PROCEEDINGS A

royalsocietypublishing.org/journal/rspa

Research

Modular representation and control of floppy networks

Siheng Chen[1], Fabio Giardina[1], Gary P. T. Choi[4] and L. Mahadevan[1,2,3]

[1]School of Engineering and Applied Sciences, [2]Department of Physics, and [3]Department of Organismic and Evolutionary Biology, Harvard University, Cambridge MA 02138, USA
[4]Department of Mathematics, Massachusetts Institute of Technology, Cambridge MA 02139, USA

SC, 0000-0002-5494-0853; FG, 0000-0002-2660-5935; GPTC, 0000-0001-5407-9111; LM, 0000-0002-5114-0519

Cite this article: Chen S, Giardina F, Choi GPT, Mahadevan L. 2022 Modular representation and control of floppy networks. *Proc. R. Soc. A* **478**: 20220082.
https://doi.org/10.1098/rspa.2022.0082

Received: 30 January 2022
Accepted: 20 July 2022

Subject Areas:
complexity, applied mathematics, mechanical engineering

Keywords:
network, floppy modes, sparsity, control

Author for correspondence:
L. Mahadevan
e-mail: lmahadev@g.harvard.edu

Geometric graph models of systems as diverse as proteins, DNA assemblies, architected materials and robot swarms are useful abstract representations of these objects that also unify ways to study their properties and control them in space and time. While much work has been done in the context of characterizing the behaviour of these networks close to critical points associated with bond and rigidity percolation, isostaticity, etc., much less is known about floppy, underconstrained networks that are far more common in nature and technology. Here, we combine geometric rigidity and algebraic sparsity to provide a framework for identifying the zero energy floppy modes via a representation that illuminates the underlying hierarchy and modularity of the network and thence the control of its nestedness and locality. Our framework allows us to demonstrate a range of applications of this approach that include robotic reaching tasks with motion primitives, and predicting the linear and nonlinear response of elastic networks based solely on infinitesimal rigidity and sparsity, which we test using physical experiments. Our approach is thus likely to be of use broadly in dissecting the geometrical properties of floppy networks using algebraic sparsity to optimize their function and performance.

1. Introduction

Many different physical systems that span multiple scales can be represented as graphs or networks. These include proteins [1], robots [2,3] and collections of

Electronic supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.6134121.

THE ROYAL SOCIETY
PUBLISHING

individual organisms or agents [4–6]. The physical, chemical and informational properties of these systems such as coordinated motion, communication and control are both enabled and constrained by the nature of the connectivity of the underlying network. Examples include task-relevant representations associated with coordination in motion planning and motor control problems in robotics and neuroscience [2,3], spatial segmentation and clustering in biomolecules [7,8] and the mechanics of protein elastic networks [9]. In all these examples, the underlying topological graphs are also automatically endowed with a geometric structure due to the embedding in two- or three-dimensional Euclidean space with edges corresponding to the distance between the nodes. This makes the understanding of the spatial structure and organization of the degrees of freedom (DoFs) in these networks an issue of critical importance, as learning how to identify, actuate, combine or remove certain DoF will provide the means to control these systems.

Since Maxwell's early work in 1864 [10], rigidity theory has been a powerful tool to model these graphs, initially treated as mechanical networks [11]. A key concept there is that of the deviation from isostaticity wherein the network is poised between being rigid and soft because the number of DoFs are just balanced by the number of constraints. Two main approaches are currently used for the study of these networks based on whether they are stiff or compliant. First, for networks with stiff links, the number of DoF can be found using infinitesimal rigidity theory and graph theory [12,13], quantities that have been leveraged in network control, metamaterial design [14–17] and determination of the minimal rigid formations [18]. Second, for networks with compliant links, mechanical properties such as bulk and shear modulus can be calculated via molecular dynamics simulation [19]. The intermediate case of networks that transition between stiff and compliant, i.e. can be transformed from mechanisms with floppy modes (zero energy deformations) to structures (finite energy deformations), has been the focus of much recent work from the perspective of phase transitions and critical phenomena. A recent question of interest in the field is the mechanical response of networks near the jamming transition, or the isostatic limit (as already defined by Maxwell more than 150 years ago) [19–21], with a particular focus on the divergence of correlation lengths near the jamming transition [22–24]. Simultaneously, there has been interest in the applications of these ideas to the control of mechanical networks [25,26], protein allostery [27–29], stiffening underconstrained networks by geometric incompatibility [24], actuating individual floppy modes using correlated noise [30], tuning bulk and shear moduli by varying the number of bonds [31,32], etc.

However, almost without exception, these studies focus on either overconstrained systems where there are no floppy modes or systems near the isostaticity transition where few floppy modes are present. But what if one is far from the isotaticity threshold with a large number of floppy modes? What is the best way to uncover modularity in these floppy modes, as manifest in their spatial structure and represented as some combination of locality, hierarchy, separability, and nestedness, for example? And what is the best way to control these floppy modes? Since actuating all the floppy modes at once is both expensive and inefficient, are there efficient representations that can guide their activation or suppression? Since many natural and artificial networks must start out as underconstrained networks, answers to these questions are likely to be of broad value. Here, we present a systematic framework to identify and represent multiple floppy modes and use this representation to construct protocols to efficiently actuate and control the network. To uncover the hidden hierarchy in the floppy modes, we use the notion of maximum sparsity in the mode representation, defined in terms of the smallest number of non-vanishing vector elements in the representation. We then demonstrate the application of these sparse mode representations in a range of different applications: robotic reaching tasks with motion primitives from the sparsity-based representation and predicting the linear and nonlinear response of elastic networks based solely on infinitesimal rigidity and sparsity.

2. Sparse basis representation

Any Euclidean graph can be represented as a mechanical network with N nodes whose movement is constrained by a set of edges $\mathcal{E}$. If the length of these edges is fixed, this leads to a set of geometric constraints that can be written as follows:

$$g_i = ||\boldsymbol{x}_p - \boldsymbol{x}_q||^2 - l_{pq}^2 = 0, \tag{2.1}$$

where node q is connected to node p by an edge $[p, q] \in \mathcal{E}$ and $\boldsymbol{x}_p$ denotes the coordinates of node p. A floppy or zero energy mode is defined as an infinitesimal mode of motion $\boldsymbol{dx}$ in the structure, which does not violate any constraints. Therefore, for any constraint g_i, we have $dg_i = \sum_j \partial g_i/\partial x_j \, \mathrm{d}x_j = 0$. By defining the elements of the rigidity matrix $\boldsymbol{R}$ by $(\boldsymbol{R})_{ij} = \partial g_i/\partial x_j$, all the constraints can be written in terms of a linear matrix relation,

$$\boldsymbol{R} \cdot \boldsymbol{dx} = 0. \tag{2.2}$$

Therefore, the null space of $\boldsymbol{R}$ contains information about all the floppy modes in the system: each basis in the null space of $\boldsymbol{R}$ corresponds to a mode of motion consistent with the constraints. More explicitly, the dimension of $\boldsymbol{R}$ is m by $n \times N$, where m is the number of constraints, n is the number of nodes and N is the spatial dimension. In two-dimensional Euclidean space, $\boldsymbol{dx}$, a vector in the null space of $\boldsymbol{R}$, can be written as $(\mathrm{d}x_{1x}, \mathrm{d}x_{1y}, \ldots, \mathrm{d}x_{nx}, \mathrm{d}x_{ny})$, representing an infinitesimal movement of the nodes that do not violate the constraints in the network. After normalization, $\sqrt{\mathrm{d}x_{ix}^2 + \mathrm{d}x_{iy}^2}$ is the infinitesimal displacement of node i and $\arctan(\mathrm{d}x_{iy}/\mathrm{d}x_{ix})$ indicates its direction. Each set of linearly independent basis vectors spanning the whole null space provides a possible representation of all the motions, and any non-degenerate rotation matrix can transform one basis to another, so that there are infinitely many equivalent representations of the null space.

