

# Collision-Free Deployments of Compact Stacks with Spring-Loaded Hinges

Jingyi Yang<sup>1</sup>, Manolis N. Chatzis<sup>2</sup> and Zhong You<sup>3</sup>

*University of Oxford, Oxford OX1 3PJ, U.K.*

**This paper proposes an optimization framework to determine spring stiffness at various hinge lines that allow compact stacks actuated by spring-loaded hinges to deploy bi-directionally into a flat surface without collision. Since the system-of-interest processes multiple degrees of freedom (DoFs), there are many potential deployment paths, some of which result in facet collisions. With various sets of initial conditions, our dynamic simulation enabled by a rigid multi-body dynamics analysis allows us to obtain the trajectory of each panel throughout the deployment process. The proximal distances between any pair of panels can thus be obtained. By maximizing the proximal distances, our optimization package searches for an optimal set of spring stiffness that deploys the compact stacks safely. Ensuring a collision-free deployment process is crucial in applications such as deploying solar panels of satellites as collisions may lead to a very large impulsive force and thus be detrimental to the system.**

## I. Introduction

In aerospace, many large and flat surfaces need to be deployed from a compact volume, such as solar panels of satellites [1] and solid-surface antennas [2]. They are folded to a minimized volume before launching and deployed to a maximized functional area in orbit. The most common mechanism to deploy solar panels of satellites expands unidirectionally, along its longitudinal direction [3]. It demonstrates superior performance, but its deployment ratio is limited. A bi-directional deployable structure shows a larger deployment ratio. It is usually designed with one DoF to simplify the morphing process [4]. When transforming from a compact configuration to an expanded functional surface, a single DoF ensures a safe and controllable deployment process. Such deployable structures are kinematically equivalent to mechanisms [5]. However, in a bi-directional deployable structure, if no voids are found within folded stacks, an uneven surface is observed when it is fully deployed [6, 7].

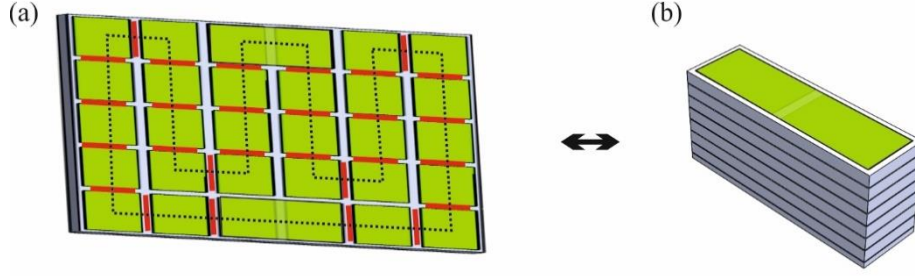
In [8], the authors have proposed a kirigami approach that folds an array of panels with uniform thickness into two compact stacks without voids. As illustrated by an example in Figure 1, each of the interior panels is adjacent to four neighboring panels but only has top and bottom facets. To package the array compactly, one panel should be stacked on each side of its facets. Therefore, revolute joints and slits are placed according to a closed path that is drawn to connect every panel in the array. The closed path is called a Hamiltonian circuit [9]. If an edge between two panels is crossed by the Hamiltonian circuit, a revolute joint is placed along the edge to connect neighboring panels; if not, a slit will be introduced. After placing all revolute joints and slits, each panel will be connected to two of its neighbors, turning the flat surface into a closed loop of rigid bodies, as shown in Figure 1(a). It can be noticed that there is no revolute joint placed between the panel in the first-row third-column and the one in the first-row fourth-column. This is because topologically they are assigned to be the top layer when the assembly is folded to stacks (Figure 1b), according to the method detailed in [8]. The two panels are merged into one rigid body. Similarly, the revolute joint between the pair of panels on the bottom layer of the stacks is removed. The two panels in the last-row third-column and last-row fourth-column are merged into one rigid body. This method allows for compact folding but in exchange, more DoFs are involved. The number of potential deployment paths increases consequently, among which, some paths end up with collisions.

---

<sup>1</sup> DPhil candidate, Department of Engineering Science

<sup>2</sup> Associate Professor of Engineering Science, Department of Engineering Science.

<sup>3</sup> Professor of Engineering Science, Department of Engineering Science.



**Figure 1. (a) A flat surface can be deployed from (b) two compact stacks. Red lines represent revolute joints, which are placed according to the dotted path, a.k.a., a Hamiltonian circuit that connects every panel. The methodology is detailed in [8].**

To safely deploy panels from a compact configuration to a flat surface, two methods can be adopted, each with pros and cons. Firstly, motors can be placed at hinges to control the deployment process. While the position and velocity of the panels can be precisely controlled at any time step, extra weight and volume are inevitably introduced to the system. In the second method, springs are placed at hinges as energy storage devices to passively deploy solar panels. In most research work, tape springs are employed to connect rigid panels [10] [11]. A tape spring can be seen as a cut-off section from a steel measuring tape. Tape springs are bent by 180 degrees in the stowed configuration. The elastic energy stored in tape springs drives the deployment process. When the tape spring unfolds to its nearly deployed configuration, it suddenly snaps due to its special geometry, which can be represented by a highly non-linear moment-rotation curve. Once the panels are released from the stowed configuration, the panels are ejected carrying significant energy which will eventually be converted to substantial kinetic energy. Excessive energy in the system can be dissipated rapidly, but a shock is also created as a side-effect. The shock may damage expensive onboard electronics. Due to the highly non-linear behavior of tape springs, only uni-directional deployments are demonstrated in the literature. Apart from tape springs, rotational springs are used to deploy solar panels on some CubeSats or nano-spacecraft [3] [12]. The deployment process is also driven by stored elastic energy, but unlike the tape spring, whose spring stiffness is changing non-linearly with respect to the angle of rotations, rotational springs have a constant stiffness within its range of motion. When the angle of rotation of a solar panel reaches its physical bound, a shock occurs and the solar panel bounces back. The non-smooth panel-to-panel interactions have been identified by past modeling efforts [12]. However, other types of collisions during the deployment process have not been studied, for instance, facet collisions amongst panels. Such collisions during the deployment may result in very high impulsive forces, which can be detrimental to the system. Although past research work has only demonstrated uni-directional deployment as well, a system using rotational spring has a greater potential of being deployed bi-directionally without collision. The spring stiffness of rotational springs can be controlled more easily as compared with that of a tape spring since it can be adjusted by changing the number of coils during manufacturing. In a bi-directional deployment such as the example demonstrated in Figure 1, rotational springs with various stiffness can be placed at various hinge locations.

