PopupCAD: a Tool for Automated Design, Fabrication, and Analysis of Laminate Devices.

Daniel M. Aukes and Robert J. Wood

Wyss Institute for Biologically Engineered Engineering
School of Engineering and Applied Sciences
Harvard University, Cambridge, MA USA

ABSTRACT

Recent advances in laminate manufacturing techniques have driven the development of new classes of millimeter-scale sensorized medical devices, robots capable of terrestrial locomotion and sustained flight, and new techniques for sensing and actuation. Recently, the analysis of laminate micro-devices has focused more manufacturability concerns and not on mechanics. Considering the nature of such devices, we draw from existing research in composites, origami kinematics, and finite element methods in order to identify issues related to sequential assembly and self-folding prior to fabrication as well as the stiffness of composite folded systems during operation. These techniques can be useful for understanding how such devices will bend and flex under normal operating conditions, and when added to new design tools like popupCAD, will give designers another means to develop better devices throughout the design process.

Keywords: Robotics, Design, Manufacturing, Dynamics, Finite Element Method, Stiffness

1. INTRODUCTION

New technologies often emerge before there is a complete understanding of how they will be used most successfully. This can be said for discrete products like smartphones and technologies like the internet, which have affected the way we socialize, communicate, and consume, as well as for new manufacturing technologies such as 3D printing. Through the proliferation of low-cost and highly-capable 3D printing technologies, companies can prototype and iterate complex engineered devices quickly and cheaply, and individuals can design and print custom parts at home. This has been facilitated by the relative ease with which it is possible to design and manufacture these devices. Typical workflows involve importing design files into proprietary machine-specific programs which perform nearly all of the tasks required to parse CAD-originated geometric shape files into machine-specific build commands which are interpreted and executed by the connected 3D printer. In many cases this process can be executed in seconds. Additionally, several websites offer upload services which automate the calculation of manufacturability and cost, as well as provide catalogs of community-designed parts which enable people without design experience to download and print existing designs.

Thus when a new manufacturing technology arises, the question of how quickly it will be adopted can be distilled into several questions. First: How complete is the fabrication technology itself? The new technology must consist of enough building blocks that one can create more than variations on a single device. Second, how accessible is the technology? Research labs and specialized manufacturing shops have access to materials and technologies that will never make it into the home, due to considerations of cost, safety, size, and energy. Finally, how accessible is the design process? If years of education or training are required in order to design a successful device or utilize a particular tool, then the design process itself is inaccessible to the casual user.

Therefore, with the relatively nascent field of laminate robots, we have a very capable process which has the potential to be accessible to many. Until now, there has not been a tool which allows one to design these devices as easily as a 3D printed part. This tool must assist the user in several key areas: it must allow us

Further author information: (Send correspondence to DMA)
DMA: danaukes@seas.harvard.edu
RJW: rjwood@seas.harvard.edu

to fully describe the device, anticipate and accomodate the fabrication techniques, be relatively expandable under different use-cases, and give the designer a sense of how a laminate devices will perform in the real world. This paper presents several new capabilites within one such laminate device design tool called "popupCAD". In this paper we discuss the development of a dynamic simulation framework and stiffness analysis tool which are compatible with the design and manufacturing environment, and the potential for time savings throughout the design workflow.

2. BACKGROUND

We have recently introduced a new design tool called "popupCAD" for the design and development of articulated laminate robots.[1,2] The development of this tool has been driven by a set of manufacturing capabilities which can be used to develop complex articulated devices,[3] self-folding devices,[4–6] and other laminate micro-robotic devices whose assembly methods are inspired by "pop-up" books.[7,8] The geometries associated with these devices are most easily described with sequences of flat shapes distributed across material-dependent layers, and the processes associated with fabricating them impose specific process constraints which heavily affect the design process, as well as the understanding of manufacturability for these devices. Through the use of mathematical frameworks tailored for these geometries and processes,[1] algorithms can be designed to automate the generation and validation of these devices. However, until this point, "popupCAD" has not provided insight into the performance of the device itself, rather the ability to manufacture it.

