Soft Cell Simulator: A tool to study Soft Multi-Cellular Robots

Jürg Germann, Andrea Maesani, Manuel Stöckli, Dario Floreano

Abstract— **Modular or multi-cellular robots hold the promise to adapt their morphology to task and environment. However, research in modular robotics has traditionally been limited to mechanically non-adaptive systems due to hard building blocks and rigid connection mechanisms. To improve adaptation and global flexibility, we suggest the use of modules made of soft materials. Thanks to recent advances in fabrication techniques the development of soft robots without spatial or material constraints is now possible. In order to exploit this vast design space, computer simulations are a time and cost-efficient tool. However, there is currently no framework available that allows studying the dynamics of soft multi-cellular systems. In this work, we present our simulation framework named Soft Cell Simulator (SCS) that enables to study both mechanical design parameters as well as control problems of soft multi-cellular systems in an time-efficient yet globally accurate manner. Its main features are: (i) high simulation speed to test systems with a large number of cells (real-time up to 100 cells), (ii) large non-linear deformations without module self-penetration, (iii) tunability of module softness (0-500 N/m), (iv) physics-based module connectivity, (v) variability of module shape using internal actuators. We present results that validate the plausibility of the simulated soft cells, the scalability as well as the usability of the simulator. We suggest that this simulator helps to master and leverage the potential of the vast design space to generate novel soft multi-cellular robots.**

I. INTRODUCTION

Multi-cellular robots are robots inspired by nature's concept of building increasingly complex systems out of multiple simpler subunits. The major potentials of multi-cellular robots compared to traditional robotic systems lie in increased adaptability, robustness and versatility because robots can adapt their morphology to task and environment [1], [2]. Furthermore, the systems are expected to reach unforeseen functionalities, self-repair themselves when damaged, efficiently reuse components and self-replicate. All these potentials however can only be reached when overcoming challenges in mechanical design and control of the system and its subunits [3], [4].

Previous work has addressed many of these challenges: a variety of solutions for different aspects of cell or module design have been proposed (e.g. connectivity, communication, etc.), as well as algorithms for control problems like self-assembly or self-reconfiguration [5]. Currently, a major limitation of multi-cellular systems is the general lack of flexibility when increasing the number of cells [3], [4], [5]. To address this problem, much research recently focused on simplifying module design and control in order to enable miniaturization [6], [7], [8]. Even though miniaturization might permit to solve several key issues of current systems, we believe that the fixed shape and the rigid materials used in state-of-the-art cells are another key problem preventing the full realization of the potentials of multi-cellular robots.

Recently, soft robotics became an emerging trend in robotics: inspired by the soft characteristics of abundant biological creatures and materials, soft robots hold the promise to achieve better and simpler robots for unpredictable situations by exploiting the "mechanical intelligence" of soft materials [9], [10]. By endowing robots with capabilities that are based mostly in the material properties and morphology of their body, soft robots are expected to be inherently more safe and adaptive than traditional hard systems [11].

We suggest to employ the soft design principle to multi-cellular robots to improve flexibility and adaption of these systems. However, studying the design of the single cell is challenging though, because the design space is vast and the influence of single parameters on the whole organisms are hard to predict. Computer simulations are an effective design tool to speed-up design time, parameter exploration, control issues, and keep costs at a lower bound. However, to the best of our knowledge there is currently no software framework available that allows studying soft multi-cellular systems. In this paper, we present a custom built 2D simulation tool named "Soft Cell Simulator" (SCS) that tackles the challenge of investigating large-numbered soft multi-cellular robots in a parametric way with high simulation speed at reasonable accuracy.

In the following, we first motivate our choice of simulation method and then describe the modeling used for the simulator in detail. After introducing the general flow of the algorithm, we test our implementation for the plausibility of the simulated soft cells and the scalability of the simulator when increasing number of cells or membrane resolution. To illustrate the practical usability of the simulator, we present results on protein-inspired self-folding of soft multi-cellular robots.

II. SIMULATION METHOD

Thanks to a continuous increase in computational power, the simulation of soft bodies gets more readily available. Still, there are few tools available that allow the systematic study and simulation of the design of soft robots. One prominent simulation environment is Hiller and Lipsons VoxCAD [12], a 3D simulation tool to design robots composed of voxels with different material properties (soft, rigid,

This research is supported by the Swiss National Foundation through the National Center of Competence in Research Robotics.

