Moving a Robot Arm by Exploiting its Complex Compliant Morphology

Helmut Hauser
Artificial Intelligence Laboratory
Department of Informatics - University of Zurich
Email: *hhauser@ifi.uzh.ch*

Gernot Griesbacher
Institute for Theoretical Computer Science
University of Technology Graz - Austria
Email: *griesbacher@igi.tugraz.at*

***Abstract*—The vision of morphological computation proposes that the complexity of compliant bodies of biological systems is not accidentally, but rather that it can contribute to the computations, which are needed for a successful interaction with the environment. We demonstrate in a simulation that a compliant, highly nonlinear body (simulated as a random network of masses and springs) can serve as a computational resource, which allows the end-effector of a two-link robot arm to move autonomously on a complex trajectory. Remarkably, simple linear and static feedback loops from the state of the compliant structure back to the robot arm torques suffice. This suggests that by outsourcing parts of the nonlinear and dynamic computation to the compliant morphology the remaining computational task is much simpler and can be even represented by some static, linear weights.**[1]

I. Introduction

The bodies of biological systems are very complex structures. They consist of numerous parts made out of various materials and they interact in a highly nonlinear and dynamic fashion. From the control theoretical point of view, the control of such dynamical systems presents a major challenge. Despite this fact, animals control their complex physical bodies with an astonishing ease. Moreover, they are able to learn and adapt the control of their complex bodies. Seeking for biological inspiration the question arises, how do biological systems deal with that challenge? One possible answer is provided by the vision of morphological computation and has been strengthen lately by related theoretical results by Hauser et al. [2], [1]. It proposes a paradigm shift: The complexity of the physical bodies of animals are not sub-optimal results of the evolutional process, but rather such bodies can potentially be used as computational resources. They can be controlled, not despite, but because of their complexity. This implies that computations (or at least parts of it), which are needed for a successful interaction with the environment, can be outsourced to the physical body. As a consequence, the task of controlling the complex body and even to learn and adapt behavior can be reduced in their complexity. Hauser et al. [2], [1] showed that randomly constructed networks[2] of masses and nonlinear springs (which could be used as a model to described complex bodies of biological systems as well of compliant robots) can be employed for a range of computation, which include

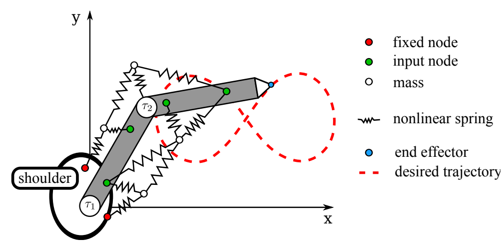


Figure 1: Scheme of the setup of the considered task.

nonlinear operations and memory. A particularly interesting class of computation can be carried out, when a feedback loop (or multiple loops) is established. After an initial supervised learning phase, such morphological computation devices are able to produce, e.g., autonomously nonlinear limit cycles, which could be then used, e.g., for locomotion. A remarkable result of Hauser et al. [1] is that if the physical body is sufficiently high-dimensional and nonlinear in their dynamics, even simple *static* and *linear* feedbacks can be used to emulate (with the help of the body) *nonlinear*, *dynamic* computations. This implies that a complex body allows to reduce the task of learning nonlinear computations (which can include even persistent memory) to the much simpler task of finding some linear, static weights. Moreover, it has been shown that noise during learning is crucial if feedback is involved, since it provides the necessary robustness of the learned limit cycle. Note that noise is the standard situation in real-world applications.

While Hauser et al. [2], [1] presented mathematical proofs and supported their results with simulations of abstract networks, we will present here a direct application of their theory. We will show, that a rigid robot arm augmented by compliant body parts, which were implemented as a randomly constructed network of masses and springs (as seen in Figure 1), is able to reproduce a complex, repetitive trajectory using only linear and static feedbacks from its compliant "body" (i.e., the state of the body) back to the torque inputs. The learned trajectory is robust to a certain extend and the proposed setup is able to recover from external perturbations.

[1]The work has been partly supported by the Swiss National Competence Center for Research in Robotics.

[2]Note that their approach is closely related the concept of reservoir computing (RC), see [3] for a review on RC.

II. Task Description

The task was to learn to move robustly the end effector of a robot arm along a desired trajectory by using a randomly constructed morphological structure, which was attached to the robot arm. Figure 1 shows a scheme of the considered setup and the desired end-effector trajectory (i.e., a figure eight trajectory). The network consisted of nonlinear springs and masses in order to approximate the complex behavior of a real biological body. It was attached to the rigid robot arm (green *input* nodes) as well to the shoulder (red, crossed *fixed* nodes). As the arm moved the attached green nodes moved with it and, as a result, the state of the compliant morphological structure (i.e., the spring lengths) changed too. The morphological computation device, which consisted of the nonlinear compliant body and two individual linear readouts of the actual state of the mass-spring network, had to produce appropriate torques τ_1 and τ_2 in order to move along the desired trajectory (red dashed line in Figure 1) in a robust fashion.