It is common to find design tools targeted specifically to a particular manufacturing paradigm or design workflow. This can be for a variety of reasons. The geometry involved may require a particular description or syntax which more generic design tools such as modern 2D or 3D CAD environments cannot efficiently provide. In the case of printed circuit board design, for example, the specific requirements of electrical circuit design, layered board layout, and circuit routing requires a tool which can describe the design across electrical and geometric domains. This tool must interface with simulation tools such as SPICE,[9] and must give users access to up-to-date libraries of commercial electronics components. Were the design to be performed in a generic CAD environment, the process would be slow and the results would have many errors. In other cases, the fabrication process may require specialized algorithms for the generation / validation of particular geometry. In the case of 3D printing, much of the early research to calculate and optimize support geometry, orient parts[10] and slice 3D geometries[11] is now widely available in a variety of open-source and proprietary programs [*] [†] [‡]. With origami and folded structures, similar needs have led to the development of a variety of design tools like as treemaker[§] and origamizer[12] for the generation of designs, as well as tools tailored to the simulation of flat folded structures.[13–16] The same can be said for sheet metal bending[17] and shape deposition manufacturing.[18] The synthesis of design and rapid fabrication techniques, and their application to milled cut structures and houses has also been studied by Sass,[19] while specialized programs exist for creating 3D structures from lamina[¶]and bent sheet metal[‖]

3. IDENTIFYING ARTICULATED SYSTEMS IN POPUPCAD

3.1 Branching Topology

In the past, we have discussed the generation of laminate designs using popupCAD, automating the generation of manufacturing geometry, analyzing manufacturability, and outputting cut files. More recently, we have developed a new tool within popupCAD which generates joint and link information for dynamic and stiffness analysis. In order to use the joint operation, the user must specify the laminate which represents the system they would like to model. The user must also specify which parts of the device are located in the world reference frame by selecting a second laminate that indicates fixed regions of the device. In addition, the user must specify the network of hinges in the device by selecting one or more joint sketches, and then identifying both the joint

[*] http://replicat.org/
[†] http://slic3r.org/
[‡] http://www.stratasys.com/
[§] http://www.langorigami.com/science/computational/treemaker/treemaker.php
[¶] http://laminadesign.com/
[‖] http://www.industrialorigami.com/

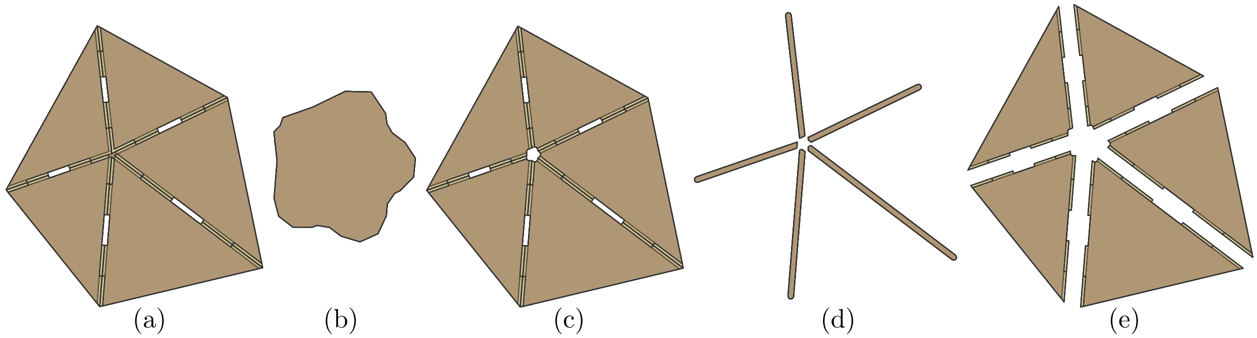


Figure 1. Steps of the Device Splitting Process. A hinged laminate device is shown in (a). Any intersecting hinge lines create material which must be removed, as seen in (b), magnified 10x relative to (a). With the hole removed (c), the device can then be split along non-intersecting hinge lines (d), fully separating the device into 5 rigid bodies (e).

layer and neighboring structural layers upon which that joint operates. With each sketch the user may supply stiffness, damping and preload information about the network of joints specified in the sketch, as well as the nominal width of the given hinges in the sketch. This may be done for multiple sets of hinges as needed, in order to fully define the different joint properties across multiple possible layers.

Once the user has specified body and joint information, popupCAD performs a series of steps to identify individual rigid bodies and the connections between them, as shown in Figure 1. First, popupCAD removes material (Figure 1b) wherever multiple joints intersect with each other (Figure 1c), using the specified joint width to determine the necessary size of the hole. Such holes are necessary to prevent rigid material from connecting links in unintended ways. Second, popupCAD removes material along each joint line (Figure 1d), in order to split the single connected laminate into a series of separate rigid bodies (Figure 1e). Any of these bodies which, when intersected with the specified fixed region produces a non-null laminate is considered fixed in the world frame. Third, popupCAD identifies the bodies connected by each joint line. Once this is accomplished, the user may export the set of rigid bodies, connections between them, the list of fixed bodies, and the list of joint properties to a file.

