Folding Cartons with Fixtures: A Motion Planning Approach

Liang Lu Srinivas Akella
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

Packaging products such as telephones and two-way radios after assembly is a common manufacturing task. Carton folding is a packaging operation performed by human operators or with fixed automation. We present a flexible method to fold cardboard cartons using fixtures; a carton blank is folded by moving it through a fixture with a robot. This method of using interchangeable fixtures enables rapid changeovers between product models. We outline a procedure to design a fixture given a carton and a folding sequence. We present an implemented motion planning algorithm that generates all folding sequences for a carton by modeling it as a many degree of freedom robot manipulator with revolute joints and branching links. A fixture constrains the carton to move along paths consisting of line segments in its configuration space, and the motion planner generates these paths. To illustrate the method, we selected a folding sequence for an example carton, designed a fixture, and demonstrated folding of the carton from blanks with an industrial robot.

1 Introduction

Packaging products after assembly is a common manufacturing task. Cartons folded from flat cardboard blanks are used to package products such as telephones and two-way radios (Figure 1). Carton folding occurs in packaging, distribution, and shipping centers. Traditional carton folding machines cannot always handle the different carton styles necessitated by product changes. Carton folding by humans requires dextrous manipulation of the carton, and leads to repetitive stress injuries.

We present a method to fold cardboard cartons using fixtures, where a carton blank is folded by moving it through a fixture with a robot. The fixtures are interchangeable and enable rapid changeovers between product models. Folding a carton with a fixture requires generating valid folding sequences for the carton, designing a fixture for a selected folding sequence, and implementing the folding operations on the fixture with a robot. A carton is an articulated structure that can be viewed as a robot manipulator with revolute joints and branching links. For a class of self-locking cartons that are folded from flat carton blanks, we present a motion planning algorithm to automatically generate all carton folding sequences. We outline a procedure to design a fixture given a carton and a folding sequence. A folding fixture constrains the carton motion to paths consisting of line segments in its configuration space. The motion planner generates feasible carton folding sequences by finding these paths. To illustrate the method, we selected a planner generated folding sequence for an example carton, designed a fixture using the design procedure, and demonstrated folding of the carton from blanks with an AdeptOne robot.

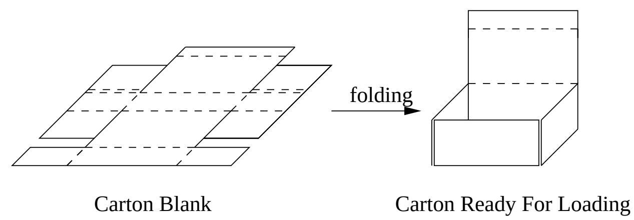

Figure 1: The Radio carton blank in its initial configuration and its goal folded configuration.

A folding carton typically has many more degrees of freedom than the robot used to fold it, and the shape of the carton changes as it is manipulated. Our approach is to use the motion planner as a tool to aid the design of minimal complexity hardware by a human designer. This work is a first step towards our long term goal of automatically designing carton folding fixtures.

We review related work in Section 2, and in Section 3 outline the fixture design procedure given a carton and a folding sequence. We present a motion planning algorithm to generate folding sequences in Section 4 and demonstrate it on different carton styles in Section 5. We describe an implemented fixture and folding sequence for an example carton in Section 6. We conclude with a summary and an outline of future work.

2 Related Work

Carton folding machines are typically available for cartons that are glued or taped in their folded configuration. We are aware of at least one commercial machine to fold some of the cartons considered in this paper [19]. While these machines can handle cartons of different sizes, they cannot handle cartons of different styles. Carton folding designs in the patent literature (for example, Capdeboscq [4], Marschke [15], McBride [16], Ward [21]) often use rotating wheels or spiral bars at different orientations to fold carton panels at their creases as the carton blank moves past.

Carton folding is similar to creating 3-D sheet metal parts from blanks by bending. de Vin *et al.* [5] describe a computer-aided process planning system to generate bending sequences for sheet metal components. It uses rules to minimize handling, select tools, and avoid tool and part collisions. Radin, Shpitalni, and Hartman [18] present a two-stage algorithm that first rapidly generates a feasible bending sequence using collision avoidance heuristics and then searches for lower cost solutions without violating its time constraints. Gupta *et al.* [8] describe a fully automated robotic system for planning and executing bends on sheet metal blanks. Wang [20] develops methods to decompose sheet metal products into components for ease of manufacturing during cutting, bending, and assembly. He unfolds 3-D products into 2-D patterns, and identifies unfolding bend sequences that avoid collisions with tools.