As a simple example, consider a robot arm shown in figure 1*a* and ignore the movement of the fingers for now. The biologically and technologically natural representations of the modes of motion in this network are the rotation around the base (shoulder) and the rotation around the first joint (elbow) (figure 1*d*,*e*). However, there are other equivalent representations, e.g. one that consists of (i) allowing the forearm and upper arm to rotate together in the same direction and (ii) allowing them to rotate in opposite directions. On a completely different scale, consider a molecule shown in figure 1*g*. When the backbone atoms are fixed in space, the movement of the side chains on the left and on the bottom right should be independent of each other, resulting in independent rotation modes (figure 1*j*,*k*). However, it is also possible to represent the motions using a linear combination of the two rotations. In both these examples, we see that while there are mathematically equivalent representations of the null space basis, some representations seem to be intuitively better as they are modular and show spatial structure embodied in locality, nestedness and hierarchy that might make them easier to control. We note that the notion of modularity here is different from that in graph theory, defined purely based on links (connections) in the network. A modular representation of floppy modes is a representation where different modes actuate different parts of networks with little overlap or interaction. A modular network can have a representation of floppy modes that is either modular or non-modular depending on the decomposition algorithm, as we shall see later. We limit our study to two-dimensional networks without edge crossings and interactions for clarity of presentation and since it is easier to compare to physical experiments with silicone elastic networks. However, our framework can be extended to three dimensions since the definition of the rigidity matrix is not restricted to two dimensions.

So how can we find a representation basis that maximizes this modularity and hierarchy? A natural response might be to use the singular value decomposition (SVD) of the rigidity matrix $\boldsymbol{R}$ to get a set of zero modes and states of self-stress [33,34]. However, the zero modes identified from SVD typically involve motion in all coordinates and are neither modular nor hierarchical (see electronic supplementary material, Section S1.3).

An approach to achieve modularity is to attain maximum sparsity in the zero mode basis, i.e. find those modes whose infinitesimal motions only involve a small number of nodes in a small

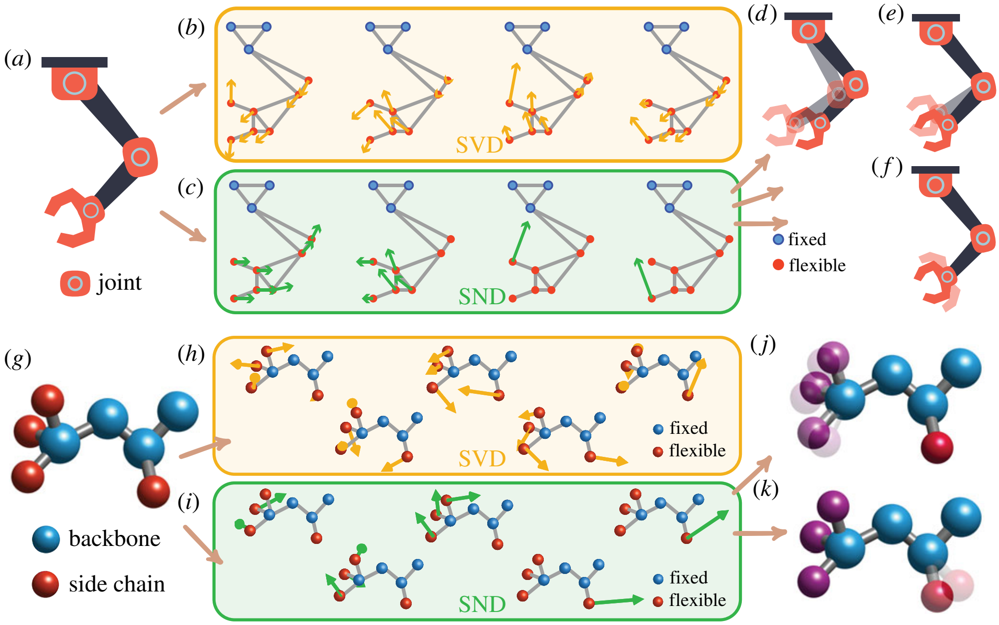

Figure 1. Mode decomposition in robotics and biology. (*a*) A simple model robot arm with three joints and four DoF. Abstracted as a node-bond network, the four floppy modes can be represented as in (*b*) using the SVD method or in a hierarchical way as in (*c*) using the sparse null space basis decomposition (SND) method. The second method provides a clear separation of all the motions (*d*, full arm, *e*, forearm and *f*, fingers). (*g*) A biological example of a molecule with backbone atoms (blue, spatially fixed), and side chains (red, flexible). The left three red atoms are pairwise connected (not shown). Using the two methods as in the example given earlier, the five floppy modes are decomposed in *h* and *i*. By using the SND method, the motions of the left side chain and the bottom right side chain are clearly separated, as shown in (*j*) and (*k*). (Online version in colour.)

spatially localized neighbourhood. These local modes will also be naturally spatially separated, as any linear combination of them will decrease the resulting sparsity in the bases. The remaining modes, if any, are more global as they involve motions of more nodes. Thus, a classification method based on maximizing sparsity has much potential in helping build a modular hierarchical representation. Mathematically our problem then translates to finding the sparsest null space basis of the rigidity matrix. Finding a sparse null space basis has been the subject of study in large-scale constrained optimization and is known to be NP-hard [35], so that all current attempts to solve this use a variety of heuristic algorithms that determine which linear recombination derived from the null space of the rigidity matrix is sparse and hence local, modular and hierarchical. Here, we adapt and generalize a recently proposed algorithm from numerical linear algebra to determine a minimal number of non-zero entries in the null space basis of a matrix [36]—the sparse null space basis decomposition [37], denoted as SND (see electronic supplementary material, Section S1.1 for details). We show that it has a natural geometric interpretation in the context of the rigidity matrix and is perfectly suited to determine the maximally sparse basis for the rigidity matrix $\boldsymbol{R}$. Furthermore, we generalize this into a multi-scale method to determine a hierarchy of zero energy modes (in terms of their participation in a motion), and this allows us to answer the question of how to determine a sparse, modular and hierarchical representation of the zero energy modes in a graph representation of a range of networks.

