HAL
open science

Adaptive and Resilient Soft Tensegrity Robots

John Rieffel, Jean-Baptiste Mouret

To cite this version:

John Rieffel, Jean-Baptiste Mouret. Adaptive and Resilient Soft Tensegrity Robots. Soft Robotics, 2018, 5 (3), pp.318-329. 10.1089/soro.2017.0066 . hal-01800749

HAL Id: hal-01800749

https://inria.hal.science/hal-01800749v1

Submitted on 27 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SOFT ROBOTICS
Volume 00, Number 00, 2018
Mary Ann Liebert, Inc.
DOI: 10.1089/soro.2017.0066

SoRo
Soft Robotics

ORIGINAL ARTICLE

Adaptive and Resilient Soft Tensegrity Robots

John Rieffel[1,*] and Jean-Baptiste Mouret[2,*]

Abstract

Living organisms intertwine soft (e.g., muscle) and hard (e.g., bones) materials, giving them an intrinsic flexibility and resiliency often lacking in conventional rigid robots. The emerging field of soft robotics seeks to harness these same properties to create resilient machines. The nature of soft materials, however, presents considerable challenges to aspects of design, construction, and control—and up until now, the vast majority of gaits for soft robots have been hand-designed through empirical trial-and-error. This article describes an easy-to-assemble tensegrity-based soft robot capable of highly dynamic locomotive gaits and demonstrating structural and behavioral resilience in the face of physical damage. Enabling this is the use of a machine learning algorithm able to discover effective gaits with a minimal number of physical trials. These results lend further credence to soft-robotic approaches that seek to harness the interaction of complex material dynamics to generate a wealth of dynamical behaviors.

Keywords: tensegrity, Bayesian optimization, vibration, resonance

Introduction

Unlike machines, animals exhibit a tremendous amount of resilience, due, in part, to their intertwining of soft tissues and rigid skeletons. In nature, this suppleness leads to several compelling behaviors that exploit the dynamics of soft systems. Octopi, for example, are able to adaptively shape their limbs with ''joints'' to perform efficient grasping.[1] Jellyfish exploit their inherent elasticity to passively recover energy during swimming.[2] *Manduca sexta* caterpillars have a mid-gut which acts like a ''visceral-locomotory piston''—sliding forward ahead of the surrounding soft tissues, shifting the animal's center of mass forward well before any visible exterior change.[3]

Taking inspiration from the natural world, the field of soft robotics seeks to address some of the constraints of conventional rigid robots through the use of compliant, flexible, and elastic materials.[4,5] Trimmer *et al.*, for instance, construct soft robots from silicone rubber, using shape memory alloy microcoil actuation, which can slowly crawl in a controlled manner[6] or roll in an uncontrolled ballistic manner.[7] Similarly, research by Whitesides *et al.* uses pneumatic inflation to produce slow, dynamically stable crawling motions[8] as well as fast, but less controlled tentacle-like grippers,[9] combustion-driven jumpers[10] and a self-contained microfluidic ''octobot.''[5]

Despite their advantages, soft-material robots are difficult to control by conventional means.[4,8] They are by their very nature high dimensional dynamic systems with an essentially infinite number of degrees of freedom. The elasticity and deformability that provide their appeal come at the cost of resonances and tight dynamic coupling between components,[6] properties that are often avoided, or at least suppressed, in conventional engineering approaches to robotic design. This complexity precludes the use of many of the traditional kinematic and inverse-dynamics approaches to robotic control.[11]

As a result, up until now, the locomotive gaits of most soft robots have been developed by hand through empirical trial-and-error.[8] This process can be both challenging and time consuming, particularly when seeking to fully exploit the dynamical complexity of soft mechanisms. Importantly, this manual process also prevents these robots from adapting their

[1]Department of Computer Science, Union College, Schenectady, New York.
[2]Inria, Université de Lorraine, CNRS, LORIA, Nancy, France.
*These authors contributed equally to this work.

© John Rieffel and Jean-Baptiste Mouret 2018; Published by Mary Ann Liebert, Inc. This Open Access article is distributed under the terms of the Creative Commons License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

control strategy when the context changes, for instance when they encounter an unexpected type of terrain, or when they are physically damaged.

In this work, we describe a new class of soft robot based upon a tensegrity structure driven by vibration. Like many other soft robots, this tensegrity robot is resilient, and can resist damage when perturbed or crushed. Unlike other soft robots, however, this particular modular tensegrity robot is easy to build, easy to control, and, thanks to a data-efficient reinforcement learning algorithm,[12] it can autonomously discover how to move, and quickly relearn and adapt its behavior when damaged.

Vibration is an increasingly popular method of sensor-free manipulation and control for automated systems.[13] Rezik *et al.*, for instance, developed a vibration-driven planar manipulator[14] able to perform large-scale distributed planar control of small parts.[15] In mobile robotics, stick-and-slip frictional motion[16,17] driven by paired vibrating motors has been used in a variety of mobile robots.[18,19] Often, these approaches use empirically derived hand-tuned frequencies to generate motion, using linear interpolation of their two motor speeds to smoothly generate a range of behaviors. One weakness of vibration-based approaches to locomotion is that vibration of this type leads to unpredictable motion, even when assuming perfectly consistent surfaces,[17] which presents a challenge to modeling and simulation.

Tensegrities are relatively simple mechanical systems, consisting of a number of rigid elements (struts) joined at their endpoints by tensile elements (cables or springs), and kept stable through a synergistic interplay of prestress forces (Fig. 1A–C). Beyond engineering, properties of tensegrity have been demonstrated at all scales of the natural world, ranging from the tendinous network of the human hand[20] to the mechanotransduction of living cells.[21] At every size, tensegrity structures exhibit two interesting features[22,23]: they have an impressive strength-to-weight ratio and they are structurally robust and stable in the face of deformation. Moreover, unlike many other soft robots, tensegrity structures are inherently modular (consisting of only struts and springs) and are, therefore, relatively easy to construct. They are simple enough to be baby toys and featured in books for children activities,[24] whereas complex enough to serve as the basis for the next generation of NASA's planetary rovers.[25]

The most common control method for tensegrity robots is to slowly change the lengths of the struts and/or cables, causing large-scale, quasi-static (rather than dynamic) structural deformations, which, in turn, make the robot move through tumbling and rolling.[26,27] As they assume that the structure is