J. Germann, A. Maesani, M. Stöckli, R. Pericet-Camara and D. Floreano are with the Laboratory of Intelligent Systems, School of Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
Email: jurg.germann@epfl.ch

For video footage of SCS: http://lis.epfl.ch/softSRrobots thermodynamic behavior, etc). This tool however is made for fixed-morphology robots and does not support the simulation of cellular robots where cells have the ability to attach and detach to another based on a specific physical connection principle. Since there is - to the best of our knowledge - currently no other framework available that would allow the study of soft multi-cellular robots, we decided to develop our own tool. Developing a robotic simulation however means facing the trade-off between adequate physical accuracy and computational efficiency. In order to simulate large-scale modular systems, high simulation speed is of prime importance. Thus, using high precision simulation methods like Finite Element Modeling or other computationally intensive physics simulation tools were not considered here.

Real-time physics engines are mostly being used for video games or other forms of interactive computing. These engines use more simplified calculations and decreased accuracy in order to enable fast computation time. Physical phenomena are only approximate and often simulations are trimmed to obtain visually correct results. Still, when high simulation speed becomes a priority over simulation precision, these engines offer a viable alternative.

Even though real-world physics are 3D, we decided to use a 2D simulation environment. Because, as a first step, it offers a technically much simpler and faster approach. For sake of generality, the formal modeling presented in the following section is still kept in 3D.

Our implementation is based on the Box2D physics engine [13], which is an open source rigid body solver written in C++. Box2D is stable, adaptable, computationally efficient and has an already built-in visualization testbed. We developed SCS by implementing a soft module modeling, enabling inter-module connectivity and module deformation mechanisms. SCS is a completely object-oriented environment, thus allowing easy integration of different types of connectors, deformation mechanisms or soft module modelings.

III. MODELING

Our goal was to develop a simulator to aid in the design and control of a system based on entirely soft cells. In particular, we wanted a simulator that allows us to explore soft cell parameters and system self-organization strategies like self-assembly, self-repair, or self-reconfiguration. The soft cells should have the capabilities of varying their softness, changing their outer shape and connecting to other cells in a physically plausible way. Furthermore, our requirements were (i) high enough simulation speed to test large numbered systems (minimum of 1000 cells), (ii) large non-linear deformations without module self-penetration, (iii) high tunability of cell softness (more than 100x stiffness change) , (iv) physics-based module connectivity (e.g. magnetic connectivity), and (v) variability of module shape using internal actuators (like e.g. using artificial muscles). In the following we describe in more detail our choice of simulation method and modeling of different mechanisms.

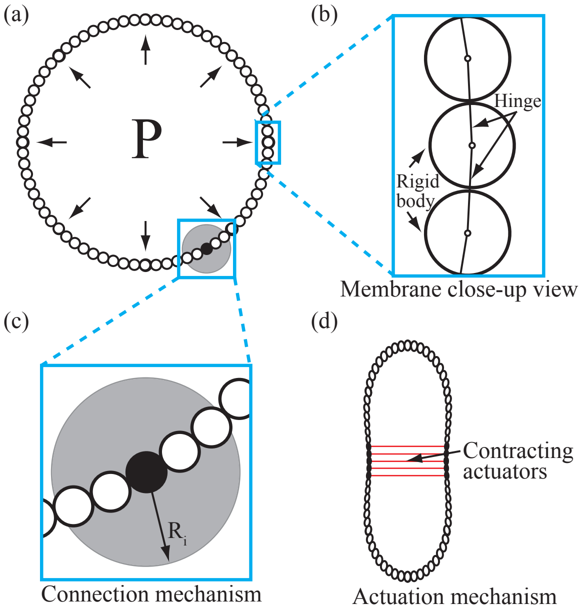


Fig. 1: Soft cell modeling: (a) cells are round in shape and have parametric softness based on internal pressure, (b) the membrane of the soft cells is composed of multiple small rigid bodies that are connected by hinges, (c) each membrane body can bear a connection mechanism with a specific interaction radius R_i, (d) contracting actuators can be placed between the membrane bodies.

A. *Pressure based model for soft cells*

The defined global shape of our cells without external interaction is round (Fig. 1a), similar to soap bubbles or many single biological cells that have a round (spherical) shape because this minimizes their total surface energy [14].

