their interactions with their environment. The experiment devises a physical and a simulated prototype. The machine-learning algorithm is implemented on the virtual model in a three-dimensional physics environment, and a solution emerges after a number of tests. The learned behavior is transferred to the physical prototype to test its performance in reality. This method succeeds in allowing the physical prototype to stand up. The findings of this process may have useful implications for developing embodied dynamic structures that are enabled with adaptive behavior.

INTRODUCTION

With the onset of kineticism in architecture, architects are interested in the notion of dynamic buildings that are able to solve problems. To do this autonomously, they are enabled with agency via sensing and actuation. A building that may perceive its environment, its users, and its own geometric and material configuration is an intriguing prospect. In this context, architects are looking to research in artificial intelligence and embodied mobile robotics for the methods and theories that may be useful to this new breed of architecture. Tristan Sterk has stated that buildings embedded with computation are indeed robotic artifacts (Sterk 2012). This architecture uses sensors to read its environment, and actuators to alter its physical properties in response to changes in its surroundings as well as human needs and desires.

The process sets out to explore the effectiveness of machine learning and digital evolution in resolving problems of uncertainty for dynamic structures. In a previous experiment, a physical prototype is built using a pair of pneumatically actuated scissor-mechanisms. Using a genetic algorithm (GA), the prototype autonomously generates—or teaches itself—the optimal actuation strategy for producing a walking gait that best fits its design. The behavior emerges without any instruction as to what combination of actions is required for locomotion (Mehanna and Sher 2012). However, the device is made of rigid metallic frames and cannot account for stability: when it falls during testing it requires manual assistance for correction. This is one of the factors that contributed to adopting flexible structures within this project to address the self-correction, or “getting back up”, problem. Contemporarily, it seems as though a desire to tread lightly, both in terms of carbon footprints and actual building footprints is taking hold. As such, this project is an exploration into lightweight, flexible, reconfigurable, and unanchored structures that are capable of both self-perception through embedded sensing and self-correction through embedded actuators, all using a combination of learning and evolution.

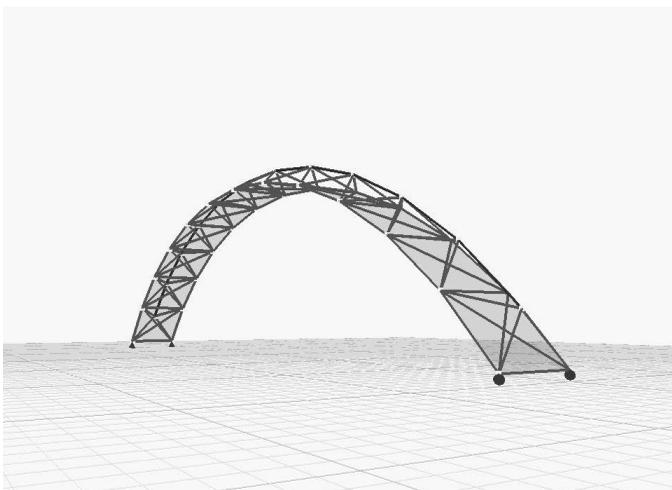
The physical model is built using a combination of rapid prototyping and a push-fit technique, and the simulated model is created using a particle-spring system approach. The central question is: how can machine learning be used to train flexible, lightweight and unanchored structures to stand up? Restoring and maintaining verticality for structures, as an act of active physical resilience, can be claimed to be an essential feature in the development of adaptive architecture. The task is to determine an appropriate actuation strategy for self-rectifying behavior, without any preset data or instruction. The hypothesis is that a neural network (NN) combined with a genetic algorithm (GA) can evolve the appropriate control system that would be required to enable the structure to learn from its sensory-motor interactions with its simulated physical environment in order to perform autonomous self-rectifying behavior upon being in a compromised position.