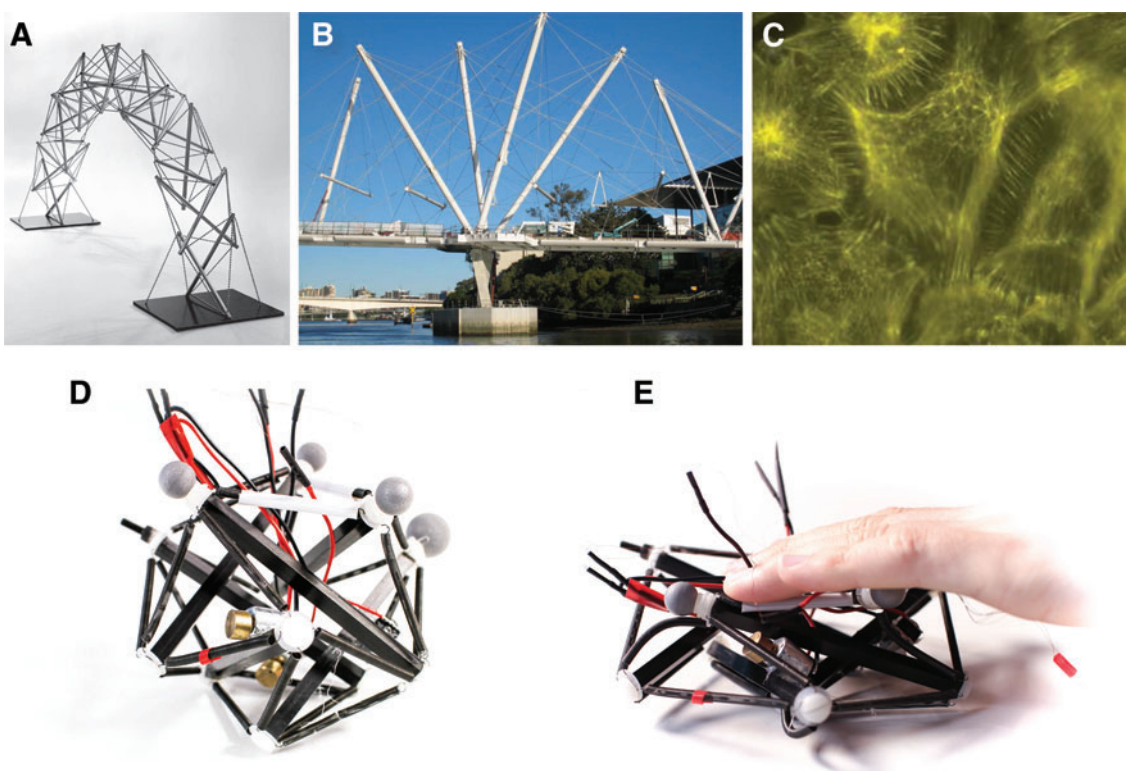

FIG. 1. Concept of our soft tensegrity robot. Tensegrity structures are combinations of rigid elements (struts) joined at their endpoints by tensile elements (spring or cables) that are kept stable by the interplay of prestress forces. **(A)** The first tensegrity structures appeared in art, with the sculptures of Kenneth Snelson.[22,23] **(B)** They have been subsequently used in architecture, for instance for the Kurilpa bridge (Brisbane, Australia). **(C)** More recently, tensegrity has been found to be a good model of the mechanotransduction of living cells.[21] **(D)** Our tensegrity robot is based on carbon struts and springs. It is actuated by three vibrators (glued to three of the struts) whose frequency is automatically tuned by a trial-and-error learning algorithm (Materials and Methods). **(E)** Thanks to the tensegrity structure and to the compliance of the springs, our robot will keep its integrity when deformed and spring back into its initial form. A video is available in Supplementary Data (Supplementary Video S1). Color images available online at www.liebertpub.com/soro

relatively stiff throughout locomotion, such control strategies are not suitable for more compliant soft tensegrity robots. In addition, they lead to slow locomotion speeds.

Lately, researchers have begun investigating more dynamical methods of tensegrity robot control. Bliss *et al.* have used central pattern generators (CPGs) to produce resonance entrainment of simulated nonmobile tensegrity structures.[27] Mirletz *et al.* have used CPGs to produce goal-directed behavior in simulated tensegrity-spine-based robots.[28] These efforts, however valuable, were all produced in simulated environments, and have not yet been successfully transferred into real-world robots. As Mirletz *et al.* point out,[27] the dynamic behavior of tensegrities is highly dependent upon the substrate they interact with—this means that results developed in simulated environments cannot necessarily be simply transferred to real robots (in Evolutionary Robotics, this is known as the "Reality Gap"[29,30]).

More recently, Böhm and Zimmermann developed a tensegrity-inspired robot actuated by a single oscillating electromagnet.[31] Although this robot was not a pure tensegrity (it rigidly connected multiple linear struts), it was, compellingly, able to change between forward and backward locomotion by changing the frequency of the oscillator. Vibration has been proposed as a means of controlling much softer robots as well.[32]

Here we explore the hypothesis that the inherent resonance and dynamical complexity of real-world soft tensegrity robots can be beneficially harnessed (rather than suppressed), and that, if properly excited,[33] it can resonate so that the robot performs step-like patterns that enable it to locomote. To test this hypothesis and demonstrate the potential of soft tensegrity robots, we designed a pocket-sized, soft tensegrity robot whose parameters were tuned to maximize resonance, and whose goal is to locomote as fast as possible across flat terrain. To find the right vibrational frequencies, we equipped the robot with a data-efficient trial-and-error algorithm, which also allows it to adapt when needed.

Materials and Methods

Our soft tensegrity robot (Fig. 1D, E) is based upon a canonical six-bar tensegrity shape consisting of equal length composite struts connected through 24 identical helical springs, with four springs emanating from each strut end. Unlike most tensegrity structures, which seek to maximize stiffness by, among other things, using nearly inelastic cables,[33] here we replace the cables with very elastic springs, with spring constants chosen with the goal of producing suitably low natural frequencies of the structure, with corresponding large displacements—in other words, to maximize suppleness. This allows the pocket-sized robot to maintain its structural shape under normal operation, and yet be easily compressed flat in one's hand. A variable speed motor coupled to offset masses was then attached to three of the struts to excite the natural frequencies of the structure. The motor and weight were chosen to be in a range consistent with preliminary models. Because of the difficulty in modeling real-world interactions of these tensegrity robots, as well as the fact that we use a real-world optimization process described hereunder, variability in the exact placement of each motor on a strut is allowed.

Like many robots, the tensegrity robot needs to use different gaits to achieve locomotion, depending on terrain. In our case, these gaits are determined by the speeds of the three vibratory motors. As the exact properties of the terrain are seldom known *a priori*, and because hand-designing gaits are time consuming (not to mention impossible when the robot is in remote or hazardous environments), this robot finds effective motor frequencies by using a trial-and-error learning algorithm whose goal is to maximize the locomotion speed.

Earlier work of ours[34,35] used interactive trial-and-error as well as automated hill climbing techniques to find optimal gaits for a tensegrity robot. These gaits could, in turn, be incorporated into a simple state machine for directional control. However, these techniques required hundreds of physical trials that were time consuming and produced significant wear on the physical robot. More importantly, the interactive procedure required a human in the loop, whereas we envision robots that can adapt autonomously to new situations (e.g., a damage or a new terrain). The work described in this article, by automating the optimization process while minimizing the number of physical trials required, substantially reduces the amount of human interaction required, and is an important step toward full autonomy.

Here, as a substantial improvement upon these earlier time-intensive methods, we employ a Bayesian optimization algorithm,[12,36,37] which is a mathematical optimizer designed to find the maximum of a performance function with as few trials as possible.

Conceptually, Bayesian optimization fits a probabilistic model (in this case a Gaussian process,[38] see Materials and Methods) that maps motor speeds to locomotion speed. Because the model is probabilistic, the algorithm can not only predict which motor speeds are the most likely to be good, but also associate it with a confidence level. Bayesian optimization exploits this model to select the next trial by balancing exploitation—selecting motor speeds that are likely to make the robot move faster—and exploration—trying combinations of motor speeds that have not been tried so far (Materials and Methods). As an additional benefit, this algorithm can take into account that observations are by nature uncertain.

The Bayesian optimization algorithm usually starts with a constant prior for the expected observation (e.g., the expected speed is 10 cm/s) and a few randomly chosen trials to initialize the model. For this robot, however, common sense, along with preliminary modeling, suggests that speeds near the motor maximums are more likely to produce successful gaits, and that near-zero motor speeds are not expected to make the robot move. This insight was substantiated in preliminary experiments: many effective gaits were produced by high motor speeds, both forward and backward. Therefore, to speed up learning, we use a nonlinear prior model as follows: (1) if the three motor speeds are close to 0, then we should expect a locomotion speed close to 0 and (2) if all the motors are close to full speed (in any direction), then we should expect the maximum locomotion speed (Materials and Methods and Fig. 2D). Thanks to this prior, the Bayesian optimization algorithm does not need any random sample points to seed the prior and instead starts with promising solutions. Despite this prior, learning is still needed, because many combinations of motors at full speeds make the robot tumble or rotate on itself, resulting in low performance; in