4. DYNAMIC ANALYSIS

In order to facilitate a better understanding of the way in which these devices will perform, it becomes necessary to capture some of the properties of the materials used in laminate devices. While in some cases the selection of engineering materials ensures that the resulting devices behave as ideal rigid bodies connected by frictionless revolute joints, it becomes evident that many devices realized by this process exhibit some form of non-ideal behavior, such as when links – and whole systems – bend under loading, when joints twist in directions other than their primary axes, or when the bending torques associated with joints begin to affect force/torque transmission through the system. In addition, actuator driving frequencies in these laminate devices can often surpass the device's resonant frequency, tying the system's motion not just to its rigid idealized motion, but to its stiffness, inertia, and damping properties.

To better understand the role dynamics plays in these composite devices, we have developed a symbolic rigid body dynamics system in Python[**] using the sympy[††] package. The Sympy package for Python allows one to create and operate upon symbolic expressions similar to Mathematica, Matlab, Maple, and Maxima, with the advantage that it is available for Python applications to be easily written using it. This project has been developed in parallel to similar efforts such as pydy[20] and MotionGenesis[‡‡], where the emphasis is placed on generating analytical equations of motion. Unlike typical dynamics engines, where efficient numerical calculations permit fast, real-time calculation of rigid body motion, these tools focus on generating efficient analytical expressions for explicitly defined system state variables. While pydy would also be a reasonable candidate for generating such

[**]www.python.com
[††]www.sympy.org
[‡‡]www.motiongenesis.com

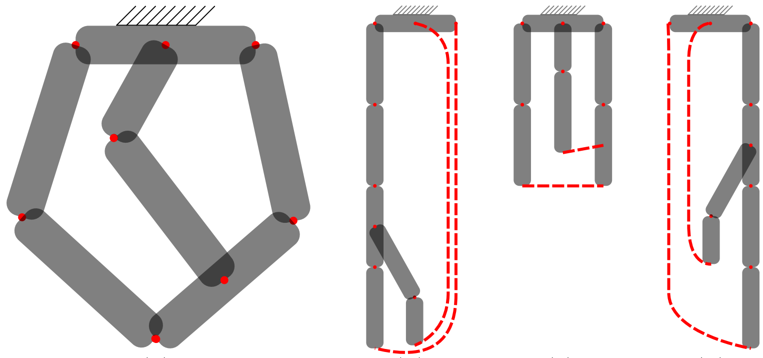


Figure 2. Equivalent Topological Variants. A system of connected rigid bodies is seen in (a). (b),(c),and (d) represent equivalent branching tree topologies, where the red lines indicate additonal constraint equations required to fully define the system seen in (a).

equations, we have developed our own dynamics system because it leaves open the possibility of switching to more efficient symbolic math systems and investigate different equation derivation methods in the future. Built upon this basic ability, we have created several Python classes which allow us to efficiently describe and operate upon elements common to dynamic systems.

4.1 Reference Frames

Reference frames allow one to efficiently compute vector calculations between groups of vectors which share common rotational relationships. This is common for rigid body systems where various points on a single rigid body can be described as fixed within a reference frame, and may be represented by simple relationships between topologically neighboring frames. There are many ways to describe the rotational relationship between reference frames; many more than are required to describe the rotational relationships between laminate devices. Euler parameters, Rodrigues parameters, and screw representations are among many rotational descriptions, but rotation matrices and their associated direction cosines provide a straightforward representation commonly found in robotics applications. When a frame B is rotated from another frame A by a certain amount q about an axis $\hat{f} = f_x \hat{a}_x + f_y \hat{a}_y + f_z \hat{a}_z = f_x \hat{b}_x + f_y \hat{b}_y + f_z \hat{b}_z$ fixed in both frames, that frame's rotation matrix ${}^{A}\mathbf{R}^{B}$ and angular velocity vector ${}^{A}\vec{\omega}^{B}$ is expressed by

$$
\begin{aligned}
{}^{A}\mathbf{R}^{B} &= \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \mathbf{f} * \mathbf{f}^{T} \right) \cos(q) + \begin{bmatrix} 0 & -f_z & f_y \\ f_z & 0 & -f_x \\ -f_y & f_x & 1 \end{bmatrix} \sin(q) + \mathbf{f} * \mathbf{f}^{T} && (1) \\
{}^{A}\vec{\omega}^{B} &= \dot{q}\hat{f}, \text{ where} && (2) \\
\mathbf{f} &= \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^{T} \text{ and} && (3) \\
\dot{q} &= \frac{\mathrm{d}(q)}{\mathrm{d}t}. && (4)
\end{aligned}
$$

4.2 Branching Topology

Branching tree topologies are one of the most common ways to describe and understand complex networks of rigid bodies, and are sufficient to describe robotic systems – including parallel systems – when paired with a method to describe additional constraints. For generic parallel rigid body systems there are often a variety of equivalent topologies, as seen in Figure 2, as the selection of how to split a parallel system into a branched topology is somewhat arbitrary. Parallel topologies are particularly common in laminate devices because such systems permit parallel kinematic loops through the network of hinges formed on one or more laminate layers.

