2014 International Symposium on Nonlinear Theory and its Applications
NOLTA2014, Luzern, Switzerland, September 14-18, 2014

Employing L-systems to Generate Mass-Spring Networks for Morphological Computing

R. Bernhardsgrütter†, C. W. Senn†, R. M. Füchslin†, C. Jaeger†, K. Nakajima‡ and H. Hauser♯

† Institute of Applied Mathematics and Physics, Zurich University of Applied Sciences, Winterthur, Switzerland
‡ Hakubi Center for Advanced Research, Kyoto University, Kyoto, Japan
‡ Nonequilibrium Physics and Theoretical Neuroscience, Department of Applied Analysis and Complex Dynamical Systems, Graduate School of Informatics, Kyoto University, Kyoto, Japan
♯ Artificial Intelligence Laboratory, University of Zurich, Zurich, Switzerland

Abstract—Robot arms equipped with an additional network of masses and springs enable a facilitation of control by the exploitation of morphological properties. Typically, these mass-spring networks are being generated randomly. This approach comes in its basic form unhandy when the network should obey specific constraints, such as spring lengths, angles between springs or, e.g., the physical expansion range of the network. We present an approach of emulating a growth process on the basis of L-systems. Therefore, we have developed a simulation environment to define L-systems and instructions to translate the produced strings into mass-spring networks.

1. Introduction

Contemporary robotics seems to experience limitations when it comes to the construction of systems that are compatible with requirements of environments and task schemes as they are found in complex environments, e.g., in a hospital. These limitations are manifold, including adaptivity to new situations, robustness and, last but not least, intrinsic safety. There is an apparent tension between contemporary design of robots and the way how natural evolution implements its most versatile moving systems: The former are most often stiff and equipped with a number of degrees of freedom (DoF) as small as possible, the latter are made of rather soft materials and exhibit consequently a large number of DoF. Conventional robot designs aim at a small number of DoF as they imply a facilitation in modelling and controlling the system (both with respect to computation as well as sensory bandwidth necessary for a sufficiently precise determination of the state of the system). In recent years, the concept of "soft robotics" has gained growing attention [1, 2, 3]. There are several reasons for this trend, among them are the fact that the concept of soft robots matches well with the spirit of bio-inspired engineering or the hope that soft robots can be made intrinsically safer than stiff and heavy systems. A main obstacle for the implementation of soft systems is the apparent need for more complex control protocols. The concept of morphological computation offers a way to overcome this difficulty: Instead of delegating control to some control unit, the dynamics of the soft body itself is made a part of the control scheme, i.e., part of the control can be outsource to the physical body of a robotic platform.

The term "morphological computation" can be loosely defined as the exploitation of the shape, material properties, and physical dynamics of a physical system to improve the efficiency of a computation; e.g. the computation needed for the control of a robot. Thereby, the physical dynamics of the robot is made part of the control process itself. A simple example is the exploitation of the self-stabilitzation property of limit cycles. For a detailed description of morphological computation see [8, 9, 10]. The term "computation" is justified, as demonstrated by Hauser et al. in [4]. was shown that a sufficiently large parameterized mass-spring system[1] is in fact able of emulating a large family of different computations, which could, e.g., be used for control.

One way for the implementation of morphological computation uses the morphology of the robot as a reservoir in the sense of reservoir computation. Compared to classical artificial neural network, reservoirs have improved learning speed as no recurrent weights have to be learned [4]. Furthermore, reservoirs can be used in a multitasking context, i.e., multiple computations can be carried out simultaneously with the same reservoir [5]. During the learning process, weights are calculated with the help of linear regression. These parameters are then used to calculate the output based on a weighted sum of the (partial) inner state of the reservoir. Hauser et al. [4, 5, 6] have demonstrated that a compliant body (a morphology consisting of mass-spring systems) can be employed as such a reservoir.

Following this approach, our proposed setup consists of a compliant network of nonlinear springs and

[1]By term mass-spring systems we refer physically realistic systems, i.e., damped mass-springs.

masses attached to a rigid robot arm. This compliant physical body – the mass-spring network – serves as a reservoir [6]. Usually such networks are created randomly [5]. This task can be challenging, as random networks might not be suitable for a real-world scenario, e.g. given shape constraints.

In this work we used L-systems [7] in order to generate networks in respect to a bio-inspired design. This allows to easily parameterize the creation process of mass-spring networks and, therefore, enables one to create networks according to specific needs.