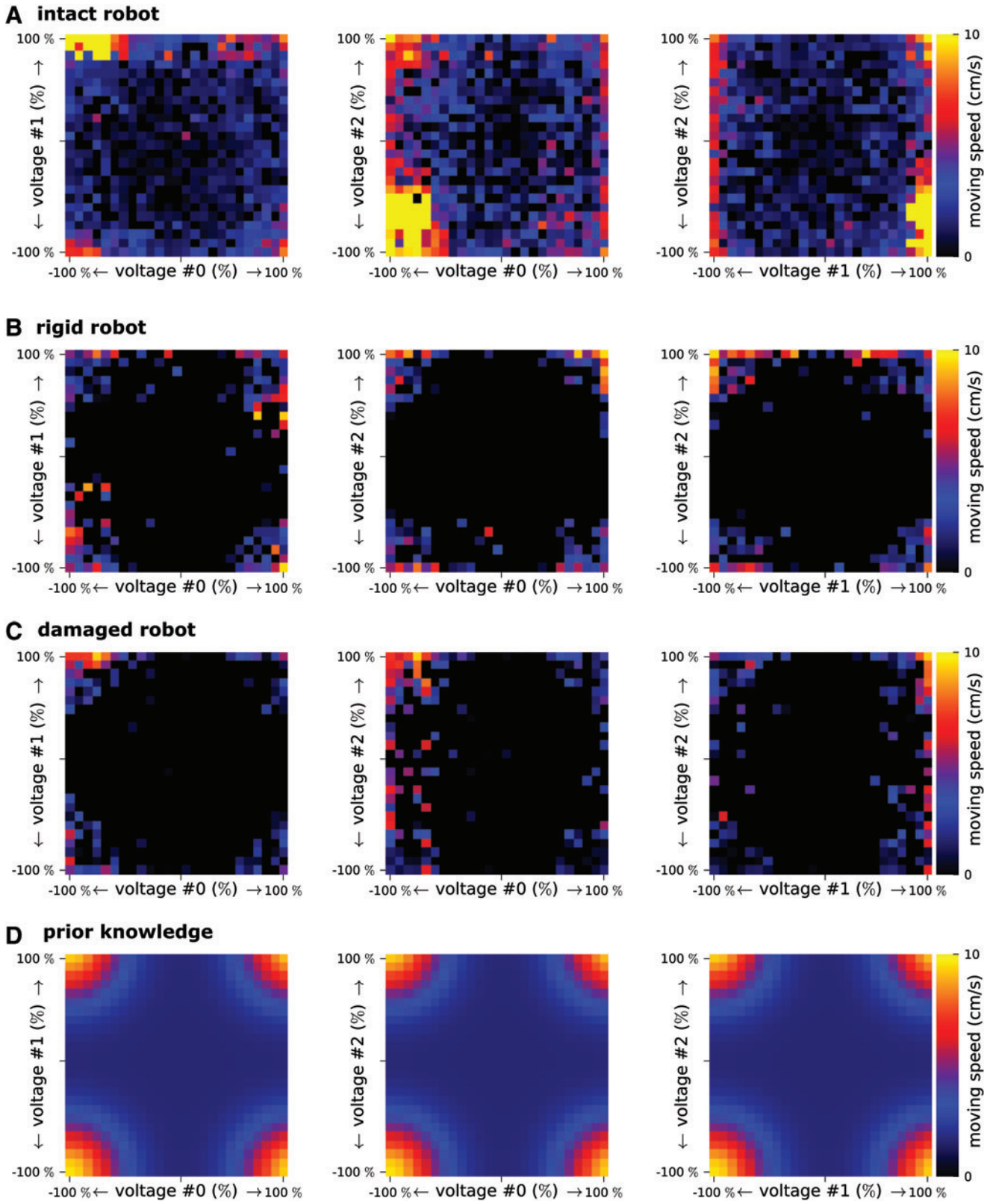

FIG. 2. Performance profiles for all the conditions. These performance profiles show the performance potential of each combination of two motor speeds (the third motor is considered as a ''free variable''). Three plots are required to get a comprehensive picture of the performance space: v_1 versus v_2, v_1 versus v_3, and v_2 versus v_3. **(A)** Intact robot (Fig. 1D). The profiles are computed with 1800 policy evaluations (20 replicates × 30 trials × 3 sets of experiments—with prior, without prior, random search). **(B)** Rigid robot (Fig. 4A). The profiles are computed with 600 policy evaluations (30 trials × 20 replicates). **(C)** Damaged robot (Fig. 5). The profiles are computed with 600 policy evaluations (30 trials × 20 replicates). **(D)** Prior knowledge. Prior knowledge used to guide the learning algorithm (Materials and Methods). Color images available online at www.liebertpub.com/soro

addition, subtle changes to motor speeds can have dramatic effects upon the resulting robot gait.

Robot

The tensegrity used follows the geometry described as TR-6 by Skelton.[23] Few actual machining operations are required to produce the tensegrity. The six 9.4 cm long composite struts are cut from 6.35 mm square graphite composite tubes (Goodwinds). The three 12 mm vibrational motors (Precision Microdrives Model 312-107) were mounted to the flat outer surface of the struts using hot melt adhesive. Both ends of each strut were then tapped for 10–24 nylon screws fitted with nylon washers. The hooked ends of the helical springs (Century Spring Stock No. 5368) were attached directly to holes drilled through the nylon washers. The motors were connected through thin gauge magnet wire to Serial Motor Controllers (Pololu Qik 2s9v1 Dual Serial Motor Controller) connected, in turn, to a USB Serial Adapter (SparkFun FTDI Basic Breakout board)

The specific spring constants were chosen to produce relatively low natural frequencies and correspondingly large displacements of the structure while at the same time limiting estimated static deflection to 5% of strut length. To determine this, a single strut was modeled as being connected to four linear springs at each end, equally spaced around the radius, each at a 45° angle. Limiting static deflection to 5% of strut length results in a spring constant value of 0.209 N/cm. Subsequently, the entire six-bar structure was modeled by assuming that one strut was to be anchored in place and then using matrix structural analysis to determine the natural frequencies. The vibrational motor was then chosen that was capable of generating sufficient centrifugal force at a suitable range of frequencies. Details of the modeling and design are provided in Ref.[34]

Control policy

Each policy is defined by three pulse width modulation (PWM) values that determine the input voltage of the three vibrating motors ($\chi = [v_1, v_2, v_3]$), which can take values between 0 (full speed, backward) and 1 (full speed, forward); 0.5 corresponds to a speed of 0, that is, to no movement.

Performance function

Each controller is tested for 3 s, then the Euclidean distance between the starting point and the end point is recorded. The performance function is the distance (in cm) divided by 3. If during the 3 s evaluation period, the yaw of the robot exceeds 1 radian, the evaluation is stopped and the recorded distance is the distance between the starting point and the point reached by the robot when it exceeded the yaw limit.

The policies are evaluated externally with a motion tracking system (Optitrack Prime 13/8 cameras), but the same measurements can be obtained with an embedded camera connected to a visual odometry system.[12,39]

Profile plots

We use the profile plots to depict the search space and the prior used by the learning algorithm (Fig. 2). For each pair of dimensions, we discretize the motor speeds into 25 bins. For each bin, we compute $p_{profile}(v_1, v_2) = \max_{v_3} p(v_1, v_2, v_3)$, where $p(v_1, v_2, v_3)$ is the performance of the robot for motor speeds v_1, v_2, v_3 and $p_{profile}(v_1, v_2)$ is the performance reported in the profile. To get comprehensive pictures, we need three plots: $p_{profile}(v_1, v_2)$, $p_{profile}(v_1, v_3)$, and $p_{profile}(v_2, v_3)$.

Learning algorithm

Our learning algorithm allows the robot to discover by trial-and-error the best rotation speeds for its three motors. It essentially implements a variant of Bayesian optimization, which is a state-of-the-art optimization algorithm designed to maximize expensive performance functions (a.k.a. cost functions) whose gradient cannot be evaluated analytically.[36,37] Like other model-based optimization algorithms (e.g., surrogate-based algorithms,[40–42] kriging,[43] or Design and Analysis of Computer Experiments (DACE)[44,45]), Bayesian optimization models the objective function with a regression method, uses this model to select the next point to acquire, then updates the model, etc. until the algorithm has exhausted its budget of function evaluations.

Here a Gaussian process models the objective function,[38] which is a common choice for Bayesian optimization.[36,37,46–48] For an unknown cost function f, a Gaussian process defines the probability distribution of the possible values $f(\mathbf{x})$ for each point $\mathbf{x}$. These probability distributions are Gaussian, and are, therefore, defined by a mean (μ) and a variance (σ^2). However, μ and σ^2 can be different for each $\mathbf{x}$; a Gaussian process, therefore, defines a probability distribution *over functions*:

$$P(f(\mathbf{x})|\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})), \tag{1}$$

where $\mathcal{N}$ denotes the standard normal distribution.

