Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2013) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2014

Rigid folding in robotic multi-agent systems

Benjamin Felbrich[a], Jörg Rainer Nönnig[a], Sebastian Wiesenhütter[a] *

[a]*TU Dresden, Dept. of Architecture, Juniorprofessur of Knowledge Architecture, Helmholtzstraße 10, Dresden 01069, Germany*

Abstract

The paper seeks to investigate the potential of biologically inspired multi-agent systems for architectural applications. The generation of 'form' through the process of 'rigid folding' represents a complex architectural procedure, which we seek to perform by employing a number of simple collaborative agents. Through a prototypic approach, closely matched with the constraints of a real world application, we want to demonstrate that the central architectural needs of form adaptability and resilience can be increased with this method. The established prototype is intended to serve as the 'embodiment' for different codes and strategies to test. Its shape transitions are based on the principal of folding, in which the triangle mesh topology serves as the basis for its folding patterns.

Keywords: bionics, architecture, free form generation, multi-agent systems, rigid folding

1. Introduction

Rigid folding in architecture has been recognized as a promising form-determining principle for a long time – yet its potential has not been fully tapped. The complex operations require the designer to pursue sophisticated geometrical computation, while the physical production of folded structures demands high technical skills resp. manufacturing technology. This has been done intuitively by hands-on experimenting on physical models, by material "sketching", and of course the spatial imagination in the designer's mind. The understanding and evaluation of the folding characteristics in a given geometry proceeds step by step. We hold that the use of algorithmic and generative modeling tools can bring a shift in the cognitive process of the designer. On this assumption, we formulated an experimental setup in which this process can be aided by artificial intelligence. The setup combines two different biological phenomena into one approach: while the

* Benjamin Felbrich. Tel.: +49 176 56794466.
E-mail address: bfelbrich@hotmail.com.

physical body and its movement is based on the biological process of folding (as it can be found in the unfolding of blossoms or leaves from a bud), its control mechanism takes inspiration from biological cognitive processes and general phenomena of collective intelligence, where characteristics of 'emergence' can be observed.

In the following paragraphs we will take a closer look at folding processes in nature. Thereupon we will introduce the combination of the two principles by way of a 'free form generating robot'.

2. Folding in nature

In developing blossoms and leaves in a compact, folded state until they are almost entirely completed within a bud, plants found a very efficient strategy to save their most fragile parts from the hazards of frosty weather and predators. It is not until these parts have reached a certain level of attractiveness to pollinators, or an effectiveness in collecting energy through sunlight, that – triggered by a certain environmental signal (e.g. increasing temperature during spring time) – the plant activates the process of unfolding. The blossom or leave reaches full size within a very short time and is immediately able to fully benefit the plant.

The correlation with architectural demands is obvious: The need to "grow" – or better: to arrange a structure to a far extend under convenient conditions, and to activate its structural potential thereafter as quickly as possible in-situ, is a common modus operandi in many architectural and engineering tasks. Prefabrication of concrete pieces, wooden framework and basically every pre-assembly operation could be interpreted as a bio-inspired process in this sense.

The folding patterns on which plant-folding is based on, are a subject of research itself widely discussed in the respective scientific fields. However it is save to say that most of these folding patterns are based on the principles of symmetry and repetition. It is also known, that most of these patterns can unfold in one step, i.e. they do not require an order or coordination in the sequence of unfolding. From the moment, the signal to unfold is given, all folding edges simultaneously unfold to their final state. These aspects have strong practical implications: On one hand a repetitive, symmetric structure is much easier to produce than a irregular, chaotic assembly of parts. This counts for both nature and man made environment. On the other hand, the control of a one step folding can't be simplified any further.

These benefits aside, the range of forms which can be erected in this simple folding process is rather limited. In other words: not every imaginable free form geometry can be generated by unfolding a symmetric, repetitive folding pattern in one step. Examples for the utilization of regular folding in design are not hard to find. Yet the generation of free form demands the utilization of irregular folding patterns and a sophisticated multi-step folding control. Here the folding pattern derives from the target shape that is to be achieved. Specifically spoken, we can discretize a free form surface which represents our target surface, to obtain the folding edges needed to (un-)fold this geometry. This process will be described in detail in section 4.1.