To address this issue we have created a branching tree topology class, which computes the relative rotation matrix and angular velocity vector between any two frames within the tree when paired with reference frame class mentioned in Section 4.1. In addition, the method for generating the topology attempts to create the simplest expressions possible by splitting parallel loops to minimize the length of each resulting branch from the world frame. For example, if a new frame C is to be connected to two frames, A, and B in an existing topology, and A has a path length of 1 to the world frame while B has a path length of 3, C will get added relative to A and a constraint equation will be added relating its motion to B.

To compute the relative rotation matrix ${}^{A_1}\mathbf{R}^{A_n}$ between frames A_1 and A_n, where the path connecting the two is $\{A_1, \ldots, A_n\}$, the topology retrieves the relative rotation matrices and velocity vectors ${}^{A_i}\mathbf{R}^{A_{i+1}}, i = \{1, n-1\}$ and ${}^{A_i}\vec{\omega}^{A_{i+1}}, i = \{1, n-1\}$, respectively, between each set of frames and recursively computes

$$
{}^{A_1}\mathbf{R}^{A_n} = {}^{A_1}\mathbf{R}^{A_{(n-1)}}\, {}^{A_{(n-1)}}\mathbf{R}^{A_n} \tag{5}
$$

$$
{}^{A_1}\vec{\omega}^{A_n} = {}^{A_1}\vec{\omega}^{A_{(n-1)}}\, {}^{A_{(n-1)}}\vec{\omega}^{A_n}. \tag{6}
$$

4.3 Vectors

A vector is represented as a Python dictionary of $m \times 1$ Sympy matrix objects each associated with a reference frame, where m=3 for three-dimensional systems. Addition of one vector with another involves adding keyed vector components together. Multiplication of a scalar with a vector is achieved by multiplying the scalar with each keyed vector component. Negating a vector is thus possible by multiplying a vector by a scalar value of -1. This enables vector subtraction, which is simply represented as the addition of one vector to a negated second vector.

Expressing a vector in another frame is made possible with (1). For a vector $\vec{v} = \vec{v}_A + \vec{v}_B + \ldots$ where $\vec{v}_A, \vec{v}_B, \ldots$ contain only components in frames $A, B, \ldots$ such that

$$
\vec{v}_A = v_{ax}\hat{a}_x + v_{ay}\hat{a}_y + v_{az}\hat{a}_z, \tag{7}
$$

$$
\vec{v}_B = v_{bx}\hat{b}_x + v_{by}\hat{b}_y + v_{bz}\hat{b}_z, \tag{8}
$$

and so on, $\vec{v}$ may then be expressed in frame C as

$$
\vec{v} = \begin{bmatrix} \hat{c}_x & \hat{c}_y & \hat{c}_z \end{bmatrix} \left({}^{C}\mathbf{R}^{A} \begin{bmatrix} v_{ax} \\ v_{ay} \\ v_{az} \end{bmatrix} + {}^{C}\mathbf{R}^{B} \begin{bmatrix} v_{bx} \\ v_{by} \\ v_{bz} \end{bmatrix} + \ldots \right) \tag{9}
$$

Expressing a generic vector in components of a specific reference frame facilitates dot and cross products to be formed between arbitrary vectors. For two vectors $\vec{v}_1$ and $\vec{v}_2$, where $\vec{v}_1 = v_{1x}\hat{c}_x + v_{1y}\hat{c}_y + v_{1z}\hat{c}_z$ and $\vec{v}_2 = v_{2x}\hat{c}_x + v_{2y}\hat{c}_y + v_{2z}\hat{c}_z$ when expressed in C using (9),

$$
\vec{v}_1 \cdot \vec{v}_2 = v_{1x}v_{2x} + v_{1y}v_{2y} + v_{1z}v_{2z} \tag{10}
$$

$$
\vec{v}_1 \times \vec{v}_2 = (v_{1y}v_{2z} - v_{1z}v_{2y})\,\hat{c}_x + (v_{1z}v_{2x} - v_{1x}v_{2z})\,\hat{c}_y + (v_{1x}v_{2y} - v_{1y}v_{2x})\,\hat{c}_z \tag{11}
$$

