PLANNING MOTIONS FOR SHAPE-MEMORY ALLOY SHEETS

MUKULIKA GHOSH, DANIEL TOMKINS, JORY DENNY, SAMUEL RODRIGUEZ, MARCO MORALES, AND NANCY M. AMATO

1. Introduction

In many settings, such as space or deep-sea exploration, the number and types of resources that a mission can accommodate are limited and simply inadequate to dynamically repurpose a component. This can be overcome using reconfigurable smart materials such as Shape Memory Alloys (SMAs). SMAs remember their original shape such that a deformed SMA can return to a trained shape upon changes in temperature [Jani et al. 14]. Components built from SMAs can adopt many shapes, allowing the same component to be used for a number of tasks, including new tasks that were not considered before deployment. For example, SMAs can be used to build shape shifting structures that morph based on environmental factors such as temperature or light exposure.

A major challenge in planning motions for an SMA robot is that the sheet can bend at an infinite set of points. Planning motions for SMA robots is important because it not only allows us to explore the solution space of a particular SMA robot, but it also simulates the folding path helpful in determining the parameters required to achieve foldings for physical SMA robots, e.g., how much and where to heat to achieve a desired fold. For example, in Fig. 1 a sheet (a) folds through an intermediate state (b) into a cylinder (c).

This work describes how an SMA folding problem is modeled so that it can be solved using a state-of-the-art sampling-based motion planning algorithm. We model the sheets similar to rigid origami [Tachi 11] as groups of inflexible regions connected by flexible regions, which we call joints, and restrict the motion of the sheet along the flexible joints. This reduces the bends the sheet can attain to single dimension. Unlike rigid origami, however, we cannot allow sharp folds these SMA sheets. Hence we model the SMA joints to have uniform curvature along the joint length. In addition to modeling the system, we consider the problem of finding not only collision free but also gravitationally stable motions (Fig. 1(b)). These are motions that keep the center of mass positioned over some panel of the structure on the ground plane so that it does not topple down. We show how we can apply the Rapidly-exploring Random Tree (RRT) [LaValle and Kuffner 01], a common sampling-based planner, to this problem. We augment the original RRT algorithm

---

M. Ghosh, D. Tomkins, J. Denny, S. Rodriguez and N. Amato are with the Parasol Lab., Dept. of Computer Science and Engineering, Texas A&M Univ., College Station, Texas, 77843, USA {mghosh, kittsil, jdenny, sor8786, amato}@cse.tamu.edu.

M. Morales is with Instituto Tecnológico Autónomo de México marco.morales@itam.mx.

This research supported in part by NSF awards CNS-0551685, CCF-0833199, CCF-0830753, IIS-0916053, IIS-0917266, EFRI-1240483, RI-1217991, by NIH NCI R25 CA090301-11, by Chevron, IBM, Intel, Oracle/Sun, by Award KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST) and by Asociación Mexicana de Cultura AC. J. Denny supported in part by an NSF Graduate Research Fellowship.
Figure 1. Folding an SMA sheet from its unfolded flat state (a) passing through a series of feasible (non-colliding and gravitationally stable) intermediate states, like (b), to a folded shape (c). (d) is an infeasible state, since it is not gravitationally stable.

with a new distance function specific to our SMA robots and a rigidity analysis to see which components can be moved without causing instability (as in Fig. 1(d)).

Our specific contributions are as follows:

- A geometric representation of SMA robots useful for sampling-based motion planning.
- A novel definition of motion feasibility that includes collision, curvature, and stability constraints suitable for SMA robots.
- An adaptation of RRTs designed specifically for folding SMA sheets into 3D shapes.

Our results validate our model and motion planning algorithm for these types of robots by folding a single SMA sheet into multiple interesting 3D shapes. These results show significant flexibility in modeling various planning problems and include physical constraints at a comparable time to not using stability constraints.

2. Preliminaries and Related Work

A robot is a movable object whose state can be described by $d$ parameters, or degrees of freedom (DOFs), each corresponding to an object component (e.g., object positions, orientations, and/or link angles). In SMA sheets, the DOFs are the angles of their flexible regions. Hence, a robot’s state, or configuration, can be uniquely represented by a point $(x_1, x_2, \ldots, x_d)$ in an $d$-dimensional space (where $x_i$ is the $i$th DOF), called the configuration space ($C_{\text{space}}$) [Lozano-Pérez and Wesley 79]. The subset of all feasible configurations is the free space ($C_{\text{free}}$), while the subset of the infeasible configurations is the obstacle space ($C_{\text{obst}}$). Thus, the motion planning problem becomes that of finding a continuous trajectory in $C_{\text{free}}$ from a given start configuration to a goal configuration. In general, it is infeasible to compute explicit $C_{\text{obst}}$ boundaries [Reif 79], but we can often determine whether a configuration is feasible or not quite efficiently, by performing a validity test, e.g., collision detection test in the workspace, the robot’s natural space.