3. Control mechanism

As mentioned before we need to coordinate several steps of unfolding in order to achieve free form geometry. The most obvious approach would be to prescribe a certain program in which the control of each folding edge is defined and followed to completion of folding. This approach, however, proves very inflexible. As soon as we change the initial state or the target state of this process, the whole program has to be revised. A promising approach is the application of origami based methods. A framework for the systematic design of the crease patterns has been proposed at RTWH Aachen [1]. The constraints for such form-finding are developability and flat-foldability. This however limits the range of possible shapes. Once the optimized shape has been determined the erection of the structure is performed by a crane-robot in a one-step-process. This

method requires the knowledge of sophisticated mathematical descriptions of origami folding such as description through crease angles [2]. We hold that omission of the constraints developability and flat-foldability results in higher range of shapes to be achieved. Thus we propose to rather apply concepts of 'self-organization' for form-finding. This will reduce the systems' reliance on the knowledge of mathematical models. It will rather be enabled to adapt its behavior to its inherent physical structure as well as the changeable given tasks represented by a multiplicity of free forms to create.

Concepts of self-organization and distributed computation have been successfully implemented in many fields. The Ant Colony Optimization algorithm [3], introduced in 1993 by Marco Dorigo et al. became a prominent example of a logical mechanism in which the interaction of simple agents leads to a relatively intelligent collective behavior to optimize food paths.

In robotics the concept of 'Swarm Intelligence' is closely related to the term 'unpredictability' and has often been conceived in the form of cellular robotic systems [4]. These concepts employ the cooperation of non-synchronized, autonomous, non-intelligent, kinetic and heterogeneous robots to achieve global tasks. The characteristics of self-reconfiguration renders these systems very potent for architectural problem solving as described above. Latest developments propose swarm robots like the integrated project "Symbrion" of a group of European universities [5] or the project "Termes" at Harvard University [6]. The translation of these concepts of 'self-organization' in architecture reaches from urban scale developments to micro-material concepts. High potential is seen in methods of 'growth and adaption' [7]. However the concepts for folding geometry are mostly concerned with origami folding, e.g. in the 'programmable matter' of shape-shifting sheets [8]. Latest material developments and technologies related to 4D-printing have fostered concepts of 'self-assembly' and especially 'self-folding' [9]. However the self-organized folding of rigid plates into free-form structures is still an unsolved problem. To approach this given task we took the basic idea to define each folding edge as a single agent with the target to reach a certain angle between its adjacent facets. There are numerous ways to achieve the final folding state. The limitations of the respective movements would lie in their physical interconnections. The means of solving would lie in direct communication and collaboration between the agents. The aim of the following work was to build a prototype with the respective physical properties (irregular fold patterns, reconfigurability, interconnection) and to test different code sets with this prototype.

4. Digital Setup

As a framework, a physical embodiment to geometrically describe, optimize, steer and to test the ideas of decentralized self-organization against, we propose a multi-agent folding robot in the topology of a triangular mesh. This device was established and built in a prototyping process at the chair of Knowledge Architecture at TU Dresden. In the following we will describe its basic function and geometry, the simulation of its behavior and how it influenced the prototyping process, its electronic setup and coding and finally evaluate its performance with a brief discussion of the main constraints.

4.1. Underlying mesh topology – the folding pattern

Simple robotic agents are aligned to form a triangular folding pattern. In fact they are located on the inner edges of a virtual triangular polygon mesh. The triangular alignment proved most suitable, since, in contrary to quad or hexagonal meshes, its facets are always planar and therefore the structure is much easier to handle. Derived from the needs of an architectural utilization this underlying mesh had to fulfill certain conditions:

1. Limitation: In computer graphics, triangle meshes usually employ an infinite variety of edge lengths and connectivity maps to regenerate and depict the respective surface as smooth and accurate as possible. If we were to rebuild this kind of mesh in actual prototypes and models, we would have to build a vast amount of unique pieces, which leads to immense cost. In order to reduce cost, we had to

reduce the amount of unique pieces by increasing the degree of repetition. To do so, we established a triangulation algorithm which only makes use of a very limited amount of available edge lengths (for example: four different edge lengths) to generate the folding pattern from a predefined free form target surface. This leads to a reduced amount of possible triangle configurations. With this operation mode, however, the accuracy of the surface triangulation and the smoothness of its result represent an issue of optimization. Specifically spoken, not all nodes of the generated mesh necessarily lie exactly on the surface to triangulate. We were willing to accept small tolerances to dramatically reduce cost. Figure 1 shows the triangulation of a free form using this method and the results in accuracy and smoothness to be achieved.