3. Hierarchical representation and spatial separation of floppy modes

To understand how this works in practice, we first consider a simple mechanical network—the robot arm mentioned earlier (figure 1*a*) that can be abstracted as an underconstrained structure with nodes resembling joints (the shoulder, elbow and wrist), with the base (shoulder) fixed. Constructing the rigidity matrix from the links in the network, we apply the SND method to find a set of basis vectors (floppy modes), and for comparison, we use a simple SVD algorithm to generate another set of basis vectors. (For simplicity, we only compare SND with SVD in the main text. See electronic supplementary material, Section S1.3 for a comparison with other matrix decomposition methods such as QR decomposition.) As shown in figure 1*b*, an SVD basis represents a mixed motion: fingers and forearm move together in different directions, and it is hard to interpret the physical meaning of the basis vectors. In contrast, the SND method finds an easily interpretable representation (figure 1*c*)—there are four modes that clearly distinguish between local and global motions. The four modes reflect the movement of the whole arm, the wrist and the two fingers, respectively, from left to right. Since SND is trying to maximize the sparsity, the simple finger modes, only involving one node (two coordinates), are preferred over more complicated modes.

In addition to separating the modes hierarchically, we now show that the SND algorithm also separates them spatially. Consider the simple molecule (figure 1*g*) with backbone atoms (blue) spatially fixed and side chains (red) being flexible. The three atoms connected to the leftmost blue atoms are connected to each other as a single functional group (these three bonds between the three red atoms are not shown in the figure). Since each atom in the side chain has two rotational DoFs, constraint counting shows that there are in total $2 \times 4 - 3 = 5$ DoF. With the SVD method, we can find one representative basis of the five modes shown in figure 1*h*. We see that the modes are hard to interpret, as each of them involves the motion of all the atoms in the side chain. With the SND method, we see another representation shown in figure 1*i*. In contrast to the SVD basis, here we clearly see a separation of the motions of two groups of atoms—the top three modes only involve the functional group on the left, while the bottom two modes only involve the atom in the bottom right.

As just described, there is no scale-dependent hierarchy in deploying the SND algorithm. However, in many real physical systems in protein networks, or robotic systems, there is a need for a scale-dependent approach to determining the optimally sparse and modular representation. To achieve this, we can implement a multi-scale version of the SND algorithm, using the concept of bi-connected components in graph theory to identify all the 'hinges' in the graph where a section can rotate freely around another. We then apply SND to each sub-graph once these rotational modes are identified. The intrinsic recursive nature of this multi-scale SND ensures that the hierarchy is built-in in the mode representation and delivers a more intuitive mode representation of the robotic arm shown in figure 1*c*, leading to a more intuitive mode involving the rotation of the wrist and the finger such that they rotate together as a rigid section (see electronic supplementary material, Section S2 for more details and an extensive discussion of this multi-scale approach).

To understand the performance of the SND relative to the SVD method, we deploy both on a random planar triangular graph of a 4×4 network with 4 DoF. The SND method clearly identifies all the floppy modes (figure 2*a*) in a hierarchical way (the left three modes) and a spatially separated way (the last mode). On the contrary, the SVD method produces a mixture of modes that are hard to interpret (figure 2*b*). For a quantitative comparison, we define the size s of a mode to be the number of non-zero entries in its vector representation, and the participation rate P of a set of modes as the sum of the sizes of all the modes: $P = \sum_j s_j$, which quantifies the sparsity as well as the modularity of the modes (different spatially separated modes involve different set of nodes, thus smaller P). We note that the SND decomposition is dependent on the row order of the rigidity matrix, and the decomposition mentioned earlier follows from one random shuffle (of row order) of the rigidity matrix; the advantage of SND can be seen by repeating the experiments and checking the average performance. By repeating the mode decomposition 100 times (each time with the rows of $\boldsymbol{R}$ randomly shuffled), we can calculate the average participation rate P for modes generated by the SND method, which is significantly lower than that of modes generated by SVD (figure 2*c*).

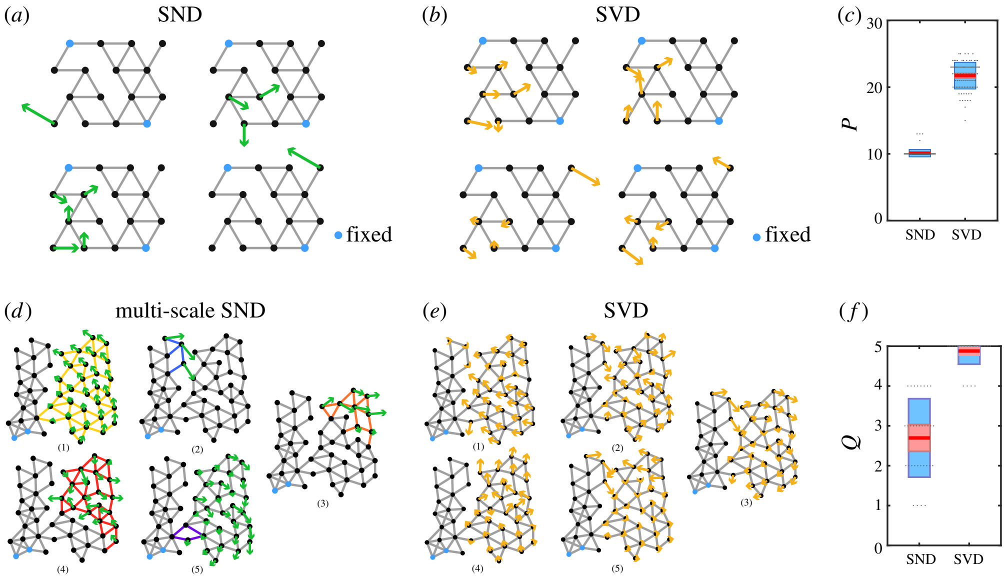

Figure 2. Mode decomposition and activation in mechanical networks. (*a*) For this 4 × 4 triangular network, the SND algorithm generates a set of modes with clear hierarchy (from mode 1 to 3) that are spatially separated (mode 4). (*b*) With the SVD algorithm, these modes are not properly separated, and it is hard to attach physical meaning to a single mode. (*c*) The participation rate *P* using the SND method is much lower than that using the SVD method. (*d*) For this complicated network, multi-scale SND can identify the rotational mode (1) and floppy regions (2–5) in a hierarchical manner. (*e*) The SVD creates neither hierarchical nor local motions since most nodes are involved in all modes. (*f*) The distribution of *Q* (number of modes a node is involved in) of SND and SVD representation. (Online version in colour.)

To further demonstrate the intuitive representations arising from SND, we consider a more complex network as shown in figure 2*d*. The multi-scale SND method can not only identify the rotational mode of the right sub-graph with respect to the left sub-graph (mode (1)) but also identify the floppy regions in a spatially separated (mode (2)) and hierarchical manner (modes (3) to (5)). This representation agrees with our intuition and furthermore reveals the non-trivial orange and red floppy sub-graphs, which are invisible otherwise. In contrast, the modes derived from SVD involve almost all nodes that can possibly move (figure 2*e*). Defining Q as the number of modes that a node is involved in, in figure 2*f*, we see that the representation with SND has a smaller average Q compared to SVD. Overall, we see that SND outperforms SVD and other matrix decomposition methods (e.g. QR decomposition) by being interpretable, naturally separable and hierarchical, and finally being computationally faster (see electronic supplementary material, section S1.3 for more details).