2. Methods

We used a simulation of a robot arm, based on the Bullet physics engine,[2] to explore the dynamics of mass-spring networks. The task of the robot arm was to follow a given trajectory.

2.1. Robot Arm and Mass-Spring Network

Networks qualify for reservoir computing when they are dynamic, complex, nonlinear, and provide a fading memory [4]. Figure 1 shows a perpendicular robot arm with a small mass-spring network.[3]

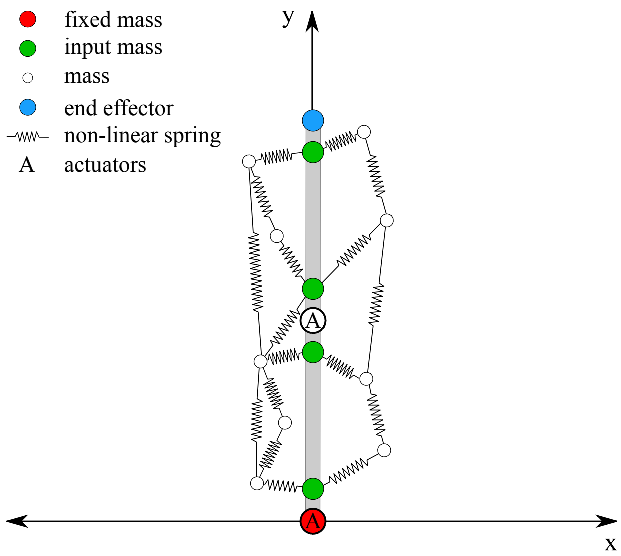

Figure 1 A robot arm with a mass-spring network, which serves as a reservoir, consisting of a shoulder, an elbow, and an end effector. The network is attached to the arm via four input nodes, located next to the joints.

At the base, a *fixed mass* represents the shoulder. It acts as joint with an actuator (A) so the arm can be moved. The elbow, in the middle of the robot arm, is also an active joint (A) and controls the angle between the upper and the lower arm.[4] The *end effector* is designated to follow a target trajectory. The mass-spring network is fixated to the arm segment by the green nodes called *inputs*. When the arm moves, forces are transmitted via these nodes to the network and, thus, change the state of the reservoir. Furthermore, there are one or more, fixed nodes, which act as an anchor and give the network an orientation in space.

2.1.1. Growing Networks Using L-systems

Typically, reservoirs of mass-spring networks are randomly generated, i.e., that a random or given number of masses are randomly placed around a robot arm and springs connect randomly selected masses with each other [4, 5, 6]. This work uses a different approach. We "grow" the networks using so called L-systems. L-systems, introduced by A. Lindenmayer [7], are based on formal languages (in its most fundamental form a parallel rewriting system in the sense of a grammar in the theory of formal languages). As being demonstrated by various authors (e.g. Prusinkiewicz and Lindenmayer [11]), the L-system approach is able of reproducing an astonishingly large variety of rather complex-looking artificial plants by using rather simple algorithmic rules. This gives the benefit of having a certain amount of control over the network structure – like the size or certain properties of the shape – and, furthermore, the string representation is fit for a usage with evolutionary algorithms. Typically, L-systems are purely deterministic. It is, however, easy to add some amount of randomness to the growth process. We applied a similar approach, since it improved the usability of the grown networks as reservoirs.

Growth Process As to use L-systems for the generation of mass-spring networks, we need a grammar with a set of production rules for terminal and non-terminal symbols, and we have to link these symbols to *instructions* – for example "create a mass" or "create a spring" – in order to *translate* the produced L-system string to a mass-spring network.

Let our grammar have the alphabet $\Sigma = \{A, B, a, b\}$, which translates the instructions

- $a \Rightarrow$ *mass creation* $[r_1, r_2]$ with inner radius $r_1 = 2$ and outer radius $r_2 = 4$, and
- $b \Rightarrow$ *spring creation* $[r_1, r_2]$ with inner radius $r_1 = 2$ and outer radius $r_2 = 4$.

[2] http://www.bulletphysics.org

[3] The mass-spring network shown in Figure 1 would probably not perform well since it might have too few masses and springs. Figure 1 has only the purpose of illustrating the concept schematically.

[4] *Upper arm* is called the segment between the shoulder and the elbow. Subsequently, the segment between elbow and end effector is called *lower arm*. This notation does not depend on the alignment of the robot arm.

Further, we provide the following production rules

- $A \rightarrow BAB$ and
- $B \rightarrow abb$.