At iteration t, if the performance $[P_1, \cdots, P_t] = \mathbf{P}_{1:t}$ of the points $[\chi_1, \cdots, \chi_t] = \chi_{1:t}$ has already been evaluated, then $\mu_t(\mathbf{x})$ and $\sigma_t^2(\mathbf{x})$ are fitted as follows[38]:

$$\begin{aligned}
&\mu_t(\mathbf{x}) = \mathbf{k}^T\mathbf{K}^{-1}\mathbf{P}_{1:t} \\
&\sigma_t^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) + \sigma_{noise}^2 - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}, \\
&\textit{where} \\
&\mathbf{K} = \begin{bmatrix} k(\chi_1, \chi_1) & \cdots & k(\chi_1, \chi_t) \\ \vdots & \ddots & \vdots \\ k(\chi_t, \chi_1) & \cdots & k(\chi_t, \chi_t) \end{bmatrix} + \sigma_{noise}^2 I \\
&\mathbf{k} = [\, k(\mathbf{x}, \chi_1) \quad k(\mathbf{x}, \chi_2) \quad \cdots \quad k(\mathbf{x}, \chi_t)\,].
\end{aligned} \tag{2}$$

Matrix $\mathbf{K}$ is called the covariance matrix. It is based on a *kernel function* $k(\mathbf{x_1}, \mathbf{x_2})$ that defines how samples influence each other. Kernel functions are classically variants of the Euclidean distance. Here we use the *exponential kernel*[12,37,38,48]:

$$k(\mathbf{x_1}, \mathbf{x_2}) = \exp\left(-\frac{1}{\beta^2}||\mathbf{x_1} - \mathbf{x_2}||^2\right), \tag{3}$$

because this is the most common kernel in Bayesian optimization and we did not see any reason to choose a different one.[37,48] We fixed β to 0.15.

An interesting feature of Gaussian processes is that they can easily incorporate a prior $\mu_p(\mathbf{x})$ for the mean function,

which helps to guide the optimization process to zones that are known to be promising:

$$\mu_t(\mathbf{x}) = \mu_p(\mathbf{x}) + \mathbf{k}^T\mathbf{K}^{-1}(\mathbf{P}_{1:t} - \mu_p(\chi_{1:t})).. \quad (4)$$

In our implementation, the prior is a second Gaussian process defined by hand-picked points (see the "prior" section hereunder).

To select the next χ to test (χ_{t+1}), Bayesian optimization maximizes an *acquisition function*, a function that reflects the need to balance exploration—improving the model in the less known parts of the search space—and exploitation—favoring parts that the model predicts as promising. Numerous acquisition functions have been proposed (e.g., probability of improvement, the expected improvement, or the upper confidence bound (UCB)[37,47,48]); we chose UCB because it provided the best results in several previous studies[47,48] and because of its simplicity. The equation for UCB is

$$\chi_{t+1} = \arg\max_x (\mu_t(\mathbf{x}) + \kappa\sigma_t(\mathbf{x})), \quad (5)$$

where κ is a user-defined parameter that tunes the tradeoff between exploration and exploitation. We chose $\kappa = 0.2$.

Prior for the learning algorithm

The learning algorithm is guided by a prior that captures the idea that the highest performing gaits are likely to be a combination of motors at full speed (in forward or in reverse). In our implementation, it is implemented with a Gaussian process defined by nine hand-picked points and whose variance is ignored (Eq. 2). The kernel function is the exponential kernel (Eq. 3), with $\beta = 0.15$.

The nine hand-picked points ($\chi_1, \cdots, \chi_9$) are as follows (Fig. 2D):

$$\begin{aligned}
&\chi_1 = [-100\%, -100\%, -100\%],\\
&\chi_2 = [-100\%, -100\%, +100\%],\\
&\chi_3 = [-100\%, 100\%, -100\%],\\
&\chi_4 = [-100\%, +100\%, +100\%],\\
&\chi_5 = [+100\%, -100\%, -100\%],\\
&\chi_6 = [+100\%, +100\%, +100\%],\\
&\chi_7 = [+100\%, -100\%, -100\%],\\
&\chi_8 = [+100\%, -100\%, +100\%], \textit{ and}\\
&\chi_9 = [0\%, 0\%, 0\%];\\
&P(\chi_1), \cdots, P(\chi_8) = 0.3; P(\chi_9) = 0.
\end{aligned}$$

Statistics

For all experiments, we report the 5*th* and 95*th* percentiles. We used a two-tailed Mann–Whitney *U* test for all statistical tests. For the box plots, the central mark is the median, the edges of the box are the 25th and 75th percentiles (interquartile range), the whiskers correspond to the range $[25\% - 1.5 \times IQR, 75\% + 1.5 \times IQR]$, and points outside of the whiskers are considered to be outliers (this corresponds to the "interquartile rule"). For each box plot, the result of the Mann–Whitney *U* test (two-tailed) is indicated with asterisks: * means $p \leq 0.05$, ** means $p \leq 0.01$, *** means $p \leq 0.001$, and **** means $p \leq 0.0001$.

Results

We first evaluate the effectiveness of the learning algorithm (Fig. 3). The performance function is the locomotion speed, measured for 3 s, in any direction. If the robot turns too much, that is if the yaw exceeds a threshold, the evaluation is stopped. The covered distance is measured with an external motion capture system, although similar measurements can be obtained with an onboard visual odometry system.[12,39] We compare three algorithms: random search, Bayesian optimization without prior (using 10 random points to initialize the

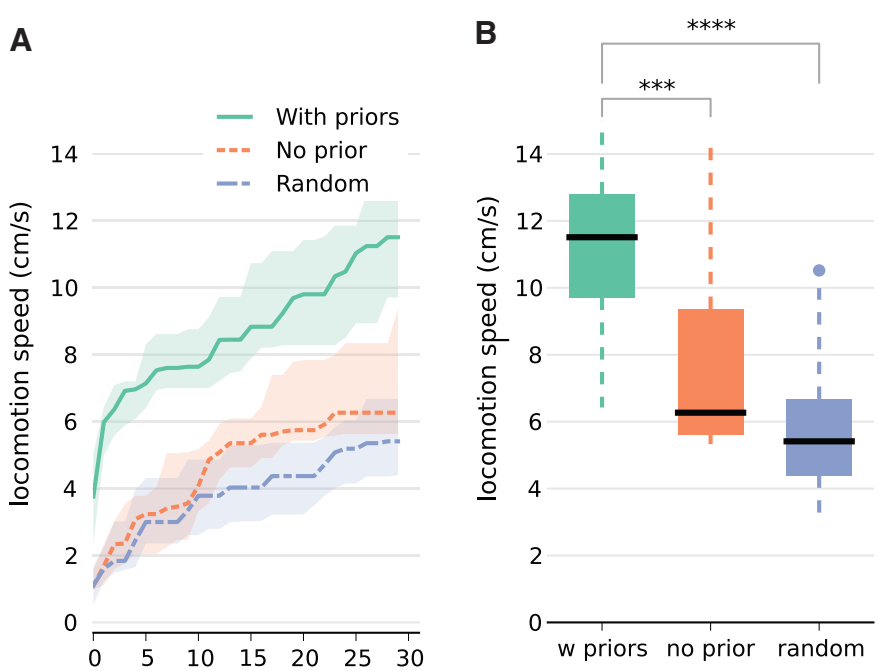

FIG. 3. Evaluation of the learning algorithm. **(A)** Locomotion speed after each of the 30 trials. The light zones represent the 25th and 75th percentiles. **(B)** Locomotion speed after 30 trials. The *central mark* is the median, the edges of the box are the 25th and 75th percentiles (IQR), the *whiskers* correspond to the range $[25\% - 1.5 \times IQR, 75\% + 1.5 \times IQR]$, and points outside of the *whiskers* are considered to be outliers (this corresponds to the "interquartile rule"). Each condition is tested with 20 independent runs of the algorithms. IQR, interquartile range. ***$p \leq 0.001$; ****$p \leq 0.0001$. Color images available online at www.liebertpub.com/soro

Table 1. Best Locomotion Speed, Power Consumption, and Cost of Transport for the Gait that Corresponds to the Maximum Speed at 5 V

Characteristic	*Value*
Weight (W)	89*g*
Body length	13 cm
Best locomotion speed (S) [after learning]	15 cm/s (1.15 body lengths per second)
Current drawn (full speed):	700 *mA*
Power drawn at 5 V (P)	3.5 *W*
Cost of transport at maximum speed (COT)	262 $\left(\text{COT} \triangleq \frac{P}{WS}\right)$

For reference, a COT of 262 is comparable with the COT of a mouse (>100), but much higher than a car or a motorcycle ($\sim$3).[64]
COT, cost of transport.

algorithm), and Bayesian optimization with prior. Each algorithm is allowed to test 30 different motor combinations (resulting in 90 s of learning for each experiment) and is independently run 20 times to gather statistics. The results show that the best locomotion speeds are obtained with the prior-based Bayesian optimization (11.5 cm/s, 5th and 95th percentiles [8.1, 13.7]), followed by the prior-free Bayesian optimization (6.3 cm/s [5.5, 12.4]). The worst results are obtained with the random search (5.4 cm/s [3.5, 9.9]). The absolute best locomotion speed (15 cm/s) was found with Bayesian optimization (Table 1) and corresponds to 1.15 body lengths per second. Overall, these experiments demonstrate that the prior-based Bayesian optimization is an effective way to automatically discover a gait in only 30 trials with this robot. Videos of a typical gait is available as Supplementary Data (Supplementary Video S1; Supplementary Data are available online at www.liebertpub.com/soro).