In robotics, collision-free motion planning amongst multiple agents is well-studied [13]. Collision-free paths and trajectories can be explored in feasible configuration space. However, they often rely on the use of actuation forces. Here, we are instead interested in avoiding collisions for an autonomous system. An additional challenge is the non-analytic nature of the system due to the behavior of the hinges. Those challenges limit the use of some existing tools employed in other problems.

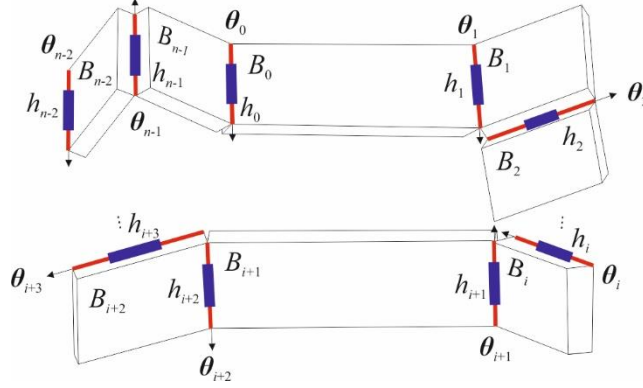
As such, this work will explore the collisions due to bi-directional deployment and suggest a means for avoiding them through proper calibration of the stiffness of rotational springs. The objective is to develop an optimization framework for deploying a closed-loop of panels with multiple DoFs from compact stacks into a flat surface while avoiding collisions before full deployment.

This paper is structured as follows. In section II, the closed-loop system is modeled taking into account the non-linear behavior at the hinges and the non-smooth panel-to-panel interactions. The forward dynamics responses of each rigid body are calculated recursively through Newton-Euler formulation via MATLAB Simscape [14]. With the position and orientation information of each panel at every timestep during the deployment process, the GJK algorithm [15] is introduced in section III to interrogate if two panels overlap, which would be indicative of a collision. If a pair of panels are found to collide, the approaching speed between them is calculated; if not, the minimum observing distance among panels will be recorded. In section IV, an optimization framework is set up to minimize the approaching speed if panels are in collision, or to maximize the observing distance if not. In the framework, the set of spring stiffness at various hinge locations are the parameters to be optimized. In section V, an example is presented to demonstrate the effectiveness of the optimization framework. Finally, the paper closes with a discussion and an outlook for future development in section VI.

## II. Dynamic Model

A closed-loop chain of rigid bodies  $\{B_0, B_1, \dots, B_{n-1}\}$  deploys to a flat surface and folds onto two stacks. Figure 2 captures a state in its deployment process. To facilitate compact folding, hinges  $\{h_0, h_1, \dots, h_{n-1}\}$  are placed along the edges of a panel. Along each rotating hinge  $h_i$ , a generalized coordinate  $\theta_i$  is introduced, allowing only the rotational DoF of the body  $B_i$  with respect to its previous one along hinge  $h_i$ . The generalized coordinate  $\theta_i$  is a vector whose direction is along its respective hinge  $h_i$  and its magnitude is within a permissible range of motion  $0 \leq \theta_i \leq \pi$ , with  $\theta_i = 0$  representing a fully deployed configuration and  $\theta_i = \pi$  representing a fully folded configuration. In our dynamics simulation model, a contact element shall be applied at each hinge to make sure that the rotations between any two connected rigid bodies are constraint within the permissible range of motion. Rotating out of the permissible range means a penetration between two connected bodies, which would not happen in the real deployment process. This contact element will be detailed in Section II.C.

Furthermore, a closed-loop of  $n$  rigid bodies processes  $n-6$  DoFs, which indicates that not all the defined rotations  $\theta_i$  are independent. The number of independent generalized coordinates describing the motion is less than the number of hinges. One approach to compute the dynamics of such a system is through solving the equations of motion under the set of extra constraints, for example, using Lagrange multipliers which would have resulted in differential-algebraic equations (DAEs) [16]. However, solving the formulation based on DAEs would pose challenges computationally.



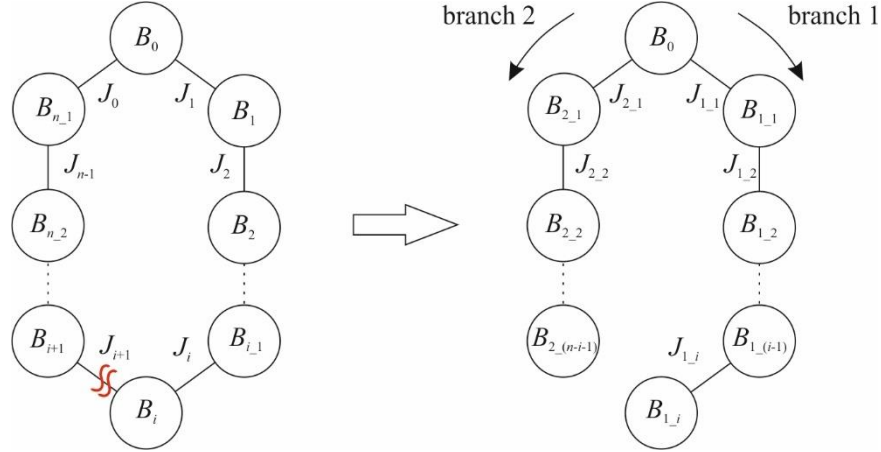
**Figure 2. A closed-loop chain consisting of  $n$  rigid bodies in the deployment process.**

Another approach to tackle the dynamics of closed-loop kinematic chains is by replacing some of the constraints with penalty springs [17]. To make a closed-loop chain into an equivalent open-loop system, the closed-loop can be cut open along any hinge line. The constraints along the hinge are replaced with translational springs of very high stiffness and dampers connecting the two open ends. Theoretically, when the spring is infinitely stiff, it can function equivalent to a revolute joint. All DoFs now become independent because the occurring system corresponds to two open-loop chains. Therefore, the equations of motion of the system can be written as a set of ordinary differential equations (ODEs) that can be integrated numerically in a more straightforward manner. Yet, the penalty springs formulation ensures that the obtained motion of this equivalent open-loop chain system is practically compatible with that of the original closed-loop chain system. Thus, the second approach is adopted to analyze the dynamics of a closed-loop system.