The motion constraints in carton folding parallel those in assembly planning. Wilson and Latombe [22] developed the non-directional blocking graph representation for assembly planning by disassembly sequencing. Of related interest is work by Goldberg and Moradi [6] on assembly planning for machines that execute a pipelined series of part rotations and vertical insertions requiring only one or two degrees of freedom. This is an example of the trend in robotics towards minimalist solutions (Canny and Goldberg [3]) that use simple and inexpensive actuators and sensors.

We model cartons as robot manipulators to generate folding sequences by motion planning. See Latombe's book [10] for an introduction to motion planning. Lozano-Perez [11] developed an algorithm to plan collision-free motions of serial manipulators. Its complexity is exponential in the robot degrees of freedom. It recursively approximates the configuration space obstacles from one-dimensional slices. Only a few practical motion planning methods have been developed for robots with many degrees of freedom. These include the use of potential field and randomized search techniques (Barraquand and Latombe [1]), construction of probabilistic roadmaps (Kavraki *et al.* [9]), and sequential planning with backtracking (Gupta and Guo [7]).

Recent work in computational geometry relates polyhedral shapes and the polygonal shapes they can be unfolded to, and draws connections to the art of origami. Lubiw and O'Rourke [14] present a dynamic programming algorithm to determine if a given polygon can be folded to a convex polytope. Biedl *et al.* [2] study unfoldings of two classes of nonconvex orthogonal polyhedra.

3 Folding with Fixtures

A carton blank is a flat cardboard cutout structure consisting of a set of panels, with creases separating adjacent

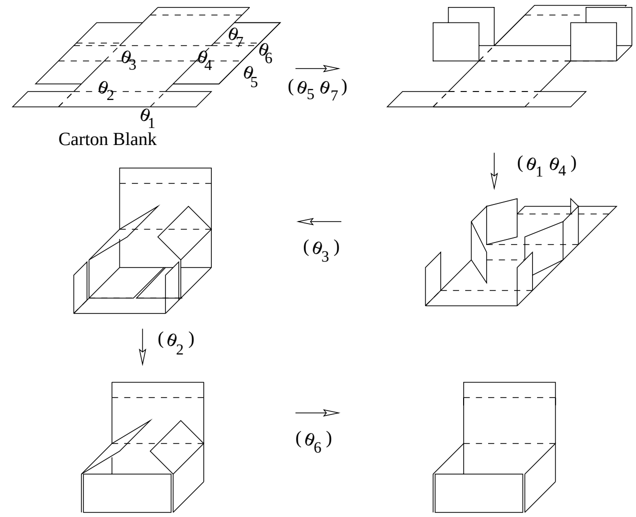

Figure 2: An example folding sequence. Fold creases are indicated by dotted lines.

connected panels. The creases are the desired fold lines for the carton. Carton folding is often performed by humans using their hands both to fold the carton and to maintain the intermediate folded configurations of the carton panels.

A carton can usually be folded to its goal state in multiple ways that differ in the sequence of folds. See Figure 2 for an example folding sequence. We present a method to fold carton blanks into cartons using fixtures. We design the fixture shape to match the carton shape for a selected folding sequence so that the carton is folded along its creases by being translated in contact with the fixture. To make the carton fold at a crease as it translates, we place a *wall* fixture element perpendicular to the carton panel at the crease location. A wall is a flat plane whose length is typically chosen to be slightly greater than the crease length. The wall causes a carton panel to rotate along the crease and maintains the folded shape of the panel. This design procedure is illustrated in Figure 3. For a chosen fold sequence, the distances between walls along the motion direction are selected to ensure the folds occur in the desired sequence.

Multiple folds of the carton can be performed simultaneously by judicious design of the fixture. Making the fixture a passive structure and using an industrial robot to move the carton through the fixture leads to a design with a small number of moving elements. Since a folding sequence specifies only the relative orientations of the carton panels, a given folding sequence can have several instantiations in the world frame, each of which may yield a different folding fixture.

4 Folding Sequence Generation

Identifying a feasible folding sequence is central to the fixture design process. Our approach is to generate all valid folding sequences for a human designer to evaluate. We formulate the generation of folding sequences as a robot motion

