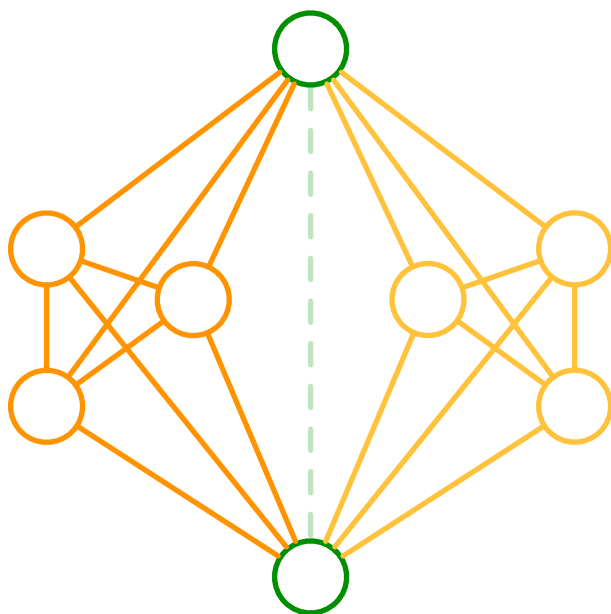# The Rigidity of Frameworks: Theory and Applications

Jessica Sidman and Audrey St. John

Communicated by Benjamin Braun

ABSTRACT. When robots fly to collectively move an object, the algebraic and combinatorial approaches of rigidity theory can provide conditions that determine whether the swarm will maintain formation. We introduce the fundamentals for the analysis of the rigidity of frameworks along with other motivating applications drawn from computer-aided design and computational biology.

Jessica Sidman is professor of mathematics and statistics at Mount Holyoke College. Her e-mail address is jsidman@mtholyoke.edu.

Audrey St. John is associate professor of computer science at Mount Holyoke College and is partially supported by NSF IIS-1253146. Her e-mail address is astjohn@mtholyoke.edu.

Picture a swarm of robots flying to move a piece of rubble from a disaster site, an engineer designing the roof of a stadium, or a protein docking an inhibiting drug. What these scenarios, pulled from robotics, computer-aided design (CAD), and structural biology, have in common is that each is governed by a finite system of geometric constraints that can be analyzed using ideas from rigidity theory.

The classical structure studied in rigidity theory is the *bar-and-joint* framework, as in Figure 1. It is composed of universal (rotating) joints connected by bars that constrain the distances between pairs of joints. Formally, let $G = (V, E)$ be an undirected graph with n vertices (representing joints) and m edges (representing bars). We define a *bar-and-joint framework* in $\mathbb{R}^d$ to be a pair $(G, \mathbf{p})$, where $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in (\mathbb{R}^d)^n$ is a *realization* of the joints in $\mathbb{R}^d$.

The fundamental question in rigidity theory is:

Question 1. Given a framework $(G, \mathbf{p})$, is it rigid?
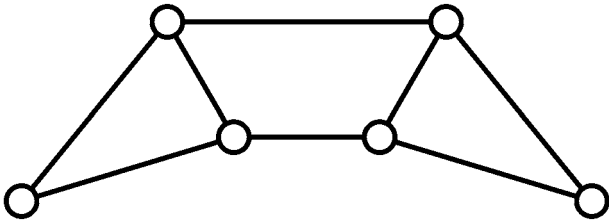
Figure 1. A bar-and-joint framework is composed of universal joints constrained by bars with fixed lengths.

Intuitively this question is asking: if one were to "push" on the framework, would it deform? To formalize this, we define a *length* function $\ell : E \to \mathbb{R}$ specifying the (squared) lengths of the bars from $(G, \mathbf{p})$. Then we are interested in understanding the set of all realizations $\mathbf{q}$ satisfying (1):

(1) $$||\mathbf{q}_i - \mathbf{q}_j||^2 = \ell(ij), \forall ij \in E.$$

These squared distance equations are constraints that form a system of quadratic polynomial equations that must be satisfied by the joint coordinates.