### A. Topology of the Multibody System

In a closed-loop kinematic chain consisting of  $n$  bodies,  $n$  hinges are connecting each rigid body with two of its neighbors. Along each hinge line  $h_i$ , a revolute joint is placed to connect two consecutive rigid bodies, which only allows a single DoF along the hinge. To facilitate a more computational efficient dynamic analysis, a cut is made along a hinge line to break the closed-loop into two open chains. The relationship showing the adjacency of bodies and joints is demonstrated by a topological graph in Figure 3 (left), whose vertices  $B_i$  represent bodies and edges  $J_i$  represent revolute joint that is placed along hinge  $h_i$ . The root  $B_0$  represents the solar panel that is fixed to the main body of a satellite. Since the set of solar panels, as a closed-loop kinematic chain, is not grounded but rather connected to a satellite that is floating in space, the root  $B_0$  can be assumed to be connected to the ground by a floating joint – a joint that allows six DoFs [18]. The dynamics simulation can be made more realistic if the action-reaction force between the solar panels and the main body of the satellite is known. However, the prerequisite of setting up a floating joint is knowing the dimensions and the weight of the satellite. To demonstrate the deployment of solar panels from stacks to a flat surface as a stand-alone system, we conveniently assume that the root  $B_0$  is grounded.

After the cut is applied, the multibody system can thus be treated as a tree with two branches that are rooted at  $B_0$  (Figure 3 right). The joint being cut open and released is referred to as a cut joint. The branch that is going clockwise is denoted as branch 1, and the other, branch 2. The last body in a branch is called a leaf. It has only one neighbor whereas all the other bodies have two. Apart from the root  $B_0$ , all the other bodies are re-numbered as  $B_{b\_i}$ , with the subscript  $b$  showing the branch that the body belongs to, and  $i$  increases consecutively from the root to the leaf in each branch. Subsequently, a joint connecting two consecutive bodies  $B_{b\_i}$  and  $B_{b\_i+1}$  is renumbered as  $J_{b\_i}$ ; the hinge that accommodates revolute joint  $J_{b\_i}$  is re-numbered as  $h_{b\_i}$ . Comparing to the number of joints in a closed-loop chain, it is reduced by one after the cut.



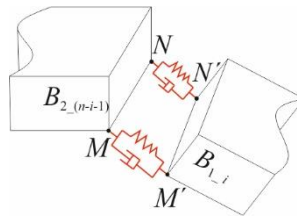
**Figure 3. Topological graphs of a closed kinematic chain turning into two open chains through applying a cut at joint  $J_{i+1}$ .**

To make the open-loop system equivalent to the original closed-loop, the leaves of the two branches are connected by very stiff translational springs. The force acting on the two rigid bodies depends not only on the spring stiffness but also on the relative distance between the two bodies at each time instance. The connecting details and the force acting on the two bodies will be explained in later contents.

### B. Force at the Cut Joint

Before the cut, it is a closed-loop and only one DoF is allowed along a revolute joint. After the cut is made, the closed-loop is transformed into two open chains. There are no motion constraints between the last bodies in the two open chains. Therefore, five DoFs shall be practically constrained via translational springs with very high stiffness.

Along the line of cut  $MN$ , two sets of virtual springs with zero natural length and dampers are connecting  $M$  and  $M'$ ,  $N$  and  $N'$ , as shown in Figure 4. Once there is a small distance between  $M$  and  $M'$ , there will be a large force generated by the translational spring that is placed between the two points and brings the distance back to zero. The translational spring at  $MM'$  constraints the relative distance of  $M'$  with respect to  $M$  along the length, width, and thickness directions of the panel. Likewise for the separation between  $NN'$ . Therefore, five DoFs are practically constraint along the line of cut  $MN$  – three translational DoFs and two rotational DoFs, left with the only rotational DoF about  $MN$ .



**Figure 4. At the cut joint, very stiff translational springs and dampers are connecting the two bodies to form an equivalent closed loop.**

At  $MM'$ , the force exerted by the spring and damper is

$$F_M = -k_c \Delta x_{MM'} - c_c \Delta \dot{x}_{MM'} \quad (1)$$

where  $k_c$  and  $c_c$  are spring stiffness and damping coefficient of the virtual element at the cut joint, respectively.  $\Delta x_{MM'}$  and  $\Delta \dot{x}_{MM'}$  are the relative distance and the time rate of change of the relative distance between  $M$  and  $M'$ , respectively.

The properties of the introduced springs and dampers at the cut joint are not to be optimized. They are pre-defined so that the separations between  $M$  and  $M'$ ,  $N$  and  $N'$  are negligible in comparison to the motion-of-interest during the deployment process, yet the system is not too stiff to cause integration errors. Besides, the damping is chosen so that the elongations of the springs are critically damped.

### C. Modeling of Hinge Elements

In the double-branch open-chain system as described in the previous contents, the dynamic equations of rigid bodies are completely decoupled and automatically include the kinematic constraints. To determine the forces and torques exerted by consecutive bodies in one branch, the branch number  $b$  in the subscript is omitted for simplicity. Elastic spring-loaded hinges of stiffness  $k_i$  are placed along hinge  $h_i$  resisting the angle of rotation  $\theta_i$ . In the stowed configuration, the initial condition for the rotation of all hinges is  $\theta_i = \pi$ , along which the corresponding rotational springs have non-zero initial torques. Within the permissible range of motion  $0 \leq \theta_i \leq \pi$  along hinge  $h_i$ , the resulted torque can be written as

$$\tau_i = -k_i \theta_i + T_i(\omega) \quad (2)$$

where  $T_i(\omega)$  is an angular velocity-dependent energy loss at  $h_i$ .