Sampling-based methods [Kavraki et al. 96, LaValle and Kuffner 01] are quite successful at solving motion planning problems. One method, Rapidly-exploring Random Tree (RRT) [LaValle and Kuffner 01], explores $C_{\text{free}}$ locally by iteratively expanding nodes of the tree towards random configurations until a goal configuration or region has been reached. Many variants have been proposed to address various weaknesses of RRTs [Kuffner and LaValle 00, Karaman and Frazzoli 11]. Sampling-based planners have been applied for planning motions for deformable robots in deformable environments [Rodriguez et al. 06], a related but distinct problem from SMA sheet folding where robots are self-deformable, and maintain their total surface area. One of the earlier approaches, f-PRM [Holleman et al. 98,
Kavraki et al. 98, Guibas et al. 99], samples the control points of a Bézier surface that models a flexible patch. Configuration validation accounts for collisions and for energy feasibility. Despite their success, however, these approaches do not maintain certain parts as rigid and do not account for contact constraints with a surface, such as gravitational stability.

Folding in robotics has been well studied. Sampling-based techniques have been applied to paper folding planning [Song and Amato 01], and Balkom and Mason designed a robotic system to fold paper [Balkcom and Mason 08]. Specialized robotic systems using SMA joints and actuators have been designed for origami folding [An and Rus 12]. Printable, self-folding robots using SMA sheets have also been designed in [Felton et al. 13].

There has been some work analyzing the properties of robots like ours. For instance, a preliminary paper introduced an algorithm to find flat unfoldings of complex geometric shapes [Hernandez et al. 13]. The unfoldings were then fed into complex analysis systems to model them as self-actuating SMA sheets. However, planning collision-free and gravitationally stable motions of these structures is an unexplored problem.

3. AN SMA PLANNER

We propose to apply an RRT planner to SMAs that accounts for gravity as a constraint and that biases the extension of the RRT towards narrow passages produced by the tight constraints in folding. In this section, we describe the algorithm used for planning motions for SMAs. Next, we describe how we model SMA robotic systems to define configurations. Then, we describe the distance function, validity test, and rigidity extender used in the planner.

3.1. AN SMA RRT PLANNER. We plan the motions of SMA robotic systems using an RRT algorithm, as shown in Algorithm 1. RRT starts from the initial configuration of the robot, \( q_{\text{root}} \), and builds a tree that randomly explores the \( C_{\text{space}} \). The nodes of this tree are valid configurations for the SMA robot, and its edges are valid paths between the nodes that they connect. Thus, the planning problem consists of generating a tree that has one leaf close enough to the goal SMA robot configuration. This tree is built by repeatedly randomly selecting a configuration \( q_{\text{rand}} \); finding \( q_{\text{near}} \), the configuration in the tree that is closest to \( q_{\text{rand}} \); and expanding from \( q_{\text{near}} \) to \( q_{\text{rand}} \) until either becoming invalid or attaining some maximum distance \( \Delta q \).

**Algorithm 1 RRT**

**Input:** number of expansion attempts \( n \), root configuration \( q_{\text{root}} \), maximum expansion length \( \Delta q \)

**Output:** Tree \( T \)

1. \( T.\text{AddNode}(q_{\text{root}}) \)
2. for \( 1 \ldots n \) do
3. \( q_{\text{rand}} \leftarrow \text{RandomCfg}() \) // generate random SMA configuration
4. \( q_{\text{near}} \leftarrow \text{NearestNeighbor}(T, q_{\text{rand}}) \) // find closest existing configuration to \( q_{\text{rand}} \)
5. \( q_{\text{new}} \leftarrow \text{RigidityExtend}(q_{\text{near}}, q_{\text{rand}}, \Delta q) \) // move from \( q_{\text{near}} \) to \( q_{\text{rand}} \)
6. \( \text{Update}(T, q_{\text{near}}, q_{\text{new}}) \) // add new SMA configuration to tree
7. return \( T \)
3.1.1. Generation of random SMA configuration. (Line 3) The generation of random SMA configurations, is performed by obtaining a random joint angle for each of its dof's using a uniform distribution. As explained in Section 3.2, each of these dof's are the angles of the flexible regions in the SMA robot.