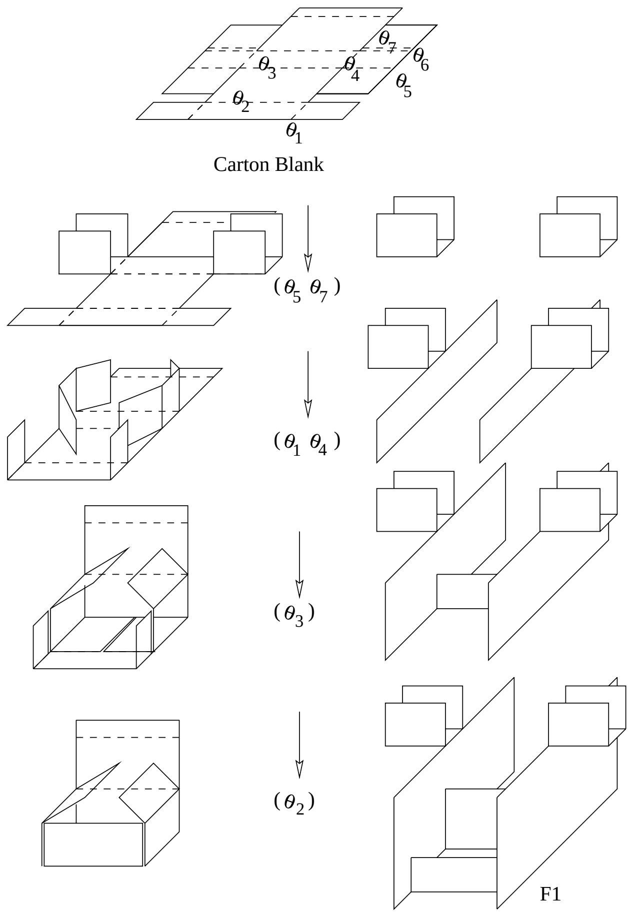

Figure 3: Designing a folding fixture. Creating and modifying fixture elements to perform each fold of the folding sequence for a downward carton translation results in the fixture F1.

planning problem where a carton is an articulated robot with revolute joints and branching sequences of links (Figure 4). We exploit symmetry in the carton shape to reduce the number of robot joints to be considered. The motions of panels on one side of the carton are assumed to mirror those on the other side. Each valid path from the initial configuration to the goal configuration determines a folding sequence. We seek to generate all possible paths, and to subsequently use one of them to design a fixture.

Generating folding sequences is challenging for several reasons. First, a carton robot has a large number of degrees of freedom, ranging from seven to nine for the cartons we studied. Second, a carton robot has branching sequences of links. Third, we seek all possible paths, unlike most motion planning algorithms that look for a single path to the goal.

We can consider the joint motions to move a carton from its unfolded configuration to its folded configuration, or from its folded configuration to its unfolded configuration. Since a folded configuration has more constraints on the permissible joint motions than an unfolded one, it is more efficient to consider unfolding a carton from its folded configuration

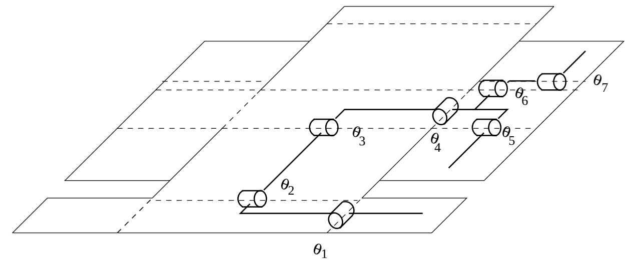

Figure 4: A carton as an articulated robot with revolute joints and a branching sequence of links. Symmetry in the carton's shape makes it sufficient to model only one half of the carton as a robot.

to its unfolded configuration. In what follows, we generate unfolding sequences rather than folding sequences.

4.1 Configuration Space Representation

The *configuration space* $\mathcal{C}$ of an object [10] is the space of all configurations of the object. Each coordinate of the configuration space represents a degree of freedom in the object configuration. The set of joint angles of a robot manipulator define a configuration space. The configurations forbidden to the robot due to collisions with other objects or due to self-collisions constitute the set of configuration space obstacles $\mathcal{C}_{obstacle}$. The set of legal configurations of the robot constitute its free space $\mathcal{C}_{free}$.

The configuration space of a carton with n joints where the ith joint has a lower limit of low_i and an upper limit of $high_i$ is $[low_1, high_1] \times \ldots \times [low_i, high_i] \times \ldots \times [low_n, high_n]$. We use a recursive tree representation of the configuration space of a manipulator with revolute joints, developed by Lozano-Perez [11], to represent the carton's configuration space. This method builds an efficient approximation of the configuration space obstacles from one-dimensional slices of the configuration space. Each level of the tree is a set of cells corresponding to intervals of one joint of the robot. A tree leaf corresponds to a range of configurations of the robot identified as being in $\mathcal{C}_{free}$ or $\mathcal{C}_{obstacle}$. We extend Lozano-Perez's tree representation to handle the branching sequence of links of the carton robot. We select a base panel and compute the tree representation for the longest chain of links. For each branching sequence of links from this main chain, we create additional levels of the tree. The levels for the first branching sequence are attached to the leaf nodes of the tree for the main chain. Each branching chain of links is represented in sequence, and the levels for the branching chains are concatenated. A leaf in this resulting tree represents a configuration interval and encodes whether or not it is in free space.