CONTEXT

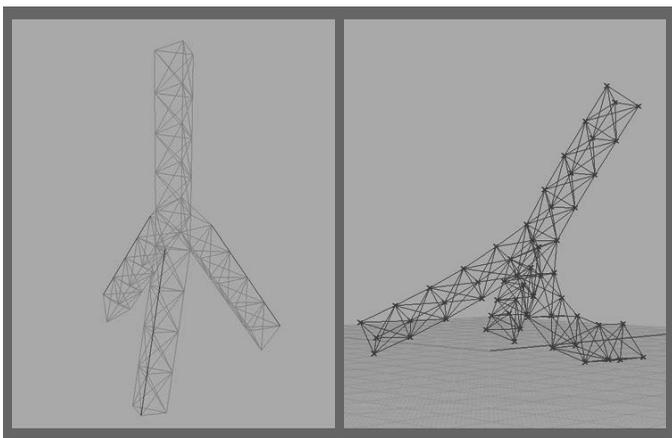
Sterk believes exploring computationally driven structures is imperative. By coupling the revived process-oriented interest, the computational-design attitude, and the physical tear-it-down approach, he envisions the new architecture to “sit at the edge of parametric design and robotics” (Sterk 2012, 159). Sterk defines Responsive Architecture as “those that employ sensing, control and actuation to effect persistent adaptation in buildings” (Sterk 2012, 156). He also points out that if there is doubt as to whether buildings ought to shape-shift or have moving parts, it’s due to practical and not conservative theoretical aims.

Sterk’s work addresses subjective as well as structural-environmental criteria. He employs actuated tensegrity assemblies to produce responsive prototypes which fulfill all three requirements for responsive architecture: “1) Controllable rigidity; 2) lightweight properties; and 3) they must be capable of undergoing asymmetrical deformations” (Sterk 2006: 3). As for control systems, Sterk suggests that Brooks’ “Subsumption Control” is useful due to its effectiveness in scaffolding several independent activities into a unified process. For Brooks, static representations of the world deter rather than aid intelligence; “the key observation is that the world is its own best model” (Brooks 1990: 3). Brooks advocates the “physical grounding hypothesis”. He points out that nature spent most of its time developing the rudimentary “essence of being and reacting” and afterwards, reason became possible (Brooks 1990: 3). Physically grounded systems are those that enjoy a coupling to their physical environment through sensation and action. Any effective forms of intelligence would be the emergent result of the system’s interactions with the physical world.

In *Resilient Machines Through Continuous Self-Modeling*, a novel approach for developing control systems is implemented directly on a physical robot (Lipson 2006). Lipson conveys that machines should have the robustness to operate autonomously and to survive in the real world, which is predominantly unstructured and chaotic. This kind of resilience, he explains, can be elicited once the machine possesses an idea about its own morphology, and that self-image in this case is made possible through motion. A self-model is entirely synthesized by the robot by selecting from a set of possible morphologies inferred from sensory readings that result from a number of randomly actuated movements. This self-model is then employed to produce a locomotion strategy in real-time, which also emerges without any pre-set instruction. This self-modeling is persistent, continuously updating itself based on the robot’s physical state. If any alteration, such as the severance of a limb occurs, the change instantly factors into the model, and the locomotion strategy is revised on the spot to adjust gait and compensate for the modification (Lipson 2006). This experiment is significant when thinking towards resilient structures. This notion of linking to the world via sensors and actuators



1 A series of actuated units combined as an adaptable truss.



2 Several actuated trusses combined to create the testing prototype

to cope with complexity is an elementary drive underlying the experiments within this research.

Nolfi and Floreano explain that behaviors emerge as a result of the interaction between an entity and its environment (Nolfi and Floreano, 2002); there is a great deal of complexity in nature as to how these interactions lead to adaptive behavior. Similarly, designing systems that rely on coordination between sensation and action is a challenging task. However, by combining learning with evolution, the control system of an artifact could autonomously develop through its interactions with the world (Nolfi and Floreano, 2002). Various combinations of NNs and GAs have been investigated (Montana 1995). Most commonly, GAs are used to find optimal synaptic weights within feed-forward NNs (Montana 1995). This is especially useful in the current scenario where there are no pre-defined input-to-output mappings required for training the network. GA's resolve this by optimizing the neural connections according to criteria external to the network (Harvey et al. 1997).

Montana points out that NNs with this kind of freedom are not limited to pattern classification and can be used in control systems for autonomous robots. In order to address the question of machine learning in unassisted self-rectification, a NN is the kind of structure needed to represent the nervous system for sensory-motor coordination, and as training data is unavailable, a GA is used with a general fitness rule.

