

How to design and control robots with stretchy, flexible bodies

November 21 2019, by Rob Matheson



An MIT-invented model efficiently and simultaneously optimizes control and design of soft robots for target tasks, which has traditionally been a monumental undertaking in computation. The model, for instance, was significantly faster and more accurate than state-of-the-art methods at simulating how quadrupedal robots (pictured) should move to reach target destinations. Credit: Andrew Spielberg, Daniela Rus, Wojciech Matusik, Allan Zhao, Tao Du, and Yuanming Hu



MIT researchers have invented a way to efficiently optimize the control and design of soft robots for target tasks, which has traditionally been a monumental undertaking in computation.

Soft robots have springy, flexible, stretchy bodies that can essentially move an infinite number of ways at any given moment. Computationally, this represents a highly complex "state representation," which describes how each part of the robot is moving. State representations for soft robots can have potentially millions of dimensions, making it difficult to calculate the optimal way to make a robot complete complex tasks.

At the Conference on Neural Information Processing Systems next month, the MIT researchers will present a model that learns a compact, or "low-dimensional," yet detailed state representation, based on the underlying physics of the robot and its environment, among other factors. This helps the model iteratively co-optimize movement control and material design parameters catered to specific tasks.

"Soft robots are infinite-dimensional creatures that bend in a billion different ways at any given moment," says first author Andrew Spielberg, a <u>graduate student</u> in the Computer Science and Artificial Intelligence Laboratory (CSAIL). "But, in truth, there are natural ways soft objects are likely to bend. We find the natural states of soft robots can be described very compactly in a low-dimensional description. We optimize control and design of soft robots by learning a good description of the likely states."

In simulations, the model enabled 2-D and 3-D soft robots to complete tasks—such as moving certain distances or reaching a target spot —more quickly and accurately than current state-of-the-art methods. The researchers next plan to implement the model in real soft robots.

Joining Spielberg on the paper are CSAIL graduate students Allan Zhao,



Tao Du, and Yuanming Hu; Daniela Rus, director of CSAIL and the Andrew and Erna Viterbi Professor of Electrical Engineering and Computer Science; and Wojciech Matusik, an MIT associate professor in electrical engineering and computer science and head of the Computational Fabrication Group.

"Learning-in-the-loop"

Soft robotics is a relatively new field of research, but it holds promise for advanced robotics. For instance, flexible bodies could offer safer interaction with humans, better object manipulation, and more maneuverability, among other benefits.

Control of robots in simulations relies on an "observer," a program that computes variables that see how the soft robot is moving to complete a task. In previous work, the researchers decomposed the soft robot into hand-designed clusters of simulated particles. Particles contain <u>important</u> <u>information</u> that help narrow down the robot's possible movements. If a robot attempts to bend a certain way, for instance, actuators may resist that movement enough that it can be ignored. But, for such complex robots, manually choosing which clusters to track during simulations can be tricky.

Building off that work, the researchers designed a "learning-in-the-loop optimization" method, where all optimized parameters are learned during a single feedback loop over many simulations. And, at the same time as learning optimization—or "in the loop"—the method also learns the state representation.

The model employs a technique called a material point method (MPM), which simulates the behavior of particles of continuum materials, such as foams and liquids, surrounded by a background grid. In doing so, it captures the particles of the robot and its observable environment into



pixels or 3-D pixels, known as voxels, without the need of any additional computation.

In a learning phase, this raw particle grid information is fed into a machine-learning component that learns to input an image, compress it to a low-dimensional representation, and decompress the representation back into the input image. If this "autoencoder" retains enough detail while compressing the input image, it can accurately recreate the input image from the compression.

In the researchers' work, the autoencoder's learned compressed representations serve as the robot's low-dimensional state representation. In an optimization phase, that compressed representation loops back into the controller, which outputs a calculated actuation for how each particle of the robot should move in the next MPM-simulated step.

Simultaneously, the controller uses that information to adjust the optimal stiffness for each particle to achieve its desired movement. In the future, that material information can be useful for 3-D-printing soft robots, where each particle spot may be printed with slightly different stiffness. "This allows for creating robot designs catered to the robot motions that will be relevant to specific tasks," Spielberg says. "By learning these parameters together, you keep everything as synchronized as much as possible to make that design process easier."

Faster optimization

All optimization information is, in turn, fed back into the start of the loop to train the autoencoder. Over many simulations, the controller learns the optimal movement and material design, while the autoencoder learns the increasingly more detailed state representation. "The key is we want that low-dimensional state to be very descriptive," Spielberg says.



After the robot gets to its simulated final state over a set period of time—say, as close as possible to the target destination—it updates a "loss function." That's a critical component of machine learning, which tries to minimize some error. In this case, it minimizes, say, how far away the <u>robot</u> stopped from the target. That loss function flows back to the controller, which uses the error signal to tune all the optimized parameters to best complete the task.

If the researchers tried to directly feed all the raw particles of the simulation into the controller, without the compression step, "running and optimization time would explode," Spielberg says. Using the compressed representation, the researchers were able to decrease the running time for each optimization iteration from several minutes down to about 10 seconds.

The researchers validated their model on simulations of various 2-D and 3-D biped and quadruped robots. They researchers also found that, while robots using traditional methods can take up to 30,000 simulations to optimize these parameters, robots trained on their model took only about 400 simulations.

Deploying the model into real soft robots means tackling issues with realworld noise and uncertainty that may decrease the model's efficiency and accuracy. But, in the future, the researchers hope to design a full pipeline, from simulation to fabrication, for <u>soft robots</u>.

Provided by Massachusetts Institute of Technology

Citation: How to design and control robots with stretchy, flexible bodies (2019, November 21) retrieved 28 April 2024 from <u>https://techxplore.com/news/2019-11-robots-stretchy-flexible-bodies.html</u>



This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.