During fold sequence generation, we assume no external obstacles collide with the carton. Obstacles in the carton configuration space correspond to collisions of the carton panels with each other as the carton is folded, and are detected by rotating each panel and checking for collisions with the stationary panels (Lu [12]).

4.2 A Motion Planning Formulation

We first characterize the paths the carton robot can take in its configuration space when folded by a virtual (uninstantiated) fixture. A joint is unactuated and begins rotating when a panel it is attached to contacts a fixture wall. We assume that all joints move at an angular velocity ω, and that once a joint starts moving, it rotates monotonically to completion. Joints may begin moving simultaneously or with delays in motion. For a robot with n joints, we represent the sequence of joint motions by $(\theta_{i_1}, \theta_{i_2}, \theta_{i_3}, \ldots, \theta_{i_n})$ which means joint θ_{i_1} starts moving before or at the same time as θ_{i_2}, θ_{i_2} starts moving before or at the same time as θ_{i_3}, and so on. For each such *joint sequence*, we also identify the rotational delay δ between successive joints. (Note that we use θ_{i_k} to represent joint i_k as well as its angular value.) A delay δ_k means joint $\theta_{i_{k+1}}$ starts rotating after joint θ_{i_k} has rotated through an angle δ_k. We represent a *delay sequence* corresponding to the joint sequence $(\theta_{i_1}, \theta_{i_2}, \theta_{i_3}, \ldots, \theta_{i_n})$ by $(\delta_1, \delta_2, \ldots, \delta_{n-1})$. For a specified joint sequence and delay sequence, the motion of joint θ_{i_k} from low_{i_k} to $high_{i_k}$ is given in terms of the time parameter t by:

$$\begin{array}{l} \theta_{i_k}(t) = \omega(t - t_k^{start}) + low_{i_k}, \qquad t_k^{start} \leq t \leq t_k^{end} \\ t_k^{start} = \sum_{j=1}^{k-1} \delta_j / \omega \\ t_k^{end} = (high_{i_k} - low_{i_k})/\omega + \sum_{j=1}^{k-1} \delta_j/\omega. \end{array} \tag{1}$$

The dependence of the joint angles of the carton robot on each other, for a specified joint sequence and delay sequence, can be equivalently expressed by the *motion constraints*:

$$\begin{array}{c} \theta_{i_k}(t) = \theta_{i_{k-1}}(t) - \delta_{k-1} - low_{i_{k-1}} + low_{i_k}, \\ t_k^{start} \leq t \leq t_{k-1}^{end},\ k = 2, \ldots, n. \end{array} \tag{2}$$

The *initial configuration constraints* are:

$$\theta_{i_k}(t) = \quad low_{i_k}, \quad t < t_k^{start},\ k = 1, \ldots, n. \tag{3}$$

The *goal configuration constraints* are:

$$\theta_{i_k}(t) = \quad high_{i_k}, \quad t > t_k^{end},\ k = 1, \ldots, n. \tag{4}$$

The delay δ_k can assume any value in the interval $[0, high_{i_k} - low_{i_k}]$. If δ_k is zero, the joints θ_{i_k} and $\theta_{i_{k+1}}$ begin moving simultaneously, and if δ_k is $(high_{i_k} - low_{i_k})$, joint $\theta_{i_{k+1}}$ begins moving only after joint θ_{i_k} has completed its motion. For the cartons we studied, $low_{i_k} = 90$ and $high_{i_k} = 180$ for all k. So the configuration space of the carton robot is $[90, 180]^n$ and $\delta_1, \delta_2, \ldots, \delta_{n-1}$ can assume any value from zero to 90 degrees.

When m of the n joints are in motion, there are $m-1$ active motion constraints and $n-m$ active initial or goal configuration constraints. Each constraint defines a constraint plane in the n-dimensional configuration space of the robot, and the $n-1$ total active constraint planes intersect along a line (Figure 5). So a path consists of a sequence of line segments defined by the above equations. The maximum number of line segments in a path is $2n-1$.