Opposing the angular velocity  $\omega_i$ , the energy loss at hinge  $h_i$  can be modeled as viscous damping, which is typically caused by packing grease in the bearing of a revolute joint. It is linearly proportional to the angular velocity  $\omega$  and is opposing to the rotational motion. The velocity dependent damping torque along hinge  $h_i$  can be written as Equation (3).

$$T_i(\omega) = -c_i \omega_i \quad (3)$$

where  $c_i$  and  $\omega_i$  are the damping coefficient and the angular velocity along hinge  $h_i$ .

When  $\theta_i$  goes beyond the permissible range of motion, contact occurs between the facets of the neighboring panels around the hinge. To avoid penetration of facets, the contact is modeled by a virtual rotational spring with spring stiffness of upper and lower bounds  $k_u, k_l \gg k_i$ , and damping coefficients of upper and lower bounds  $c_u, c_l$ . The overall torque  $\tau_i$  is summarized in Equation (4).

$$\tau_i = \begin{cases} -k_u(\theta_i - \pi) - c_u \omega_i - c_i \omega_i & \text{if } \theta_i > \pi \\ -k_i \theta_i - c_i \omega_i & \text{if } 0 \leq \theta_i \leq \pi \\ -k_l \theta_i - c_l \omega_i - c_i \omega_i & \text{if } \theta_i < 0 \end{cases} \quad (4)$$

Again, the values  $c_u, c_l$  and  $k_u, k_l$  are not to be optimized. They are chosen to represent contact springs and dampers to avoid penetration between two connected rigid bodies. The springs are stiff and the dampers dissipate energy to prevent significant bounce-back of the panels. When  $\theta_i$  goes beyond the permissible range, the contact elements act in parallel with the rotational spring which provides a nominal torque as elaborated in Equation (4). However, while the contact is engaged within the aforementioned penetration range, it dominates the responses.

This system is implemented with the aid of MATLAB Simscape Multibody Multiphysics Library [19]. The dynamics of the deployment process is computed via an explicit Runge-Kutta (2,3) pair in which a variable time-stepping integration is performed [20]. The variable increment at each time step is determined by evaluating if the configuration is close to the singularity of the system. A Runge-Kutta (2,3) pair is chosen instead of Runge-Kutta (4,5) pair for it is more computationally efficient at a given tolerance [21].

## III. Collision Detection

At this point, we should separate the ‘‘contacts’’ occurring when  $\theta_i$  attains cross the values of 0 or  $\pi$ , which are discussed in the previous section, from all the other collisions. The latter is the focus of this section and what the authors refer to as collisions. It should further be noticed that in this work, what happens to the system after such a collision is not of interest, as the priority is to avoid its occurrence. This allows the authors not to model the response of the system after collisions occur. As a result, the response of the system until the first occurrence of such a collision is accurately simulated with the tools presented in the previous section.

This then becomes a problem of detecting the first occurrence of collision in the system, by post-processing the simulation results. At each time instance, the configurations of any two bodies  $B_S$  and  $B_Q$  that are within the closed-loop can be extracted from the simulation results. As rigid bodies are convex polytopes, the configurations of  $B_S$  and  $B_Q$  can be represented by two sets  $S$  and  $Q$  containing their corresponding coordinates of vertices. We then calculate the relative translations between the two convex polytopes. The set of translations that could cause

interference of the two polytopes forms a new convex polytope, which is called Minkowski Difference of the two convex polytopes  $S$  and  $Q$  [15, 22]. Minkowski Difference is a binary operation that minuses all the points in  $Q$  from that of in  $S$ . The operation is detailed in Equation (5).

$$S \oplus (-Q) = \{s - q \mid s \in S, q \in Q\} \quad (5)$$

If the two convex polytopes interfere, the Minkowski Difference will contain the origin because one of the differences must be a zero vector if all points in  $Q$  are subtracted from all points in  $S$  and they share a common coordinate. Minkowski Difference transforms the problem of intersection detection between two polytopes into determining if a point is within a new polytope or not. The geometrical collision detection can be conducted through GJK algorithm [15], which calculates the Minkowski Difference of any given convex polytopes. It outputs proximal distance between the two bodies, i.e. the minimal distance between them, if they do not interfere with each other.

In a loop of  $n$  bodies as described in Figure 2, collision will be checked at every time instance between body  $B_i$  and the other panels, except the ones it shares a hinge with, as “contacts” with neighboring bodies are not to be avoided. Hence at each time step, a vector  $D$  is obtained and updated, with its elements being the proximal distances between any pair of bodies that are not connected by a hinge. The smallest value in  $D$  is denoted as  $d$ . In case of penetration between bodies, i.e. a collision, the proximal distance  $d$  is assigned to be zero. That is a feature of the GJK algorithm, which does not output a penetration depth. In such cases, additional information is provided by calculating the approaching speed between colliding panels at the time of the collision. This approaching speed is calculated using a finite difference scheme: When a collision is detected at time instance  $t_k$ , the distance between proximal points of the two panels at the last consecutive time steps  $t_{k-1}$  and  $t_{k-2}$  can be extracted, denoting as  $d_{k-1}$  and  $d_{k-2}$  respectively. The approaching speed of the pair of panels at  $t_k$  can thus be approximated by Equation (6).

$$v_k = \begin{cases} \frac{d_{k-2} - d_{k-1}}{t_{k-1} - t_{k-2}} & \text{if } d_k = 0 \\ 0 & \text{if } d_k > 0 \end{cases} \quad (6)$$

When a collision occurs at time instance  $t_k$ , the maximum approaching speed between any pair of disconnected panels is denoted as  $v_a$ . If no collision is detected throughout the simulation,  $v_a = 0$ . Therefore, the collision checker will either result in a non-zero value of  $v_a$  or a non-zero value of  $d$  depending on if a collision is detected during the simulation.

#### IV. System Optimization

As stated in Section I, the objective of this paper is to determine a set of spring parameters at various hinge lines that deploy panels from compact stacks to a flat surface without the type of collisions that are defined in Section III. It is preferable if panels are as further apart as possible during the whole deployment process.

