Ribosomal Robots: Evolved Designs Inspired by Protein Folding

Sebastian Risi, Daniel Cellucci and Hod Lipson
Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, USA 14853
{sebastian.risi, dwc238, hod.lipson}@cornell.edu

ABSTRACT

The biological process of ribosomal assembly is one of the most versatile systems in nature. With only a few small building blocks, this natural process is capable of synthesizing the multitude of complex chemicals that form the basis of all organic life. This paper presents a robotics design and manufacturing scheme which seeks to capture some of the versatility of the ribosomal process. In this scheme, a custom "printer" folds a long ribbon of material in which control elements such as motors have been embedded into a morphology that is capable of accomplishing a pre-defined task. The evolved folding patterns are encoded with a special kind of *compositional pattern producing network* (CPPN), which can compactly describe patterns with regularities such as symmetry, repetition, and repetition with variation. This paper tests the efficacy of this design scheme and the effects of different ribbon lengths on the ability to produce walking robot morphologies. We show that a *single strip of material* can be folded into a *variety of different morphologies* displaying different forms of locomotion. Thus, the results presented here suggest a promising new method for the automated design and manufacturing of robotic systems.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – connectionism and neural nets

General Terms

Algorithms

Keywords

Design Automation, Robotics, Neuroevolution, NEAT, Compositional Pattern Producing Networks

1. INTRODUCTION

Ribosomal assembly in biological systems enables the manufacturing of compounds that display unparalleled diversity and utility [10]. Using only a few basic building blocks, this natural process enables the construction of the multitude of varied, complex chemicals that form the basis of all organic life. However, the immense potential of this design and manufacturing scheme has not yet been realized artificially; most robots are still manually designed and constructed, which can be a difficult and time-consuming task.

While recent advances in 3D printing have allowed robotic prototypes to be produced with greater ease and less expense [9, 11], the 3D printing of active systems such as motors and batteries has achieved only limited success. The major goal of a robot "walking out of the printer" is therefore not realizable with current hardware and software. Additionally, current 3D printed technology is not yet adapted to handling *recycled material*. Biological matter, on the other hand, can be recycled within an ecosystem because the same molecular building blocks are common to all live forms.

To address these challenges, this paper introduces a ribosomal-inspired assembly process which begins with a strip of *pre-manufactured source material* that has been augmented with actuators and other control elements and folds this ma-

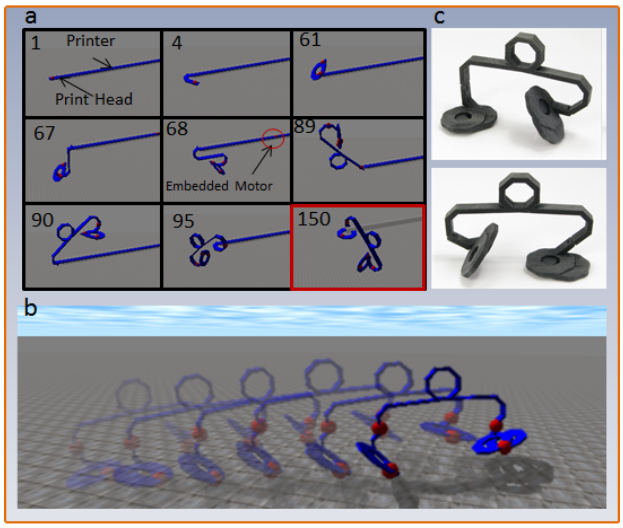


Figure 1: Ribosomal Manufacturing Example. (a) Printing the robot involves running a pre-assembled length of material through the printer, and bending it into a useful configuration. (b) A composite image shows the resulting folded bipedal robot walking from left to right. (c) A non-actuated version of the evolved model was transferred to reality via a 3D printer.

GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands.

terial into a useful configuration by a custom "printer" (Figure 1).

An important part of this process is that it allows the act of *assembly* to be decoupled from the act of *construction*. Since the ribbon of material comes pre-assembled, construction merely involves folding the material into the desired shape, and since different combination of mutations can be applied to the ribbon, one pre-assembled strip can produce a multiplicity of viable forms.

To create functioning robots out of a single piece of ribbon with hundreds of parts requires complex folding patterns. These folding patterns are genetically encoded by a special kind of compositional pattern producing network (CPPN; Stanley 2007), which can compactly describe patterns with regularities such as symmetry, repetition, and repetition with variation [13, 14, 15].

This paper tests the idea of ribosomal-inspired assembly in a simple, simulated locomotion experiment. In this task, three-dimensional robots are first folded out of a one-dimensional piece of ribbon and must then travel as far as possible in a limited amount of time.