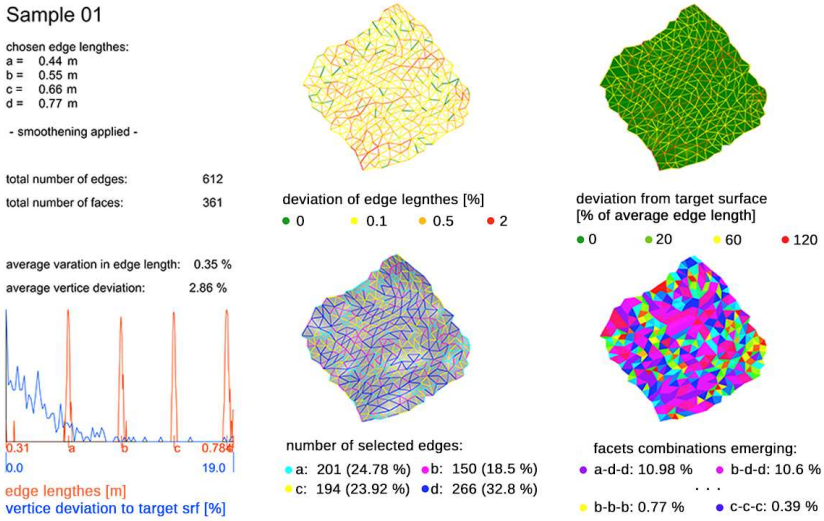

Fig. 1. Triangulation of a sample surface

2. Perforation: Every triangular mesh consists of two kinds of nodes: “closed” nodes which are entirely surrounded by adjacent mesh facets and “open” nodes which are adjacent to a hole or the border of the mesh. In [10] the discrete Gaussian curvature K in a node p in polyhedral surfaces is defined through the sum of the angles α_i between neighboring edges adjacent to this node p:

$$K(p) := 2\pi - \sum_i \alpha_i$$

This equation obviously only applies to “closed” nodes, in which the angle sum is fixed. Thus, “closed” nodes have a precisely defined discrete Gaussian curvature. In “open” nodes, however, there is no fixed discrete Gaussian curvature. Thus, these nodes are much more flexible in their folding behavior. We increased the amount of these “open” nodes in our mesh by leaving certain mesh edges out. In other words: We perforated the mesh and could thereby increase its overall folding flexibility.

Through physical mockups, we found, that a suitable physical behavior and flexibility is achieved, when every closed node is adjacent to at least one open node.

Due to this modifications, we decided to call this geometrical structure the perforated, limited, triangular mesh (short: *p-l-t mesh*) of the rank *x*, where *x* defines the number of unique edge lengths utilized in this mesh.

4.2. Simulation

Beside the prototypic approach in which the model is physically built, tested and improved, a sophisticated simulation can represent a short cut in the development of a dynamic model. If we refrain from elaborating an extremely complex simulation, however, this tool fails in depicting the physical behavior in the sense of material yielding, motor strength and force transition. Nonetheless we could use it to check some of the principles we wanted to apply.

The topological equivalence of our folded structure to triangular polygon meshes reveals a great potential in simulating the folding behavior. It enabled us to use the particle spring engine Kangaroo in Grasshopper for Rhino 5 for the dynamic simulation of rigid folding.