4.4 Scalar and Vector Derivatives

Sympy facilitates the creation of symbolic expressions which can be differentiated with respect to symbols. Symbols which depend on time – such as state variables or input functions – are explicitly defined as differentiable to a certain order when created. In doing so, the system creates and stores a symbol for each derivative of the given variable. For example, when a user specifies a state variable "qi", the system creates three separate variables, q_i,$\dot{q}_i$, and $\ddot{q}_i$, representing the variable and its first and second derivatives. The system can then find the derivative of a scalar expression with respect to each explicitly defined differentiable variable using the chain rule. For an expression e and a set of differentiable variables $q_1, \ldots, q_i, \ldots, q_n$,

$$
\dot{e} = \frac{\mathrm{d}(e)}{\mathrm{d}t} = \frac{\mathrm{d}(e)}{\mathrm{d}q_1}\dot{q}_1 + \ldots \frac{\mathrm{d}(e)}{\mathrm{d}q_i}\dot{q}_i + \ldots + \frac{\mathrm{d}(e)}{\mathrm{d}q_n}\dot{q}_n. \tag{12}
$$

For a vector expression $\vec{e}$ with components in only one reference frame B such that $\vec{e} = e_x\hat{b}_x + e_y\hat{b}_y + e_z\hat{b}_z$, the derivative of that vector with respect to B, and any frame A can be expressed as

$$\frac{{}^B\mathrm{d}(\vec{e})}{\mathrm{d}t} = \dot{e}_x\hat{b}_x + \dot{e}_y\hat{b}_y + \dot{e}_z\hat{b}_z \tag{13}$$

$$\frac{{}^A\mathrm{d}(\vec{e})}{\mathrm{d}t} = \frac{{}^B\mathrm{d}(\vec{e})}{\mathrm{d}t} + {}^A\vec{\omega}^B \times \vec{e}. \tag{14}$$

Thus for any frames connected through a series of defined rotations and angular velocities between axis, (14) with (5) or (13) with (9) may be selected, depending on the efficiency of each representation, to calculate the derivative of vectors with components in multiple frames.

4.5 Calculation of Forces and Torques

Joints, idealized as revolute joints between bodies, define the relative rotations between bodies, and permit torques to be applied at the joints according to available stiffness and damping models for bending laminate materials. Thus for a vector $\vec{v}$ which defines the fixed axis of rotation between two connected rigid bodies, a state variable q_r which measures the angle of rotation between a proximal body A and a distal body B, and its initial rest angle q_{0_r}, the joint torques $\vec{\tau}_k$ and $\vec{\tau}_b$ related to the stiffness and damping properties of the material can be characterized as

$$\vec{\tau}_k = -k(q_r) * (q_r - q_{0_r})\hat{e} \quad \text{and} \tag{15}$$

$$\vec{\tau}_b = -b(q_r)\dot{q}_r\hat{e}, \quad \text{where} \tag{16}$$

$$\hat{e} = \frac{\vec{v}}{|\vec{v}|}. \tag{17}$$

These forces which characterize a particular type of interaction between neighboring bodies must be considered in context of the velocities at which they are applied. This permits the calculation of generalized active forces for a system. The contribution of these torques to the generalized active force can be calculated for a torque $\vec{\tau}$ as[21]

$$f_r = {}^N\vec{\omega}_r^A \cdot (-\vec{\tau}) + {}^N\vec{\omega}_r^B \cdot \vec{\tau}, \tag{18}$$

The appropriate hinge model may then be selected from existing theoretical beam-bending equations. For linear models, $k(q_r) = k$ and $b(q_r) = b$.

4.6 Computation of Dynamic Equations using Kane's Method

Using Kane's method we can derive the equations of motion once we have described the particles and bodies in a system, and the forces and moments acting upon them. For each r indicating a unique state variable, the corresponding dynamic equation can be formulated using

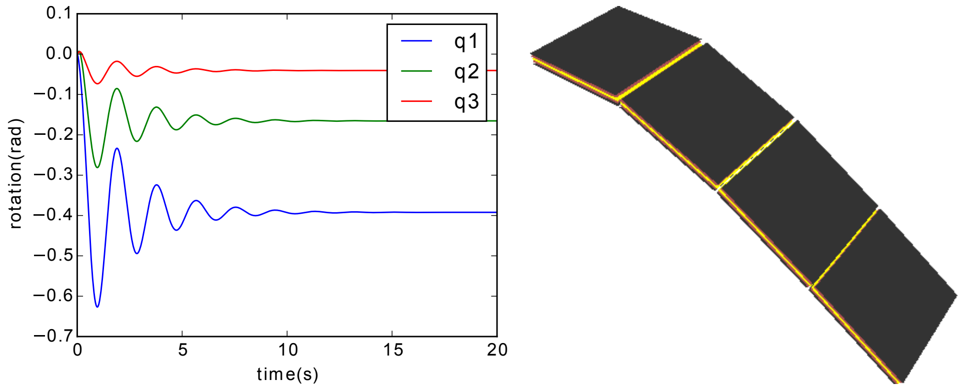
$$F_r + F_r^* = 0 \quad (r = 1, \ldots, n), \tag{19}$$

where F_r indicates the generalized active forces acting upon the system and F_r^* indicates the generalized inertial forces acting upon the system.