As the experiments in this paper will show, it is indeed possible to evolve a variety of different robots – all created from the same source material – that display distinct forms of locomotion. Additionally, the CPPN encoding can effectively transform a one-dimensional structure into a three-dimensional actuated robot with regularities like symmetry and repetition. Finally, this paper analyzes different ribbon configurations and shows that there exists a complex relationship between the (1) number of segments, (2) number of actuators, and (3) the way the evolutionary algorithm handles designs that collide during the folding process.

The results presented here are significant for research in artificial life and evolutionary robotics because the fabrication platform can act as the foundation of an entire robotic ecology. Since any robot generated with this manufacturing scheme will share the same material, interactions between competing members, consumption and recycling of detritus (unsuccessful designs), and constructions that involve multiple interweaving ribbons of material are possible.

The paper begins with a review of ribosomal-inspired assembly methods, CPPNs, and NEAT in the next section. The ribosomal robot approach is then detailed in Section 3. Sections 4 and 5 describe the experimental setup and present results in a simulated locomotion domain. Section 6 then discusses the implications of these results and outlines future work, followed by conclusions in Section 7.

2. BACKGROUND AND RELATED WORK

This section reviews related work on ribosomal-inspired assembly and also the technologies that enable the ribosomal-robot encoding introduced in this paper.

2.1 Ribosomal-Inspired Assembly

Previous examples of ribosomal-inspired assembly include the thesis work of Griffith [5], which involved the construction of two-dimensional structures using sequences of specially manufactured square tiles. These square tiles were connected to one another with hinges at each of the corners, forming a flexible sequence of structural elements that could bend into different configurations. Assembly involved pushing a sequence of tiles out of a narrow channel, where different patterns of magnets on the tiles dictated the direction that the hinge connecting pairs of elements would bend. The magnets caused the structure to fold into specific configurations depending on the order of these magnet patterns, allowing the formation of complex two-dimensional shapes. This work was extended with three-dimensional chains, and showed that such chains could approximate any three dimensional shape [1].

Additionally, researchers such as Onal et al. [12] generated three-dimensional robotic structures from flat, two-dimensional planes through the use of laser-engraved origami patterns. This work demonstrated that one two-dimensional sheet of specially prepared material could generate multiple unique forms including those resembling a plane and a boat. This was accomplished through the creation of a series of panels containing hinges in a box-pleat pattern. The elements of the pattern could then bend along these hinges using shape-memory alloy, producing a robot that could switch between forms depending on the environment.

Finally, the most recent implementation of assembly via folding is the Milli-Motein, a product of the work of Knaian et al. [6]. This chain of novel electropermanent wobble motors demonstrates the ability to dynamically fold into multiple distinct configurations, and the unique design of the wobble motor allows the system to hold a chosen position indefinitely without power.

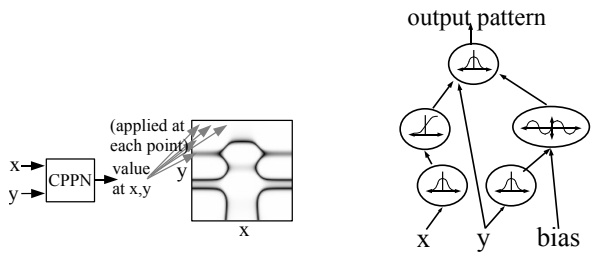
While all of these previous works demonstrate the potential of this avenue of inquiry, each has notable limitations. In the first case, the tiles contained no actuation other than the magnets that enabled assembly, so therefore the final structure was static. In the second case, the proof-of-concept and theoretical potential of the structure was demonstrated, but the final product was limited by the resolution of the pattern that could be implemented. In the final case, the desire to combine the elements of both assembly and locomotion into one system produced a platform that could be effectively miniaturized, but involved a system of highly complex motors which was limited in length by the ability of an individual motor to lift multiple segments.

The advantage of the method introduced in this paper is that it enables larger and more complex structures, since the material that forms the robot's structure can be static, but without losing a large amount of the versatility, since segments can be recycled back into the system and reprinted to address different environmental objectives.

2.2 Compositional Pattern Producing Networks (CPPNs)

Capturing the structural motifs that appear in ribosomal synthesis requires a robust representation capable of encapsulating the phenomena of symmetry, repetition, and variation. Several studies have demonstrated the ability of CPPNs to represent patterns with motifs reminiscent of those seen in natural organisms [13, 14, 15]. To this end, this paper employs CPPNs for the encoding of the fold patterns that represent a morphology. CPPNs are a variation of artificial neural networks (ANNs) that differ in the set of possible activation functions and the manner in which these functions are applied [15]. While CPPNs are similar to ANNs, there are vital differences between the two types of networks that will be described in this section.