4.2.1. Check for approach sufficiency

Figures 2 to 5 show the key frames of the simulation of the collapse and re-erection of a rank 4 p-l-t mesh. It was intended to avoid any deviations in the lengths of the mesh edges during the simulation. Thereby any movement and shape transition of the structure is solely based on the rotation of adjoining mesh facets around their common edges – folding. Since the utilized particle spring engine was merely designed to simulate elastic networks, it fails to depict the behavior of 100% rigid material. In other words, when strong forces are applied to the fold structure, it is not possible in the simulation to avoid edge length dilatations. We utilized this property to interpret edge length deviations as cracks within the respective agents when they exceed a certain threshold value.

In the very beginning (figure 2) the folded structure has the shape of the free form to be regenerated (the target shape). For this folding structure the information about the number of nodes, the way they are connected (described with a connectivity map) and the distance between nearby nodes (mesh edge lengths) form the basic unalterable information we need to describe and simulate how the shape can be transformed through folding. In our assumption, the knowledge of the angles between adjoining facets of the mesh is enough information to explicitly describe the mesh's state, and thereby the actual form of the folded structure. In other words: While node connectivity and edge length remain invariant during the simulation, every transformation through folding can be described explicitly with the variation of the angles between adjoining mesh facets. We assigned this angle δ_i to the respective mesh edge i about which these adjacent facets rotate and to the robotic agent located on this edge. If our assumption is true, the target state of the folding structure can be reached when every agent i has reached its individual target angle. Then a hinge force applied over every mesh edge, pushing the two adjoining mesh facets until they span their individual target angle, should, if it's strong enough, force the folding structure to generate its target shape.

During the simulation the trajectory of the angles δ_i were recorded. Figure 6 exemplarily shows the angle trajectory curve of edge (=agent) number 108. Note that, since the simulation starts in the target state, the angle δ_{108} in $t = 0$ is the target angle for this agent (green in fig. 6).

From the initial state (fig. 2) to the moment of total collapse (fig. 3; the moment marked "A" in figures 6 and 7) no other force than gravity is applied. The structure collapses only through folding. Note how the angle value of agent nr. 108 arbitrarily changes. After this collapse, in figure 3 (resp. the moment "A") the hinge force mentioned earlier is applied between neighbor mesh facets to restore their initial angle value. Regardless of the mesh's topology this force is equal in every edge and constant over time starting from moment "A" (resp. the logic of a one step folding, see section 2). Figure 5 shows the complete re-erection of the structure into its initial state. Note how the angle nr. 108 in fig. 6 jumps back towards this value in green. In fig. 7 you can see how the edge length nr. 108 abruptly changes in the moment A. The value in magenta marks a deviation of 1% of the edge's length. This value was defined as a threshold not to be exceeded to avoid cracks in the system.

The complete re-erection of the structure shows, that the information about the individual target angles is enough to achieve the overall target state. On the other hand, a simple hinge force applied constantly and equally in all agents, will most likely cause cracks in such a folding mechanism.

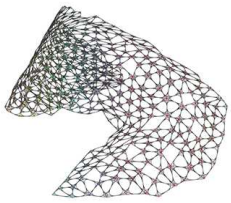

Fig. 2. Initial shape equal to target shape

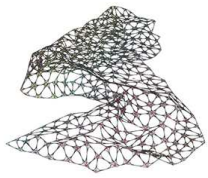

Fig. 3. Moment of total collapse

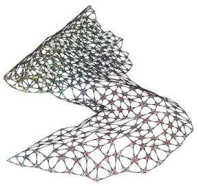

Fig. 4. Moment of first hinge force application (moment “A”)