The membrane of a soft cell is composed of multiple small rigid bodies, connected by hinge joints to form a closed chain (Fig. 1.b). The hinges enable highest flexibility between the membrane bodies, thus allowing all kinds of inelastic transformations. Deliberately, we did not enable elasticity between the small membrane bodies, because this would easily result in cell inter-penetrations when trying to bring multiple soft cells together.

In order to give the cells parametric softness, we implement a simple yet effective model based on thermodynamics that enables fast physical simulation [15]. Most other existing models for modeling softness are rather complex to implement and come with slow solving speed [16]. In principle, pressure is a force acting on incremental surface elements and is normal to the surface. The expression for pressure in a specified point on a surface is then:

$$\vec{P} = P\vec{N}$$

where P is a pressure value and N is the normal vector to the surface which the pressure force is acting. In order to obtain the pressure force, we multiply the pressure value with the area of the surface A. This leads to the expression:

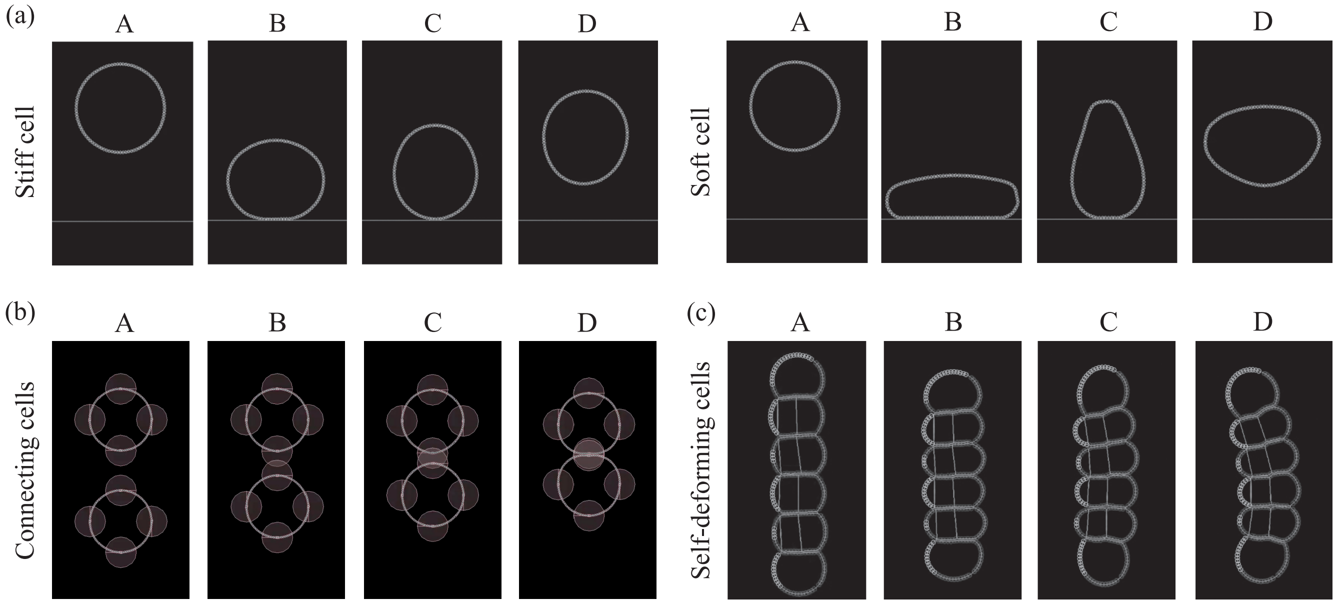


Fig. 2: a) Sequence of a soft cell bouncing on a surface under high internal pressure (left) and low internal pressure (right), (b) sequence of two cells forming a connection once their interaction radii overlap, (c) sequence of a self-deforming multi-cellular structure: the four cells in the middle of the organisms have actuation mechanisms (artifical muscles) that can contract on command.

$$d\vec{F} = \vec{P}dA$$

To calculate the pressure force value P, we use the thermodynamic approximation known as the Ideal Gas Law. This approximation focuses more on the macroscopic effects of gas presence, and does not take into account actual interactions among gas particles. The Ideal Gas Law gives a simple relation between pressure P, gas temperature T, body volume V, amount of substance of gas n and the ideal gas constant R:

$$pV = nRT$$

From this equation we can extract a pressure value if we know the volume of the body, gas temperature and the number of gas molecules:

$$P = \frac{nRT}{V}$$

We assume temperature to be constant and only the volume of the soft body to change. Thus, for every simulation step we compute the variation of the soft body volume and apply a reactive force due to internal pressure on every membrane body

$$F_{react} = C(\frac{V_{init}}{V} - 1)$$

where C is the constant defining the internal pressure, V_{init} the initial volume and V the actual volume of the soft body.