METHODOLOGY

I) DESIGN

The objective of the experiment is to examine the idea of using machine learning to teach structures to autonomously self-rectify. First the testing device is designed, and then virtual and physical versions of it are constructed. A learning algorithm is then written and implemented on the virtual prototype. Once a result emerges in the simulation, the outputs for that behavior are tested as an actuation strategy for the physical structure. When allocating a design for the structure to be used, research into simulated actuated structures led to look into the Workshed project (Crowther, McCann, and Senatore 2011). The actuated truss becomes a template, which is adopted as the building block for the design of the testing device (Figure 1).

Kangaroo, a physics component under Grasshopper, is used to quickly test out several geometrical options. This facilitates determination of the kind of structural framework suitable for this experiment. When thinking towards any structure to be tested for stability, the primary force to be considered is the downward pull of its own weight. Being a subject immersed in this directional environment governed by gravity, it may then assume local vertical polarity—a “down” and an “up”; if the structure in question is not random, it has to assume a top and a bottom. This becomes the primary constraint over the shape of the testing device—to have a discernible top from bottom (Figure 2).

The top consists of a vertical mast. The bottom part is comprised of a base of three members acting as supports. Rather than being one monolithic element, the base is articulated as independent supports so that they may be actuated separately. The number of legs is chosen as the minimum required by a structure to achieve, or restore, vertical stability. The design also limits the ways in which the structure can fall. Configuring the legs around three axes of symmetry allows the structure to rest into the same position. Once the main elements of the structure have been defined—head, core, and three supports—they become the components that constitute both the physical and virtual prototype.

II) PHYSICAL PROTOTYPE

In order to build a lightweight, dynamic, and reconfigurable structure, carbon fiber is chosen to create the primary elements for its flexibility, lightness and durability (Figure 3). 3D printed nylon parts constitute the pylons and core. Tension wire is used to alter the

geometry. The core houses three multi-turn servos, which pull and release the tension wire upon actuation. The entire prototype can be assembled in minutes as it uses simple push-fit connections. This allows for experimenting with alternate geometries. For instance, it is possible to try out various leg designs by changing number and position of pylons or by altering the length of the carbon fiber element; the same applies to the mast. This makes the structure completely reconfigurable and permits potentially expanding the experiment into various scenarios.

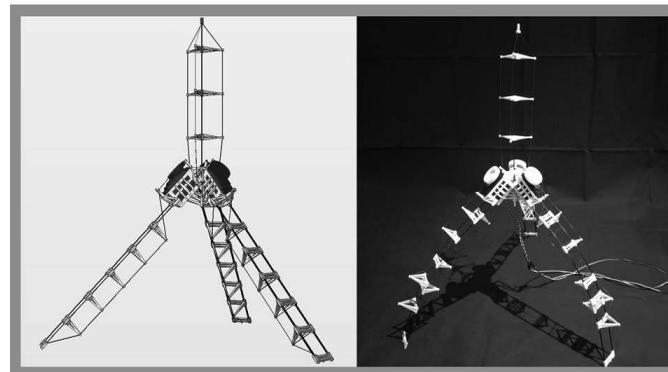
As general guidelines, the device is intended to address the three criteria for adaptive structures suggested by Sterk: controllable rigidity, lightweight properties, and capable of undergoing asymmetric deformations (Sterk, 2006). Rigidity is altered when servos are actuated; the carbon fiber and nylon materials provide flexibility, as well as lightweight properties; and when servos are at different positions, the overall deformation becomes asymmetrical.

III) SIMULATION

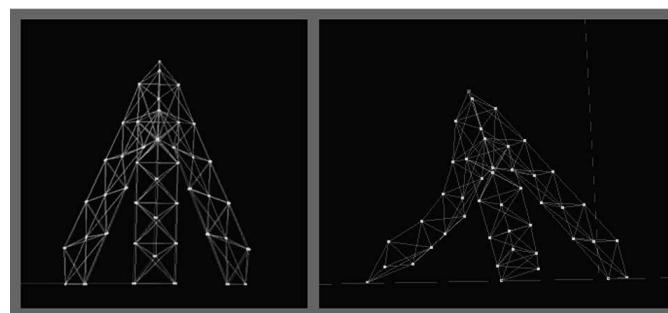
A particle system is used to build the digital model in Processing (Figure 4). The three-dimensional physics environment is based on Curtain, a two-dimensional cloth simulator using a particle system with verlet integration created by Jared Counts (Counts, 2011). The algorithm is augmented to operate in three-dimensions by transposing all functions from two to three coordinates. The node and strut classes are based on the particle and link classes of Curtain, and are also adapted to three-dimensions. An actuator class is created in order to simulate the testing device. The strut class is modified so that instead of using each linear element's constant rest length to calculate its constraints on two particles, a dynamic variable called current length is introduced. Current length changes with actuation factor, thus turning the strut into an actuator.