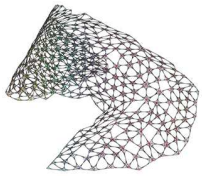

Fig. 5. Complete re-erection

Fig. 6. angle curve of agent nr. 108

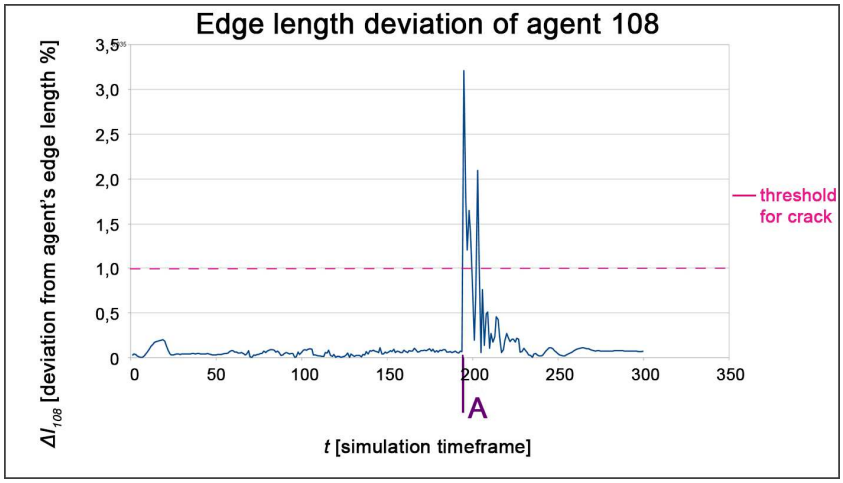

Fig. 7. Length deviation of edge nr. 108

4.2.2. Guide simulation for the prototype

The purpose of this simulation setup was to depict the behavior of the actual prototypic setup. It is the digital counter part of the physical prototype that was built later. While the simulation described in 4.2.1 served to give a general insight in the p-l-t mesh's behavior and therefore utilized a very large amount of agents, this simulation served as a guideline for the prototype to be built. Therefore a smaller amount of mesh facets was used (to reduce cost in the prototype). Nonetheless it is also built on the principles of a p-l-t mesh of rank 3. In this digital model we could play with the geometry, fold it virtually and observe the individual changes in edge lengths and agent angles δ_i. Figure 8 shows different states of an exemplary folding process with 8.a) showing the state defined as the initial state and 8.c) as its target shape. This particular shape transition was recorded in the agents' angle trajectories to be repeated by the prototype as an exemplary folding program.

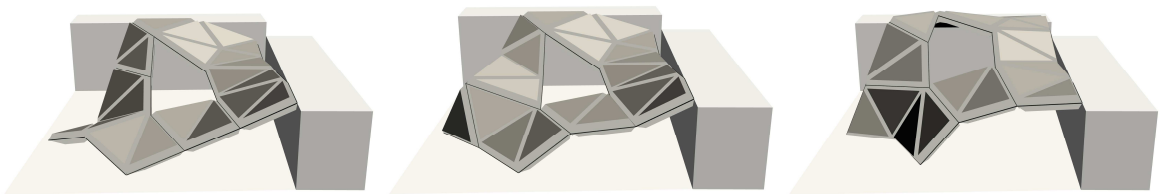

Fig. 8. Folding mechanism in different state: a) start, b) intermediate state, c) final state

5. Physical Prototype

Through a spindle drive mechanism mounted to a frame on each mesh face, the agents are able to apply a push or pull force to the frame and thereby change the angle δ_i between two adjacent mesh facets. With this method each agent is capable to modify the overall state of the device. Figure 9 shows the basic idea.

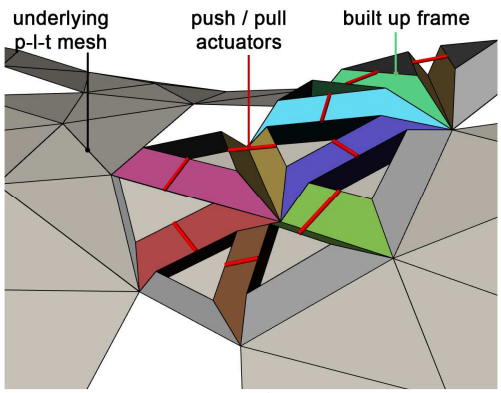

Fig. 9. Basic idea of the mechanism

Each of the 21 inner edges of the mesh structure is equipped with an electric motor integrated in a gear box and mounted to a rotating threaded shaft to push and pull its counter piece. Furthermore every agent (fig. 10) provides a potentiometer with its rotation axis aligning with the folding edge. With this orientation the potentiometer gives a feedback over the angular state and rotation of the agent. The model mainly consists of light weight foam board and the more stable MDF in the highly stressed parts (the motors' gear boxes, the potentiometers mounts etc.). It was designed as a construction kit to be reconfigurable and to be arranged in any manner, as long as the three different predefined edge lengths are being used. Figure 11 shows the group of 21 agents combined to the mesh structure resp. to figure 8 a). The target for this specific setup was to shift into the state resp. to figure 8. c).