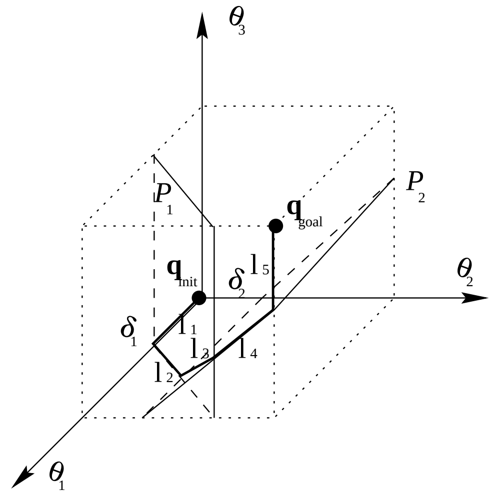

Figure 5: Constraint planes and line segments for a 3-D configuration space with $\mathbf{q}_{init} = (90, 90, 90)$ and $\mathbf{q}_{goal} = (180, 180, 180)$. Path τ, composed of line segments l_1, l_2, l_3, l_4, and l_5, corresponds to joint sequence $(\theta_1, \theta_2, \theta_3)$ and delay sequence (δ_1, δ_2). Planes P_1 and P_2 correspond to the constraints $\theta_2 = \theta_1 - \delta_1$ and $\theta_3 = \theta_2 - \delta_2$ respectively. Line segment l_1 lies along the intersection of planes $\theta_2 = 90$ and $\theta_3 = 90$, and has length δ_1. l_2 lies along the intersection of P_1 and plane $\theta_3 = 90$, and the length of its projection on the θ_2 axis is δ_2. l_3 lies along the intersection of planes P_1 and P_2, l_4 lies along the intersection of P_2 and $\theta_1 = 180$, and l_5 lies along the intersection of $\theta_2 = 180$ and $\theta_1 = 180$.

4.3 Search Algorithm

Our goal is to identify collision-free paths for the carton robot in its configuration space. A path or a folding sequence is defined by a joint sequence and a corresponding delay sequence. A robot with n joints has $n!$ joint sequences. To identify the valid δ ranges for each feasible joint sequence, we discretize the δ ranges into s intervals sampled at their midpoints. So each joint sequence has s^{n-1} possible delay sequences. We must identify the valid folding sequences among the $n!s^{n-1}$ sequences.

The simplest algorithm would be to generate all $n!$ joint sequences, and for each joint sequence, generate the path for each distinct delay sequence. The delay sequences would be generated by looping over the δ values, which determine the path line segments. For each path, we would test if any of its line segments collides with a configuration space obstacle. If no collision occurs, the folding sequence is feasible.

We modify the above algorithm to more efficiently prune infeasible paths. For a given joint sequence, multiple paths to the goal may share a sequence of line segments. If a line segment collides with an obstacle, none of the paths it belongs to are feasible. Every time we select a δ value, we test the defined line segment for a collision. If it does collide, we can classify the corresponding set of paths as infeasible without instantiating the rest of the delay sequence. We thus prune the entire set of paths emanating from the line segment

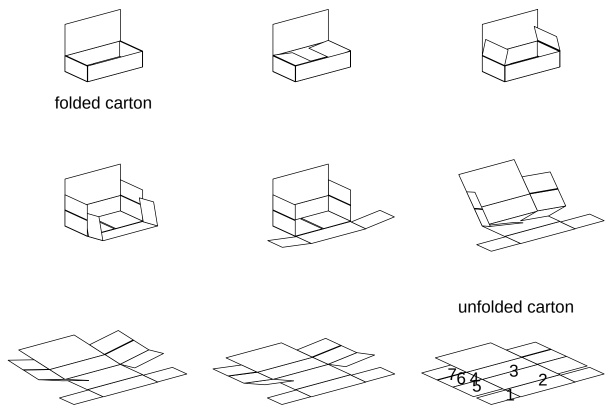

Figure 6: Example (un)folding sequence for the Radio carton with joint sequence $(\theta_6, \theta_7, \theta_2, \theta_1, \theta_3, \theta_4, \theta_5)$ and delay sequence $(90, 90, 90, 90, 0, 10)$, depicted from left to right, top to bottom.

with the collision. Once the elements $\delta_1, \ldots, \delta_{n-2}$ of the delay sequence have been instantiated and the corresponding line segments found to be collision-free, we find valid values for δ_{n-1} by exploiting spatial coherence of the obstacles. If the line segment for some value of δ_{n-1} collides with an obstacle, we test the neighboring cells at the incremented value of δ_{n-1}. If those are collision-free, we test the new line segment for feasibility. Else we increment δ_{n-1} and repeat the process. See Figure 6 for a folding sequence generated by the algorithm.