III. Simulations

The robot arm was simulated in Matlab at a time step of 1 ms based on a full dynamic model as used in [2]. It was a standard two link robot arm restrained to the two-dimensional plane (i.e., no gravity). The morphological structure consisted of a random network of masses connected by nonlinear springs. The structure of the network was based on a construction policy, which was found to perform well and which is described below. A typical network, resulting from this process and which was used for the presented plots, can be seen in Figure 2. The gray shaded blocks depict the two links of the robot arm. The so-called input nodes[3] (green) were equally distributed and attached along the axes of the two links. In order to stabilize the network some additional (purple) nodes were added. They were fixed relatively to either one of the links by keeping a constant distance to them. The red, crossed nodes were fixed in a global reference frame (i.e., were attached to the robot's shoulder). Finally, the red shaded circles denote the areas in which we randomly positioned additional nodes (i.e., mass points). As in [1], we used a Delaunay triangulation to find non-crossing connections between the nodes. These connections were then simulated as nonlinear springs. Their parameters, which defined the physical properties of the springs, were randomly drawn from a defined range. Note that this implies that the compliant structure was not constructed for this specific task.
The task of the morphological structure was to produce robustly torque trajectories for the two degrees of freedom, i.e., τ_1 and τ_2. Hence, we had two linear readouts, and two corresponding feedback loops. The linear readouts were defined as weighted sums of all L actual spring lengths ($l_i(t), i = 1, 2, \ldots, L$), hence, $\tau_1 = (t) \sum_{i=1}^{L} w_{1,i} \cdot l_i(t)$ and $\tau_2(t) = \sum_{i=1}^{L} w_{2,i} \cdot l_i(t)$.

The target trajectories τ_1^* and τ_2^* for the torques were found by the following procedure: Based on the trajectories,

[3]We called them input nodes, since through these nodes the movement of the arm introduced forces to the compliant morphological structure.

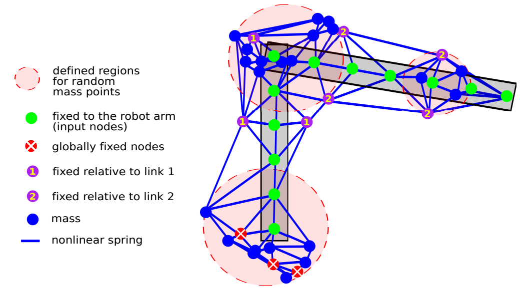
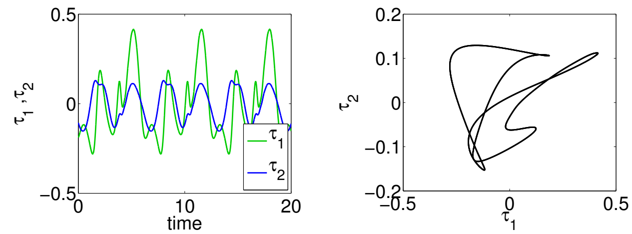


Figure 2: Explanation of the construction policy with the network, which was used for the presented experiments. The compliant structure consisted of $N = 44$ nodes and $L = 122$ springs. For a more detailed description please refer to the text.



(a) (b)

Figure 3: Target torque trajectories, which lead to a figure eight trajectory in Cartesian space for the end effector. **(a)** The trajectories in time and **(b)** its corresponding limit cycle, i.e., τ_1 vs. τ_2.

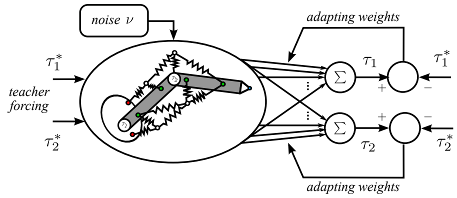
defined by the figure eight trajectory in Cartesian space, a chosen starting position and the Jacobian of the robot arm we calculated the corresponding trajectories of the joint angles. Subsequently, the corresponding torques were found by the use of PD-controllers[4] in order to follow those joint angle trajectories. Figure 3 shows the resulting target torques and the corresponding limit cycle they produce.
Figure 4 shows the setups in the learning and the exploitation phase. During the learning process the loops were open and noise ν was added directly to the morphological structure in form of sensory noise, i.e., it was superimposed on the readouts of the actual state of the morphological structure. Additionally, the robot was driven with the target torques, i.e., τ_1^* and τ_2^*. The lengths of all springs l_i with $i = 1, \ldots, 122$ were collected over the learning time and the ideal output weights $\mathbf{w}_1^*$ and $\mathbf{w}_2^*$ (in order to produce the desired torque signals) were calculated with a simple linear regression. Note that the described learning approach (i.e., *teacher forcing*) is a standard approach used in the context of reservoir computing (see [3]). For the exploitation phase the loops were closed and the system ran autonomously (Figure 4b).