Patterns in nature can be described, at a high level, as compositions of functions, wherein each function in the composition represents a stage in development. CPPNs employ



(a) Mapping (b) Composition

Figure 2: CPPN Encoding. (a) The function f takes arguments x and y, which are coordinates in a two-dimensional space. When all the coordinates are drawn with an intensity corresponding to the output of f, the result is a spatial pattern, which can be viewed as a phenotype whose genotype is f. (b) The CPPN is a graph that determines which functions are connected. The connections are weighted such that the output of a function is multiplied by the weight of its outgoing connection.

a connectionist framework similarly to ANNs, but they allow nodes to posses a unique activation function. Each of these functions represents a common regularity, and together their synthesis can produce results of nearly arbitrary complexity.

The indirect-CPPN encoding can compactly encode patterns with regularities such as symmetry, repetition, and repetition with variation [13, 14, 15]. For example, a symmetric function such as a Gaussian curve can create an output pattern that is symmetric, and a periodic function such as sine is able to create segmentation through repetition.

Most importantly, *repetition with variation* (e.g. such as the fingers of the human hand) is attainable by combining regular coordinate frames (e.g. sine and Gaussian) with irregular ones (e.g. the asymmetric x-axis). For example, a function that takes as its input the sum of a symmetric function and an asymmetric function outputs a pattern with imperfect symmetry. In this way, CPPNs produce regular patterns with subtle variations.

Specifically, CPPNs produce a phenotype that is a function of n dimensions, where n is the number of dimensions in physical space. For each coordinate in that space, its level of expression is an output of the function that encodes this phenotype. Figure 2 shows how a two-dimensional phenotype can be generated by a function of two parameters that is represented by a network of composed functions.

Instead of two-dimensional images the approach in this paper evolves the folding pattern of a piece of ribbon encoded by CPPNs with the NEAT algorithm, reviewed next.

2.3 Neuroevolution of Augmenting Topologies (NEAT)

The next step in the implementation of the design process is a method by which the CPPN-encoded robotic morphologies can be optimized for specific tasks. To this end, NEAT [16, 17] is a neuroevolution method which can evolve ANNs (and therefore also CPPNs) with an evolutionary algorithm. It begins with a population of simple networks and then *complexifies* them over generations by adding new nodes and connections through mutation. By evolving networks in this way, the topology of the network does not need to be known *a priori*; NEAT searches through increasingly elaborate networks to find a suitable level of complexity.

Because it starts simply and gradually adds complexity, NEAT tends to find solution networks close to the minimum size necessary for a given system, and can evolve *both* the topology and the weights of a network. This process of complexification, which resembles how genes are added over the course of natural evolution, allows NEAT to establish high-level features early in evolution and then later elaborate on them. In this paper, robots whose structures are encoded by CPPNs are evolved by NEAT.

For a complete overview of NEAT see Stanley and Miikkulainen [16, 17]. The next section introduces modifications to the general CPPN representation that allows them to encode the folding pattern that turns a one-dimensional ribbon into a three-dimensional actuated robot.

3. APPROACH: RIBOSOMAL ROBOTS

Individual ribbons that form the basis of the robots manufactured with the *ribosomal manufacturing scheme* consist of segmented lengths of foldable material in which motors and other control structures have been embedded. Printing the robot involves running a pre-assembled length of this material through an external printer, which uses a set of molds to bend the mutable portions of the ribbon into one of five possible configurations: four 45-degree bends in the upward, downward, leftward, and rightward directions, and no bend. Each fold involves an equal amount of material, producing a digital structure composed of (unbent) control elements connected with differently shaped but equivalent lengths of mutable material.

Because an individual strip of material can take nearly any shape simply by varying the pattern of folds separating the control elements, a multiplicity of robotic morphologies can coexist on a single, generic length of material. In this way, the printer can use a minimal length of material to produce custom robotic designs tailored for specific design criteria, simply by altering the sequence of folds imposed by the printer. Finally, this manufacturing scheme has the potential to recycle robots that have accomplished their function by flattening the material and drawing it back into the printer. This recycled material can then be refolded into new designs when the need arises.

Once the folding of the robot is complete the structure is disengaged from the printer and its motors can begin actuating. In this paper, the actuators behave as motorized hinge joints (rotating around the in-plane axis perpendicular to the feed direction) and are uniformly distributed along the ribbon. Because the focus of this study is to isolate the effects of evolving different morphologies (instead of examining the effects of different control structures), each hinge joint simply changes its angle θ according to a modified sine wave function: $\theta(t) = 5sin(t)$.

The sine waves for each two consecutive hinge joints are set in antiphase to each other (i.e. their phase difference is 180 degrees; this setup facilitates the evolution of oscillatory movement patterns) and the angle of each hinge joint has an upper and lower limit of $\pm$ 45 degrees.