4. Reaching control with sparse motion primitives

The advantages of a sparse mode representation of a floppy network are most naturally seen in the context of network control: our multi-scale generalization of SND has the desirable properties for motion primitives relevant for motor control and robotics. We therefore set out to use these derived modes as motion primitives in a simple grasping task using the robotic arm shown in figure 3*a* whose task is to pinch a randomly placed target position with its two fingers. The initial arm position is randomized, and at every time step, the control algorithm selects and activates the mode, which minimizes the distance of the two fingers to the target, the activation time of which is shown in figure 3*b*,*c*. Mode 1 corresponds to the activation of the whole arm, mode 2 to the lower arm and modes 3 and 4 to the two fingers. On average, we see a cascading activation of

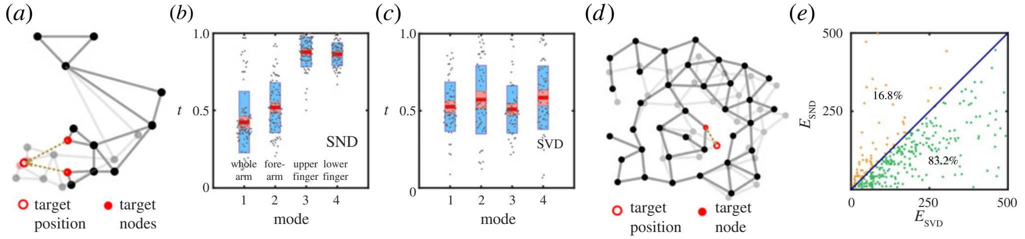

Figure 3. Mode decomposition and activation in mechanical networks. (*a*) Grasping control task for a simple manipulator arm with two fingers. (*b*) Activation of the sparsity maximizing (SND) modes in the grasping task leads to hierarchical activation of the modes over time from largest (whole arm, mode 1) to intermediate (lower arm, mode 2) to smallest (fingers, modes 3 and 4). The red line denotes the mean, the red box denotes a standard error of 1.96 times the mean (95% confidence interval) and the blue box shows the standard deviation. Initial arm positions and target locations are randomized. (*c*) Activation of the SVD modes in the grasping task shows no hierarchical activation of modes for randomized initial arm positions and target locations. (*d*) Reaching control task for a random network. (*e*) Energetic cost of the random network control task with motion primitives derived using SVD and SND for randomized target positions and nodes. For the same control task, SND finds a solution that uses less energy 83.2% of the time. See electronic supplementary material, Fig. S6 for a more detailed comparison. (Online version in colour.)

the modes in time, starting with the largest arm mode and ending with the two finger modes, as shown in figure 3*b* where the time for the activation of mode 1 and 2 is shorter than that of 3 and 4. For comparison, we also use SVD to solve the problem and see no clear distinction in terms of activation time of the modes (see electronic supplementary material, Movie 3).

The hierarchical activation may appear obvious in the simplified robotic arm, but for complex networks, our intuition may fail while modes derived using the sparsest approach can still provide spatially separated and hierarchical motion primitives. We deployed our approach for sparse motion primitives on the random network shown in figure 3*d* in a reaching control task. A node chosen at random is required to reach a randomized target location. Again we select the mode that minimizes the distance to target at every time step. We track the required kinetic energy until task completion summed over all nodes for the motion primitives. As shown in figure 3*e*, for an identical control task, a comparison with the SVD based primitives shows that the use of SND motion primitives leads to a reduced energetic cost to reach the target in 83.2% of the cases compared to the SVD motion primitives. This is perhaps not surprising because most SND modes are local and do not involve the motion of the whole network and thus minimize irrelevant and redundant motions (see electronic supplementary material, Movie 4 for an example and electronic supplementary material, Fig. S6 for a more detailed comparison). It is worth noting that the SND modes are comparable to motor primitives found using unsupervised learning [38] and show a significant improvement in performance of reinforcement learning agents. Our approach might thus provide a general interpretable framework for the determination of sparse modes for motion primitives in motor control.

5. Control and sequential tuning of a mechanical network

The sparse mode representation not only allows us to control rigid systems but also provides insight into the control of deformable networks, which we turn to next. The separation of the modes into local and global ones provides a simple and efficient guideline for the rigidification of a system. When the network is underconstrained, it is better to first freeze the largest mode (here ‘large’ refers to the size of the mode *s* defined above); within the largest mode, it is better to freeze the node with the largest infinitesimal displacement. A simple robot arm illustrates this idea (figure 4*a*). To rigidify the whole arm, it is better to freeze the motion of the wrist relative to the base since it has the largest infinitesimal displacement.

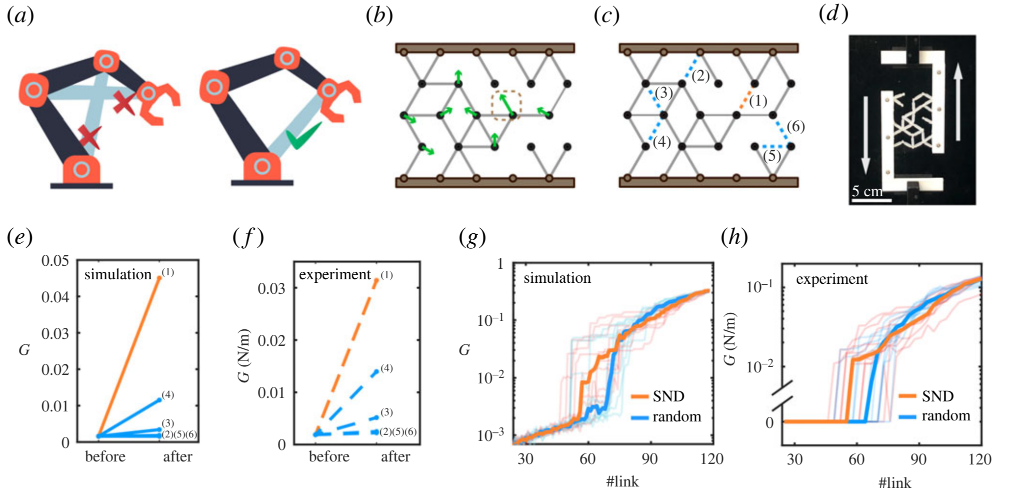

Figure 4. Efficient rigidification and sequential tuning in mechanical networks. (*a*) When rigidifying a structure, the most efficient way is to find the 'biggest mode', and freeze it by adding a link to the node with largest movement. (*b*) A triangular network with the top and bottom boundary fixed and the largest floppy mode shown in green arrows. The dashed box shows the node with the largest infinitesimal displacement. (*c*) To rigidify the network, the orange link (1) can be added. Five other links are randomly chosen ((2) to (6)) for comparison. (*d*) The experimental setup to measure the shear modulus. (*e*) The numerical simulation of the shear modulus *G* of the network (rigid bonds now treated as elastic springs) before and after each of the six links is added. (*f*) The experimental measurement of the shear modulus before and after each of the six links is added. (*g*) For a 7 × 7 network, links are added one by one using the efficient (orange) or random (blue) rigidification, respectively. The shear modulus is simulated at each time step for 10 runs, and the thick line shows the median. (*h*) The experimental verification of the result in *g* using a similar setup as in (*d*). (Online version in colour.)