We then investigate our hypothesis that the interplay between a flexible tensegrity structure and vibration is the key for effective locomotion. To do so, we designed a rigid replica of our robot that does not contain any springs: the carbon fiber struts are held in place with small carbon fiber rods (Fig. 4A). All the dimensions, strut positions, and motor positions are the same as for the tensegrity version (Fig. 1D, E). We used the same learning algorithm as for the tensegrity robot and the same prior, since we have the same intuitions about good control policies for the rigid robot as for the soft one. We replicated the learning experiment 20 times. The results (Fig. 4B) show that the rigid replica moves at about 60% of the speed of the tensegrity robot (7.1 cm/s [5.6, 9.3] vs. 11.5 cm/s [8.1, 13.7]), which suggests that the flexibility of the tensegrity structure plays a critical role in its effective locomotion. In addition, we measured the amplitude of movement along the vertical axis for the end of four struts, both with the soft tensegrity robot and the rigid replica; we repeated this measure with 50 random gaits in both cases. These measurements (Fig. 4C) show that the markers move at least twice more when the structure is flexible (2.3 [1.5, 4.8] cm vs. 0.99 [0.61, 2.1] cm), which demonstrates that the structure amplifies the movements induced by the vibrators.

In addition to being deformable, tensegrity structures often maintain most of their shape when a link (a spring or a strut) is missing, leading to relatively smooth failure modes. We evaluate the ability of our robot to operate after such damage by removing a spring (Fig. 5A). As the shape of the robot is changed, we relaunch the learning algorithms. The experiments reveal that successful, straight gaits can be found in 30 trials, although they are significantly lower performing than

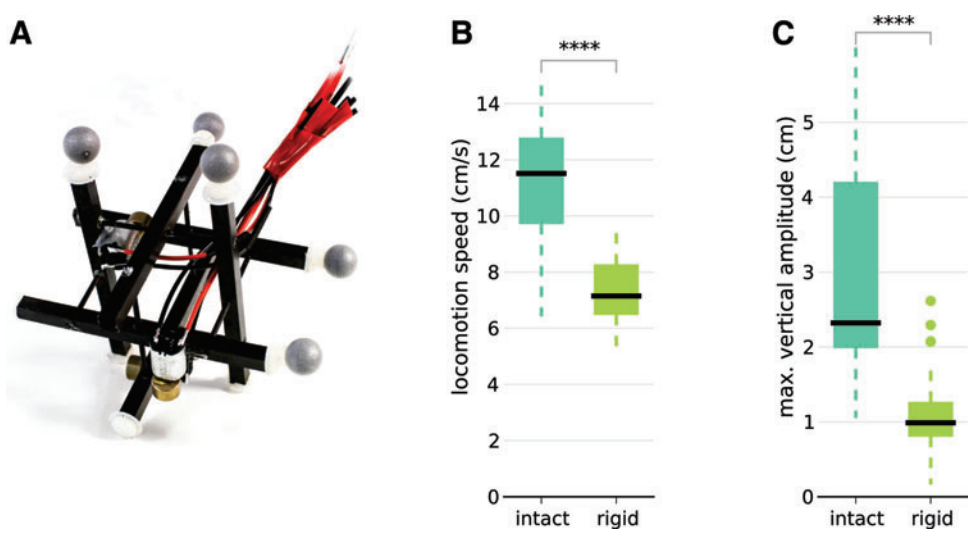

FIG. 4. Experiments with a rigid robot (with priors). **(A)** Rigid replica of our soft tensegrity robot. This robot is identical to the robot shown in Figure 1, except that it contains no spring: the carbon fiber struts are held in place with small carbon fiber rods. All the dimensions, strut positions, and motor positions are the same as for the tensegrity version. **(B)** Locomotion speed after 30 trials for the intact (Fig. 1) and the rigid robot **(A)**. Each condition is tested with 20 independent runs of the algorithm (Bayesian optimization *with priors*). **(C)** Maximum amplitude of the markers for random gaits. In each case, we captured the vertical position of the four markers for 50 random gaits of 3 s. We report the maximum height minus the minimum height (over the four markers). For the box plots, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the *whiskers* extend to the most extreme data points not considered outliers, and outliers are plotted individually. Color images available online at www.liebertpub.com/soro

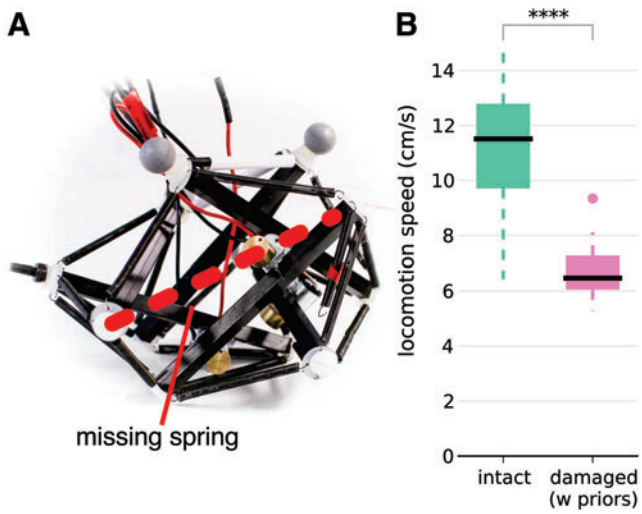

FIG. 5. Experiments with a damaged robot. **(A)** Damaged robot. A spring is disconnected from the robot as shown in Figure 1. **(B)** Locomotion speed after 30 trials. The *central mark* is the median, the edges of the box are the 25th and 75th percentiles (IQR), the *whiskers* correspond to the range $[25\% - 1.5 \times IQR, 75\% + 1.5 \times IQR]$, and points outside of the *whiskers* are considered to be outliers (this corresponds to the "interquartile rule"). Each condition is tested with 20 independent runs of the algorithms. ****$p \leq 0.0001$. Color images available online at www.liebertpub.com/soro

gaits obtained with the intact robot (11.5 cm/s [8.1, 13.7] versus 6.5 cm/s [5.6, 8.2] Fig. 5B).

During all the reported experiments, we evaluated $20 \times 30 \times 3 = 1800$ different gaits on the intact robot, $20 \times 30 = 600$ gaits on the rigid robot (20 replicates, 30 trials for each replicate, and 3 treatments), and $20 \times 30 = 600$ gaits on the damaged robot. We can use these points to draw a picture of the search space that does not depend on the learning algorithm (Fig. 2). Since the search space is too high dimensional to be easily visualized (three dimensions + performance, resulting in a four-dimensional plot), we compute performance profiles[49,50]: for each combination of two motor speeds $\{v_1, v_2\}$, we report the best performance measured regardless of the speed of the third motor (Materials and Methods). The performance profiles (Fig. 2A) for the intact robot reveal that there are two high-performing regions, roughly positioned around $\{-100\%, 100\%, -100\%\}$ and $\{-100\%, -100\%, 100\%\}$ and that the first region ($\{-100\%, 100\%, -100\%\}$) is where most high-performing solutions can be found. This finding is consistent with the prior given to the learning algorithm (Fig. 2D), which models that the best performance should be obtained with a combination of -100% and $+100\%$ values. It should be emphasized that the best gaits do not correspond to the most extreme values for the motor speeds: the most reliable optima is around $\{-90\%, 100\%, -90\%\}$, mostly because too extreme values tend to make the robot tumble. The best solutions for the rigid robots are also found in the corners, that is, for combinations of $+100\%$ and -100% motor speeds, but the measurements suggest that the optimum might be different from that obtained with the intact robot (more data would be needed to conclude). The data for the damaged robot show more clearly that the best solutions are around $\{-100\%, -100\%, 100\%\}$, which corresponds to the second optimum found for the intact robot (the lowest performing robot).