> # *Given a framework $(G, \mathbf{p})$, is it rigid?*

It is clear that, given $(G, \mathbf{p})$, there are infinitely many realizations of G satisfying the same length function ℓ, as we can use a Euclidean motion of $\mathbb{R}^d$ to obtain another via translation, reflection, or rotation; such realizations are *congruent*. If there is a unique framework $(G, \mathbf{p})$ in $\mathbb{R}^d$, up to congruence, then we say that $(G, \mathbf{p})$ is *globally rigid*. We will not delve further into the topic of global rigidity, but note that there are many interesting and challenging questions posed around unique realizations. To capture the intuitive notion of rigidity we are interested in the framework's *local* behavior. If $(G, \mathbf{p})$ is the unique realization of (G, ℓ) up to congruence in some neighborhood of $\mathbf{p}$, then we say that $(G, \mathbf{p})$ is *(locally) rigid*; otherwise, it is *flexible*.

We also consider additional structural properties of a framework, as illustrated in Figure 2. If the removal of any edge in a rigid framework causes it to flex, the framework is *minimally rigid*. If, on the other hand, the removal of any edge in a rigid framework does not cause a flex, the framework is *redundantly rigid*. In a flexible framework, there may be rigid *components*: vertex-maximal induced subgraphs that are themselves rigid. These properties are typically studied via the *rigidity matroid*, whose independent sets are the sets of edges of G that impose independent geometric constraints. In general, a matroid is a combinatorial object abstracting the notion of linear independence, appearing in many other guises within mathematics. The rigidity matroid may be defined via the rigidity matrix introduced below or algebraically via the equations defining the Cayley–Menger variety.

Since (squared) distance constraints satisfied by the joint coordinates are given by quadratic polynomials, it is possible to try to analyze the set of all realizations of a given pair (G, ℓ) symbolically using computational algebra. However, a Gröbner basis for an ideal generated by quadratic polynomials in k variables can require generators of degree $O(2^{2^k})$, so computations with the squared distance constraints may quickly become intractable. Moreover, care must be taken in applying results from algebraic geometry in this setting as we are interested in realizations over the real numbers, and many results in algebraic geometry require an algebraically closed ground field.

Therefore, we turn to the study of "infinitesimal rigidity theory" in which we analyze the Jacobian, or *rigidity matrix*, of the quadratic system specified by Equation 1 in order to understand a framework's first-order behavior. As we will see shortly, the dimension of the null space of a framework's rigidity matrix is $\binom{d+1}{2}$ if and only if the framework is infinitesimally rigid. Since infinitesimal rigidity implies rigidity [1], answering the following question gives a sufficient condition for rigidity in the algebraic setting.

Question 2. Is a given framework $(G, \mathbf{p})$ infinitesimally rigid?

The rigidity matrix of a framework with n joints and m bars has dn columns and m rows, and its null space consists of the *infinitesimal motions* permitted by the framework. For example, the rigidity matrix of the 2-dimensional framework depicted in Figure 2(c), without the dashed edge, has joint coordinates $((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4))$, and its rigidity matrix is the following:

$$\begin{pmatrix} x_1-x_2 & y_1-y_2 & x_2-x_1 & y_2-y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_2-x_3 & y_2-y_3 & x_3-x_2 & y_3-y_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_3-x_4 & y_3-y_4 & x_4-x_3 & y_4-y_3 \\ x_1-x_4 & y_1-y_4 & 0 & 0 & 0 & 0 & x_4-x_1 & y_4-y_1 \\ 0 & 0 & x_2-x_4 & y_2-y_4 & 0 & 0 & x_4-x_2 & y_4-y_2 \end{pmatrix}.$$

Every infinitesimal motion consists of a velocity vector assigned to each vertex such that each bar is orthogonal to the relative velocities of its joints. Hence the infinitesimal motion instantaneously preserves the bar lengths. If the only infinitesimal motions permitted are the $\binom{d+1}{2}$ *trivial motions*, then the framework is *infinitesimally rigid*. In the plane, there are three trivial degrees of freedom corresponding to translation of the whole framework in the *x*- and *y*-directions and rotation about the origin. Indeed, the dimension of the null space of the example rigidity matrix is generically 3, and the framework is infinitesimally rigid. Note that a rigid framework may not be infinitesimally rigid, as demonstrated by realizing the (rigid) triangle in the plane with its vertices collinear; the null space of its rigidity matrix has dimension 4.

A pivotal theorem of Laman from 1970 [4] states that almost all realizations of a graph will have the same infinitesimal behavior, providing a characterization of "combinatorial rigidity" in the plane.