The worst-case time complexity of the algorithm is exponential in the number of joints. By identifying the set of joints that can move first, we can eliminate some invalid combinations and reduce the search time. We test if a joint can move first by checking if it has a collision-free configuration when rotated by a small amount, keeping all other joints at their start orientations. For each of the joints that can move first, we can identify the set of the joints that can potentially move second. Rapidly identifying the first two joints that can move means we explore only a subset of the $n!$ joint sequences.

4.4 Multiple Angular Velocity Formulation

Since all joints may not rotate at the same angular velocity, we can permit each joint θ_{i_k} to rotate with an angular velocity $\gamma_{i_k}\omega$, $\gamma_{i_k} \geq 1$. For a specified joint sequence and delay sequence, the motion of joint θ_{i_k} is now given by:

$$\begin{aligned} &\theta_{i_k}(t) = \gamma_{i_k}\omega(t - t_k^{start}) + low_{i_k}, \qquad t_k^{start} \leq t \leq t_k^{end} \\ &t_k^{start} = \textstyle\sum_{j=1}^{k-1} \delta_j/\gamma_{i_j}\omega \\ &t_k^{end} = (high_{i_k} - low_{i_k})/\gamma_{i_k}\omega + t_k^{start}. \end{aligned} \tag{5}$$

The initial and goal configuration constraints remain unchanged. The search algorithm is similar to that for equal angular velocities, with the additional consideration that for

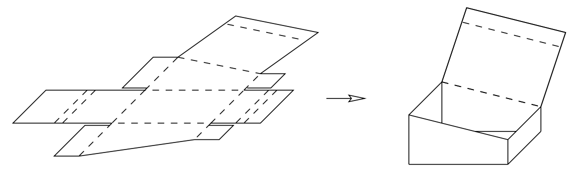

Figure 7: The Slope carton.

each joint sequence we must consider all possible permutations of valid angular velocity instantiations. We discretize the set of allowed angular velocities for each joint. When p joints can each have r angular velocities, the worst-case running time increases by a factor of $O(p^r)$.

4.5 Evaluating the Folding Sequences

The folding sequence planner returns a potentially large number of valid sequences. The human designer has to select a sequence based on its foldability by a fixture. Folding sequence generation can be made interactive by having the designer specify preferred joint sequences to explore. The designer must also consider the degrees of freedom of the actuating robot when selecting a folding sequence for fixture design. Since only the relative orientations of the carton panels in a folding sequence are specified by the motion planner, the sequence may be instantiated in several ways whose foldability on a fixture can differ significantly based on whether the carton can only be translated or both rotated and translated. Thus several different folding fixtures may be designed for a given folding sequence.

5 Carton Styles

The motion planning algorithm can generate folding sequences for any carton style that can be modeled as a robot with revolute joints and branching sequences of links with no kinematic loops. Examples include the Slope carton (Figure 7) and the HP carton (Figure 8). See Figure 9 for a folding sequence for the HP carton. Since the search complexity is exponential in the number of joints, we seek to reduce the number of modeled joints. Modeling a carton as a robot with fewer joints also reduces the number of valid folding sequences to be evaluated. The HP carton has additional symmetry that we use to model it as a serial manipulator, reducing the number of robot joints from 7 to 5. The Slope carton can be modeled as two serial robots with different link dimensions and mirrored joint motions to reduce the number of joints from 9 to 5 for each robot. The configuration space trees for the Slope robots are computed independently and used jointly for path planning.

6 Implementation

We implemented the motion planner to generate folding sequences in C++. We discretized each joint range into 10 intervals. To compute the configuration space tree, the planner took 206 minutes for the Radio carton with 7 joints, 167

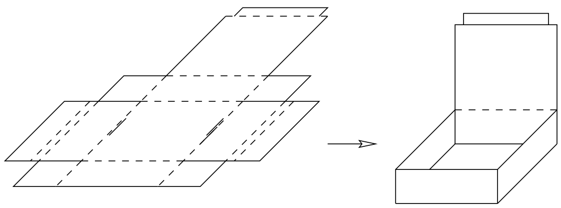

Figure 8: The HP carton.

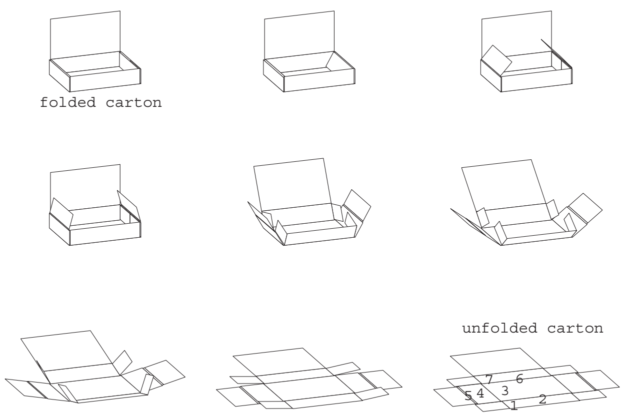

Figure 9: An (un)folding sequence for the HP carton.