4.7 Implementation

Combined with the method described in Section 3 for describing the joint networks in a design, a dynamic system can be generated automatically, following a standard rubric.

1. Import the laminate bodies, joints, joint parameters, and world reference frame using the joint definition file exported from popupCAD.



(a) (b)

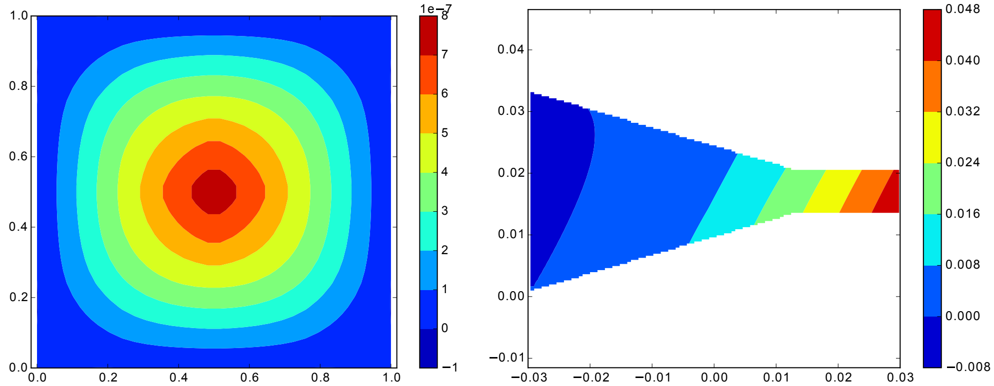
Figure 3. Triple Pendulum Simulation. (a) shows the three relative rotation angles of a triple pendulum as it comes to rest in gravity. (b) shows a screenshot from an animation of the device in its final rest position.

2. Create reference frames for each independent body, and, using the connections between bodies created by a joint, enumerate equivalent branching topologies for the system. Select the most efficient expression for unit vectors in the terminating reference frames when expressed in the world (Newtonian) reference frame.
3. Create and store a system of kinematic constraint equations for all remaining loop equations not used in the selected topology.
4. Starting from the world (Newtonian) reference frame, recursively search through the topology and create the rigid body definitions (mass and inertial distributions) and system of kinematic translations between proximal and distal joints required to define the kinematics. Mass properties may be derived from the laminate geometries combined with each layer's material properties, as specified in popupCAD.
5. Add stiffness and damping torques, as well as any joint preloads and limits, and their corresponding angular velocities to the system.
6. Add non-contact forces (such as gravity) to each body.
7. Generalize and sum the list of active and inertial forces acting upon the system to generate a system of dynamic equations using Kane's method.[21]
8. Apply the second derivatives of the additional, previously stored constraint equations to the system and solve for acceleration and velocity variables.
9. Integrate along the equations of motion, starting from the initial state provided by popupCAD.

This implementation has been performed on an exemplar triple pendulum for the purposes of validation. Figure 3a displays the state variables representing the relative rotation of each link over time as they fall in gravity and come to rest. Each joint has a stiffness of 100 Nm/rad and damping of 10 Nm-s/rad, while each link has unit mass and is 1m by 1m in size. The simulation tool also creates a visualization of the motion, which can be exported to video or viewed in real time by the user. Figure 3b shows a screenshot from that visualization of the pendulum at rest.

5. STIFFNESS ANALYSIS

Given the range of materials compatible with the laminate process and the planar geometries typical of the materials, the difference in stiffness between joints and links in a laminate system cannot always be treated as infinite. Though materials like metal, carbon fiber, and fiberglass can often be used to stiffen links, and flexible layer thicknesses can be adjusted to reduce bending moments, these systems often exhibit unintended flexibility. This can affect performance in a number of ways. First, non-rigid links can store energy in bending instead of transmitting unattenuated forces from between joints. Slight changes in kinematics due to these deformations can also affect the nonlinear kinematics of such linkage-based transmissions. This can have the effect of distorting an intended particular velocity or force profile. In addition, perfect one-dimensional pin joints may permit a certain amount of twisting in other directions, affecting both performance and lifetime.