The performance profiles thus demonstrate that the prior knowledge given to the learning algorithm is consistent with the three different robots (intact, rigid, and damaged), which suggests that it might be helpful in other situations (e.g., different damage conditions). They also demonstrate that gaits that work the best on the intact robot do not work on the damaged robot (Fig. 2A versus C, second column): this shows that the learning algorithm is needed to adapt the gait if the robot is damaged.

Discussion

Soft tensegrity robots are highly resilient, easy to assemble with the current technology, and made with inexpensive materials. In summary, vibratory soft tensegrity robots recast most of the complexity of soft robotics—building and actuating soft structures—into a much simpler class of robots—easy to build and to actuate—while keeping many of the attractive properties of soft robots—for example, resilience and deformability. Thanks to the learning algorithm, our prototype can achieve locomotion speeds of >10 cm/s (more 1 body length per second) and learn new gaits in <30 trials, which allows it to adapt to damage or new situations. To our knowledge, this places it among the fastest soft robots. Our soft tensegrity robots achieve this speed because they uniquely harness the flexibility and the resonance of tensegrity structures. Discovering methods of exploiting flexibility and resonance in this manner opens new research avenues for future tensegrity structures, in particular when mechanical design can be coupled with machine learning algorithms that automatically identify how to control the resonances.

Although our soft tensegrity robots also to a large extent benefit from anisotropic friction, our effort is distinct from other vibration-based robots such as Kilobot[18] and RatChair[19] in several important ways. First, because of the nature of the structure, opposing pairs of vibrating motors are not effective—and as our results show, small changes to our robot's motor speeds can have large and nonlinear effects upon its behavior. This renders the linear interpolation approach of

that work ineffective. As a consequence, rather than relying upon hand tuning, we instead employ Bayesian optimization to determine the most effective vibrational frequencies in a minimum number of physical trials.

Another distinction is that our soft tensegrity robot's intrinsic resonance is tuned to respond to the vibratory input of its actuators. The benefit of this tuned resonance is particularly noticeable when the performance of the soft tensegrity robot is compared with that of the rigid mock tensegrity robot described in experiments (Fig. 4). These soft tensegrity robots also stand in contrast to other more rigid tensegrity robots,[25,26] which generally try to suppress their resonance. Harnessing flexibility and resonance opens new research avenues for future soft robots, in particular, when mechanical design can be coupled with machine learning algorithms that automatically identify how to control the resonances.

One of the more thought-provoking illustrations of the potential of soft tensegrity robots is best observed on the Supplementary Video S1, at slow speed: once properly tuned by the learning algorithm, the vibrations induce large, visible deformations of the structures that create a step-like pattern for the ''feet'' at the end of the rigid struts (more quantitative results can be seen in Fig. 4C). These step-like patterns have the potential to allow tensegrity robots to step over small irregularities of the ground like a walking robot. Importantly, these patterns are made possible by the mix of soft and rigid elements in the same structure: they are likely to be much harder to induce and control both with a fully soft robot and with a fully rigid robot. A promising research avenue is to focus on how to control the movement of the feet explicitly and make steps that are little disturbed as possible by the irregularities of the floor.

An added benefit of vibrational locomotion for soft robotics is that, although our current robot is tethered, it could in principle be easy to power soft tensegrity robots with an embedded battery, by contrast with the many fluid-actuated soft robots,[4,8] which need innovative ways to store energy.[5] Nevertheless, soft tensegrity robots could excite their structure by other means; for instance, a flywheel that is rapidly decelerated could help the robot to achieve fast movements,[51] or high-amplitude, low-frequency oscillations could be generated by moving a pendulum inside the structure.[52]

Earlier work of ours on mobile tensegrities[34,35] used a rather simple interactive hill climber to discover effective locomotive gaits; however, this type of simplistic stochastic search was suboptimal. Although there may be little qualitative difference between our earlier gaits and those described here, there are profound differences in terms of the time and data efficiency of this Bayesian optimization approach. Most significantly, the hill climber places no emphasis on reducing the number of physical trials performed, and as a consequence required hundreds of trials and hours of experimentation before discovering effective gaits. These repeated physical trials put unnecessary wear on the robot, and required a substantial amount of human effort in resetting the robot between trials. Furthermore, the OpenCV-based optical tracking of the robot was rudimentary and lacked the spatial precision required of more effective algorithms. The Bayesian optimization approach we have used here, along with the high precision Optitrack system, profoundly reduces the number of physical trials and the corresponding wear on the robot, thereby increasing its capacity for faster and more autonomous resilience and adaptivity.

We purposely designed the robot so that the search space is as small as possible, which, in turn, makes it more likely for the robot to be capable of adapting in a few trials. Put differently, one of the main strengths of vibration-based locomotion is to make the search problem as simple as possible. Although, in principle, a variety of optimization techniques (e.g., simulated annealing[45]) might have been used, there are compelling reasons why our adaptation algorithm is based on Bayesian optimization, namely because (1) it is a principled approach to optimize an unknown cost/reward function when only a few dozen of samples are possible[37] (by contrast, the simulated annealing algorithm relies on the statistical properties of the search space, which are valid only with a large number of samples[53]), (2) it can incorporate prior knowledge in a theoretically sound way (including trusting real samples more than prior information),[12] and (3) it takes into account the acquisition noise.[38] For instance, Bayesian optimization is the current method of choice for optimizing the hyperparameters of neural networks,[37,54] because evaluating the learning abilities of a neural network is both noisy and time intensive. The downside of Bayesian optimization is a relatively high computational cost: the next sample is chosen by optimizing the acquisition function, which typically requires using a costly, nonlinear optimizer such as DIRECT[55] or Covariance Matrix Adaptation Evolution Strategy (CMA-ES)[56] (our implementation uses CMA-ES, see Materials and Methods). Put differently, Bayesian optimization trades data with computation, which makes it data efficient, but computationally costly. As we mostly care about data efficiency, we neglect this cost in this work, but it could be an issue on some low-power embedded computers.

Most black-box optimization (e.g., CMA-ES[56]) and direct policy search algorithms (e.g., policy gradients[57]) could substitute Bayesian optimization as an adaptation algorithm by directly optimizing the reward (instead of first modeling it with Gaussian process). Although they would not need time-intensive optimizations to select the next sample to acquire, these algorithms are tailored for at least a 1000 evaluations (e.g., 10^4 to 10^5 evaluations in benchmarks of two-dimensional functions for black-box optimizers[58]), are not designed to incorporate priors on the reward function, and are, at best, only tolerant to noisy functions. As a consequence, although algorithms such as CMA-ES could work as an adaptation algorithm, they appear to be a suboptimal choice for online adaptation when only a few dozen of evaluations are possible.