The parameters r_1 and r_2 allow to limit the *area for random creation* to a circle ($r_1 = 0$) or a ring ($r_1 > 0$) around the position of the *interpreter*, which reads the string to translate symbol-by-symbol and *executes* it, thus creates new masses and springs.

2.2. Learning Phase

The learning follows the supervised procedure explained in [4], which we summarize briefly. The procedure is described for one actuator and must be repeated for each actuator used. The desired trajectory is discretised with a suitable time step. At each step, the angular velocity $v^{(\text{pre})}$ of the actuator is calculated and the spring lengths are recorded while moving along the trajectory. This leads to a matrix $L = (l_{ij})$ containing at position (i, j) the length of spring j at time t_i. These spring lengths are superimposed with white noise, a procedure which has been shown to increase robustness against perturbations [4]. The data from an initial period (the *washout-time*) is dismissed to exclude any transients from the start of the learning process. Based on the recorded settling times after a fast 1° swiveling movement, the washout-time was set to $t_w = 300$ simulation steps. The weights $w = (w_j)$ for each spring are then calculated by solving the linear equation system $L\,w \stackrel{!}{=} v^{(\text{pre})}$ in the least squares sense [4].

After this learning procedure, the robot should be able to follow the learned trajectory without guidance and controlled solely by the mass-spring network, using the angular velocities $v^{(\text{morph})} := \sum w_j l_j$ calculated from the actual spring lengths l_j measured during the motion and this movement is expected to be robust against perturbations.

2.3. Exploitation Phase

We call the stage, where the system runs freely and without guidance the *exploitation phase*. To start this phase, we need an initialization as follows: the robot arm is guided along one complete cycle of the target trajectory. After this, at time $t = 0$, all initial transients in the mass-spring network are expected to have died out and control is smoothly transferred to the network by applying the angular velocity $v^{(\text{trans})} := \lambda\, v^{(\text{pre})} + (1 - \lambda)\, v^{(\text{morph})}$ to the actuators for a suitable monotonous transition function λ with $\lambda(t = 0) = 1$ and $\lambda(t \geq t_1) \equiv 0$. Empirical evaluation has shown that a fade-over time t_1 corresponding to half a cycle of our trajectory led to stable results. This smooth fade-over of the control to the morphology was necessary in all our simulations to guarantee stable morphological control.

3. Trajectory Reproduction

We will show a successful trajectory reproduction, accomplished with a mass-spring network generated by an L-system.

By *successful* we mean, that the robot arm was able to approximately follow a learned trajectory in a stable manner.

Mass-Spring Network We used the mass-spring network shown in Figure 2. This specific network performed best among other tested ones. It consists of 19 masses and 43 springs. The illustration shows the noteworthy *airiness* of the network – most of the springs are longer than 7 length units, which equals to $\frac{7}{10}$ of a arm segment, and only a few network masses are connected via more than three springs.

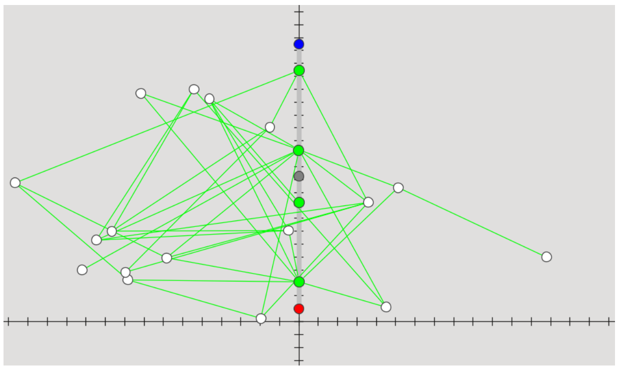

Figure 2 The mass-spring network grown using an L-system and used in this simulation.

Exploitation Phase The robot arm managed to follow the learned trajectory, although the whole circle *drifted* away over time, as shown in Figure 3.

The dashed line represents the target trajectory. The robot arm was initialized with the end effector located at the (x/y)-coordinates (0/21) and was then guided through the *warm-up* sequence. Thus, the first cycle is exactly the same as the learning trajectory. The control is then slowly taken over by the robot arm and its morphology, again during one cycle. After five and a half cycles, the simulation was stopped.

The reservoir was able to control the robot arm and create a stable approximation of the learned trajectory, if we ignore the drifting.[5]

[5] A detailed analysis of different mass-spring networks – also randomly generated ones – may be reasonable in order to evaluate performance varieties and estimate how different aspects of a network effect its capability to control the robot arm.