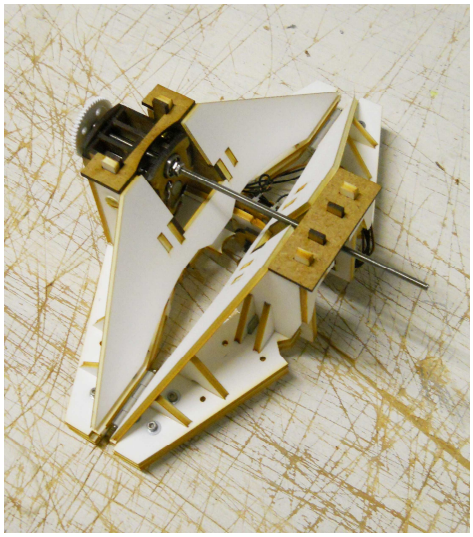

Fig. 10. Single agent

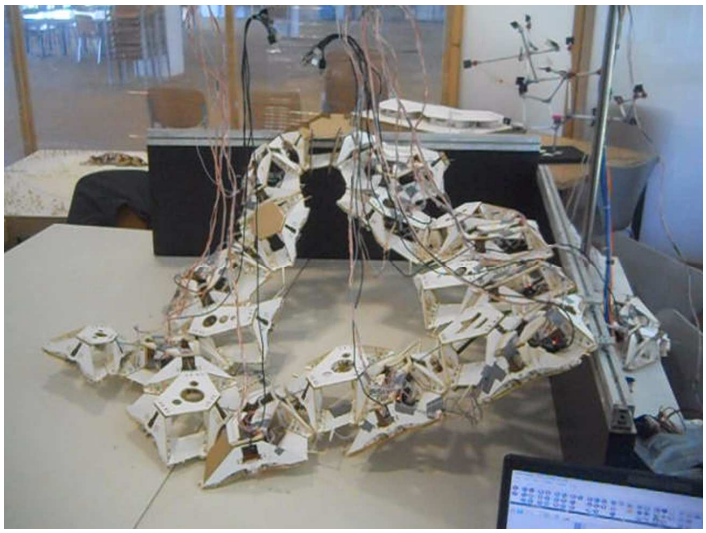

Fig. 11. Entire device of 21 interconnected agents

5.1. Electronic components

Following electronic components were installed in the device:

Motors: 21 x Johnson 20543, 21 x Murata Z5U-5 ceramic capacitor 1 µF
Motor drivers: 12 x L298N two channels, up to 2A each
Potentiometer: 21 x Piher PT-15 10 kΩ
Control: 1 x Arduino Duemilenove (for global control), 11x SparkFun Pro Micro 5V 16MHz
Power supply: 2x LPK2-23 400W power supply, 12V 22A each

A notebook is used to send global timing and ON/OFF signals to the Arduino Duemilenove using the Firefly interface in Grasshopper for Rhino 5. From here these signals are distributed to the smaller SparkFun Pro Micro boards, each of which control two motors. The commands of the two motors run independently within the code resulting in 21 quasi-independent robotic agents with an input from the respective potentiometer, an ON/OFF switch and a global timing input.

5.2. Current code setup

The first coding uses the simplest strategy as described in section 4.2.1. in which a hinge force constantly pushes the agent towards its target angle value. As presumed from the simulation this strategy caused a very agile movement of the structure but also led to cracks. The initial target set for this experiment – to reach the target shape – could not be achieved with this setup. In a certain point the device stops moving and can't move further towards its destined position due to lack in power.
The second coding setup, however, is based on angle control curves recorded from the simulation described in 4.2.2. These curves were implemented in the code in the form of 10-bit integer arrays and intended to be followed by each agent synchronized by the global timing signal. This kind of choreography merely represents a predefined program, the device follows to reach its goal. However, due to its lack in strength and stability the robot was not yet able to reach its goal this code either. It gets stuck in the same manner as it does for.