IV. Results

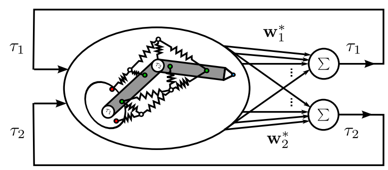
A. Reproducing the trajectories

For the following plots we used the network of Figure 2. Figure 5 shows the trajectories, which were autonomously produced in the exploitation phase (closed loops). The system

[4]The used P and D values were empirically found to have a reasonable performance.

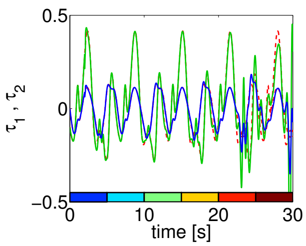


(a) learning phase

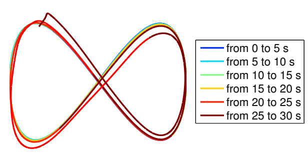


(b) exploitation phase

Figure 4: Schematic figures of **(a)** the learning phase with open loops (teacher forcing) and additional noise ν and **(b)** the exploitation phase with closed loops.



(a)



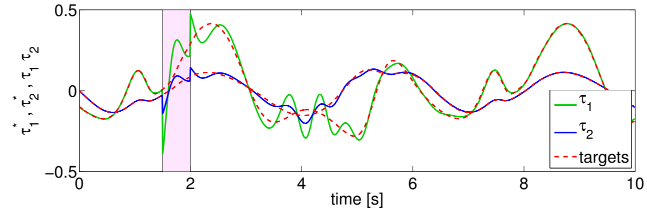
(b)

Figure 5: Output of the morphological computation device after learning and the resulting end-effector trajectory.

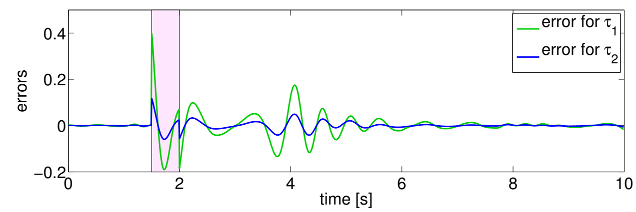
was able to reproduce the desired limit cycle and the corresponding end-effector trajectory for a little more than three and a half cycles. After that the system lost track and drifted away. However, as our robustness test in the next Section IV-B suggests, the learned trajectory was stable to a certain extent. Prior to the presented simulation we were able to successfully demonstrate that abstract random networks (as in [1], i.e., without the robot arm) are able to emulate the necessary computation in a robust fashion. Since in the present robot arm experiments the ranges, over which the springs were operated, are much bigger than with the abstract networks, we speculate that the very simple Matlab implementation of the simulation of the physics is responsible for the instability. We plan to implement more sophisticated physical simulations in the future. Therefore, the presented results should be considered as preliminary.

B. Robustness

A stability test has been conducted and the results are summarized in Figure 6. We applied an external perturbation, which started at 1.5 s and ended at 2.0 s (marked by the red time window). At every time step we added a value of 0.005 to the angle of the first link. The morphological computation device was able to recover from this perturbation and found its way back to the desired trajectories.



(a) torque trajectories



(b) errors

Figure 6: Results of the robustness test.

V. Conclusions and Possible Future Work

We have demonstrated that, in principle, the morphological computation setup of [1] is applicable to a robot arm with a compliant morphology. We were able to employ the compliant parts as a computational resource, which allowed us to produce nonlinear and time-dependent signals (i.e., τ_1 and τ_2) using only static, linear output weights. Although the desired trajectory could not be followed for a long time, our robustness experiment and the experience with the simulations conducted in [1] make us confident that it is possible to improve the results by using a more sophisticated physical simulation. We even think that a real-world application of the approach is feasible. Note that in that case the noise would be provided naturally by the sensors and by the demonstrator.

Another application could be to use the same setup for a quadruped robot, e.g., in order to learn different gaits. Note that the results in [1] suggest that even a switching between different gaits (e.g., walk and trot) should be possible. The switching could be initiated either by some internal signal (e.g., desired walking speed) or by some external force, e.g., bending angles of the knees due to a change of the load, which has to be carried. The fact that the readouts are defined as some linear weights allows to use directly well-established linear online adaptation algorithms.

Another direction of improvement could be to try to mimick better the biological system by using, e.g., antagonistic muscle pairs and "bones” as rigid body parts. Additionally, the torques would be then applied by certain muscles instead of artificial servos.

References

[1] Helmut Hauser, Auke Ijspeert, Rudolf M. Füchslin, Rolf Pfeifer, and Wolfgang Maass. The role of feedback in morphological computation. submitted to Biological Cybernetics, August 2011.

[2] Helmut Hauser, Auke Ijspeert, Rudolf M. Füchslin, Rolf Pfeifer, and Wolfgang Maass. A theoretical foundation for morphological computation. submitted to Biological Cybernetics, April 2011.

[3] Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*, pages 471–482, 2007.