Figure 3.a illustrates the difference in dynamics between a soft and a stiff cell when colliding with an even surface.

B. Connection mechanism model

In order to give soft modules the ability to interact and to attach together to form various structures, we required the implementation of a physically meaningful connection mechanism system between two or more cells.

The implementation permits to bound a connection mechanism to any rigid body of the membrane (see Fig. 1.c). Every connector has parametric connection strength and interaction radius. If two or more mechanisms from different cells overlap with their interaction radii, the connectors become active. Then, forces according to a specific force-distance relation are applied to the centers of the rigid membrane bodies that carry the connectors. Connections break either if at least one of the two mechanisms is turned off or if the cells are pulled apart due to external forces.

To give a more practical example of a mechanism, the connection type used in the upcoming sections of this paper is based on magnetic forces. The force-distance relation between two small magnets can be approximated as [17]:

$$F = \frac{\mu q_{m1} q_{m2}}{4\pi r^2}$$

where μ is the permeability of the intervening medium (here: air), q_{m1} and q_{m2} are the magnitudes of magnetic poles and r is the separation of the magnets.

Figure 2.b shows an example of two cells coming in contact within their interaction radius and connecting.

C. Actuation mechanism model

Soft cells should not only have the ability to passively deform, but to actively alter their outer global shape similar to biological muscle cells. Coordinated active shape change of multiple cells can then result in specific motion or behaviors of the system.

Shape change could happen in multiple ways. In order to have a plausible and practically realizable model, we enable shape change through actuators that may be set between any two bodies of the membrane (Fig. 1.d). In simulation the controlled parameter is the length of the actuator between the two membrane bodies. This modeling obviously is simplistic, still in engineering practice it is somewhat similar to linking two points of a membrane with an artificial muscle (e.g. Shape Memory Alloy). As long as the muscle is not actuated, there is no major influence on the global shape of the cell nor on its deformability. Contracting the muscle results in deformation of the global shape of the cell.

Figure 2.c gives a qualitative example of a self-deforming soft-multicellular organism. Here, the artificial organism can globally change shape from a straight to a curved line.

IV. ALGORITHM

During one timestep of the simulated environment, SCS performs the following operations:

```
for one timestep do
    /* update single cells                      */
    get position and velocity of all membrane bodies;
    get center of gravity and compute surface of cell;
    compute and apply forces on every membrane body
    according to internal pressure model;
    /* update connection mechanism              */
    apply connection mechanism forces;
    /* update actuation mechanism               */
    apply actuation mechanism forces;
    /* update other                             */
    perform collision detection, integration, and
    constraint solution;
end
```

Algorithm 1: SCS algorithm.

V. EXPERIMENTS

We tested our implementation against our most important requirements: first, we characterized the soft behavior of a single cell; second, we tested the scalability of the simulator when increasing the number of cells and number of membrane bodies; third, we present a practical example that illustrates a potential application and demonstrates the usability of the simulator.

A. Experimental setup

For all experiments the number of rigid bodies per soft cell is kept at 100 units if not otherwise mentioned. This empirical value proved for us to be a good match between simulation speed and accuracy

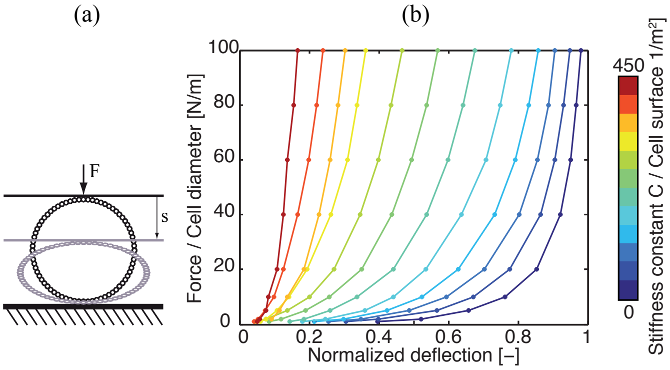


Fig. 3: a) Sketch of the force-deflection measurement setup in simulation, (b) obtained data for different internal pressure constant settings and gradually increased force.

For the integrator of the simulator we use a timestep set to $1/50s$. Below this value we observe unstable simulations with high vibrations or other unrealistic physical behaviors.