3.1 Encoding Folding Patterns

The CPPN-based ribosomal robot encoding is shown in Figure 3. The inputs to the CPPN are the current segment number p and the scaled distance from the ribbon center

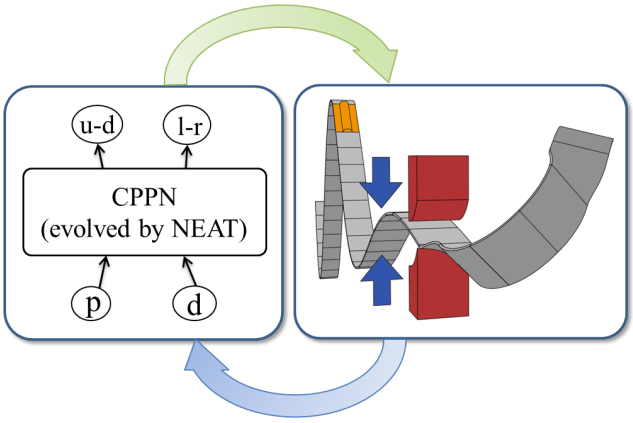


Figure 3: Ribosomal CPPN Encoding. (a) The printer sends the current position p and distance from center of the ribbon d to the CPPN (evolved by NEAT), which in turn determines the type of fold that the printer should perform. After the folding process is complete the resulting three-dimensional robot is able to interact with its environment.

$d = \frac{p-N/2.0}{N/2.0}$, where N is the total number of ribbon segments. The input d is specifically added to encourage the formation of symmetrical patterns. The outputs indicate the type of desired fold (e.g. left/right or up/down). The desired 45 degree fold (if any) at a particular position p is determined based on the highest activated output (left/right or up/down). If $|ud| > |lr|$ the ribbon is either folded upwards 45 degrees if $ud > 0.4$ or folded downwards if $ud < -0.4$. The printer performs no fold if neither $|ud|$ nor $|lr|$ is greater than 0.4. The right/left folds are determined in an equivalent manner.

This paper tests and compares the effect of different ribbon lengths on the ability to produce walking robot morphologies. In particular, two different methods are employed to determine the length of the ribbon being fed into the printer: a fixed-length method and a variable-length method. In the **fixed-length** method, the CPPN folds a ribbon whose number of segments have been determined *a priori* (by the experimenter). Conversely, in the **variable-length** setup, the CPPN is augmented with an additional output m that allows it to dynamically determine the maximum number of segments. To determine m the CPPN is initially queried with both p and d set to zero.

The ability of CPPNs to represent a variety of different patterns [13, 14, 15], together with the insight that a one-dimensional string can theoretically approximate any three-dimensional shape [5], suggest that the ribosomal CPPN encoding will enable a multitude of different three-dimensional robot designs.

4. EXPERIMENTAL SETUP

In order to test the efficacy of the introduced manufacturing scheme and design methodology, the system is tasked with evolving designs that can effectively accomplish a simple locomotion task. Locomotion tasks allow the evolution of a variety of different behaviors (e.g. crawling, jumping, pedalism) [8, 9] and are therefore a good domain to test the versatility of the proposed system. This experiment and the underlying ribosomal assembly system is implemented in a realistic physics simulation using the freely available Open Dynamics Engine (http://www.ode.org).

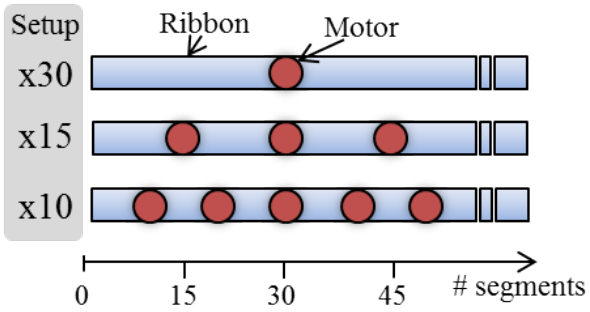


Figure 4: Ribbon Actuator Spacing. This figure shows the three different motor spacings that are compared in this paper. For example, in the x30 setup, the ribbon is augmented with a motorized hinge joint every 30 segments.

The **fixed-length** CPPN setup is evaluated with ribbons consisting of 60, 150, and 300 segments. In the **variable-length** setup the CPPN can automatically determine the number of ribbon segments (see Section 3.1) in the range of [60, 300] segments.

In addition to the number of segments in an individual strip of material, this paper also examines the effect of the total number of hinge actuators on the overall design process. Specifically, the material contains hinge-actuators spaced equally either every 30 segments (**x30**), every 15 segments (**x15**), or every 10 segments (**x10**). This separation and the overall length of the ribbon are varied together to examine the full effect of both degrees of freedom. For example, a ribbon with 150 segments with an actuator spacing of x10 has a total of 14 motors each separated by 10 segments of foldable material. Figure 4 shows an overview of the three different ribbon designs.

The fitness function, which is the same for all aproaches, is defined as follows:

$$F = \sqrt{(sx-ex)^2 + (sy-ey)^2} - \sqrt{(sz-ez)^2}, \qquad (1)$$

where (sx, sy, sz) describes the robot's center of gravity at the beginning of the trial and (ex, ey, ez) is the center of gravity after the evaluation period of $1,500$ time steps. The fitness function rewards robots that travel in straight lines (instead of circles) and discourages designs that have a high initial center of gravity and move by simply falling forward (which would increase the value of $\sqrt{(sz-ez)^2}$).