This can be inspected via maximizing  $d$ , which is the minimum proximal distance between any pair of panels throughout the deployment process. However, when collisions occur,  $d$  always equals zero. For such cases, it is instead desirable to minimize the approach speed  $v_a$ . Despite the non-differentiable nature of both  $d$  and  $v_a$  around the occurrence of a collision, the objective function is formulated in Equation (7).

$$f(K) = v_a - d \quad (7)$$

The set of variables to be optimized,  $K = \{k_0, k_1, \dots, k_{n-1}\}$ , comprise the stiffness of rotational springs at all hinge lines. It should be noted that in the objective function, either  $d$  or  $v_a$  is zero for a given realization of  $K$ . If a collision is detected,  $f(K) = v_a$ ; if not,  $f(K) = -d$ . Therefore, the objective function is either minimizing the approaching speed or maximizing the proximal distance, even though it is written as a difference between the two.

An initial guess of the rotational spring stiffness  $K_0$  is given as input to initialize the optimization process, which is afterward advanced iteratively via an interior-reflective Newton approach [23]. To obtain a set of realistic spring stiffness that leads to a collision-free deployment process, the nonlinear objective function  $f(K)$  is minimized subject to upper and lower bounds on the variables [24]. If a local optimal realization is obtained by the end of the optimization process, from which the objective function is turned out to be positive, the result shall be rejected because a positive value of the objective function indicates a collision. The workflow of the optimization process is summarized in Figure 5.

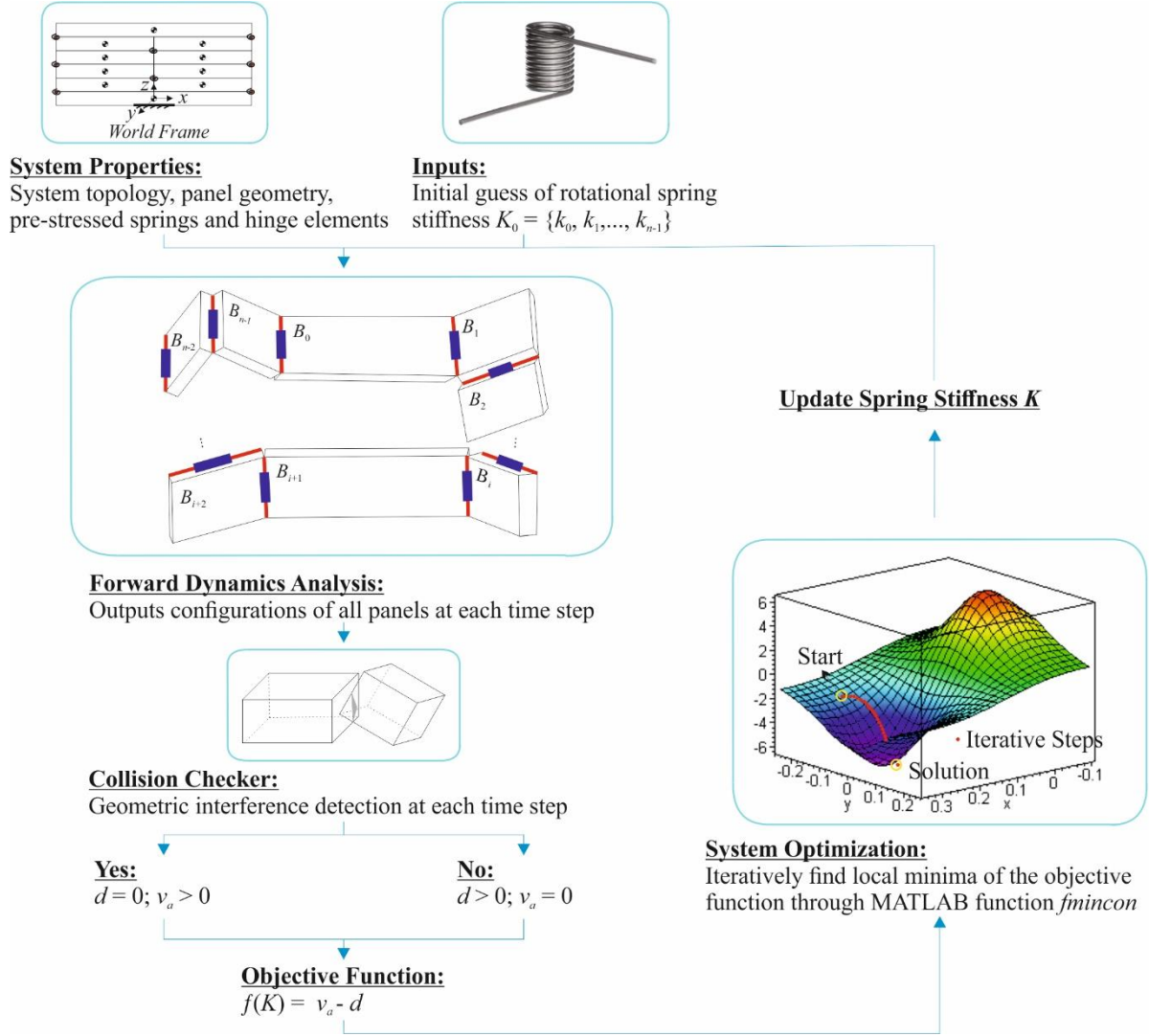


Figure 5. The system architecture of the optimization process

## V. An Example

In this example, the deployment process is modeled for 10 panels that are connected in a closed-loop kinematic chain as shown in Figure 6. Each panel is modeled as a rigid body  $B_i$  as shown in Figure 7, which is a rectangular prism with centroid  $C_i$ , length  $l$ , width  $w$ , and thickness  $a$ . The geometry and density of each rigid body are summarized in Table 1. The gravitational acceleration is set to 0 due to the micro-gravitational environment in space.