5.3. Evaluation

In our opinion this failure does not necessarily call for stronger motors or stronger frames but for a more intelligent control mechanism. Just like ants are not able to optimize their food paths on their one, the agents in this setup need to distribute loads or find an optimized strategy to reach their goal even if they are very weak. In a simulation this goal can be abstracted into the task for the agents to coordinate their control curves with those of their neighbors under inconsistent constraints. The advantage of our prototypic approach is that we are now able to directly test the sufficiency of such code experiments in the actual device. Also an approach without any previous simulation but with try and error is possible.

6. Conclusion

An important advantage of the setup lies in the parallelism of its components. While the agents act independently through their own processing unit, it was shown in 4.2.1, that if every single agent achieves its individual goal to reach its target angle value, the common goal of regenerating the target surface is also reached. This aspect simplifies the issue. On the other hand, this goal can't be reached without the agents collaborating, as the simulation in section 4.2.1 and the failure in the prototype suggest. The agents are theoretically able to send, receive and interpret signals among each other. This gives the device the ability to

act as a multi-agent system. Besides it gives us the chance to play with algorithmic tools under realistic conditions and with a very sharply defined goal.

The chosen geometry of the mesh has shown to be suitable for architectural applications. Further development needs to continue on two levels: a) the integration into building technology and b) the improvement of the algorithms for control. Some options for a) include:

* folded roof structures which could interact directly with the buildings climatic parameters
* sound modulation through adaptive geometries of architectural spaces
* a modular, reusable concrete form-work which would render concrete shells efficient through drastic reduction of the otherwise immense formwork costs
* construction automation and change of relationship between assembler and assembly

Since the physical characteristics greatly influence the behavior of the system, the use of elastic materials could be interesting as well as the application of high performance textiles. This might aid in overcoming the observed rigidity in the system and lead to different scenarios of problem-solving. For the aspects of b) a broad range of optimization algorithms could be benchmarked and compared. In our opinion, promising approaches lie in the utilization of machine learning strategies and swarm optimization.

Acknowledgements

This work could not have been done without the cooperation with Professor Daniel Lordick at the Institute of Geometry at TU Dresden – we are sincerely grateful for the guidance and support.

References

[1] R.Chudoba, J. van der Woerd, J. Hegger, „Numerical modeling support for form-finding and manufacturing of folded-plate structures made of cementitious composites using origami pinciples", Computational Modeling of Concrete Structures, CRC Press, 2014, page 251-261

[2] Tachi, Tomohiro, Simulation of Rigid Origami, University of Tokio, 2009, http://www.tsg.ne.jp/TT/cg/ SimulationOfRigidOrigami_tachi_4OSME.pdf (found: 2013/11/24)

[3] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed Optimization by Ant Colonies. In F. J. Varela and P. Bourgine, editors, Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, pages 134-142. MIT Press, Cambridge, MA, 1992

[4] Gerardo Beni, Jing Wang, „Swarm Intelligence in Cellular Robotic Systems", Robots and Biological Systems: Towards a New Bionics? NATO ASI Series Volume 102, 1993, pp 703-712 [5] SYMBRION Integrated Project – Symbiotic Evolutionary Robot Organism, Prof. Dr. rer. nat. habil. Paul Levi University of Stuttgart

[6] Designing Collective Behavior in a Termite-Inspired Robot. Construction Team: Justin Werfel, Kirstin Petersen, Radhika Nagpal. Science, Vol 343, no 6172, 14 Feb 2014

[7] Taro Narahara, "Self-organizing Computation A Framework for Generative Approaches to Architectural Design", Harvard University Cambridge, MA, 2010, ISBN: 978-1-124-97871-0

[8] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, R. J. Wood," Programmable matter by folding", Proceedings of the National Academy of Sciences, 2010, 10.1073/pnas.0914069107

[9] Skylar Tibbits, " 4D Printing: Multi-Material Shape Change", Architectural Design, Volume 84, Issue 1, pages 116–121, January/February 2014

[10] Alexander I. Bobenko, Yuri B. Suris, "Discrete Differential Geometry: Integrable Structure", Graduate Studies in Mathematics, Vol. 98, AMS, 2008, page 27-28

.