Evaluation is either terminated after the allotted time expires or if collisions between different ribbon parts occur during the folding process (these robots automatically receive a fitness of zero). The details of how collisions are handled are highlighted in the next section.

4.1 Collision Handling

The employed collision scheme in this paper tries to strike a balance between computational expense, design considerations, and realistic simulation. During the **folding process** designs are discarded if parts of the ribbon that are *separated by more than one motor* intersect. For example, in a sixty segment structure containing two motors, the first twenty segments could intersect with themselves and the sec-

Table 1: Simulation Parameter Settings

Parameter	Value
Proportional Constant	9.0
Gravity	-9.8
Maximum Torque	5.0 newton meters
Segment Length	0.5 meters
Segment Width	0.4 meters
Segment Density	0.1 kilograms per cubic meter
Hinge Joint Limit	±45 degrees

ond twenty, but not with the third twenty. In this way, the folding process can selectively disable individual motors by creating overlapping designs that intersect between parts. However, the entire design cannot be rendered completely immobile by causing the first part to intersect with the last.

While it is important to note that such overlapping configurations are not directly physically realizable by a ribbon printer, the chosen collision scheme allows an initial wider exploration of the space of CPPN-encoded ribbon robots.

During the **actuation of the robot**, collision is also checked for segments that are directly connected to each other, which means that segments can not freely move through each other.

4.2 Experimental Parameters

Because CPPN-NEAT differs from original NEAT only in its set of activation functions, it uses the same parameters [16]. The size of each population was 100 with 10% elitism. Sexual offspring (50%) did not undergo mutation. Asexual offspring (50%) had 0.75 probability of link weight mutation, 0.1 chance of link addition, and 0.05 chance of node addition. The NEAT coefficients for determining species similarity were 1.0 for nodes and connections and 0.1 for weights. The available CPPN activation functions were sigmoid, Gaussian, absolute value, cosine, and sine, all with equal probability of being added. Parameter settings are based on prior reported settings for NEAT [16, 17]. They were found to be robust to moderate variation through preliminary experimentation. Some of the ODE simulation parameters are shown in table 1.

Note that the seed CPPN for all approaches directly connects the distance from center input d to the weight output of the CPPN but lacks direct connections from the position input p. However, mutations on the seed to create the initial generation can connect arbitrary inputs to arbitrary outputs. This setup has shown to encourage the evolution of symmetric morphologies while also allowing asymmetrical morphologies to evolve.

5. RESULTS

Figure 5 shows the final performance of each treatment averaged over 40 independent runs consisting of 500 generations of evolution. The x30 ribbon with 60 segments reaches an average fitness of 33.11 (σ = 3.37) and outperforms all other fixed-length approaches ($p < 0.001$; according to the Student's t-test) except the x10 setup with the same number of segments. Figure 6 shows the training performance over evaluations for the 60- and 300-segment setup (the 150-segment setup was omitted for readability).

The performance for a specific ribbon type (e.g. x10, x15, x30) is never significantly worse the shorter the ribbon but often significantly better (in 12 out of 18 pair-wise comparisons; $p < 0.001$). This result suggests that the probability of finding a viable design is negatively correlated with the likelihood of collisions during the folding process.

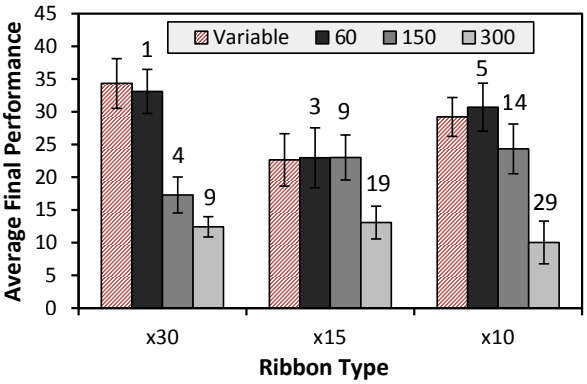


Figure 5: Average Final Performance. This figure shows the average final fitness and standard deviation averaged over 40 runs. The number of motors for each ribbon is shown above each treatment for the fixed-length setups. The variable-length setup can automatically decide on the length of the ribbon which in turn determines the number of embedded motors.

The variable-length setup performs similarly to the fixed-length setup with 60 segments regardless of actuator density (x10, x15, x30) and the final average fitness values are not significantly different. A closer look at the robot morphologies created through this approach reveals that the variable-length CPPN almost consistently, in 114 out of the total 120 runs (40 for each ribbon type), chooses to construct robots out of 60 segments, which is the smallest number of segments possible (Section 4). These results suggest that the variable-length setup converges on the structure that is easiest to optimize and therefore shows the highest performance.