(a) Square Plate (b) Actuator

Figure 4. Two FEA Case Studies using a BCIZ element. A simply-supported square plate fixed along its edges with a point load in the center is seen in (b), while (b) shows a shape typical of a piezoelectric actuator, with non-symmetric laminate construction, cantilever support on the left and end-loading on the right.

In order to bring an understanding of the role stiffness plays in such systems, we are developing a finite-element analysis tool for popupCAD. This tool will accommodate a variety of element types from first order triangular elements, to quadratic or cubic square elements. Like the dynamics system, this tool uses Python and Sympy to derive element stiffness equations analytically from basic shape functions.

5.1 Case Studies

Because of the geometries common to laminate devices, we have used classical composite laminate theory[22] in conjunction with triangular BCIZ elements[23] within our FEA program to study shapes output by popupCAD. Any laminate shape within popupCAD can be exported from the main menu, and imported into this new FEA tool. In the future, higher levels of integration will facilitate full analyses directly from the user interface, but currently, the user specifies the input shape along with any loading and constraint conditions within a simple script. Figure 4 shows two example analyses. Figure 4a shows a canonical problem: the displacement of a simply-supported, square, uniform plate with a point load in the center. This result matches fairly well with other common FEA programs such as LISA and ABAQUS. Figure 4b shows a shape common to devices: that of a piezoelectric actuator. In this study we have specified a laminate carbon-fiber construction of 45°, 22.5°, and 45°. Despite symmetric loading conditions along the central axis, the plate displacement graph shows twisting in the beam, which is to be expected with asymmetric laminate constructions. While this particular design is not typical for piezoelectric actuators, this example displays the capabilities of the program to deal with atypical laminate designs.

6. FUTURE WORK

Future work will require characterizing both of these tools performance in relation to other well-known and more-established tools. While our goal is not to completely replace other, potentially more-complete dynamics and FEA packages, we hope to provide designers with a first-order understanding of the expected performance of their designs. Given the common nature of hinges, manufacturing processes, layup methods, and planar geometries within laminate devices, we expect a more streamlined process for generating analyses than with external tools, as they would require more setup and specification just to reproduce the conditions already given by the popupCAD design. We hope this will allow a designer to study the impact of changes in kinematics and laminate construction across a wide range of devices and materials.

Many improvements to the dynamics tool are expected. First, a better method of creating loop constraints must be selected. While a variety of methods are being explored, this may require shifting to a different method for deriving the equations of motion, from Featherstone's method[24] to Lagrange Multiplier[25] methods. While Kane's method is useful for generating concise, symbolic equations of motion, these other methods may provide faster simulation speeds for the complex networks of rigid bodies typical of these devices, allowing the user to try more designs within the same amount of time.

While the BCIZ element has a well-established track record for reliability in many conditions, there are many different element types which could be explored within our FEA tool that may work better for specific cases. In addition, should an entire laminate robot be simulated with this tool, the difference in stiffness between link and joint regions may produce results which do not simulate large deflections realistically. This may require shifting away from classical laminate theory and simple plate bending assumptions to more realistic models, such as the generalized laminate plate theory[26] or hyperelastic material models.

7. CONCLUSIONS

We have presented two new tools actively being developed for use within the popupCAD development environment. First we discuss the beginnings of a dynamic analysis system for understanding the ideal rigid body motion generated by networks of hinges in laminate devics. Second, we discuss the analysis of laminate stiffness and deflection with a new FEA tool. As these tools mature we anticipate a high level of integration with the main design tool, so that non-experts can visualize the motion and deformation of their designs during development, and integrate that knowledge into more-successful designs.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation (grant numbers EFRI-1240383 and CCF-1138967) and the Wyss Institute for Biologically Inspired Engineering. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Aukes, D. M., Goldberg, B., Cutkosky, M. R., and Wood, R. J., "An analytic framework for developing inherently-manufacturable pop-up laminate devices," *Smart Materials and Structures* **23**, 094013 (Sept. 2014).

[2] Aukes, D. M. and Wood, R. J., "Algorithms for Rapid Development of Inherently-Manufacturable Laminate Devices," in [*ASME Conference on Smart Materials, Adaptive Structures and Intelligent Systems*], V001T01A005–V001T01A005, ASME, Newport, RI, USA (2014).

[3] Aukes, D. M., Ozcan, O., and Wood, R. J., "Monolithic Design and Fabrication of a 2-DOF Bio-Inspired Leg Transmission," in [*Third International Conference, Living Machines 2014, Milan, Italy, July 30 August 1, 2014.*], 1–10 (2014).