IV) THE MACHINE LEARNING ALGORITHM

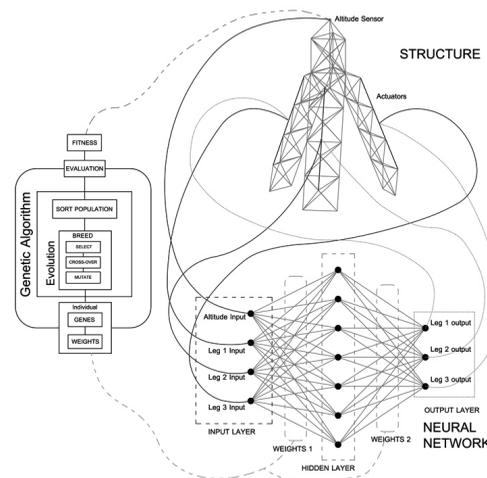
The NN is based on Alasdair Turner's Artificial Neural Network program created to recognize handwritten digits. It uses back-propagation to train the network weights, which relies on a database of handwritten digits, each accompanied by a corresponding label. This network operates in isolation, and in order to become the nervous system embodied within a physical structure, it needs to be coupled directly to the world through this structure's sensory motor affordances. In order to embed the network into the virtual model, its input and output neurons are directly linked to certain nodes and to the actuators within the structure. Though this experiment involves testing around one scenario and simulation, in principle, interaction with the world would allow the network to learn to adapt the structure to any task permitted by its geometry, actuator limits and material properties. If data sets were manually encoded to achieve such behavior using back propagation, they would have to encompass a laborious amount of mapping information that would defeat the purpose of this exercise. The aim