3.1.2. Finding the closest existing SMA configuration. (Line 4) In order to find the closest existing SMA configuration to $q_{rand}$, we apply the distance function described in Section 3.3 to try to identify the configuration already in the tree that is the “easiest” to transform into $q_{rand}$.

3.1.3. Expanding the tree. (Line 5) The tree is expanded by moving from $q_{near}$ towards $q_{rand}$. This is done by following a straight path in the $C_{space}$ in small increments, each of which is tested for validity as explained in Section 3.4. If a maximum distance $\Delta q$ is reached without encountering any invalid configurations, the last configuration found is returned as $q_{new}$. Otherwise, the last valid configuration is returned as $q_{new}$. In this expansion, we use a Rigidity Extender, a modification (explained in Section 3.5) introduced in this paper, to better deal with complex physical constraints.

3.1.4. Updating the tree. (Line 6) The final step of the algorithm is to add the configuration $q_{new}$ to the tree and an edge from $q_{near}$ to $q_{new}$ weighted by the distance between these two configurations.

3.2. Modeling an SMA Robotic System. An SMA sheet is composed of a thin, compliant elastomer layer sandwiched between two grids of SMA wires, as shown in Fig. 2(a). This composition allows for both strength and stability as stated in [Peraza-Hernandez et al. 13] and allows selective application of heat to achieve a particular folding path. The physical and geometric properties of these sheets are explored in-depth in [Peraza-Hernandez et al. 13, Hernandez et al. 13].

To plan the motions of this robot, we need to be able to model each of its potential configurations, i.e., define a mapping from the robot’s workspace to its $C_{space}$. However, an SMA surface can bend at any subset of its infinite points and therefore has infinite degrees of freedom. Since this is both computationally infeasible, we introduce a restricted model of an SMA sheet which still preserves many usable configurations.

In this paper, we model SMA sheets as being composed of inflexible SMA regions and flexible SMA joints similar to rigid origami [Tachi 11]. We can then assume
a uniform curvature along each joint, perpendicular to the surface of the inflexible regions and parameterized by one \( \text{DOF} \). In other words, each joint will attain a uniform curvature throughout its length. In a physical implementation, heat is applied to actuate a region of the SMA; thus, our representation assumes an even application of heat to the joint region, causing this uniform curvature. It should also be noted that, because SMA sheets cannot have sharp folds like rigid origami \cite{Tachi11}, it is important to model smooth curves.

Therefore, a single degree of freedom representing an SMA joint \( i \) is just an angle \( \theta_i \in [-\pi, \pi] \), which is the total angle of curvature through the joint, subject to some physical constraints required by the material’s physical properties.

As an example, consider the robot shown in Fig. 2(b). This robot is a 21cm by 10cm SMA sheet, which is modeled by two inflexible 10cm by 10cm squares connected by a single 1cm by 10cm joint. This robot has one degree of freedom, which is the angle in the SMA joint; in the figure, this joint is actuated to \( \frac{\pi}{2} \) radians.

### 3.3. Balance Distance.

The distance between two configurations is a numerical representation of the difference between them; i.e., two configurations should have a low distance, if one can be easily transitioned into another.

Consider two configurations \( \langle \theta_1, \theta_2, \ldots, \theta_n \rangle \) and \( \langle \theta_1', \theta_2', \ldots, \theta_n' \rangle \) (where \( n \) is the \( \text{DOFS} \) of the robot). Typically, sampling based algorithms will use the Euclidean distance in \( C_{\text{space}} \), \( \sqrt{\sum_{i=1}^{n} (\theta_i - \theta_i')^2} \). However, two configurations which have low difference in joint angles may have very different centers of mass (Fig. 3(a,b)); for the sake of gravitational stability, we do not want to consider these configurations to have an easy transition between them as the intermediate configurations might not be gravitationally stable. Additionally, configurations which have noticeable difference in joint angles can have close (or even identical) centers of mass (Fig. 3(b,c)), so it is not sufficient to only look at the difference between the centers of mass of two configurations.