The performance of the 150 segment setup increases significantly from an average of 17.28 (σ=2.74) with 4 motors (x30) to an average fitness of 24.33 (σ=3.79) with 14 motors (x10). The performance of the 60 and 300 segment setups display a non-linear relationship with a change in the number of motors. The 300 segment setup performs best with ribbon type x15 but its performances decreases with lower and higher motor densities (though not significantly). The performance of the 60 segment setup shows the opposite behavior and reaches a local minimum with the x15 ribbon. This surprising difference in the relationship between actuator density and segment length requires explanation.

A possible source of this disparity involves the particular manner in which collisions are treated in the evolutionary search algorithm (see Section 4.1). In order to test the effect of this treatment, 20 additional runs were performed with collisions disabled (i.e. evaluation of the robots was not terminated upon collision and parts of the robot separated by motors were able to intersect each other during simulation). Figure 7 shows that disabling collisions significantly ($p < 0.001$) increases the final performance of all but the x30 ribbon.

5.1 Example Evolved Solutions

A variety of different robots all created from the same strip of material (a 150-segment ribbon with four motors)

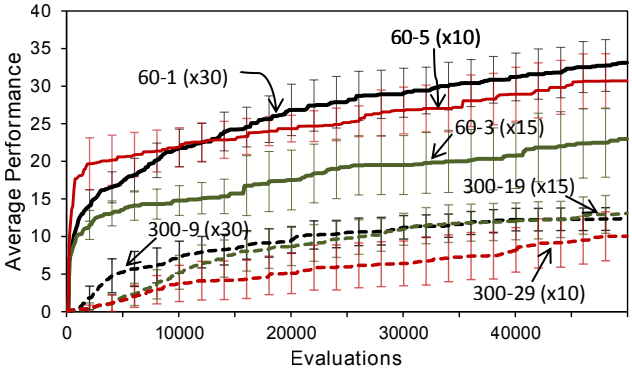


Figure 6: Training Performance Over Evaluations. This figure shows the average performance over generations (including standard deviation) for the 60 and 300 segment setups over all three different ribbon types. For each treatment the number of segments, total number of motors and ribbon type are shown (e.g. *60-1 (x30)* = 60 segment ribbon of type x30 with one motor).

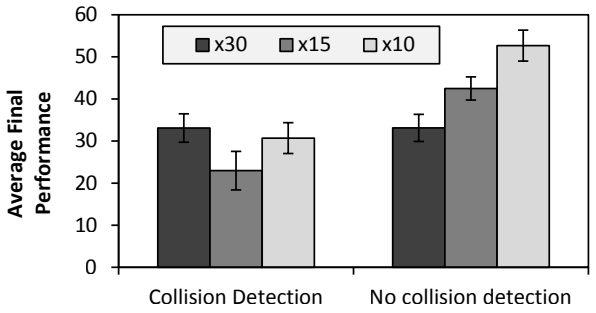


Figure 7: Average Final Performance With and Without Collision Detection. This figure shows the average final performance for the 60 segment setup with and without collision detection. Disabling collisions during the simulation significantly increases the performance of all but the x30 ribbon.

are shown in Figure 8. The evolved robots exhibit various methods of locomotion (including crawling and pedalism) and their morphologies clearly show regularities like symmetry and repetition[1]. Figure 9 shows an example of the simulated folding process of a three-dimensional robot with 300 segments and nine motors over the course of multiple stages. A composite image of a bipedal robot locomoting from left to right is shown in figure 1b.

In order to test the feasibility of structures evolved using this scheme, a non-actuated, 3D-printed model of a simple biped design was produced. This simple model, shown in Figure 1c, illustrates that the structure is capable of both supporting itself, and standing on its own. Future models will be printed using a custom printer, and will include actuation pre-embedded in the medium.

The main conclusion is that the CPPN encoding is able to effectively encode a variety of different robots with repeating patterns and symmetries. Additionally, the manner in which collisions are treated during the evolution of morphologies plays a vital role in the overall fitness of the resulting forms. This was indicated by the non-linear relationship between optimal ribbon length and actuator density, and confirmed by the linear relationship exhibited when collisions did not play a role in the design process.

[1]Videos of the evolved designs can be found at `http://www.youtube.com/user/NEmachines`

6. DISCUSSION AND FUTURE WORK

Through a simple vocabulary of 20 amino acids, the ribosome enables the assembly of the multitude of protein structures that form the foundation of all organic life. Capturing the unique confluence of versatility, complexity, adaptability, and robustness displayed by the products of ribosomal assembly is an important goal of artificial life research.