Theorem (Laman 1970). *A graph $G = (V, E)$ with n vertices is generically minimally rigid if and only if*
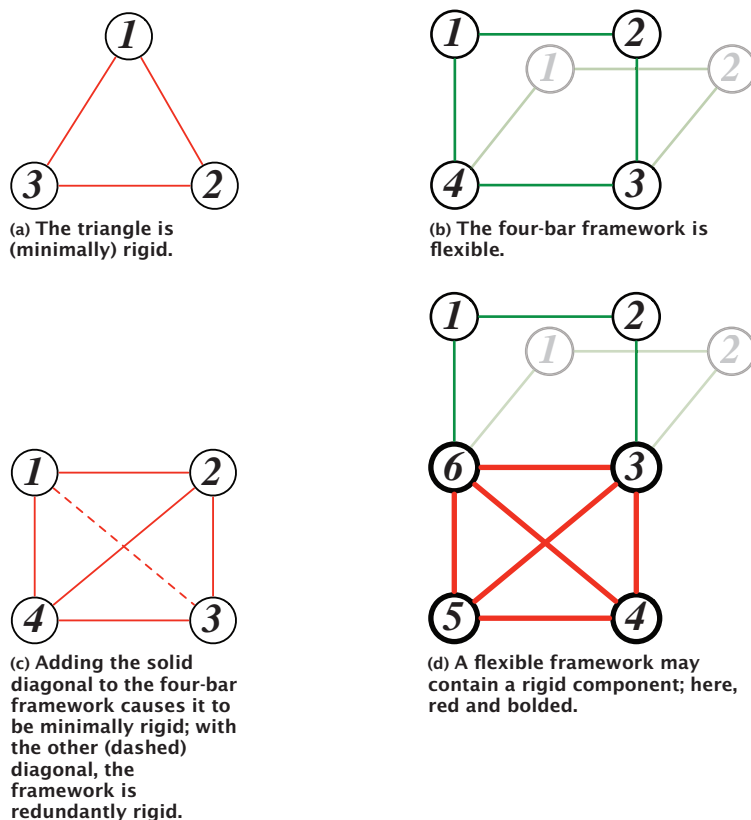
- $m = 2n - 3$, *and*

(a) The triangle is (minimally) rigid.

(b) The four-bar framework is flexible.

(c) Adding the solid diagonal to the four-bar framework causes it to be minimally rigid; with the other (dashed) diagonal, the framework is redundantly rigid.

(d) A flexible framework may contain a rigid component; here, red and bolded.

Figure 2. Bar-and-joint frameworks in the plane.

- $m' \leq 2n' - 3$ for all sets of $n' \geq 2$ vertices, where m' is the number of induced edges.

Notice that the framework in Figure 2(d) satisfies the first condition, but fails the second on the subset of vertices $\{3, 4, 5, 6\}$. In an arbitrary dimension d, the first condition generalizes to $m = dn - \binom{d+1}{2}$; assuming this condition holds for a graph, the notion of *genericity*[1] can be captured by a polynomial called the "pure condition" expressing the determinant of the matrix obtained by appending $\binom{d+1}{2}$ rows to the rigidity matrix chosen so that the rows eliminate the trivial motions. In the plane, Laman's theorem holds precisely when this polynomial is not identically zero; in
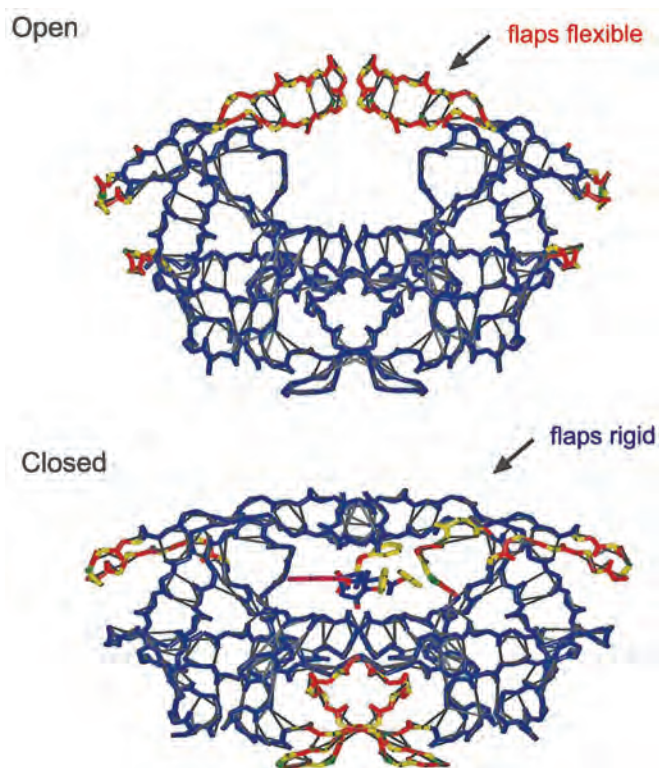
> *Detection of rigid components can vastly reduce the complexity of analyzing large structures, such as proteins.*

this case, almost all realizations of G are infinitesimally rigid and hence rigid.