So far, we have based our analysis on the study of the rigidity matrix alone and thus have been limited to considerations of infinitesimal motions. In a mechanical setting, this is equivalent to either rigid-bond networks or spring networks with infinite stiffness k. We now explore the possibility of generalizing our approach for the control of rigidification of a spring network with finite stiffness k. Inspired by the robot arm in figure 4*a*, we propose identifying the largest mode (with the largest s, i.e. involving the largest number of nodes) and then adding edges between the node with the largest infinitesimal displacement $\left(\sqrt{\mathrm{d}x_{ix}^2 + \mathrm{d}x_{iy}^2}\right)$ and one of its neighbours to rigidify the network (which we refer to as the MS protocol, for maximizing stiffness). Intuitively, freezing or rigidifying this large floppy mode adds more low-frequency modes to the system, thus making the structure more connected and rigid. (See electronic supplementary material, Section S4 for more details, improvements to the protocol, and comparison with other graph theory-based methods.)

To test this idea for efficient rigidification, we first apply it to a 5 × 5 network with the link pattern shown in figure 4*b*, with the top and bottom boundary fixed on two bars, the biggest floppy mode represented in green arrows and the node with the largest infinitesimal displacement identified in the dashed box. To freeze this mode, link (1) can be added (figure 4*c*), while links (2) to (6) are randomly chosen among the remaining free links. To test whether our MS control protocol from the infinitesimal approach works in a finitely extensible spring network sense (finite k), we use the shear modulus G to characterize the overall rigidity (stiffness). If the spring stiffness is k and the rest length is l_0 for all the springs, and the strain is γ, G can be calculated as follows [19]: $G = (2/A)(E/\gamma^2)$, where A is the area of the network, and E is the stretching energy $E = \frac{1}{2}(k/l_0)\sum_{[i,j]\in\mathcal{E}}(\boldsymbol{x}_i - \boldsymbol{x}_j)^2$. We also carry out a physical experiment with cast silicone rubber networks (see electronic supplementary material, Section S4.9 for details), with the force measurement setup as shown in figure 4*d*. In figure 4*e*,*f*, we show that both numerical simulations and the experiments confirm that adding link (1) increases the shear modulus much more significantly than adding other links, i.e. the MS protocol based on infinitesimal rigidity remains of value for finitely deformable elastic networks where geometric nonlinearities cannot be neglected.

With the successful test of the MS protocol at the single bond level, we proceed to apply the MS control protocol on the sequential tuning of system rigidity. As an example, we apply the tuning process to a 7 × 7 triangular network. Starting from 10 network configurations with 20% randomly chosen links, we sequentially add links one by one using the MS protocol and compare the result with a random protocol. With the same number of links added, the network tuned with MS protocol (the orange line) indeed has larger shear modulus than the case where links are added randomly (the blue line) when the network has a non-zero shear modulus (50–70 links) (figure 4*g*, the thick line shows the median for each case). Physical experiments with cast networks (see electronic supplementary material, Section S4.9 for details) verify our numerical results (figure 4*h*), suggesting our method can be applied to find the links that rapidly stiffen an underconstrained mechanical network. All together, our results show that the infinitesimal approach (on a rigid network with *infinite k*) using the rigidity matrix **R** is valuable even in determining control protocols for a spring network (with *finite k*). (See electronic supplementary material, Section S4 for some improvements to the protocol and a more detailed comparison with other methods.)

As a side note, we notice that once the network has some non-zero shear modulus (associated with 80–110 links), the random protocol actually outperforms the MS protocol. To understand this, we observe that once a connected percolating cluster has formed, it is better to connect within that cluster to strengthen this network, rather than adding links to freeze the remaining floppy modes, which most likely are near the boundaries. We also notice that when the network has few links (less than 50), there is a small non-zero shear modulus. This artefact comes from the random noise added to facilitate the simulation (see electronic supplementary material, Section S4A for more details).

6. Identifying critical links in a mechanical network

So far, we have shown the power of the hierarchical representation of modes based on infinitesimal rigidity on the control of global network rigidity, as measured in terms of tuning the overall stiffness or shear modulus. Since floppy networks have a large number of zero energy modes, as we tune the global stiffness of the network, the heterogeneity of the local response of the network also begins to vary. We now closely examine the response of individual bonds to external loading and characterize how this heterogeneity might be amplified. In particular, when an underconstrained mechanical system is subject to deformation/external forcing, we ask if it might be possible to detect critical bonds, i.e. bonds subject to more stretching and more prone to failure using our sparse mode representation.

To create a heterogeneous network, we use a bi-disperse disk-packing algorithm used to generate jammed configurations [39,40] and randomly remove some links to allow the structure to have multiple DoF (figure 5*a*). When a uniform outward stretching displacement is applied to this underconstrained network, bonds that can rotate easily do so to accommodate this. For example, in the top right area of the network in figure 5*a*, the bonds rotate in response to deformation. To quantify this local flexibility, we use the SND method to determine the sparse representation of the rigidity matrix and find m sets of sparse representations of floppy modes. Defining s_j^h as the size of jth mode in the hth set (the size of a mode s_j defined previously is the number of non-zero entries in the vector representation) allows us to quantify the globality for a node as follows:

$$f_i = \frac{1}{m}\sum_{h=1}^{m} \min_{j\in \boldsymbol{B}_i^h} s_j^h, \tag{6.1}$$