In a first step towards this goal, this paper presented a ribosomal robotics platform that is inspired by the folding of proteins. An important difference to related work in ribosomal-inspired assembly [5, 6, 12] is the fact that the robotics system in this paper decouples the act of *construction* from the act of *assembly*. In the future, this fabrication process can allow the on-demand creation of specialized robots capable of solving a wide variety of complex problems, but all derived from a single, generic strip of material.

The evolved designs show that it is indeed possible to create a variety of different robots, all from the same source material, that display different methods of locomotion. Also, the employed CPPN encoding can effectively transform a one-dimensional structure into a three-dimensional robot containing actuation.

Through the testing of different methods of ascertaining ribbon length and fold patterns, the variable-length CPPN (Section 3.1) converges to the smallest possible length of 60 segments. This result indicates that the simple locomotion task in this paper can efficiently be solved by a robot that is not exceedingly complex. Therefore, future work will include applying the approach to more elaborate tasks and environments, which would *require* designs that involve more complex folding patterns.

It is worth noting that designs using ribbons consisting of 300 segments perform significantly worse than those consisting of 60 segments (Figure 5), regardless of the number of actuators included on the structure (e.g. x10, x15, and x30). One might argue that the decrease in performance is due to an increase in the space of genetically encoded parameter values of the folds. However, an indirect encoding like a CPPN is not the same as a direct encoding with a one-to-one mapping between the parameter values and the genes. Rather, the longer the ribbon, the more likely collisions will occur during the folding process (resulting in an automatic termination of the design evaluation), thereby making the domain very fragile (i.e. many mutations are fatal).

Indeed, many of the robots in the initial generations of the 300 segment approach received a fitness of zero (creating no gradient for an objective based search method) and disabling collisions increases the performance of almost every treatment (figure 7). While it is important to note that the employed collision scheme allows overlapping configurations that are not directly physically realisable by a ribbon printer, it enables us to study the effects of collision handling on fitness together with an initial wider exploration of the space of CPPN-encoded ribbon robots.

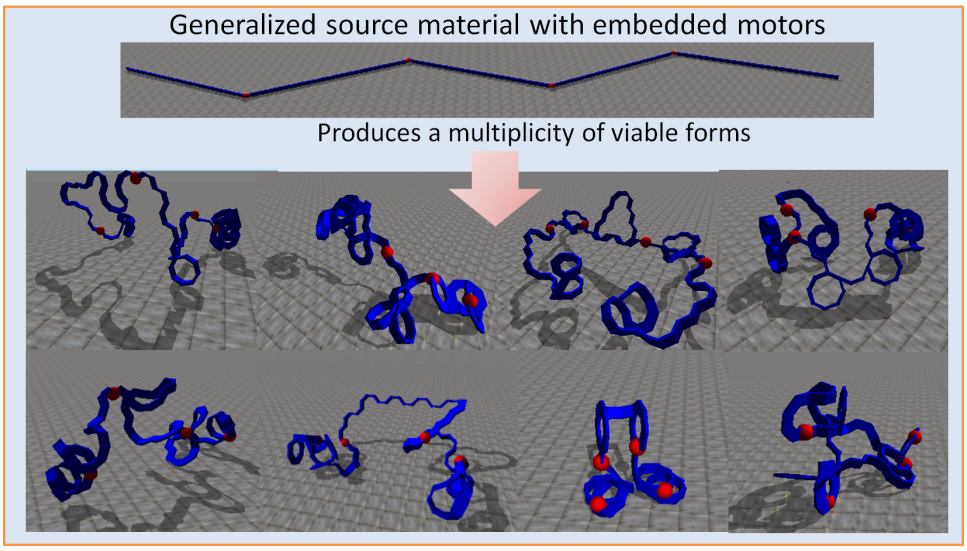


Figure 8: Ribosomal Robot Examples. Shown are eight different robots all created from *the same 150-segment ribbon* that exhibit different methods of locomotion. The four embedded hinge motors are shown in red. The CPPN encoding is able to effectively encode a variety of different robots with repeating patterns and symmetries.