Traditional Bayesian optimization uses a constant mean as a prior,[46,47] that is, the only prior knowledge is the expectation of the cost/reward. By contrast, we show here that it is effective to introduce some basic intuitions about the system as a nonconstant prior on the reward function. We thus increase the data efficiency while keeping the learning algorithm theoretically consistent. Cully *et al.*[12] also used a nonconstant prior; however, (1) they generated it using a physics simulator, which is especially challenging for a vibrating tensegrity robot and (2) they only computed this prior for a discrete set of potential solutions, which, in turn, constrain Bayesian optimization to search only in this set. Here we follow a more continuous approach as our prior is a continuous function, and we show that relevant priors can be defined without needing a physics simulator. The more general problem of how to generate ''ideal'' priors is far from trivial. Intuitively, priors should come from a meta-learning process,[59]

for instance, an evolution-like process,[60] which would search for priors that would work well in as many situations as possible (i.e., instincts). Effectively implementing such a process remains an open grand challenge in machine learning.[59]

Putting all these attractive features altogether, soft tensegrity robots combine simplicity, flexibility, performance, and resiliency, which makes this new class of robots a promising building block for future soft robots. Of course, additional work is needed to have a more complete theory of the ''optimal suppleness'' of soft robots. Intuitively, too much suppleness would absorb the energy transmitted by the vibrator and prevent effective gaits, but, at the other end of the spectrum, a rigid robot cannot generate the form changes that are necessary for the most interesting gaits. This may be what we observed when we damaged the robot: by making the structure less constrained, the shape of the robot may have become looser, and ''softer,'' which impacted the maximum locomotion speed (alternatively, the removal of a spring might have prevented the transmission of some oscillations or some resonance modes). Nevertheless, for every kind of suppleness that we tried, the Bayesian optimization algorithm was always capable of finding some effective gaits, which means that the ''optimal softness'' does not need to be known *a priori* to discover effective locomotion. In this regard, trial-and-error approaches like those used here provide a valuable ability to respond and adapt to changes online in a rather robust manner, much like living systems.[12]

Several exciting open questions remain. So far, we have only demonstrated the effectiveness of this technique on a single substrate rather than across an entire range of environments. A compelling question we look forward to exploring in future work, for instance, is the extent to which the locomotive gaits we have discovered are robust and self-stabilizing in the face of external perturbations and changes in the substrate. Of course, the general problem of robust locomotion of any robot, much less soft robots, across multiple substrates and environments remains a relatively open topic. Recent work has, for instance, explored hand-picked strategies for the quasi-static locomotion of a cable-actuated tensegrity on inclined surfaces.[61] Our own ability to harness tensegrity vibration to induce large-scale and dynamic structure offers a compelling and promising method of discovering much more dynamic gaits for these environments. Indeed, our robot design is already capable of interesting behavioral diversity, including several unique rolling behaviors, which might be beneficial across environments—however, we were unable to explore these more deeply due to the tethered nature of this design. Nonetheless, the speed with which our algorithm can learn effective gaits, especially when damaged, provides a glimpse into how future soft robots could adapt to new and unexpected environments *in situ*, with no pre-existing knowledge or experience of that environment.

This leads to the recognition that the present prototype, although more than sufficient to demonstrate the claims of this article, is not yet fully autonomous: it relies on a tether for power, uses an external motion capture to evaluate its performance (locomotion speed), and uses an offboard computer for the learning algorithm. We are in the process of designing a fully wireless and autonomous tensegrity robot, as illustrated in Figure 6. This next generation of robot will be capable of substantially more dynamical behaviors, such as rolling and jumping, and more capable of exploring complex environments. Evaluating the performance of locomotion techniques using on-board processing could in principle be achieved either with accelerometers or with an embedded camera paired with a visual odometry algorithm,[12,62] but the vibrations and the fast movements of the struts are likely to disturb many visual algorithms. In addition, the modular nature of this wireless strut design means that we could explore an entire range of tensegrity robot morphologies, including those with considerably more than six struts.

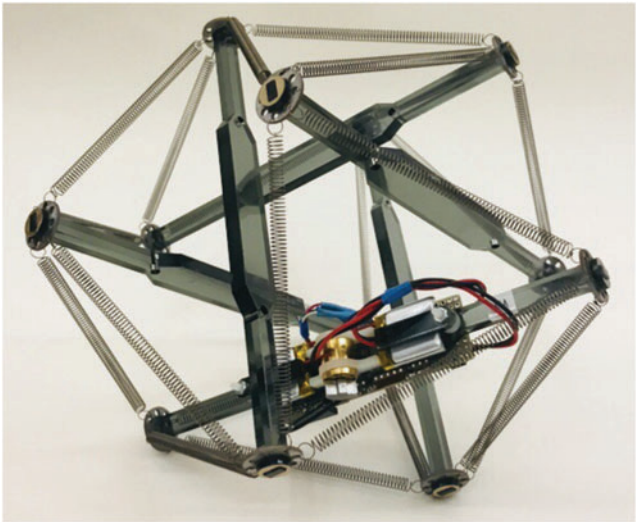

FIG. 6. An untethered version of the tensegrity robot. This new robot, still under development, will allow for more interesting dynamical behaviors such as rolling, as well as complex environments. This could in principle allow for completely on-board learning as well. Color images available online at www.liebertpub.com/soro

Overall, our soft tensegrity robots move thanks to the complex interactions between the actuators (vibrators), the structure (springs and struts), and the environment (the ground). This kind of emergent behavior is central in the embodied intelligence theory,[63] which suggests that we will achieve better and more life-like robots if we encourage such deep couplings between the body and the ''mind''—here, the controller. However, as demonstrated in this work, trial-and-error learning algorithms offer a strongly viable approach to discovering these emergent behaviors.

Acknowledgments

This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Project: ResiBots, grant agreement No 637972); J.R. received funding for a 1-month visit at University of Lorraine, France. The authors thank Dorian Goepp for his invaluable help and the ResiBots team for their comments on this article. The authors also thank Bill Keat for his help and insight into the design of the robot, and all the undergraduates of Union College's Evolutionary Robotics Group.

Computer code

(https://github.com/resibots/rieffel_mouret_2018_soft_tensegrity); all related code is available with an open-source license on GitHub.

Data availability

(http://dx.doi.org/10.5281/zenodo.1196304); these data are available on Dryad.

Author Contributions

J.R. designed the robot and J.R. and J.-B.M. designed the study, performed the experiments, analyzed the data, and wrote the article.

Author Disclosure Statement

No competing financial interests exist.