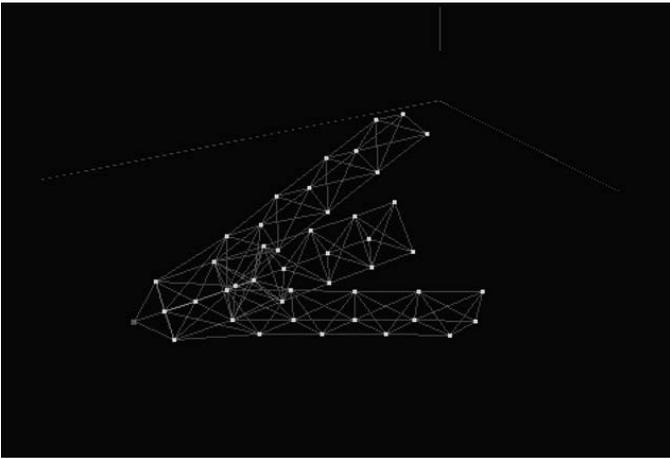
3 Physical prototype



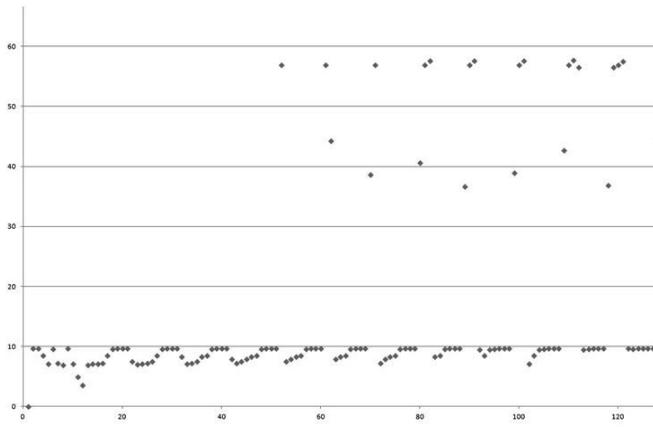
4 Virtual prototype



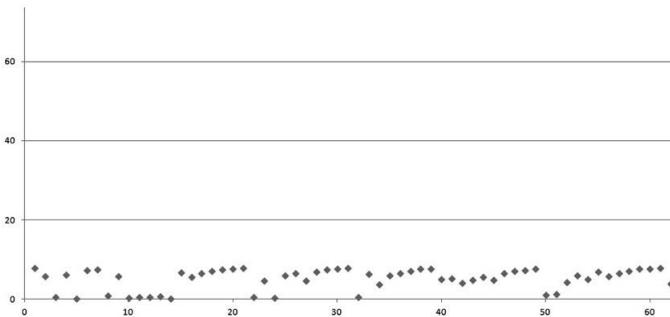
5 Overall algorithmic setup



6 Virtual structure in fallen position



7 Control experiment results



8 First output strategy results

is to teach this structure with as little instruction as possible, and this is where the role of the GA comes in (Figure 5).

The network is comprised of three layers: input, hidden, and output layers consecutively. The input layer neurons act as the receptors. They gather data about the structure's physical status through functions written to act as sensors. Two input strategies are tested.

First Input Strategy: the overall height is measured using the node at the apex of the structure, and is fed into one input neuron. The current length of its actuators indicates the position of each leg. In total, four input neurons are used (Figure 5).

Second Input Strategy: data on the structure's orientation in space is approximated by measuring the altitude of three nodes that are at the same level when the structure is upright. The data is fed into three input neurons, in addition to the same leg inputs included in the previous method. This input layer is made of six neurons.

The output layer has three neurons each corresponding to a linear array of actuators in one leg. Two different output strategies are tested:

First Output Strategy: the first strategy treats the network's output data as rates of actuation; different output values produce different speeds of motion applied consistently for the entire duration of one test.

Second Output Strategy: the output data is conveyed as amounts of actuation, resulting in different durations of actuation, and therefore different leg positions.

The optimization is based on the Genetic Algorithm program by Alasdair Turner (Turner, 2009), which is designed to evolve three-dimensional boxes that have the maximum sum of edge length while having the smallest volume. For the current experiment, and as a means for training the network in the place of back propagation, this GA is adapted to evolve the synaptic weights using a general fitness rule that represents a simple and direct test of competence—a target height in space equal to the height of the structure when perfectly upright. As this GA is geared towards weight optimization, the genotype encodes real numbers within the range of the possible values that the weights of the network's synaptic connections may have. The fitness rule records the height achieved by each individual using the same node as that for altitude input in the NN. This is a direct indication of the success of the each individual—the closer to the standing up position at the end of each test, the higher they will be ranked in the population.

Within the first generation of the evolution sequence, ten individuals are created with random genes. These are stored in the population class. The primary function within the population class is evolution. Its sub-methods are population sorting, selection, and breeding. The population is sorted from weakest to fittest. Selection chooses parents randomly with a bias toward the fitter

side of the spectrum. This prevents the evolution from converging prematurely to a local optimum. Breeding consists of crossover and mutation. Crossover takes the genes of the selected parents and merges them at random. Mutation takes the new genotype and performs a 5 percent chanced randomization. As an opposite to the convergent pull of selection, the mutate operator occasionally introduces random variations to the gene pool, widening the search and avoiding any premature convergence to a non-solution. For the control experiment, a GA is built separately. This is to be compared to the combined algorithms. The genes directly encode values that would translate as amounts of actuation.

RESULTS

Note that for all experiments, individuals have identical starting conditions (Figure 6). Fitness is mapped onto a percentage, with 100 percent representing a height of 260 pixels, which is the height of the structure when upright.

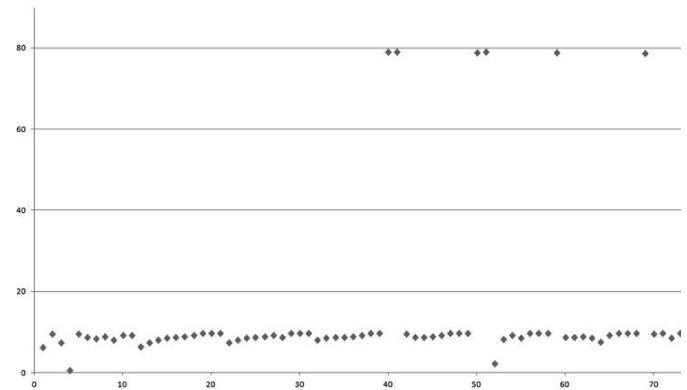
Experiment 1 / Control Experiment—GA: after running for approximately four generations, the first solution appears. Around the eighth generation, more individuals are standing up. By the fourteenth generation, approximately half the population is standing up. Although these structures somewhat stand, they remain within a 60 percent margin (Figure 7).