minutes for the HP carton with 7 joints, 91 seconds for the HP carton with 5 joints, and 196 seconds for the Slope carton with 5 joints. To generate all folding sequences for the equal angular velocity case, the planner took approximately 28 minutes for the Radio carton with 7 joints, 56 minutes for the HP carton with 7 joints, and 8 seconds each for the HP and Slope cartons with 5 joints. The number of valid joint sequences are 48, 384, 20, and 20 respectively. All computations were on a Sun SPARC 10 with 64 MB of memory.

We demonstrated folding of the Radio carton style using an AdeptOne SCARA robot with an additional fifth joint (Figure 10). The implemented folding sequence includes an intermediate rotation of the carton (Figure 11). We designed a fixture for the folding sequence by simultaneously considering the fixture shape and actuating robot motions. The fixture has two pneumatic actuators to aid the folding process. The robot uses a *holding plate* with vacuum suction cups to pick up the carton. Since the holding plate is dimensioned to enable the desired folds without causing collisions with the fixture, it may need to be interchanged along with the fixture when the carton style or size changes.

We used the same fixture and holding plate for experiments on two carton models that differed slightly in their stiffness and crease locations. One model was successfully folded on 9 of 10 trials while the other was successfully folded on 10 of 16 trials. Failures resulted from undesired folds, noticeable scuffing, accidental collisions of a panel and the holding plate, or incomplete insertion of the self-

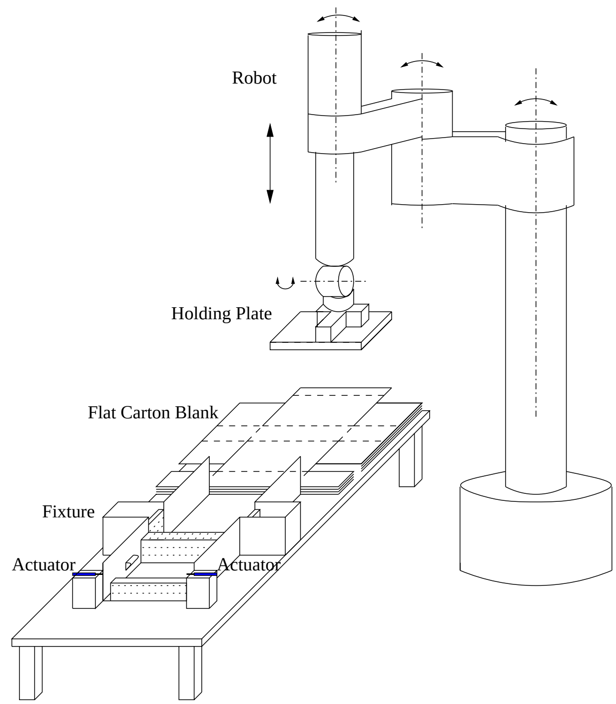

Figure 10: Implemented fixture and AdeptOne robot.

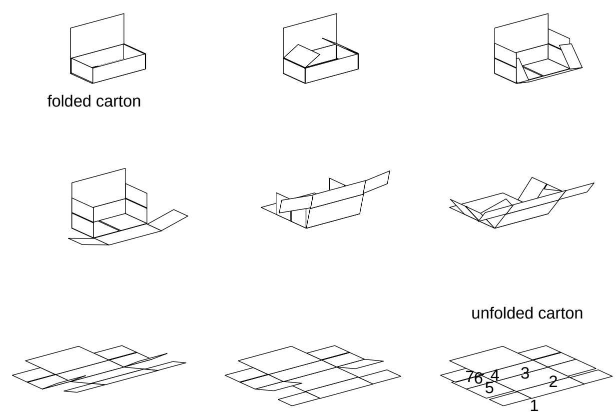

Figure 11: The implemented folding sequence includes a rotation of the carton. The joint sequence is (θ_6, θ_7, θ_2, θ_1, θ_3, θ_4, θ_5), the delay sequence is (0, 90, 90, 70, 20, 0), and the angular velocity of θ_4 is thrice that of all other joints.

locking tabs, and can be eliminated with better fixture fabrication. The robot can fold a carton in about a minute in our current implementation ([13]). Pipelining the folding operations or using multiple actuators can significantly reduce the execution time.