For the constraint solver it is important to use a relatively high iteration count both for the velocity phase and the position phase. This is because membranes tend to interlock when two cells come into contact if these values are set too low. Here, we set the count to 200 for velocity and 100 for position. Obviously, using that many iterations decreases performance but improves the quality of our simulation.

We run SCS on a MacBook Pro Model 2009 (Apple Inc.) for all tests carried out in this paper.

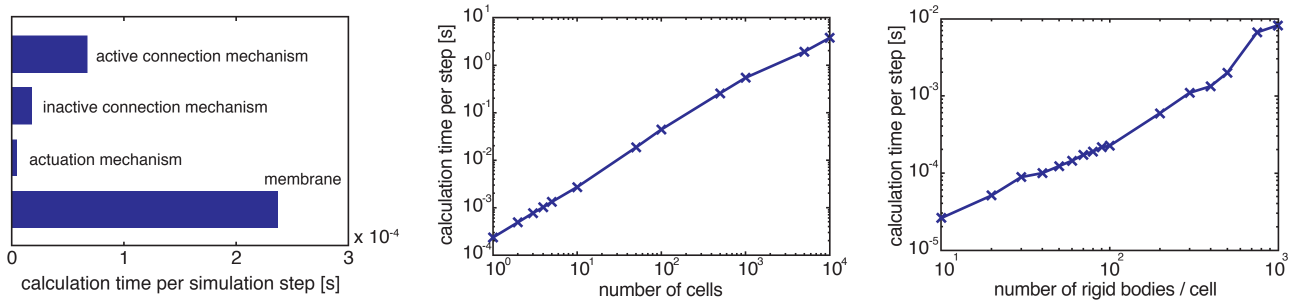
B. Soft cells

In order to test the implementation of the soft cells and to characterize the influence of the pressure constant C, we perform a force-deflection measurement as illustrated in Fig. 3.a. In the experiment, a soft cell is bound to a flat horizontal ground and then put under different compression loads by a horizontal bar. The pressure constant is increased to a maximum value of 450 m^{-2}, which marks the limit for stable simulations.

The obtained results are shown in Fig. 3.b. As can be seen in the graph, the relation between force and deflection resembles the one of a progressive spring [20], which is reasonable given the pressure-based modeling. Also, the deflection decreases soundly with an increase of the internal pressure. The maximum measured stiffness of a cell under maximum pressure is higher than $500N/m$, which satisfies our requirement of having a widely tunable softness.

C. Scalability

In order to test the scalability of the simulator, we first characterize the repartition of calculation time for one simulation step between soft membrane, connection mechanism and actuation mechanism. The obtained data is shown in Fig. 4.a. The calculation of the soft membrane with its internal pressure takes up the most time whereas connection and actuation mechanism are less computationally expensive. As can be seen in the figure,



(a)

(b)

(c)

Fig. 4: a) Calculation time per simulation step for the individual components of a soft cell: the soft membrane with its internal pressure takes up the most time whereas connection and actuation mechanism are less expensive. b) Calculation time per simulation step for gradually increased the number of cells. The time increases linearly up to the tested maximum number of cells. Real-time simulation speed is reached at approximately 100 cells. c) Calculation time per simulation step for the membrane of a soft cell with gradually increased resolution (number of rigid bodies). The calculation time increases to a large extend linearly within the tested domain.

there is a higher calculation time for an active connection mechanism compared to an inactive (not connected) one.

In a second experiment we evaluate the calculation time for one simulation step for an increasing number of soft cells. The obtained data is presented in Fig. 4.b. The calculation time increases linearly in the tested domain up to 10'000 cells. Considering the set simulator speed set at $50Hz$ and an average of $0.3ms$ calculation time per simulation step for one cell means that real-time simulation speed is ensured up to approximately 100 cells.

In a third experiment we vary the number of rigid bodies in the membrane of one soft cell. The resulting calculation times per simulation step are illustrated in Fig. 4.c. As can be identified in the figure, there is a slight tendency of non-linearity for a large number of bodies in one element. This is most probably due to the decreasing size of the individual rigid bodies. Nevertheless the simulator is reasonably scalable up to the tested limit of 1'000 rigid bodies per soft cell.

D. Application: Self-Folding of Soft Multi-Cellular Robots

To illustrate the usability of the simulator, we present qualitative results of our work on self-folding soft multi-cellular robots.