The counting condition characterizing rigidity in Laman's Theorem is called $(2, 3)$-*sparsity*. More generally, (k, ℓ)-sparsity conditions require that the induced subgraph on every subset of n vertices has at most $kn' - \ell$ edges. These conditions arise as necessary (and sometimes also sufficient) conditions for rigidity in other constraint systems. For example, $(2, 2)$- and $(2, 3)$-sparsity conditions characterize the rigidity of *direction-length* frameworks, composed of points that may be pairwise constrained with a specified direction (the slope of the line joining the points) or length (a bar). To analyze a graph for Laman's counting condition, a quadratic-time algorithm, the *pebble game* of Jacobs and Hendrickson [3], determines if a framework is generically rigid or flexible. If it is flexible, the algorithm detects the rigid components in the structure.

Detection of rigid components can vastly reduce the complexity of analyzing large structures, such as proteins. These macromolecules are composed of thousands or even hundreds of thousands of atoms whose strongest chemical interactions, such as covalent bonds, can be modeled by distance constraints. Since a protein's function is determined by its 3-dimensional structure, understanding the motion or flexibility near its *native* (folded) state

---

[1] *A framework is referred to as generic in rigidity theory if it is "general" in the sense of algebraic geometry, meaning that it lies in the complement of the Zariski closed set defined by the vanishing of the pure condition.*

Figure 3. Rigidifying the flaps necessary for the HIV protease to cleave the virus required proteins, as in its closed state, would inhibit the protein's function.

Figure 4. The double banana is a flexible 3D bar-and-joint framework, although it satisfies generalized $(3,6)$-sparsity counts. It is composed of 18 bars (solid) on 8 joints; the "bananas" can rotate about the center axis (green, dashed).

may hold the key for developing a drug that can inhibit or activate its function.

For example, the HIV-protease protein depicted in Figure 3 cleaves additional proteins required by the virus with a hinge-like motion similar to a pair of scissors. Blocking this motion by docking a drug that rigidifies the two flaps would inhibit the protease's function and prevent the virus from replicating.

It is natural to model proteins and other macromolecules as 3-dimensional bar-and-joint structures. However, efficient rigidity analysis is blocked by the following well-known open problem.

Question 3. Is there a combinatorial characterization for bar-and-joint rigidity in dimension 3 and higher?

The natural generalization of Laman's condition, the $(3,6)$-sparsity counting property, fails to capture rigidity, as demonstrated by the example of Figure 4 known as the "double banana."