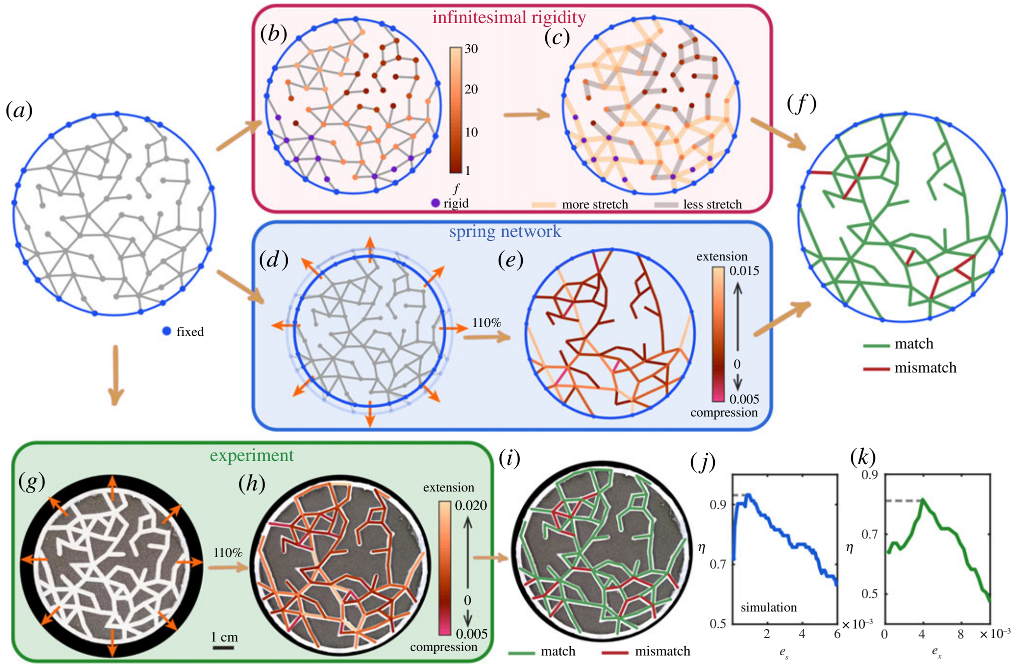

Figure 5. Load prediction in spring networks. (*a*) A spring network generated from the standard jamming algorithm. Boundary nodes (blue) are spatially fixed. (*b*) When the network is treated as a frame with rigid bonds, each node is coloured based on the globality *f*. The purple nodes are not involved in any infinitesimal modes. (*c*) Prediction of loads: links connecting nodes that are more 'global' or rigid nodes are more likely to bear loads (yellow), when the network is deformed. Other links (dark red) are less likely to bear loads. (*d*,*e*) When the network is treated as a spring network, we apply a uniform outward stretching of 10% to the boundary nodes (*d*) and let it equilibrate (*e*). Brighter links indicate more stretching. (*f*) The prediction from the infinitesimal approach and the numerical simulation agree well. (*g*) Experimental verification: Real network cast with elastic silicone rubber stretched uniformly outwards by 10%. (*h*) Brighter colours indicate more stretching. (*i*) Similarly, the green colour indicates a match between the prediction and the real experiment. (*j*) The matching accuracy between the prediction and the numerical simulation with varying threshold e_S. (*k*) The matching ratio between the prediction and the experiment with varying threshold e_X. (Online version in colour.)

where $\boldsymbol{B}_i^h$ is the set of modes that involve the motion of node i in the hth set. Using this definition, the nodes that are mostly involved in smaller modes (small globality, darker nodes in figure 5*b*) are likely locally flexible, while the nodes that are within larger modes (large globality, brighter nodes in figure 5*b*) are less likely to adjust their positions to accommodate the stretching. There are also nodes that are not involved in any modes (purple nodes in figure 5*b*). These nodes, together with the boundary nodes and global nodes, are likely to be nodes among which connecting links bear more loads (figure 5*c*, links with lighter shades are predicted to bear more load), as these links are less flexible to rotate and adjust. A threshold t in the globality allows us to carry out a binary classification for all the links into 'more stretched' or 'less stretched' ones (see figure 5*c* for the classification with $t = 12$ and electronic supplementary material, Fig. S11 for the effect of different t).

To test our prediction based on infinitesimal rigidity, we numerically simulate the deformation of the network as one made of springs (finite k) to determine which links bear loads under uniform stretching (figure 5*d*). After the network equilibrates (figure 5*e*), the difference between the initial and the final length of each spring is calculated (shown via a colour gradient that is equivalent to the force on a spring). For comparison with the prediction, we set a threshold e_S for the absolute value of the change of length (scaled by the diameter of the circle) and again carry out a binary classification of the links into 'more stretched' or 'less stretched' ones. Defining the number of links that are both predicted and simulated to stretch more as n_b, the number of links that are both predicted and simulated to stretch less as n_o and the number of total links as n_t, we calculate the matching ratio between simulation and prediction as $\eta = (n_b + n_o)/n_t$. By adjusting the threshold e_s, we see that the matching ratio can be as high as 93.2% (figure 5*j*, where *t* is chosen to be 12; see electronic supplementary material, Section S5.5 for more details). To test these results experimentally, we cast an elastic silicone rubber network having the same geometry (figure 5*g*, electronic supplementary material, Section S5.4). By stretching it uniformly outwards by 10% (figure 5*h*), we can measure how much each link is stretched. Defining the threshold e_x in the spring extension (scaled by the diameter of the circle) above which a link is classified as 'more stretched,' we compare the prediction and the experiment (figure 5*i*), and by varying the threshold e_x, we find that the maximum prediction accuracy is about 81.6% (figure 5*k*). The discrepancy between the simulation and the experiment might come from (1) the experimental networks not being made of perfect linear springs (because of material nonlinearities and elastic energies associated with nodal deformations) and (2) the thickness of the links in the experiment being non-negligible, resulting in the errors in length measurement. While it has recently been shown that a purely graph theory-based approach can also help in identifying rigid clusters and predicting the network failure [41,42], our approach provides a more general understanding of network flexibility and zero modes. Altogether our approach based on sparse mode representation provides an efficient way to predict local failure in a network, and it does not require consideration of the network energetics.

7. Discussion

Many physical and biological systems can be modelled as underconstrained networks embedded in real Euclidean space. A necessary step towards functional control of the network minimally requires a representation of the (topological) connectivity and (geometric) rigidity of the network, as this enables and constrains the efficient transmission of mechanical, electrical, chemical and hydrodynamic information through the network. By couching the quest for hierarchy and modularity in networks in terms of geometric rigidity and algebraic sparsity, we have framed the question as one of identifying, representing and understanding the flexibility (floppy modes) in underconstrained networks. By interpreting, adapting and generalizing a sparse null space basis algorithm into a multi-scale variant, we have shown how to create a sparse representation of the floppy modes that uncovers the hidden modularity in the network and describes the combination of hierarchy and spatial localization within the DoFs of the system. Our network approach is agnostic to the application area, and we demonstrate this by using it to determine and deploy minimal motion primitives in robotics and also use it to predict and control the heterogeneous mechanical response of finitely extensible mechanical networks at the level of both the entire network and individual edges.

It is likely that our methods will find use wherever there is a need for modular representations in underconstrained networks with a given topological and geometrical structure. We chose to work with a Euclidean metric and the associated infinitesimal rigidity matrix given the nature of our geometric/physical networks. More generally, it would be interesting to ask what the results would be if a linearized constraint is used for non-Euclidean geometries, or perhaps even stochastic variations thereof. From a practical perspective, how these modular representations might be engineered, learned or optimized in motor control [2], used for motion planning in robots [43], or the design and characterization of mechanical networks across scales in DNA assemblies, protein allostery [29], soft materials [1,27,28] and spatial structures [44,45] remain questions for the future.

8. Methods

Below we describe the methods used for our computational and experimental framework briefly, with further details available in the accompanying electronic supplementary material.

(i) Rigidity matrix

The number of rows of the rigidity matrix is equal to the total number of constraints. Besides the length constraint of links described in the main text, the spatial confinement is also reflected in the rigidity matrix. For any node j that is spatially fixed,

$$g = \boldsymbol{x}_p - \boldsymbol{x}_{p_0} = 0, \tag{8.1}$$

where $\boldsymbol{x}_{p_0}$ is the fixed position for node p. In the two-dimensional (three-dimensional) case, this corresponds to two (three) rows in the rigidity matrix, with only one non-zero entry per row.

All rows are normalized to have unit norm. Random shuffling of the row order is done each time before the SND calculation.