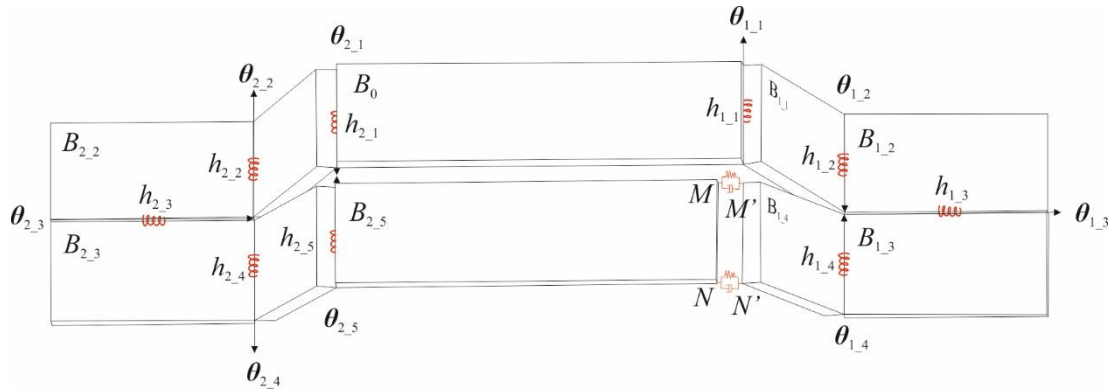
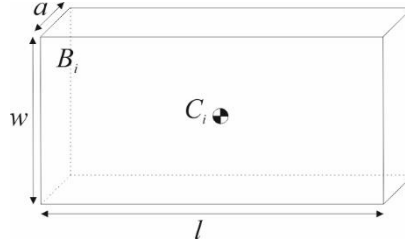


Figure 6. A closed-loop chain consisting of 10 rigid bodies in the deployment process.



**Figure 7. Body properties of a rigid panel (not to scale).**

**Table 1. Properties of rigid panels**

Length $l$	Width $w$	Thickness $a$	Density $\rho$	Gravitational Acceleration $g$
$0.3m$	$0.15m$	$0.01m$	$1800kg/m^3$	$0m/s^2$

As shown in Figure 6, with the size of rigid bodies  $B_0$  and  $B_{2-5}$  being twice of the other rigid bodies in the loop, they are the top and bottom layers when the closed-loop is folded into stacks. Rigid body  $B_0$  is connected to the main body of the satellite. The other rigid bodies are annotated according to section II. Along each hinge, a rotational spring with zero natural length is pre-tensioned to an initial angle of  $\pi$  at  $t = t_0$ . When the assembly is fully deployed, the rotations of all springs become zero. To obtain a symmetric deployment, the same values of spring stiffness and damping coefficients are adopted at symmetrical hinge locations when carrying out the optimization. The rotational springs are illustrated as red coils in Figure 6 and dampers are placed at the same locations but not shown in the figure. To obtain a set of realistic values, the stiffness of the springs is allowed to vary from  $0.1Nm/rad$  to  $0.2Nm/rad$ , and the exact values at various hinge locations are determined later through the optimization process. Energy dissipation is modeled by a viscous damper with damping coefficient  $0.05m/(rad/s)$  opposing to rotational motion. The values used in the simulation are according to realistic data when deploying spring hinge-loaded solar panels for nano-spacecraft [12].

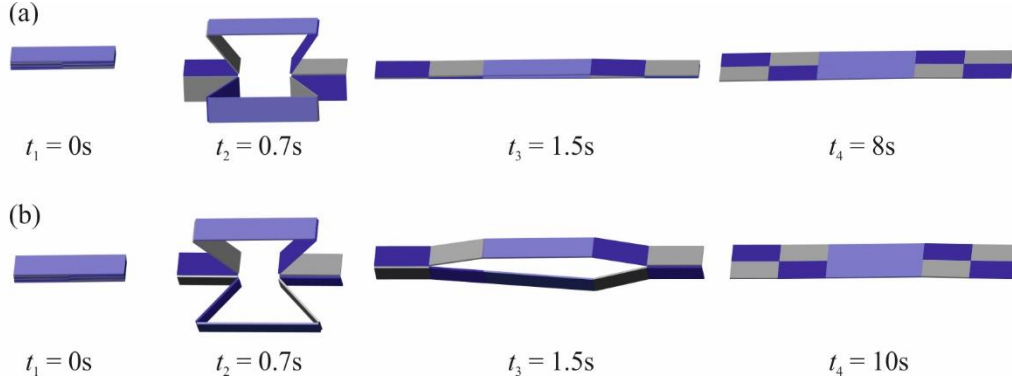
When the rotational angle goes beyond the permissible range, facet contact occurs to simulate a hard stop. The damping coefficients  $c_u, c_l$  and spring stiffness  $k_u, k_l$  are chosen to reflect the actual situation upon contact. More specifically,  $k_u, k_l$  are set to allow penetrations of panels with no more than 0.1% of their side length, while  $c_u, c_l$  are set based on material properties of the panels, shear modulus, and density in particular. In this example, the spring stiffness and damping coefficient are chosen as  $10^6Nm/rad$  and  $100Nm/(rad/s)$ , respectively.

To make the system more computationally efficient, a cut is made along a hinge line  $MN$  and the revolute joint is replaced by two sets of very stiff translational springs and dampers. It should be noted that the choice of  $k_M, k_N$  and  $c_M, c_N$  have a significant effect on the accuracy and computational speed. For instance, at  $MM'$ , the larger the value of  $k_M$ , the more accurate simulation results will be, for the closed-loop can be better represented by two open chains without any gaps where the two ends are connected. However, an extremely large value of  $k_M$  will make the system too stiff to affect computational efficiency, and sometimes can lead to ill-conditioned dynamic equations. Regarding the damping coefficient, a large value of  $c_M$  allows a speedy convergence. However, a proper value of  $c_M$  shall be chosen to damp out excessive oscillations between points  $M$  and  $M'$ , without unrealistically losing too much energy. Several values of  $k_M$  and  $c_M$  are tested while tracking the relative distances between  $M$  and  $M'$  during the deployment process. Finally, the spring stiffness  $k_M$  is chosen as  $10^7N/m$  and the damping coefficient  $c_M$  is  $5 \times 10^4N/(m/s)$  for a minimum error and a reasonable computational speed. The same set of parameters is to connect  $NN'$ .

Having set all parameters in the system, one can then initiate the optimization process, starting from assigning the same rotational spring stiffness  $k_i = 0.2Nm/rad$  to all hinges. Collision between any pair of rigid bodies will be checked geometrically after each simulation via GJK algorithm. Each rigid body is represented by a bounding box whose dimensions are 99.9% of its actual dimensions. In other words, the collision checker allows penetration that is no more than 0.1% of its actual dimension, which is compatible with the computational discrepancy at the cut joint and the facet contacts due to hard stop at hinges.