As a compromise, we use a weighted distance metric called balance distance. With this metric, the weight \( w_i \) of joint \( i \) is proportional to the fraction of the sheet’s mass actuated by that joint. (For instance, the joint in Fig. 2(b) has weight \( \frac{1}{2} \), since it divides the sheet in half.) Thus, the weight \( w_i \) encodes the contribution of the joint \( i \) in the center of mass of the robot. We then evaluate the total distance between two configurations as the sum of the weighted differences, \( \sum_{i=1}^{n} w_i \cdot |\theta_i - \theta_i'| \).

This metric ties the two ideas, difference in angles and difference in center of mass, together, since joints which actuate more mass will contribute more to changes in the position of the center of mass.

---

**Figure 3.** Balanced distance considers both difference in joint angles and difference in center of gravity: Configurations in (a) and (b) are closely related to each other with respect to joint angles. Whereas configurations in (b) and (c) are closely related to each other with respect to position of center of mass.
3.4. **Validity.** A validity checker is a method for classifying configurations as belonging to either $C_{\text{free}}$ or $C_{\text{obst}}$, a basic unit of work for most sampling-based planners. Our validity checker is composed of three parts. First, because the SMA material is relatively stiff and cannot obtain a large radius of curvature, our planning algorithm limits the fold angle based on the length of the joints. Second, we do not allow the robot to collide with itself. To quickly detect collisions, we arrange the inflexible regions into a configuration based on the angles through each joint. Finally, we require the folds of the sheet to be gravitationally stable; that is, the center of mass must be suspended over some panel on the ground plane.

3.4.1. **Radius of Curvature.** The SMA sheets considered are relatively stiff; they are not able to obtain a small radius of curvature [Peraza-Hernandez et al. 13]. Obtaining sharp angles is not possible unless the joint has a large length. Therefore, we limit the angle $\theta$ in each joint to the range $[-\frac{l}{r}, \frac{l}{r}]$, where $r$ is the radius of curvature of the material. For our simulations, we use $r$ to be $\frac{l_{\text{max}}}{\pi}$ where $l_{\text{max}}$ is the maximum of joint lengths for the SMA robot. This constraint is imposed implicitly in the sampling phase of our configurations.

3.4.2. **Collision Detection.** Since collision detection is much faster for polygons than for curved surfaces, we approximate each curved joint with a series of very thin panels along its length. The approximation error is negligible as compared to the improvement in performance obtained due to approximation. After configuring the robot in the workspace, we use a standard collision detection library [Gottschalk et al. 96] to check its validity.

3.4.3. **Gravitational Stability.** The stability of a configuration is checked against the gravitational force. For folding, we do not consider other forces, like friction nor external actuators to hold the structure. Also, we assume that the structure is always in contact with the ground in such a way that it does not fall down. In many cases this means that at least one full panel is resting on the ground. A body is considered to be in stable equilibrium under gravity if its weight, acting from the center of mass, intersects with the base of the body. Therefore, to determine the stability of an SMA robot at a particular configuration, we compute the center of mass and the base of the robot at the configuration. The center of mass is approximated as the geometric mean of the configuration, assuming uniform density throughout the SMA joints and inflexible regions. The global position of one of the panels, known as the base panel, is fixed (not necessarily fixed to the ground) as input to our system. In future, we will relax this constraint to achieve a more broad set of motions. The global position of the SMA robot at a configuration is then computed using the joint angles and the position of the base panel. As mentioned before, we assume that the structure is always in contact.
with the ground, so we find all the points where the SMA robot touches the ground plane, and the convex hull of these points defines the stable base region for the configuration. If a ray generated from the center of mass, in the direction of the force of gravity, intersects the base region, the configuration is considered to be stable (Fig. 4(a)); that is, it is valid in gravity. Otherwise the configuration is deemed unstable, or invalid in gravity (Fig. 4(b)). This introduces narrow passages in $C_{free}$ as many self-collision free configuration such as Fig. 4(b) are considered gravitationally unstable.

3.5. Rigidity Extender. In the tree expansion step of the planner (Line 5 Algorithm 1), we use a Rigidity Extender to better deal with complex physical constraints like gravitational stability that create narrow passages in $C_{free}$. In many cases, moving a single joint can cause the robot to collide or become gravitationally unstable. However, most other joints can move freely, and must do so in order to attain a new configuration. Rigidity analysis [Thomas et al. 07] allows us to lock joints which cause problems while attempting to adjust the other joints.