Experiment 2 / First Output Strategy: in the second experiment, the combined algorithm is applied using the first output strategy. It runs for six generations without any signs of improvement. As the fitness graph above shows, none of the individuals make it past the 10 percent margin (Figure 8).

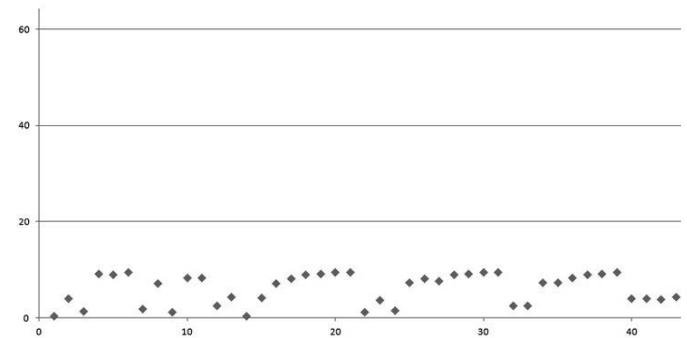
Experiment 3 / Second Output Strategy with First Input Strategy: in the third experiment, the second output strategy is employed along with the first input strategy. A solution emerges within the third generation. Not much progress takes place afterwards, but the difference with those that stood up using the GA is that in this case individuals reach an 80 percent stance (Figure 9), which is closer to the target, and they appear earlier by one generation.

Experiment 4 / Leg Input Removed: in order to identify whether the three leg inputs are contributing to the behavior, in this test they are removed, leaving only the altitude measure as input. This experiment produces no solutions, and the fitness levels do not exceed the 10 percent margin (Figure 10) This confirms the leg input's contribution to the desired behavior.

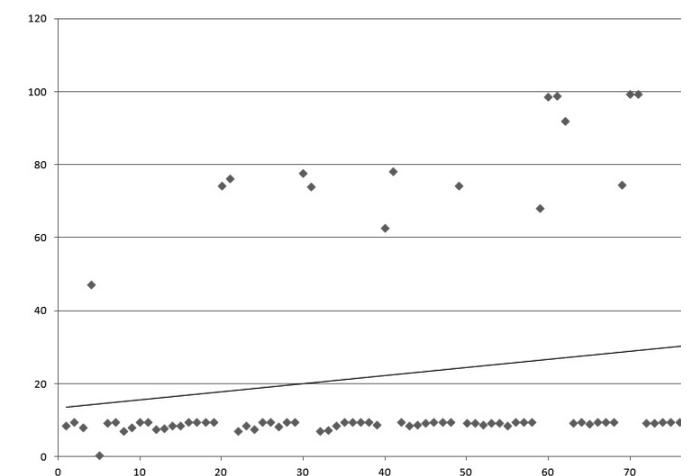
Experiment 5 / Second Output Strategy with Second Input Strategy: finally in the full experiment, applying both the second output and second input strategies allows a semi-solution to appear within the first generation before any evolution. However, it is then forgotten for an entire generation. At the beginning of the third generation, immediate progress is recorded at 80%, and by the sixth generation the 100 percent target is reached



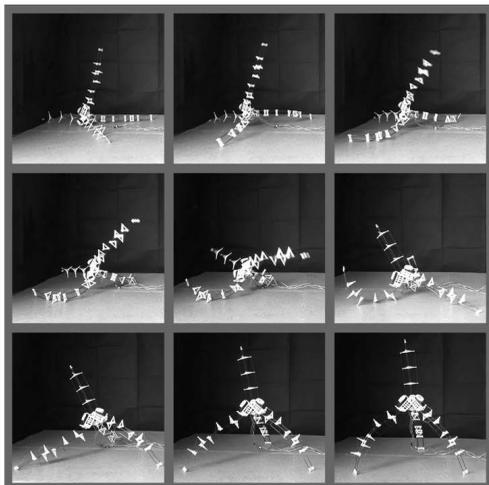
9 Results of combining second output strategy with first input strategy



10 Results when sensation in the legs is removed



11 Results of combining second output with second input strategies



12 Physical standing sequence

for the first time, making this setup a clear improvement over the preceding tests (Figure 11).