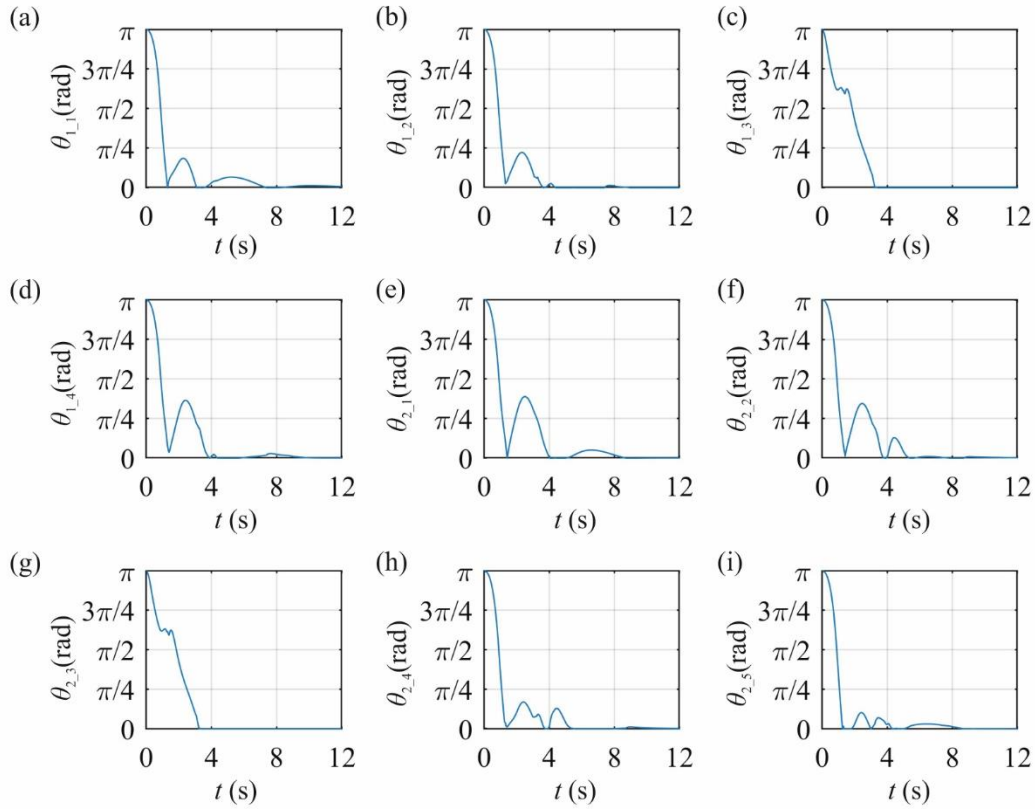
Before optimizing, when the same spring stiffness is assigned at all hinges, a collision is found halfway through the deployment process. Some snapshots of the deployment are shown in Figure 8. A facet collision occurs between the first- and the second- row of panels, as shown at  $t_3$  in Figure 8(a). To avoid collision, the set of spring stiffness at various hinges  $K$  is optimized by minimizing the objective function through *fmincon* that is elaborated in section IV. However, the solution obtained via the continuous optimization solver *fmincon* might not be easily achieved in manufacturing. For instance, the spring stiffness along the two horizontal hinges  $h_{1-3}$  and  $h_{2-3}$  are calculated to be  $0.1532Nm/rad$  and  $0.1528Nm/rad$ , respectively. They are both rounded to  $0.15Nm/rad$ , while the stiffness at other hinge lines is rounded to be  $0.2Nm/rad$ . By the end of the optimization process, another simulation is conducted with the rounded values to check if the deployment process is indeed collision-free. Some snapshots are presented in Figure 8(b). It could be observed that the optimized spring coefficients allow the panels

that are connected by horizontal hinges  $h_{1,3}$  and  $h_{2,3}$ , to deploy slower than the others. Thus the collision is avoided.



**Figure 8. The deployment process of a loop of ten rigid panels (a) with a collision at  $t_3$ , and (b) without collisions.**

In this example, there are 10 rigid bodies connected by 10 rotational hinges. Except for the cut joint along  $MN$ , there are 9 generalized coordinates  $\theta_{b,i}$  defined along hinge  $h_{b,i}$ . The time histories of all the defined rotational angles are measured, as shown in Figure 9, which demonstrates the trajectories of all panels throughout the collision-free deployment. All the angles started from  $\pi$  at  $t = t_0$  and settled to 0 when fully deployed. In the middle of the deployment process, the sharp turns at 0 rad are due to the activation of contact elements at the hinge between two side facets along the thickness direction. Penetrations of panels are therefore not observed. The panels swing back and forth before settling to their corresponding final configurations. It can also be observed that none of the angles reaching  $\pi$  after the deployment was initiated. This is a good indication as the rotational springs are not overly stiff.



**Figure 9. (a) - (i) Time history of rotation angles throughout the collision-free deployment process.**

## VI. Discussion and Conclusion

In the past, most literature investigated the design and kinematics of deployable structures with single DoF, whose morphologies were easy to predict. This paper, for the first time, adopts an optimization approach incorporating the non-linear dynamics of thick panel kirigami to deploy bi-directionally. It should be noted that the time taken from folded to fully deployed configurations is not considered in this optimization scheme. This is intentional as the time differences that are less than a couple of minutes for deployment do not matter in a space mission. Rather, a collision-free deployment is the focus of this paper, as it protects the panels from damage.