Before extending $q_{near}$ towards $q_{rand}$ to produce $q_{new}$, the Rigidity Extender tests for rigidity of all joints, as shown in Fig. 5, by slightly perturbing each individual joint in $q_{near}$ in the direction of $q_{rand}$. If that perturbation produces an invalid configuration, that joint is identified as rigid. Once the rigid joints have been identified, the extension from $q_{near}$ towards $q_{rand}$ to produce $q_{new}$ is tried by keeping the rigid joints locked.

4. Experiments

This planning approach has been implemented in a C++ motion planning library developed in the Parasol Lab at Texas A&M University.

We demonstrate the effectiveness of our approach in a variety of models shown in Fig. 6 and described below:

- **Tube.** (Fig. 6(a)) The left flap is longer than the right flap, making this not only an ordering problem (right then left) but also an interesting stability problem.
- **Latin Cross.** (Fig. 6(b)) In this problem, we model a reconfigurable Latin cross which begins as a cube and must be reconfigured into a different shape, an octahedron. This example shows how a single cut-out can be used to fold into multiple shapes.
- **Bird and Trap.** (Fig. 6(c)- 6(d)) Cut-outs which are folded into interesting shapes (bird and trap).
- **4-Flap.** (Fig. 6(e)) Each flap must be folded in a particular order, a difficult task for the planner.
The experiments were run on a Rocks Cluster running CentOS 5.1 with Intel XEON CPU 2.4 GHz processors with the GNU gcc compiler version 4.1. The video results of the folding of the above models can be found at [SMA]. Our video result also contains another example where a single cut-out can be used for multiple shapes. As the start configuration of the additional example is not gravitationally feasible, we do not include its result here. We present the total planning time for each model, averaged over 10 trials, in Table 1. The number of samples and the total planning time for planning without gravity constraint and with rigidity extender is stated in columns 3-4 (labeled “Without Gravity”) in the table. The same for planning with gravity but without rigidity extender is given in columns 5-6 (labeled “Without Rigidity Extender”) and for planning with both gravity and rigidity extender is given in columns 7-8 (labeled “With all Extensions”) in Table 1. As shown in the table, our method can plan for a wide variety of models with complex constraints within reasonable time which is less than 10 seconds. Hence, the simulation results can be readily used to determine physical properties like amount of heat applied to each joint to fold a physical SMA robot. Without the gravity constraint, many of the intermediate configurations in the folding path are invalid under gravity (See Fig. 7). These infeasible intermediate configurations creates narrow passages in $C_{free}$ contributing to the total planning time (as in Tube and Latin Cross).

Although planning with gravitational stability takes more time than without, we observe that the gravitational stability check does not heavily contribute to the total time. 4-Flap and Bird models, which are very stable in all configurations, have the same number of samples and very similar times without gravity.

Unmodified RRT [LaValle and Kuffner 01] could not plan the motions for most of our robots. Table 1 demonstrates that the rigidity extender is required for folding of
certain SMA robots under gravity. For example, even after 10 minutes the planner was not able to fold the Tube and Vertical Trap to their goal configurations.

Planning time depends on both the number of flexible SMA joints and the number of samples required to find a path (which is in turn dependent on the difficulty of the specific planning problem). For example, Bird has many DOFs, but its time is comparable to 4-Flap, which is a more difficult planning problem.

We do not compare with any other approach as no other planning method is general enough to handle these complex robotic systems, to the best of the authors’ knowledge as detailed in Section 2.

Our results show that not only is our model general enough to handle a broad spectrum of robot systems, but also it can plan for these models in reasonable time.

5. Conclusion

In this work, we show how an existing motion planning algorithm can be adapted to fold SMA sheets from a flat state to a three dimensional shape with constraints such as collision-free motion and gravitational stability. We model the SMA sheet as a set of flexible joints connecting inflexible panels to allow selective heating of the parts of the sheet to obtain a particular folding. We introduced a new distance function for SMA robots that allows us to better capture the movement from one configuration to another while accounting for gravitational stability, an important constraint in physical systems. We also introduced a rigidity analysis that allows us to lock joints whose motion might lower the probability of finding feasible motions between configurations. Our results show that our model can fold a variety of SMA sheets with the constraints in reasonable time. Hence, generation of folding paths for a variety of shapes can be used further in generation of physical parameters like...
heat applied to each joint for a physical SMA sheet to fold into a desired shape and also to evaluate how difference in heat applied would reconfigure the sheet into a different shape.

References