Experiment 6 / Physical Prototype Behavior: the network output values of the best individual from the previous experiment are recorded and implemented as a pre-set actuation strategy for the physical prototype (Figure 12). The structure stands up.

DISCUSSION

A standard feed-forward NN limits behavior in the temporal sense. Ideally the net would have internal feedback, which may allow it to exhibit more complex behavior with respect to time. However, actuating in amounts rather than speeds gives this network's behavior a temporal quality. Any combination of values given to the legs will allow them to operate for different amounts of time within one behavioral sequence. This is a key element in producing the solution to the problem at hand. Tracking the altitude of one node at the tip of the structure is enough input for the system to eventually reach a solution. However, when altitudes of three nodes are tracked, this gives the network a sense of orientation, and as a result, produces a solution more promptly. Further, this strategy combined with the second output strategy not only produces a solution quicker than other tests, but also provides more successful standing postures reaching the 100 percent target. As the control, the GA alone is able to produce the solution, but after twice the number of generations it takes when using the primary algorithm. In addition, the fitness produced by the GA does not exceed 60 percent, whereas the primary algorithm reaches the 100 percent target.

In the physical experiment, when the winning outputs are fed to the actuators of the prototype, the behavior is remarkably similar to that of the virtual prototype. The resulting actions succeed in enabling the physical structure to pick itself up and stand fully. Across the digital and physical aspects of this research, there

is a dynamic interdependence fundamental to both the design process and the experiment. The fact that such a scenario with ambiguous or minimal starting information can be addressed within simulation, and its results transferred to physical reality suggests that such an approach may be efficient in terms of time and resource requirements for even more complex situations. Rather than building each version of the design physically in order to test its properties, the use of parametric software enabled with physics simulation expedites the design process by allowing for accurate simulated testing. In terms of developing the behavior, rather than conducting the entire learning sequence on the physical device, an appropriate behavioral strategy is exploited digitally. This saves on having to perform multiple runs, and potentially consuming more energy and materials for physical parts that may wear down. The digital tools of evolution and learning enable the control system to be embodied within the physical artifact by having its behavior generated solely by the potential structural deformation that the geometry permits. With this approach, the object designs its own behavior. The distribution of the values for the synaptic weights within the winning individual becomes the mapping between inputs and outputs which is usually unavailable for complex problems. This data implicitly represents the appropriate relationship between those inputs and outputs within the context of the problem of standing up.

FURTHER WORK

The following step is to implant the evolved network into the physical prototype by embedding sensors for height and orientation data and linking the network outputs to the servos. It would be informative to test whether this generates similar behavior to the model in simulation, and whether having connected to the real world for the first time, the same NN would be able to continue learning and further tune its responses for real physics. Recurrence will be introduced to the network structure by building additional hidden layers and creating feedback loops among them with evolvable time delays on their synapses. This will turn the system into a dynamically recurrent NN. More internal states would enable the system to perform more sophisticated sequences of movements, and may produce solutions to the problem of the self-rectifying structure when having an asymmetrical and a more complex morphology.

CONCLUSION

Experimenting with new computational tools is essential for fulfilling the emerging requirements and desires imposed on the discipline of architecture. These are in turn influenced by the digitization of physical matter itself through technologies like rapid prototyping, automated fabrication, and potentially even automated construction processes. Architects look to artificial intelligence and embodied robotics for methodological and theoretical guide-

lines. As machine learning produces positive outcomes within design conditions of uncertainty, its application to physical structures needs to be vigorously investigated. The work questions whether machine learning may allow a fallen structure to self-rectify without assistance or instruction. The hypothetical claim is that an evolved neural network may allow the structure to learn to stand up through its interactions with its environment.

An experiment is conducted using both a digital and physical prototype. The algorithm is implemented on the digital structure using various combinations of input and output strategies. Methods that provide temporal complexity are the most successful, and more complex couplings to the environment, such as the use of three instead of one altitude measure, also generate better

results. This could reflect the physical grounding hypothesis, as the findings suggest that the more vivid the perception of reality, the more likely successful adaptation is to occur. This is also reflected by testing with and without sensation in the legs. The winning output values are also capable of rectifying the physical prototype; such a coupling between digital and physical aspects within the process is valuable when thinking towards designing structures with dynamic output. Simulation techniques are becoming increasingly robust and accurate: when combined with machine learning and evolution, along with physical experimentation through rapid prototyping, they present the opportunity to investigate the kinds of behaviors that may become essential in developing adaptive architecture.