This framework is composed of two "bananas," each of which is made up of two tetrahedra glued along one triangular face. The expected $(3,6)$-sparsity counts hold for the associated graph, but the structure is flexible, as the bananas may rotate relative to one another about the axis defined by their two attachment points. While not sufficient, these counts are necessary, but even they are challenging to verify from an algorithmic perspective. They fall just outside a matroidal range; the efficient
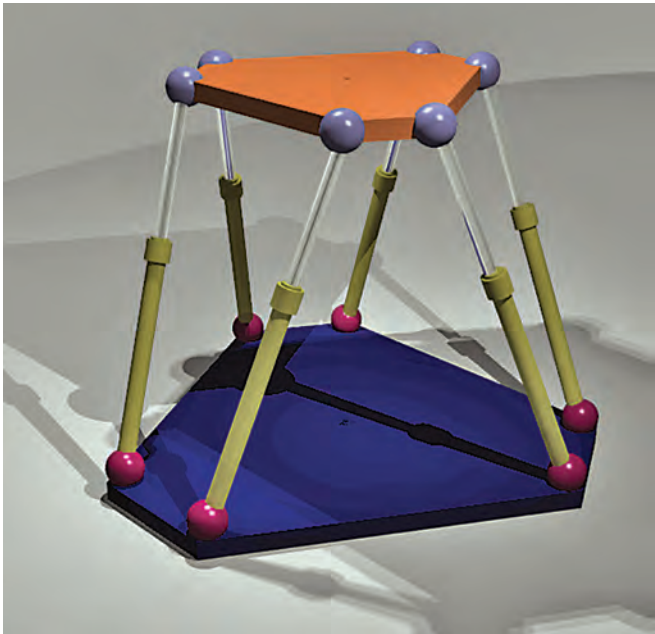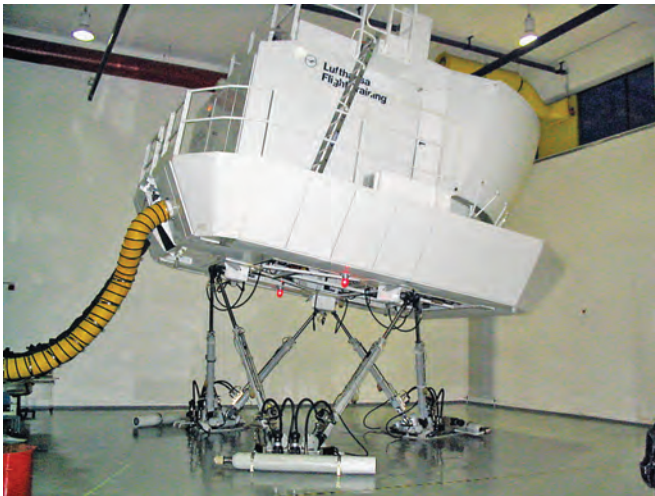
pebble game algorithm's approach is greedy and cannot be easily extended to these non-matroidal counts.

Because characterizing 3-dimensional bar-and-joint rigidity remains an open challenge, researchers turn to a different model of rigidity called the *body-and-bar* model. A body-and-bar framework is composed of rigid bodies, constrained by bars placed between pairs of bodies and attached at universal joints. A combinatorial characterization for d-dimensional body-and-bar rigidity proved by Tay in 1984 [5] uses a $(\binom{d+1}{2}, \binom{d+1}{2})$-sparsity condition, and a generalized pebble game has led to efficient analysis tools. Note that, in this model, a pair of bodies may have more than one bar between them. Indeed, a rigid framework in the plane consisting of two rigid bodies must have at least three bars. In 3D, engineers use the famous Stewart-Gough platform depicted in Figure 5 to build hexapods that support a range of positions by changing the lengths of the legs. Once the leg lengths are fixed, the structure becomes (generically) rigid and may be modeled as a body-and-bar framework with each platform a body and each leg a bar; as expected, the 6 bar constraints rigidify the framework.

The motivation to develop other rigidity models does not stem solely from the challenge of 3D bar-and-joint rigidity. For the mechanical engineer using CAD software, geometric constraints provide the language that describes a design. A stadium roof is composed of many parts, each of which is naturally modeled as a rigid body; the engineer
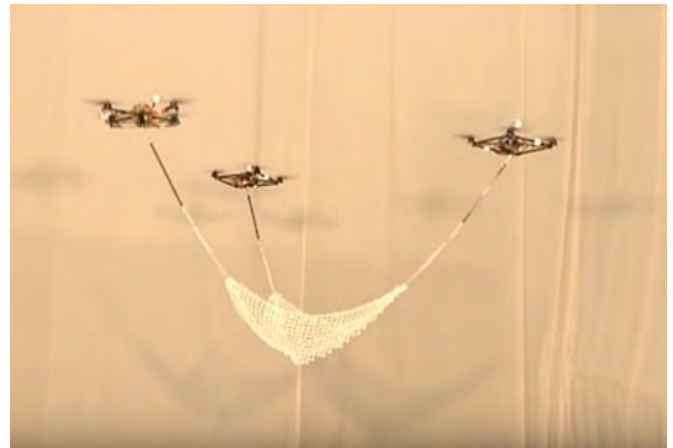
(a) Setting the lengths of the 6 prismatic legs of the platform allows different rigid realizations to be explored.

(b) Engineers use the platform to build hexapods, as seen in this flight simulator.

Figure 5. The Stewart-Gough platform can be modeled as a body-and-bar framework with each platform a rigid body and each leg a bar.