References

1. Sumbre G, Fiorito G, Flash T, Hochner B. Neurobiology: motor control of flexible octopus arms. Nature 2005;433: 595–596.
2. Gemmell BJ, Costello JH, Colin SP, Stewart CJ, Dabiri JO, Tafti D, *et al.* Passive energy recapture in jellyfish contributes to propulsive advantage over other metazoans. Proc Natl Acad Sci U S A 2013;110:17904–17909.
3. Simon MA, Woods WA Jr., Serebrenik YV, Simon SM, van Griethuijsen LI, Socha JJ, *et al.* Visceral-locomotory pistoning in crawling caterpillars. Curr Biol 2010;20:1458–1463.
4. Lipson H. Challenges and opportunities for design, simulation, and fabrication of soft robots. Soft Robot 2014;1:21–27.
5. Wehner M, Truby RL, Fitzgerald DJ, Mosadegh B, Whitesides GM, Lewis JA, *et al.* An integrated design and fabrication strategy for entirely soft, autonomous robots. Nature 2016;536:451–455.
6. Trimmer BA. New challenges in biorobotics: incorporating soft tissue into control systems. Appl Bionics Biomech 2008; 5:119–126.
7. Lin H-T, Leisk G, Trimmer BA. Goqbot: a caterpillar-inspired soft-bodied rolling robot. Bioinspir Biomim 2011; 6:026007.
8. Shepherd RF, Ilievski F, Choi W, Morin SA, Stokes AA, Mazzeo AD, *et al.* Multigait soft robot. Proc Natl Acad Sci U S A 2011;108:20400–20403.
9. Martinez RV, Branch JL, Fish CR, Jin L, Shepherd RF, Nunes R, *et al.* Robotic tentacles with three-dimensional mobility based on flexible elastomers. Adv Mater 2013;25:205–212.
10. Bartlett NW, Tolley MT, Overvelde JTB, Weaver JC, Mosadegh B, Bertoldi K, *et al.* A 3d-printed, functionally graded soft robot powered by combustion. Science 2015; 349:161–165.
11. Craig JJ. Introduction to robotics: mechanics and control. Volume 3. Upper Saddle River, NJ: Pearson/Prentice Hall, 2005.
12. Cully A, Clune J, Tarapore D, Mouret J-B. Robots that can adapt like animals. Nature 2015;521:503–507.
13. Berretty R-P, Goldberg K, Overmars MH, van der Stappen AF. Trap design for vibratory bowl feeders. Int J Rob Res 2001;20:891–908.
14. Reznik D, Canny J. The coulomb pump: a novel parts feeding method using a horizontally-vibrating surface. In: Proceedings of the 1998 IEEE International Conference on Robotics and Automation. IEEE, 1998, Volume 1, pp. 869–874.
15. Reznik D, Moshkovich E, Canny J. Building a universal planar manipulator. In: Distributed Manipulation. Böhringer KF, Choset H (Eds.) Boston, MA: Springer: 2000, pp. 147–171.
16. Vartholomeos P, Papadopoulos E. Analysis, design and control of a planar micro-robot driven by two centripetal-force actuators. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation. ICRA 2006. IEEE, 2006, pp. 649–654.
17. Joe WY. Vibration learning and control towards vibration actuated robotic systems. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015, pp. 375–377.
18. Rubenstein M, Ahler C, Nagpal R. Kilobot: a low cost scalable robot system for collective behaviors. In: Proceedings of the IEEE International Conference Robotics and Automation (ICRA). IEEE, 2012, pp. 3293–3298.
19. Parshakova T, Cho M, Casinelli A, Saakes D. Ratchair: furniture learns to move itself with vibration. In: ACM SIGGRAPH 2016 Emerging Technologies. ACM, 2016, p. 19.
20. Valero-Cuevas FJ, Yi J-W, Brown D, McNamara RV, Paul C, Lipson H. The tendon network of the fingers performs anatomical computation at a macroscopic scale. IEEE Trans Biomed Eng 2007;54:1161–1166.
21. Wang N, Naruse K, Stamenović D, Fredberg JJ, Mijailovich SM, Tolić-Nørrelykke IM, *et al.* Mechanical behavior in living cells consistent with the tensegrity model. Proc Natl Acad Sci U S A 2001;98:7765–7770.
22. Snelson K. The art of tensegrity. Int J Space Structures 2012;27:71–80.
23. Skelton RE, de Oliveira MC. Tensegrity Systems, Volume 1. New York: Springer, 2009.
24. Ceceri K. Making Simple Robots: Exploring Cutting-Edge Robotics with Everyday Stuff. San Francisco: Maker Media, Inc., 2015.
25. Caluwaerts K, Despraz J, Işçen A, Sabelhaus AP, Bruce J, Schrauwen B, *et al.* Design and control of compliant tensegrity robots through simulation and hardware validation. J R Soc Interface 2014;11:20140520.
26. Koizumi Y, Shibata M, Hirai S. Rolling tensegrity driven by pneumatic soft actuators. In: Proceedings of the IEEE International Conf. on Robotics and Automation (ICRA). IEEE, 2012, pp. 1988–1993.
27. Bliss TK, Iwasaki T, Bart-Smith H. Resonance entrainment of tensegrity structures via cpg control. Automatica 2012; 48:2791–2800.
28. Mirletz BT, Bhandal P, Adams RD, Agogino AK, Quinn RD, SunSpiral V. Goal-directed cpg-based control for tensegrity spines with many degrees of freedom traversing irregular terrain. Soft Robot 2015;2:165–176.
29. Jakobi N, Husbands P, Harvey I. Noise and the reality gap: The use of simulation in evolutionary robotics. In: European Conference on Artificial Life. Berlin, Heidelberg: Springer, 1995, pp. 704–720.
30. Koos S, Mouret J-B, Doncieux S. The transferability approach: crossing the reality gap in evolutionary robotics. IEEE Trans Evol Comput 2013;17:122–145.
31. Böhm V, Zimmermann K. Vibration-driven mobile robots based on single actuated tensegrity structures. In: 2013 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2013, pp. 5475–5480.
32. Berger B, Andino A, Danise A, Rieffel J. Growing and evolving vibrationally actuated soft robots. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation. ACM, 2015, pp. 1221–1224.
33. Oppenheim IJ, Williams WO. Vibration of an elastic tensegrity structure. Eur J Mech A Solids 2001;20:1023–1031.

34. Khazanov M, Humphreys B, Keat W, Rieffel J. Exploiting dynamical complexity in a physical tensegrity robot to achieve locomotion. Adv Artif Life ECAL 2013;12:965–972.
35. Khazanov M, Jocque J, Rieffel J. Evolution of locomotion on a physical tensegrity robot. In: ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems. Cambridge, MA: MIT Press, 2014, pp. 232–238.
36. Ghahramani Z. Probabilistic machine learning and artificial intelligence. Nature 2015;521:452–459.
37. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N. Taking the human out of the loop: a review of bayesian optimization. Proc IEEE 2016;104:148–175.
38. Rasmussen CE. Gaussian processes in machine learning. Advanced Lectures on Machine Learning. Berlin, Heidelberg: Springer, 2004. pp. 63–71.
39. Davison AJ, Reid ID, Molton ND, Stasse O. Monoslam: real-time single camera slam. IEEE Trans Pattern Anal Mach Intell 2007;29:1052–1067.
40. Booker AJ, Dennis JE Jr., Frank PD, Serafini DB, Torczon V, Trosset MW. A rigorous framework for optimization of expensive functions by surrogates. Struct Optim 1999; 17:1–13.
41. Forrester AIJ, Keane AJ. Recent advances in surrogate-based optimization. Prog Aerosp Sci 2009;45:50–79.
42. Jin Y. Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 2011; 1:61–70.
43. Simpson TW, Mauery TM, Korte JJ, Mistree F. Comparison of response surface and kriging models for multidisciplinary design optimization. Am Inst Aeronaut Astronaut 1998;98:1–16.
44. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Global Optim 1998;13:455–492.
45. Sacks J, Welch WJ, Mitchell TJ, Wynn HP, *et al.* Design and analysis of computer experiments. Stat Sci 1989;4: 409–423.
46. Lizotte DJ, Wang T, Bowling MH, Schuurmans D. Automatic gait optimization with Gaussian process regression. In: Proceedings of the the International Joint Conference on Artificial Intelligence (IJCAI). Palo Alto: AAAI Press, 2007, Volume 7, pp. 944–949.
47. Calandra R, Seyfarth A, Peters J, Deisenroth MP. An experimental comparison of bayesian optimization for bipedal locomotion. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, China, 2014, pp. 1951–1958.
48. Brochu E, Cora VM, De Freitas N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv 2010; DOI: arXiv:1012.2599.
49. Mouret J-B, Clune J. Illuminating search spaces by mapping elites. arXiv 2015; DOI: arXiv:1504.04909.
50. Reuillon R, Schmitt C, De Aldama R, Mouret J-B. A new method to evaluate simulation models: the calibration profile (cp) algorithm. JASSS-J Artif Soc S 2015;18:12.
51. Romanishin JW, Gilpin K, Rus D. M-blocks: momentum-driven, magnetic modular robots. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2013, pp. 4288–4295.
52. Chase R, Pandya A. A review of active mechanical driving principles of spherical robots. Robotics 2012;1:3–23.
53. Kirkpatrick S, Gelatt CD, Vecchi MP, *et al.* Optimization by simulated annealing. Science 1983;220:671–680.
54. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems. Red Hook: Curran Associates, Inc., 2012, pp. 2951–2959.
55. Finkel DE. Direct Optimization Algorithm User Guide. Center for Research in Scientific Computation, North Carolina State University. March 2, 2003.
56. Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). Evol Comput 2003; 11:1–18.
57. Peters J, Schaal S. Policy gradient methods for robotics. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006, pp. 2219–2225.
58. Hansen N, Auger A, Ros R, Finck S, Pošík P. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation. ACM, 2010, pp. 1689–1696.
59. Lemke C, Budka M, Gabrys B. Metalearning: a survey of trends and technologies. Artif Intell Rev 2015;44:117–130.
60. Kirschner MW, Gerhart JC. The Plausibility of Life: Resolving Darwin's dilemma. New Haven: Yale University Press, 2006.
61. Chen L-H, Cera B, Zhu EL, Edmunds R, Rice F, E. Tang A. Bronars, S.R. Malekshahi, O. Romero, and A.K. Agogino. Inclined surface locomotion strategies for spherical tensegrity robots. In: Proceedings of the International Conference on Intelligent Robotics and Systems (IROS 2017). Vancouver, Canada, 2017, pp. 4976–4981.
62. Forster C, Pizzoli M, Scaramuzza D. Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 15–22.
63. Pfeifer R, Lungarella M, Iida F. Self-organization, embodiment, and biologically inspired robotics. Science 2007; 318:1088–1093.
64. Tucker VA. Energetic cost of locomotion in animals. Comp Biochem Physiol 1970;34:841–846.

Address correspondence to:
John Rieffel
Department of Computer Science
Union College
Schenectady, NY 12308

E-mail: rieffelj@union.edu