ACKNOWLEDGEMENTS

Thanks to Sean Hanna, Ruairi Glynn, Martha Tsigkari, Angelos Chronis, Maria Eleni Skavara, and the Said Foundation.

WORKS CITED

Mehanna, Ryan and Elite Sher. 2012. *An Exploration into Computational Optimization for Motive Architecture*. Paper presented at the ACADIA Synthetic Digital Ecologies conference, San Francisco, California. October 18-21.

Beer, Randall and John C. Gallagher. 1992. *Evolving Dynamical Neural Networks for Adaptive Behavior*. *Adaptive Behavior* 1:91-122.

Brooks, Rodney. 1990. *Elephants Don't Play Chess*. *Robotics and Autonomous Systems* 6:3-15.

Brooks, Rodney. 1991. *Intelligence Without Representation*. *Artificial Intelligence* 47:139-159.

Counts, Jared. 2011. *Curtain. Open Processing Sketch*. Last accessed September 08, 2012. <http://www.openprocessing.org/sketch/20140>

Counts, Jared. 2012. *Simulate Tearable Cloth and Ragdolls With Simple Verlet Integration*. Online tutorial. Last accessed July 27, 2013. <http://gamedev.tutsplus.com/tutorials/implementation/simulate-fabric-and-ragdolls-with-simple-verlet-integration/>

Crowther, Angela, Ed McCann, and Gennaro Senatore. 2011. *Expedition Workshed: Engineering Education for the 21st century*. Paper presented at the 35th Annual Symposium of IABSE / 52nd Annual Symposium of IASS / 6th International Conference on Space Structures, London, September.

Harvey, Inman, Phil Husbands, Dave Cliff, Adrian Thompson, Nick Jakobi. 1997. *Evolutionary Robotics: the Sussex Approach*. *Robotics and Autonomous Systems* 20, 2-4:205-224.

Lipson, Hod, Victor Zykov, Josh Bongard. 2006. *Resilient Machines Through Continuous Self Modelling*. *Science* 314:1118-1121.

Montana, David J. 1995. *Neural Network Weight Selection Using Genetic*

Algorithms, in *Intelligent Hybrid Systems*, edited by Suran Goonatilake and Sukhdev Khebbal, 85-105. Sussex: John Wiley & Sons.

Nolfi, Stefano and Dario Floreano. 2002. *Synthesis of Autonomous Robots Through Evolution*. *Trends in Cognitive Sciences*, 6, 1:31-37.

Nolfi, Stefano and Domenico Parisi. 2003. *Evolution of Artificial Neural Networks*, in *The Handbook of Brain Theory and Neural Networks*, 2nd Edition, edited by Michael A. Arbib, 418-421. Cambridge, Massachusetts: MIT Press.

Sterk, Tristan d'Estrée. 2006. *Shape Control in Responsive Architectural Structures: Current Reasons & Challenges*. Paper presented at the 4th World Conference on Structural Control and Monitoring, San Diego, California, July 11-13.

Sterk, Tristan d'Estrée. 2012. *Beneficial Change: The Case for Robotics in Architecture*, in *Persistent Modelling: Extending the Role of Architectural Representation*, edited by Phil Ayres, 155-170. London: Routledge.

Turner, Alasdair. 2009. *Artificial Neural Network*. Open Processing Sketch. Last accessed September 08, 2012: <http://www.openprocessing.org/sketch/2292>

Turner, Alasdair. 2009. *Genetic Algorithm*. Open Processing Sketch. Last accessed September 08, 2012: <http://www.openprocessing.org/sketch/3101>

RYAN MEHANNA is a licensed, practicing architect.

He received a Bachelor of Architecture degree in 2004 from the American University of Beirut, and has since worked in various design firms. Ryan completed the MSc Adaptive Architecture and Computation program at the Bartlett School of Graduate Studies at UCL in 2012. He is currently interested in machine learning, genetic optimization and machine perception with the intention of transforming passive kinetic structures into adaptive architectural entities that can perceive and respond to their environments and users through physical sensing and actuation.