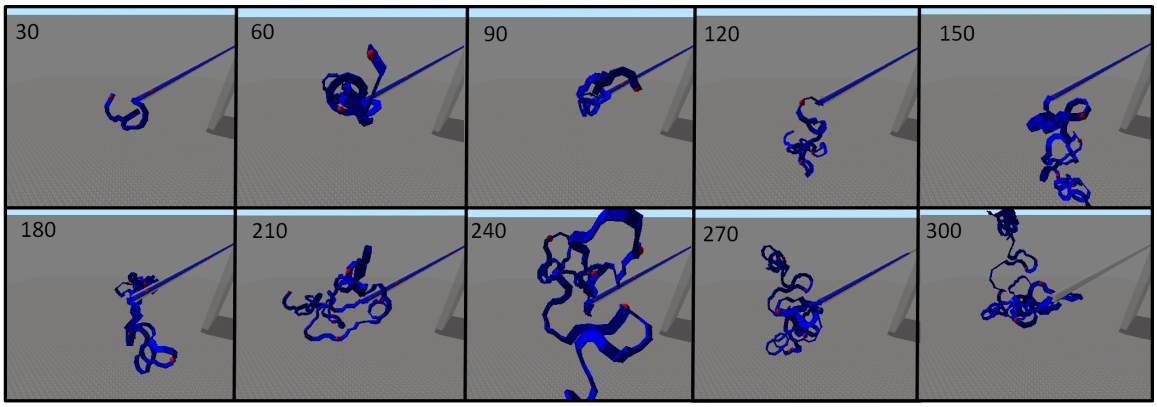


Figure 9: Folding Simulation. This figure shows the folding of a robot with 300-segments and nine motors. The current segment number is displayed at the top left corner of each frame.

An interesting alternative method to explore will be to perform folds in a manner similar to a self-avoiding walk [7], whereupon it will be determined during construction if a fold requested by the CPPN would result in a collision. In this event, the algorithm will choose the next most desired fold, producing populations that will only rarely have to settle the question of collision and therefore retain much of their morphological diversity.

Additionally, an important future direction is to evolve the fold pattern and the control of the robot together, so that the evolved controls will be uniquely tailored to their robot morphologies, allowing full autonomy. A potential approach is the *Hypercube-based NEAT* algorithm (HyperNEAT [18]), which showed that CPPNs can also represent the connectivity patterns of brain-like neural networks [2, 3, 4, 18]. In this way, NEAT can evolve CPPNs that represent large-scale ANNs with their own symmetries and regularities.

While the experiments in this paper are promising, the construction of a printer capable of implementing these designs in reality is an important next objective of this project. This printer will adapt the well-established manufacturing technique known as guided bending in order to form lengths of material into the desired shape. It will consist of a set of motors capable of driving the material into a print head consisting of hydraulic elements and pre-shaped presses that can deform the material in a consistent and structurally sound manner. Central to this process will be the discovery and implementation of different material types and cross-sectional configurations that are capable of producing structurally sound but overall very light designs. The material and cross-sectional shape choices will, in turn, inform the design processes that are tasked with discovering new robotic morphologies and controllers.

7. CONCLUSIONS

This paper presented a new ribosomal-inspired approach for the autonomous design of robotic life forms. Starting with a strip of material that has been augmented with actu-

ators a "printer" folds this structure into a three-dimensional actuated robot. Results in a simple locomotion domain confirm that this fabrication process together with a novel CPPN representation can encode a variety of different walking robot morphologies. Furthermore, because the robotics system in this paper decouples the act of construction from the act of assembly, construction merely involves folding the robot into a useful shape. Thus the main conclusion is that the ribosomal-inspired assembly method is a promising new approach that could lead to a sophisticated alternative in the autonomous design of reusable robots.

Acknowledgments

This work was supported by a McMullen fellowship and fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

References

[1] J. Bachrach, V. Zykov, and S. Griffith. Folding arbitrary 3d shapes with space-filling chain robots: Folded configuration design as 3d hamiltonian path through target solid, 2009.

[2] J. Clune, B. E. Beckmann, C. Ofria, and R. T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2009) Special Session on Evolutionary Robotics*, Piscataway, NJ, USA, 2009. IEEE Press.

[3] J. Gauci and K. O. Stanley. A case study on the critical role of geometric regularity in machine learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008)*, Menlo Park, CA, 2008. AAAI Press.

[4] J. Gauci and K. O. Stanley. Autonomous evolution of topographic regularities in artificial neural networks. *Neural Comput.*, 22:1860–1898, 2010.

[5] S. Griffith. *Growing machines*. PhD thesis, Massachusetts Institute of Technology, 2004.

[6] A. Knaian, K. Cheung, M. Lobovsky, A. Oines, P. Schmidt-Neilsen, and N. Gershenfeld. The milli-motein: A self-folding chain of programmable matter with a one centimeter module pitch. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1447–1453. IEEE, 2012.

[7] D. Landau and K. Binder. *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press, 2005.

[8] J. Lehman and K. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.

[9] H. Lipson and J. B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

[10] H. Lodish, D. Baltimore, A. Berk, and J. Darnell. *Molecular cell biology*. WH Freeman New York, NY:, 1995.

[11] S. Lohmann, J. Yosinski, E. Gold, J. Clune, J. Blum, and H. Lipson. Aracna: An open-source quadruped platform for evolutionary robotics. In *Artificial Life*, volume 13, pages 387–392, 2012.

[12] C. Onal, R. Wood, and D. Rus. Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4608–4613. IEEE, 2011.

[13] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley. Picbreeder: Evolving pictures collaboratively online. In *CHI '08: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1759–1768, New York, NY, USA, 2008. ACM.

[14] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3): 373–403, 2011.

[15] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162, 2007.

[16] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

[17] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

[18] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.