At each time step of one forward dynamic simulation, any occurring intersection between any two panels could be detected geometrically via the GJK algorithm. The optimization package can thus be written to maximize the proximal distance between the panel pair that came closest to each other during the deployment process, or, to minimize the approaching speed if panels collide, by tuning the spring stiffness at various hinge lines. By the end of an optimization process, a collision-free deployment can be identified by checking if the objective function is less than zero. In some cases, no collision-free trajectories can be found through the system optimization via *fmincon*. This may be because of an improper initial guess. The nonlinear solver *fmincon* can only search for a local minimum instead of a global extremum. One can restart the optimization process by re-setting the initial guess. In the example demonstrated in Section V, a collision-free solution was successfully found. It can be observed from Figure 9 that the time history profiles in (c) and (g) have similar trends, but they are different from that of the others. Symmetrical deployment is observed because the spring stiffness at the two horizontal hinges,  $h_{1,3}$  and  $h_{2,3}$  are set to be  $0.15Nm/rad$  whereas that of the other hinges are set to be  $0.2Nm/rad$ . Besides, the optimization package might output very small numbers. When the optimized spring stiffness  $k_i$  is found to be less than 1% of the spring stiffness at the other locations, it can be rounded up to  $0Nm/rad$ . For instance, if the spring stiffness at some hinge  $h_i$  is calculated to be  $k_i < 1.5 \times 10^{-3} Nm/rad$  in the example shown in Section V, no spring is placed along that hinge physically.

Due to the geometric complexity involved in our system, multiple equilibria positions might occur during the deployment. This brings the possibility of the system not being flat by the end of the simulation. The possibility of incomplete deployment requires the optimization process to incorporate a method to determine if the compact stacks are fully deployed to a flat surface. It can be done by checking the configuration of every panel by the end of the dynamics simulation against its theoretical fully deployed configuration. If the difference in configuration is not arbitrarily small, the deployment is considered incomplete. Some examples of panels not being fully deployed will be further investigated in our future works.

## Acknowledgments

The authors wish to acknowledge the support of the Air Force Office of Scientific Research (FA9550-16-1-0339). J. Yang would like to thank the Agency for Science, Technology and Research (A\*STAR) for sponsoring her DPhil studies.

## References

- [1] "JAXA IKAROS Mission," Japan Aerospace Exploration Agency, 2003. [Online]. Available: [https://global.jaxa.jp/countdown/f17/overview/ikaros\\_e.html](https://global.jaxa.jp/countdown/f17/overview/ikaros_e.html). [Accessed 16 June 2019].
- [2] S. G. Guest and S. Pellegrino, "A New Concept for Solid Surface Deployable Antennas," *Acta Astronautica*, vol. 38, no. 2, pp. 103-113, 1996.
- [3] F. Santoni, F. Piergentili, S. Donati, M. Perelli, A. Negri and M. Marino, "An Innovative Deployable Solar Panel System for Cubesats," *Acta Astronautica*, vol. 95, pp. 210-217, 2014.
- [4] G. E. Fenci and N. G. R. Currie, "Deployable structures classification: a review," *International Journal of Space Structures*, vol. 32, no. 2, pp. 1-19, 2017.
- [5] Y. Chen, R. Peng and Z. You, "Origami of thick panels," *Science*, vol. 349, no. 6246, pp. 396-400, 2015.
- [6] R. J. Lang, K. A. Tolman, E. B. Crampton, S. P. Magleby and L. L. Howell, "A Review of Thickness-Accommodation Techniques in Origami-Inspired Engineering," *Appl. Mech. Rev.*, 2018.
- [7] B. J. Edmondson, R. J. Lang, M. R. Morgan, S. P. Magleby and L. L. Howell, "Thick rigidly foldable structures realized by an offset panel technique," in *Origami 6*, Tokyo, Japan, 2015, pp. 149-161.
- [8] J. Yang and Z. You, "Compactly Folding Rigid Panels with Uniform Thickness through Origami and Kirigami," in *ASME IDETC*, Anaheim, CA, 2019.
- [9] W. R. Hamilton, "Letter to John T. Graves on the Icosian," in in: *The Mathematical Papers of Sir William Rowan Hamilton*, Cambridge University Press, 1931, 17 Oct., 1856, pp. eds. H. Halberstam and R.E. Ingram, Vol. 3 (Algebra).

- [10] K. A. Seffen and S. Pellegrino, "Deployment dynamics of tape springs," *the Royal Society*, vol. 455, pp. 1003-1048, 1999.
- [11] A. M. Watt and S. Pellegrino, "Rigid Deployable Solar Array," University of Cambridge, Cambridge, 2004.
- [12] F. Santoni, "Dynamics of Spring-Deployed Solar Panels for Agile Nanospacecraft," *J. Aerosp. Eng.*, vol. 28, no. 5, 2015.
- [13] L. E. Kavraki and S. M. LaValle, "Motion Planning," in *Handbook of Robotics*, Berlin, Heidelberg, Springer, 2008, pp. 109-131.
- [14] "Simscape - Matlab & Simulink," Mathworks, R2019b. [Online]. Available: <https://uk.mathworks.com/products/simscape.html>. [Accessed 06 Mar 2020].
- [15] E. G. Gilbert, D. W. Johnson and S. S. Keerthi, "A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, 1988.
- [16] C. Fuhrer and B. J. Leimkuhler, "Numerical solution of differential-algebraic equations for constrained mechanical motion," *Numerische Mathematik*, vol. 59, p. 55-69, 1991.
- [17] J. Wang, C. M. Gosselin and L. Cheng, "Modeling and Simulation of Robotic Systems with Closed Kinematic Chains Using the Virtual Spring Approach," *Multibody System Dynamics*, vol. 7, p. 145-170, 2002.
- [18] P. E. Nikravesh, "Newtonian-based methodologies in multi-body dynamics," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-Body Dynamics*, vol. 222, no. 4, p. 277-288, 2008.
- [19] S. Miller, "Simscape Multibody Multiphysics Library," MATLAB Central File Exchange, 2020. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/37636-simscape-multibody-multiphysics-library>. [Accessed 10 01 2020].
- [20] P. Bogacki and L. F. Shampine, "A 3(2) pair of Runge-Kutta formulas," *Appl. Math. Letters*, vol. 2, p. 321-325, 1989.
- [21] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE Suite," *SIAM Journal on Scientific Computing*, vol. 18, p. 1-22, 1997.
- [22] C. Ericson, Real-Time Collision Detection, CRC Press, 2005.
- [23] T. F. Coleman and Y. Li, "On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds," *Mathematical Programming*, vol. 67, p. 189-224, 1994.
- [24] "MATLAB Optimization Toolbox - fmincon," Mathworks, R2019b. [Online]. Available: <https://uk.mathworks.com/help/optim/ug/fmincon.html#References>. [Accessed 15 Feb 2020].