A major challenge for multi-cellular robots is to self-assemble from a number of unconnected single cells to a connected organism with a functional shape. A powerful strategy is protein-inspired assembly often simply called "Self-Folding" [18]. Typically, subunits are connected through permanent free joints, which means the subunits must remain attached at these points, but are free to move about them. Local interactions between adjacent subunits then decide in which way the chain is being folded. Given the system composed of different subunits, a folding sequence for many different shapes can be specified. The main advantages of that approach are its deterministic process, its simplicity in module design and possibility to fold asymmetric structures easily.

Our simulation tool helps us to investigate the design of the cells as well as the folding process. In order to enable the soft cells to self-assemble by folding, our design for the distribution and choice of connection sites is inspired by the design presented in [19]: four different cell types have programmed magnetic attracting (or repelling) areas and allow to fold a wide variety of shapes. To enable compatibility with soft cells, we put multiple connections per attraction site. This has important advantages: more connectors allow to reduce the attraction magnitude as well as the interaction radius of the single connector. Long-range interactions could potentially lead to undesired connections and unforeseen structure shapes, because the softness might allow attractions that are otherwise prevented in rigid cells.

For the folding process, cells are first pre-aligned in a chain before applying any forces to the system. The cells are sorted given the programmed sequence and connected through permanent joints. Once set, the magnetic attractions of the the cells are activated. Thus, all the folds between all the cells happen in parallel. For the simulation we assume a totally frictionless, gravity free and unconstrained environment.

Figure 5.a shows snapshots of the folding process of a linear chain folding into the letter 'E'. Using the same basic cells but different sequences, we present three other instantiations of this process in figure 5.b forming the letters 'P', 'F' and 'L'.

VI. CONCLUSION

We believe that by exploiting the soft design paradigm enables to overcome essential limitations (e.g. lack of flexibility) as well as to improve functional properties (e.g. adaptability) of current multi-cellular robotic systems. On the way to validate this hypothesis, we developed and presented here a novel tool for the simulation of soft multi-cellular robots named Soft Cell Simulator (SCS). SCS enables fast

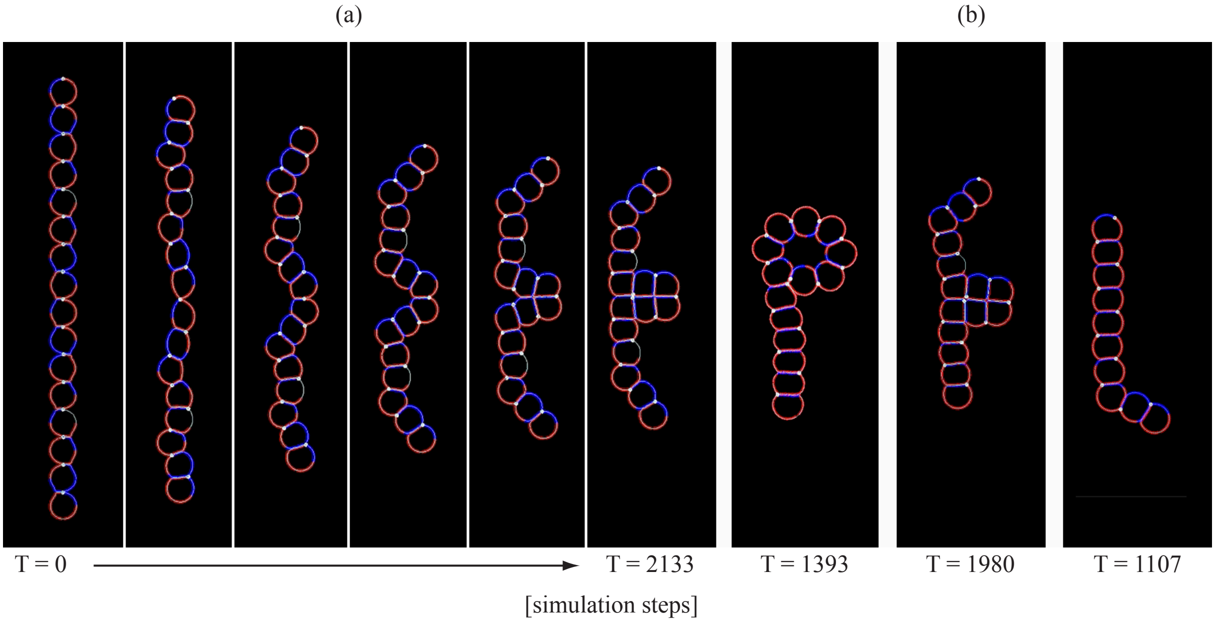


Fig. 5: Our custom built simulation tool helped us to investigate the design and the folding process of self-folding soft multi-cellular robots. (a) Folding steps of a sequence that maps into the letter 'E'; (b) three other instantiations of the product of the folding process, namely the folded letters 'P', 'F', 'L'.