Robustness Interestingly, despite the drift away from the nominal trajectory the system was to a certain extent robust against external perturbations. This was shown in a simple robustness test, where we applied a force of 5 N to all masses parallel to the x-axis for a short amount of time to simulate a *blow*.

Figure 3 shows the corresponding experimental results. The morphological computation setup was able to recover from two applied blows, approximately within one cycle.

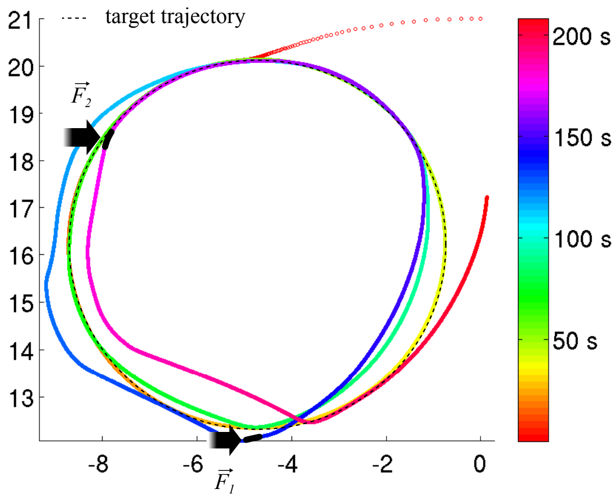

Figure 3 The mass-spring was able to handle two strong externally applied perturbations and did not lose control. This is visualized by the forces $\vec{F}_1$ and $\vec{F}_2$, applied during the black colored periods next to the arrows.

Less stable networks typically lost control after such a perturbation. An often occurred behavior was that the end effector moved off the target trajectory and stopped there. Hence, the setup shown in Figure 3 seems to be capable of handling externally influenced forces without losing control.

4. Outlook

For future research we suggest the systematic evaluation of the performance of grown mass-spring networks in comparison to randomly generated ones. Together with more extensive research regarding the structure of networks grown by L-systems could us provide potentially with general growing rules and, hence, with implicit design rules.

Acknowledgments

This paper is based on parts of the bachelor's thesis with the title "Exploitation of Morphological Properties to Control a Robot Arm," written by R. Bernhardsgrütter and C. Senn at the Zurich University of Applied Sciences under the supervision of R. Füchslin and C. Jaeger and in scientific collaboration with H. Hauser and K. Nakajima.

References

[1] D. Trivedi et al., "Soft robotics: biological inspiration, state of the art, and future research," *Appl. Bionics. Biomech.*, vol. 5, no. 3, pp. 99–117, Sept. 2008.

[2] R. Pfeifer et al., "The challenges ahead for bio-inspired 'soft' robotics," *Commun. ACM.*, vol. 55, no. 11, pp. 76–87, Nov. 2012.

[3] S. Kim et al., "Soft robotics: a bioinspired evolution in robotics," *Trends Biotechnol.*, vol. 31, no. 5, pp. 287–294, May 2013.

[4] H. Hauser et al., "The role of feedback in morphological computation with compliant bodies," *Biol Cybern*, vol. 106, p. 603, Feb. 2012.

[5] H. Hauser et al., "Towards a theoretical foundation for morphological computation with compliant bodies," *Biol Cybern*, vol. 105, p. 355, Apr. 2012.

[6] H. Hauser et al., "Moving a Robot Arm by Exploiting its Complex Compliant Morphology," in *Proceedings of the 2nd International Conference on Morphological Computation*, Venice (Italy), 2011, p. 1.

[7] A. Lindenmayer, "Mathematical Models for Cellular Interactions in Development, Parts I and II," *J. Theoret. Biol.*, vol. 18, pp. 280–315, 1968.

[8] R. Pfeifer et al., *How the Body Shapes the Way We Think: A New View of Intelligence*. Cambridge, MA: MIT Press, 2007.

[9] R. Pfeifer el al., "Self-Organization, Embodiment, and Biologically Inspired Robotics," *Science*, vol. 318, pp. 1088–1093, 2007.

[10] R. Füchslin et al., "Morphological Computation and Morphological Control: Steps Toward a Formal Theory and Applications," *Artificial Life*, vol. 19, pp. 9–34, 2013.

[11] P. Prusinkiewicz, A. Lindenmayer, "The Algorithmic Beauty of Plants," New York: Springer-Verlag, 1990.