(ii) The SND algorithm

Given a sparse matrix $M \in \mathbb{R}^{m\times n}$, the SND algorithm [37] starts with the identity matrix $H_1 = I_{n\times n} \in \mathbb{R}^{n\times n}$. Then for $i = 1, 2, \ldots, m$, the algorithm iteratively updates H_{i+1} such that the rows of H_{i+1} form a basis with a minimal number of non-zeros for the null space of the first i rows of M. Ultimately, H_{m+1} becomes a sparse basis for the null space for M (see electronic supplementary material, Section S1 for more details).

(iii) Spring network simulation

For the shear modulus calculation, at the beginning of the simulation, the nodes on the top row are displaced to the right by 8% of the width from top to bottom ($(N_y - 1)l_0 \times \sqrt{3}/2 \times 0.08$) and are fixed spatially. The motions of the nodes are simulated 20 000 steps using an overdamped Verlet integration scheme. Random uncorrelated noise (uniformly sampled from -0.0001 to $+0.0001$, much smaller than l_0) is added to the displacement for each node to facilitate the equilibrium process.

(iv) Network rigidification

In the network rigidification process, the MS protocol is as follows: starting from the initial network configuration,

(a) Find the set of floppy modes using the SND algorithm.
(b) Find the mode with the largest s.
(c) Within this mode, find the node with the largest infinitesimal displacement.
(d) Connect this node to one of its neighbours. If there are multiple closest neighbouring nodes available, randomly choose one.
(e) (For sequential tuning) Repeat steps (a) to (d) until the network is fully connected.

See electronic supplementary material, Section S4 for more variations of the MS protocol.

(v) Designing the elastic network

An initial configuration is generated using an algorithm for random close packing of bi-disperse disks [39]. Disk centres are transformed into nodes of the network, among which links are mapped from overlapping disks. About 35% of links are removed thereafter, making the network underconstrained with 18 DoF. We use a circular boundary to avoid having too many free ends. The nodes on the boundary are fixed.

(vi) Load prediction

The procedure of the load prediction in figure 5 is as follows:

(a) Apply SND to find a set of floppy modes. Calculate the size of each mode s_j.
(b) For each node i, find out the set of modes $\boldsymbol{B}_i$ that involve the movement of node i.
(c) Repeat (a)–(b) m times. Calculate the globality as $f_i = \frac{1}{m}\sum_{h=1}^{m}\min_{j\in \boldsymbol{B}_i^k} s_j^h$. If node i is not involved in any mode, f_i is set to 0.
(d) To do a binary classification, set a threshold t for the globality: nodes with f above t are called global nodes. Define the set of eligible nodes as the nodes that are (1) boundary nodes, (2) fixed nodes, or (3) global nodes.
(e) Start from the boundary nodes and do a breadth-first search until reaching another boundary node. The search path where node i and node j are connected has to follow two requirements: (1) node i and node j are connected by a link. (2) Both node i and node j are eligible nodes.
(f) Mark all the links along the shortest path as the potential links to bear loads (more stretched).
(g) Repeat (e)–(f) until all the boundary nodes have been visited.
(h) All marked links are predicted to bear more stress, and the remaining are predicted to bear less stress.

(vii) Experiment

For the network rigidification experiment, physical networks are cast using three-dimensional printed moulds and silicone rubber (Dragon Skin$^{\mathrm{TM}}$ 30, Young's modulus 5.93×10^5 Pa). Each network is fully connected initially, and we cut connections sequentially if required by the experimental protocol. Shear forces were measured on an Instron 5566 with a 10N load cell by clamping the outermost edges of the network along its length with a three-dimensional printed clamp and applying a displacement (at a rate of $1\,\mathrm{mm\,s^{-1}}$) to one side until a displacement of 10% of the network length is reached. More information in electronic supplementary material, Section S4.

For the load prediction experiment, a circular network was cast using the same material as earlier. The physical network diameter is 100 mm, and the thickness of an edge is 2.5 mm. The outermost edge of the circular network is glued on a stretchable black spandex cloth, which is uniformly stretched to reach an 11% size increase. Network nodes were extracted and evaluated from image data (see electronic supplementary material, Section S5 for more information).

Data accessibility. The data are provided in electronic supplementary material [46].

Authors' contributions. L.M.: conceptualization, formal analysis, funding acquisition, investigation and project administration; S.C.: conceptualization, data curation, formal analysis, investigation and methodology; F.G.: data curation, formal analysis, investigation and methodology; G.P.T.C.: data curation, formal analysis, investigation and methodology.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. This work was partially supported by the Swiss National Science Foundation (F.G.) the National Science Foundation under grant nos. DMS-2002103 (G.P.T.C.) and DMR-2011754, EFRI-1830901, DMR-1922321, the Simons Foundation and the Seydoux Fund (L.M.).

References

1. Jacobs DJ, Rader AJ, Kuhn LA, Thorpe MF. 2001 Protein flexibility predictions using graph theory. *Proteins* **44**, 150–165. (doi:10.1002/prot.1081)
2. Bernstein N. 1966 *The co-ordination and regulation of movements*. Oxford: Pergamon Press.
3. Khatib O. 1987 A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J. Robot. Autom.* **3**, 43–53. (doi:10.1109/JRA.1987.1087068)