and efficient study of the mechanical cell design as well as control of large numbered systems. Specifically and uniquely, the simulated cells can undergo large passive non-linear deformations, change its softness, connect to other cells and alter its shape globally.

In the future, when multi-cellular robots further decrease in cell size and increase in cell number, prediction of the influence of single cell parameters on the whole system becomes more and more difficult. Thus, we expect that design explorations based on efficient simulation tools as the one presented here will become increasingly important for the development of scalable design and control approaches.

VII. OPEN SOURCE CODE

All the Soft Cell Simulator (SCS) software is available at: https://github.com/lis-epfl/scs

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge Bryan Schubert, Stefano Varricchio and Ramon Pericet-Camara for the helpful comments on the manuscript.

REFERENCES

[1] R. Pfeifer, M. Lungarella and F. Iida, "Self-Organization, Embodiment, and Biologically Inspired Robotics", *Science*, vol. 318, 2007, pp 1088-1093.

[2] S. Murata and H. Kurokawa, "Self-Organizing Robots", *Springer Tracts in Advanced Robotics*, vol. 77, 2012.

[3] M. Yim, W-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins and G. Chirikjian, "Self-Reconfigurable Robot Systems", *IEEE Robotics & Automation Magazine*, vol. 14, pp. 43-52, 2007.

[4] S. Murata and H. Kurokawa, "Self-Reconfigurable Robots", *IEEE Robotics & Automation Magazine*, vol. 14, pp. 71-78, 2007.

[5] K. Stoy, D. Brandt and D.J. Christensen, "Self-Reconfigurable Robots: An Introduction", *MIT Press*, 2010.

[6] E. Diller, P. Chytra, S. Floyd, and M. Sitti. "Assembly and disassembly of magnetic mobile micro-robots towards deterministic 2-D reconfigurable micro-systems." *The International Journal of Robotics Research* vol. 30, no. 14, pp. 1667-1680, 2011.

[7] P. White, K. Kopanski, and H. Lipson, "Stochastic self-reconfigurable cellular robotics", *in IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2888-2893, 2004.

[8] D. Rus and K. Gilpin, "Modular Robot Systems From Self-Assembly to Self-Disassembly", *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 3855, 2010.

[9] R. Pfeifer, F. Iida and J. Bongard, "New robotics: design principles for intelligent systems", *Artificial Life*, vol. 11, pp. 99-120, 2005.

[10] D. Trivedi, C.D. Rahn, W.M. Kier and I.D. Walker, "Soft robotics: biological inspiration, state of the art, and future research", *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99-117, 2008.

[11] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics", *Trends in Biotechnology*, vol. 31, no. 5, pp. 287-294, 2013.

[12] J. D. Hiller and H. Lipson, "Evolving Amorphous Robots", *Transform*, pp. 717-724, 2010.

[13] E. Catto, "Box2d: A 2d physics engine for games.", *available at www.box2d.org*, 2013.

[14] T. Lecuit and P.-F. Lenne, "Cell surface mechanics and the control of cell shape, tissue patterns and morphogenesis.", *Nature reviews. Molecular cell biology*, vol. 8, no. 8, pp. 633-644, 2007.

[15] M. Matyka, and M. Ollila. "Pressure model of soft body simulation." *in Proceedings of Sigrad*, pp. 29-34, 2003.

[16] G. Desbunne, M. Desbrun, M-P. Cani and A. Barr, "Adaptive simulation of soft bodies in real-time", *Computational Animation 2000 Proceedings*, pp. 15-20, 2000.

[17] R. Resnick, D. Halliday, and J. Walker, "Fundamentals of physics". *John Wiley*, 2008.

[18] Pelesko, John A. "Self assembly: the science of things that put themselves together.", *CRC Press*, 2007.

[19] S. Griffith, "Growing machines", *PhD thesis, MIT*, 2004.

[20] Society of Automotive Engineers, Inc., "Spring Design Manual", *SAE International*, 1996.