7 Conclusion

We presented a flexible method to fold a class of self-locking cartons using interchangeable fixtures; a carton blank is folded by moving it through a fixture with a robot. We outlined a procedure to design a fixture given a carton and a folding sequence. By modeling cartons as robots with revolute joints and branching links and exploiting constraints on the carton motion imposed by the fixtures, we developed a motion planning algorithm to automatically generate all folding sequences. The planner generated sequences enable a human designer to select fixture-foldable sequences and design corresponding fixtures. Careful selection of the panels to be modeled as a carton robot and exploiting carton symmetry improve planner efficiency significantly. To illustrate the method, we selected a folding sequence for an example carton, designed a fixture, and demonstrated folding of the carton from blanks with an AdeptOne robot. This carton folding method enables rapid changeovers between different carton styles as it requires only a change in the fixture, holding plate, and robot program.

The motion planner can be used interactively by a human designer specifying folding sequences to focus on. Identifying criteria to rank folding sequences and eliminating sequences that are infeasible due to tool constraints would be useful. Interesting problems to be addressed in the future include developing automatic planners to design the fixtures, characterizing the effect of actuator robot degrees of freedom on fixture designs, and designing cartons for efficient folding. This motion planning approach can potentially be extended to sheet metal bending and the design of 3-D microelectromechanical structures (MEMS) created from 2-D hinged elements (Pister *et al.* [17]).

Acknowledgments

Funding was provided by Motorola through the Manufacturing Research Center at the University of Illinois at Urbana-Champaign, and by the Beckman Institute. We thank Juan Nguyen, Swee Mok, and Tom Babin for their guidance. Conversations with Seth Hutchinson and Jean Ponce helped clarify the ideas in this paper.

References

[1] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research*, 10(6):628–649, Dec. 1991.

[2] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, M. Overmars, J. O'Rourke, S. Robbins, and S. Whitesides. Unfolding some classes of orthogonal polyhedra. In *Tenth Canadian Conference on Computational Geometry*, Aug. 1998.

[3] J. F. Canny and K. Y. Goldberg. RISC for industrial robotics: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.

[4] B. Capdeboscq. Sheet folding machine. US Patent 4614512, July 1985.

[5] L. J. de Vin, J. de Vries, A. H. Streppel, E. J. W. Klaassen, and H. J. J. Kals. The generation of bending sequences in a CAPP system for sheet-metal components. *Journal of Materials Processing Technology*, 41:331–339, 1994.

[6] K. Y. Goldberg and H. Moradi. Compiling assembly plans into hard automation. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Apr. 1996.

[7] K. K. Gupta and Z. Guo. Motion planning for many degrees of freedom: Sequential search with backtracking. *IEEE Transactions on Robotics and Automation*, 11(6):897–906, Dec. 1995.

[8] S. K. Gupta, D. A. Bourne, K. H. Kim, and S. S. Krishnan. Automated process planning for sheet metal bending operations. *Journal of Manufacturing Systems*, 17(5):338–360, 1998.

[9] L. E. Kavraki, P. Svetska, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug. 1996.

[10] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, 1991.

[11] T. Lozano-Perez. A simple motion-planning algorithm for general robot manipulators. *IEEE Journal of Robotics and Automation*, RA-3(3):224–238, June 1987.

[12] L. Lu. Folding cartons with fixtures. Master's thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1999. Forthcoming.

[13] L. Lu and S. Akella. Folding cartons with fixtures. In *Video Proceedings, IEEE International Conference on Robotics and Automation*, Detroit, MI, May 1999.

[14] A. Lubiw and J. O'Rourke. When can a polygon fold to a polytope? In *AMS Conference*, Lawrenceville, NJ, Oct. 1996.

[15] C. R. Marschke. Folding of paperboard sheets and the like. US Patent 4834696, May 1989.

[16] J. McBride and R. Lile. Corner laminating apparatus and method for cartons. US Patent 4624653, Nov. 1986.

[17] K. S. J. Pister, M. W. Judy, S. R. Burgett, and R. S. Fearing. Microfabricated hinges. *Sensors and Actuators A*, 33:249–256, 1992.

[18] B. Radin, M. Shpitalni, and I. Hartman. Two-stage algorithm for determination of the bending sequence in sheet metal products. *ASME Journal of Mechanical Design*, 119:259–266, 1997.

[19] Dorell Equipment Inc. Automatic tray and box former BEM-102. North Brunswick, NJ, 1999.

[20] C.-H. Wang. *Manufacturability-Driven Decomposition of Sheet Metal Products*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Sept. 1997. Robotics Institute Technical Report CMU-RI-TR-97-35.

[21] H. D. Ward. Box blank folding apparatus. US Patent 4295841, Oct. 1981.

[22] R. H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2):371–396, Dec. 1994.