Figure 6. Robots work collaboratively to transport and manipulate objects, as seen in Raffaello D'Andrea's "Flying Machine Arena." Rigidity theory can be used to develop efficient algorithms for controlling a multi-robot formation autonomously.
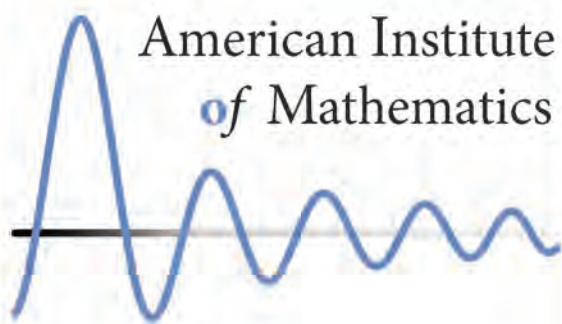
for structural integrity and for ensuring that its geometry is completely specified.

So, what about when robots fly? If a swarm of robots is tasked with picking up a piece of rubble or something more delicate like a net (as in Figure 6), local communication and sensing can be used to enable autonomous maintenance of the required formation [2]. For example, robots may follow others using specified geometric constraints, such as fixed distance or orientation, resulting in global behavior. Efficient algorithms for analyzing the rigidity of this *multi-robot formation* can minimize communication and sensing costs, allowing the swarm to effectively perform its rescue operation.

References

[1] LEONARD ASIMOW and BEN ROTH, The rigidity of graphs II, *J. Anal. Appl.* **68** March 1979, 171–190. MR 511410

[2] JULIEN M. HENDRICKX, BRIAN D. O. ANDERSON, JEAN-CHARLES DELVENNE, and VINCENT D. BLONDEL, Directed graphs for the analysis of rigidity and persistence in autonomous agent systems, *Int. J. Robust. Nonlin.* **17** (2007), no. 10–11, 960–981. MR 2333294

[3] DONALD JACOBS and BRUCE HENDRICKSON, An algorithm for two-dimensional rigidity percolation: The pebble game, *J. Comput. Phys.* **137** (CP975809) (1997), no. 2, 346–365. MR 1481894

[4] GERARD LAMAN, On graphs and rigidity of plane skeletal structures, *JENMA* **4** (1970), 331–340. MR 0269535

[5] TIONG-SENG TAY, Rigidity of multigraphs. I. Linking rigid bodies in *n*-space, *JCTB* **36** (1984), no. 1, 95–112. MR 742389

EDITOR'S NOTE. See the related "WHAT IS...a Tensegrity?" by Robert Connelly in the January 2013 issue of *Notices*.

may specify the design by requiring, for example, that "this wooden beam must be at a particular angle to the outer wall." Such a structure may be modeled as a *body-and-cad framework*, composed of rigid bodies with geometric constraints beyond the point-point distance constraints allowed in the body-and-bar model. For example, we can impose constraints that fix the angle between the sides of two bodies or fix the distance between a point on one body and a line on another. The engineer expects that the CAD software will verify that the final design is rigid, both

Image Credits

Figures 1, 2, and 4 courtesy of Jessica Sidman and Audrey St. John.

Figure 3 reproduced with permission from: Mykyta Chubynsky, Brandon Hespenheide, Donald J. Jacobs, Leslie A. Kuhn, Ming Lei, Scott Menor, A. J. Rader, M. F. Thorpe, Walter Whiteley, and Maria I. Zavodszky, Constraint Theory Applied to Proteins, *Nano-Scale Materials: From Science to Technology*, edited by S. N. Sahu, R. K. Choudhury, and P. Jena, Novascience (2006), pages 77–88.

Figure 5(a) reproduced from `en.wikipedia.org/wiki /Parallel_manipulator`.

Figure 5(b) reproduced from `en.wikipedia.org/wiki /Stewart_platform` and is courtesy of Ethan Arnold.

Figure 6 courtesy of Raffaello D'Andrea. Image captured and reproduced from `youtu.be/hyGJBV1xnJI` .

Photo of Audrey St. John courtesy of John Dolan.

Photo of Jessica Sidman courtesy of Leah Masci.

ABOUT THE AUTHORS

Both authors are passionate about supporting women in STEM. They knew each other for years, but it took an undergraduate thesis to jump-start their collaboration. Along with Meera Sitharam, they are co-editors of the forthcoming *Handbook of Geometric Constraint Systems: Principles*.

Jessica Sidman

Audrey St. John