4. Camazine S, Deneubourg JL, Franks NR, Sneyd J, Bonabeau E, Theraula G. 2003 *Self-organization in biological systems*. Princeton, NJ: Princeton University Press.
5. Sumpter DJT. 2010 *Collective animal behavior*. Princeton, NJ: Princeton University Press.
6. Olfati-Saber R, Fax JA, Murray RM. 2007 Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **95**, 215–233. (doi:10.1109/JPROC.2006.887293)
7. Halabi N, Rivoire O, Leibler S, Ranganathan R. 2009 Protein sectors: evolutionary units of three-dimensional structure. *Cell* **138**, 774–786. (doi:10.1016/j.cell.2009.07.038)
8. He Y, Ye T, Su M, Zhang C, Ribbe AE, Jiang W, Mao C. 2008 Hierarchical self-assembly of DNA into symmetric supramolecular polyhedra. *Nature* **452**, 198–201. (doi:10.1038/nature06597)
9. Tirion MM. 1996 Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys. Rev. Lett.* **77**, 1905. (doi:10.1103/PhysRevLett.77.1905)
10. Maxwell JC. 1864 On the calculation of the equilibrium and stiffness of frames. *Lond. Edinb. Dubl. Phil. Mag.* **27**, 294–299. (doi:10.1080/14786446408643668)
11. Thorpe MF, Duxbury PM. 1999 *Rigidity theory and applications*. New York, NY: Springer Science & Business Media.
12. Jacobs DJ, Thorpe MF. 1995 Generic rigidity percolation: the pebble game. *Phys. Rev. Lett.* **75**, 4051. (doi:10.1103/PhysRevLett.75.4051)
13. Guest S. 2006 The stiffness of prestressed frameworks: a unifying approach. *Int. J. Solids Struct.* **43**, 842–854. (doi:10.1016/j.ijsolstr.2005.03.008)
14. Kim JZ, Lu Z, Strogatz SH, Bassett DS. 2019 Conformational control of mechanical networks. *Nat. Phys.* **15**, 714–720. (doi:10.1038/s41567-019-0475-y)
15. Kim JZ, Lu Z, Bassett DS. 2019 Design of large sequential conformational change in mechanical networks. (http://arxiv.org/abs/1906.08400)
16. Chen S, Mahadevan L. 2019 Rigidity percolation and geometric information in floppy origami. *Proc. Natl Acad. Sci.* **116**, 8119–8124. (doi:10.1073/pnas.1820505116)
17. Chen S, Choi GPT, Mahadevan L. 2020 Deterministic and stochastic control of kirigami topology. *Proc. Natl Acad. Sci.* **117**, 4511–4517. (doi:10.1073/pnas.1909164117)
18. Krick L, Broucke ME, Francis BA. 2009 Stabilisation of infinitesimally rigid formations of multi-robot networks. *Int. J. Control* **82**, 423–439. (doi:10.1080/00207170802108441)
19. Broedersz CP, Mao X, Lubensky TC, MacKintosh FC. 2011 Criticality and isostaticity in fibre networks. *Nat. Phys.* **7**, 983. (doi:10.1038/nphys2127)
20. van Hecke M. 2009 Jamming of soft particles: geometry, mechanics, scaling and isostaticity. *J. Condens. Matter Phys.* **22**, 033101. (doi:10.1088/0953-8984/22/3/033101)
21. Liu AJ, Nagel SR. 2010 The jamming transition and the marginally jammed solid. *Annu. Rev. Condens. Matter Phys.* **1**, 347–369. (doi:10.1146/annurev-conmatphys-070909-104045)
22. Düring G, Lerner E, Wyart M. 2013 Phonon gap and localization lengths in floppy materials. *Soft Matter* **9**, 146–154. (doi:10.1039/c2sm25878a)
23. Rens R, Villarroel C, Düring G, Lerner E. 2018 Micromechanical theory of strain stiffening of biopolymer networks. *Phys. Rev. E* **98**, 062411. (doi:10.1103/PhysRevE.98.062411)
24. Merkel M, Baumgarten K, Tighe BP, Manning ML. 2019 A minimal-length approach unifies rigidity in underconstrained materials. *Proc. Natl Acad. Sci.* **116**, 6560–6568. (doi:10.1073/pnas.1815436116)
25. Liu Y, Dehmamy N, Barabási AL. 2021 Isotopy and energy of physical networks. *Nat. Phys.* **17**, 216–222. (doi:10.1038/s41567-020-1029-z)
26. Dehmamy N, Milanlouei S, Barabási AL. 2018 A structural transition in physical networks. *Nature* **563**, 676–680. (doi:10.1038/s41586-018-0726-6)
27. Rocks JW, Pashine N, Bischofberger I, Goodrich CP, Liu AJ, Nagel SR. 2017 Designing allostery-inspired response in mechanical networks. *Proc. Natl Acad. Sci.* **114**, 2520–2525. (doi:10.1073/pnas.1612139114)
28. Yan L, Ravasio R, Brito C, Wyart M. 2017 Architecture and coevolution of allosteric materials. *Proc. Natl Acad. Sci.* **114**, 2526–2531. (doi:10.1073/pnas.1615536114)
29. Wodak SJ *et al.* 2019 Allostery in its many disguises: from theory to applications. *Structure* **27**, 566–578. (doi:10.1016/j.str.2019.01.003)
30. Woodhouse FG, Ronellenfitsch H, Dunkel J. 2018 Autonomous actuation of zero modes in mechanical networks far from equilibrium. *Phys. Rev. Lett.* **121**, 178001. (doi:10.1103/PhysRevLett.121.178001)
31. Goodrich CP, Liu AJ, Nagel SR. 2015 The principle of independent bond-level response: tuning by pruning to exploit disorder for global behavior. *Phys. Rev. Lett.* **114**, 225501. (doi:10.1103/PhysRevLett.114.225501)

32. Hexner D, Liu AJ, Nagel SR. 2018 Role of local response in manipulating the elastic properties of disordered solids by bond removal. *Soft Matter* **14**, 312–318. (doi:10.1039/C7SM01727H)
33. Pellegrino S. 1993 Structural computations with the singular value decomposition of the equilibrium matrix. *Int. J. Solids Struct.* **30**, 3025–3035. (doi:10.1016/0020-7683(93)90210-X)
34. Pellegrino S, Calladine CR. 1986 Matrix analysis of statically and kinematically indeterminate frameworks. *Int. J. Solids Struct.* **22**, 409–428. (doi:10.1016/0020-7683(86)90014-4)
35. Coleman TF, Pothen A. 1986 The null space problem I. Complexity. *SIAM J. Algebra. Discr.* **7**, 527–537. (doi:10.1137/0607059)
36. Abaffy J, Spedicato E. 1989 *ABS projection algorithms: mathematical techniques for linear and nonlinear equations*. Upper Saddle River, NJ: Prentice-Hall.
37. Khorramizadeh M, Mahdavi-Amiri N. 2013 An efficient algorithm for sparse null space basis problem using ABS methods. *Numer. Algorithms* **62**, 469–485. (doi:10.1007/s11075-012-9599-1)
38. Todorov E, Ghahramani Z. 2003 Unsupervised learning of sensory-motor primitives. In *Proc. 25th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc. (IEEE Cat. No. 03CH37439), Cancun, Mexico, 17–21 September 2003*, vol. 2, pp. 1750–1753. New York, NY: IEEE.
39. Skoge M, Donev A, Stillinger FH, Torquato S. 2006 Packing hyperspheres in high-dimensional Euclidean spaces. *Phys. Rev. E* **74**, 041127. (doi:10.1103/PhysRevE.74.041127)
40. Ellenbroek WG, Zeravcic Z, van Saarloos W, van Hecke M. 2009 Non-affine response: jammed packings vs. spring networks. *Europhys. Lett.* **87**, 34004. (doi:10.1209/0295-5075/87/34004)
41. Berthier E, Porter MA, Daniels KE. 2019 Forecasting failure locations in 2-dimensional disordered lattices. *Proc. Natl Acad. Sci.* **116**, 16 742–16 749. (doi:10.1073/pnas.1900272116)
42. Berthier E, Kollmer JE, Henkes SE, Liu K, Schwarz JM, Daniels KE. 2019b Rigidity percolation control of the brittle-ductile transition in disordered networks. *Phys. Rev. Mater.* **3**, 075602. (doi:10.1103/PhysRevMaterials.3.075602)
43. Kirillova S, Cortés J, Stefaniu A, Siméon T. 2008 An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins. *Proteins* **70**, 131–143. (doi:10.1002/prot.21570)
44. Sussman DM, Goodrich CP, Liu AJ. 2016 Spatial structure of states of self stress in jammed systems. *Soft Matter* **12**, 3982–3990. (doi:10.1039/C6SM00094K)
45. Lerner E. 2017 Comment on ‘Spatial structure of states of self stress in jammed systems’ by D. M. Sussman, C. P. Goodrich, and A. J. Liu, *Soft Matter*, 2016, 12, 3982. *Soft Matter* **13**, 1530–1531. (doi:10.1039/C6SM01111J)
46. Chen S, Giardina F, Choi GPT, Mahadevan L. 2022 Modular representation and control of floppy networks. Figshare. (doi:10.6084/m9.figshare.c.6134121)