[4] An, B., Miyashita, S., Tolley, M. T., Aukes, D. M., Meeker, L., Demaine, E. D., Demaine, M. L., Wood, R. J., and Rus, D., "An end-to-end approach to making self-folded 3D surface shapes by uniform heating," in [*2014 IEEE International Conference on Robotics and Automation*], 1466–1473 (2014).

[5] Felton, S., Tolley, M., Demaine, E., Rus, D., and Wood, R., "A method for building self-folding machines," *Science* **345**, 644–646 (Aug. 2014).

[6] Felton, S. M., Becker, K. P., Aukes, D. M., and Wood, R. J., "Self-folding with shape memory composites at the millimeter scale," *Journal of Micromechanics and Microengineering (in Review)* (2015).

[7] Whitney, J. P., Sreetharan, P. S., Ma, K. Y., and Wood, R. J., "Pop-up book MEMS," *Journal of Micromechanics and Microengineering* **21**, 115021 (Nov. 2011).

[8] Sreetharan, P. S., Whitney, J. P., Strauss, M. D., and Wood, R. J., "Monolithic fabrication of millimeter-scale machines," *Journal of Micromechanics and Microengineering* **22**, 055027 (May 2012).

[9] Nagel, L. W. and Pederson, D. O., "Simulation Program with Integrated Circuit Emphasis (SPICE)," (1973).

[10] Alexander, P., Allen, S., and Dutta, D., "Part orientation and build cost determination in layered manufacturing," *Computer-Aided Design* **30**, 343–356 (Apr. 1998).

[11] Kulkarni, P. and Dutta, D., "An accurate slicing procedure for layered manufacturing," *Computer-Aided Design* **28**, 683–697 (Sept. 1996).

[12] Tachi, T., "Origamizing polyhedral surfaces.," *IEEE transactions on visualization and computer graphics* **16**(2), 298–311 (2010).

[13] Stellman, P., Buchner, T., Arora, W. J., and Barbastathis, G., "Dynamics of Nanostructured Origami," *Journal of Microelectromechanical Systems* **16**, 932–949 (Aug. 2007).

[14] Tachi, T., "Simulation of Rigid Origami," in [*Origami4:Proceedings of 4OSME*], (2009).

[15] Mullineux, G. and Matthews, J., "Constraint-based simulation of carton folding operations," *Computer-Aided Design* **42**, 257–265 (Mar. 2010).
[16] Gao, W., Ramani, K., Cipra, R. J., and Siegmund, T., "Kinetogami: A Reconfigurable, Combinatorial, and Printable Sheet Folding," *Journal of Mechanical Design* **135**, 111009 (Oct. 2013).
[17] Wang, C.-H. and Bourne, D. A., "Design and manufacturing of sheet-metal parts: Using features to aid process planning and resolve manufacturability problems," *Robotics and Computer-Integrated Manufacturing* **13**, 281–294 (Sept. 1997).
[18] Binnard, M. and Cutkosky, M. R., "Design by Composition for Layered Manufacturing," *Journal of Mechanical Design* **122**(1), 91 (2000).
[19] Sass, L., "Synthesis of design production with integrated digital fabrication," *Automation in Construction* **16**, 298–310 (May 2007).
[20] Gede, G., Peterson, D. L., Nanjangud, A. S., Moore, J. K., and Hubbard, M., "Constrained Multibody Dynamics With Python: From Symbolic Equation Generation to Publication," in [*Volume 7B: 9th International Conference on Multibody Systems, Nonlinear Dynamics, and Control*], V07BT10A051 (2013).
[21] Kane, T. R. and Levinson, D. A., [*Dynamics: Theory and Applications*], The Internet-First University Press (1985).
[22] Chou, T.-W., [*Microstructural design of fiber composites*], Cambridge University Press, New York (1991).
[23] Bazeley, G. P., Cheung, Y. K., Irons, B. M., and Zienkiewicz, O. C., "Triangular elements in plate bending - conforming and non-conforming solutions," in [*Proceedings of the First Conference on Matrix Methods in Structural Mechanics*], 547–576, Wright Patterson AF Base (1966).
[24] Featherstone, R. and Orin, D., "Robot dynamics: equations and algorithms," in [*IEEE International Conference on Robotics and Automation. Symposia Proceedings*], **1**, 826–834, IEEE (2000).
[25] Baraff, D., "Linear-time dynamics using Lagrange multipliers," in [*Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*], **30**(January), 137–146, ACM (1996).
[26] Reddy, J. N., Barbero, E., and Teply, J., "A Plate Bending Element Based on a Generalized Laminate Plate Theory," *International Journal for Numerical Methods in Engineering* **28**(July 1988), 2275–2292 (1989).