Metal Wire Manipulation Planning for 3D Curving — How a Low Payload Robot Can Use a Bending Machine to Bend Stiff Metal Wire

Ruishuang Liu\textsuperscript{1}, Weiwei Wan\textsuperscript{1,*}, Emiko Isomura\textsuperscript{2}, and Kensuke Harada\textsuperscript{1,3}

Abstract—This paper presents a combined task and motion planner for a robot arm to carry out 3D metal wire curving tasks by collaborating with a bending machine. We assume a collaborative robot that is safe to work in a human environment but has a weak payload to bend objects with large stiffness, and developed a combined planner for the robot to use a bending machine. Our method converts a 3D curve to a bending set and generates the feasible bending sequence, machine usage, robotic grasp poses, and pick-and-place arm motion considering the combined task and motion level constraints. Compared with previous deformable linear object shaping work that relied on forces provided by robotic arms, the proposed method is suitable for the material with high stiffness. We evaluate the system using different tasks. The results show that the proposed system is flexible and robust to generate robotic motion to corporate with the designed bending machine.

I. INTRODUCTION

This paper presents a combined task and motion planning method for a robot to curve metal wires into 3D shapes. We assume a collaborative robot that is safe to work in a human environment. Such a robot cannot provide enough force to bend objects with large stiffness. To solve the problem, we designed a bending machine using a stepper motion and a 1:80 gear set. The bending machine can provide 240Nm bending force to bend stiff objects. We develop a combined task and motion planner to plan the bending sequence of the bending machine at the task level and the robot’s grasping poses and joint trajectories at the motion level. The planner enables the robot to use the bending machine like a professional human worker and curve stiff and various-shaped metal wires without manually defined rules or programming routines.

Previously, using robots to curve metal wires was widely studied, as robots provided a promising solution to flexible or personalized manufacturing of metal-wire products. Existing robotic methods use robots as independent machines. They program a robot to hold a metal wire and bend it by pressing against a fixture or an external robotic gripper. The payload of the robot limits the maximally affordable metal stiffness. Unlike the existing methods, we ask a robot to collaborate with an external folding machine. With the support of our proposed combined task and motion planning method, the robot can automatically determine the folding sequence, the grasping poses, and the robot joint trajectories to conduct 3D curving of stiff metal wires.

Fig. 1 shows an example of our system in action. In the first step, we use polygonal approximation to represent a desired 3D curve as a set of bending candidates. Then, we perform a combined task and motion planning to simultaneously generate the bending sequence and machine usage, the grasping poses, and the pick-and-place motion of the robot. The robot will follow the planned results to use the bending machine and bend metal wire.

Our main contribution is the close-loop exploration considering combined task and motion level constraints. At the task level, we study the geometric constraints of the bending machine and the changing shapes of the metal wire. These constraints help us to determine several tentative bending sequences. At the motion level, we study the logical relations of robotic grasping poses and the availability of robot joint trajectories that move the metal wire from a previous bending pose to the next one. The motion level results will be recorded during exploration to prune the task level tentative sequences and improve planning speed. They together make the impossible exploded combinatorics tractable.

In the experimental section, we carry out both simulations and real-world experiments to study the performance of the proposed method. The results show that our method can

1Department of System Innovation, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan.
2First Department of Oral and Maxillofacial Surgery, Graduate School of Dentistry, Osaka University, Japan.
3National Inst. of AIST, Japan.
*Contact: Weiwei Wan, wan@sys.es.osaka-u.ac.jp
find a solution for metal wire with less than eight bending candidates in two minutes if a solution exists. The developed robot system can curve different 3D shapes in the real world with satisfying performance.

II. RELATED WORK

A. Manipulation of Deformable Linear Objects

Robotics research communities have paid considerable attention to the problem of the automatic manipulation and shaping of deformable linear objects. The objective is to reshape an object into the desired one using a mechanical system. Different from a rigid object, a deformable object is more complicated due to its infinite degree of freedom. The large configuration space of deformable objects makes it difficult to model using traditional approaches [1]. Also, since the object shape changes while being manipulated, additional methods need to be developed to track the deformation and conduct online collision checking during planning.

State-of-the-art deformable objects studies in robotic manipulation literature can be separated according to their physical properties (with and without compression strength) and geometric properties (linear, planar, and solid) [2]. Especially for DLOs, they are further classified into five categories in [3] considering its deformable type (plastic or elastic) and whether force less than gravity can result in deformation.

This paper focuses on elastoplastic DLOs (e.g. metal wire, rod). Exemplary studies on this category are as follows. Jin et al. [4] designed an automation device to perform the archwire bending task. The device could form several desired shapes by predefined processes. Xia et al. [5] achieved the same task with a robotic arm and an external gripping unit. A sample-based planner is applied for bending path generation. More recently, Laezza et al. [6] proposed a reinforcement learning-based elastoplastic DLOs shape control method. When the material’s stiffness is high, it will be difficult for general collaborative robots to reshape. Solutions include Lu et al. [7], which leveraged a heating technique in cooperation with an industrial robotic arm to soften shape metal rods, and Zhang et al. [8], which designed a special multi-axis special-purpose machine to form titanium alloy strip used in oral and maxillofacial surgery.

Unlike the studies above, we propose using a collaborative robot arm and a bending machine to achieve the curving task. Our difference and novelty are two-fold. First, the proposed system can cope with high stiffness metal wire, as bending forces do not directly affect the robot arm. Second, as the metal wire is picked and held by the robot arm rather than fixed on the machine or special end-effectors, diverse bending sequences are allowed to support combined task and motion planning.

B. Combined Task and Motion Planning

Combined exploration and planning considering the mutual influence of task and motion level constraints is the key technique for efficiently generating robotic manipulation motion using the bending machine and bending metal wire. The method extends combined task and motion planning, which was used to generate action sequences at a high level and motion for specific actions at a low level. While combined task and motion planning has been conventionally studied for decades, recent literature attempts to extend it with uncertainty, force constraints, and deeply learned heuristics. For example, Holladay et al. [9] extended an existing task and motion planner with controllers that exert wrenches and constraints while considering torque and frictional limits to achieve Forceful Manipulation. Silver et al. [10] proposed a bottom-up relational learning method for operator learning and demonstrated how the learned operators could be used for planning in a TAMP system. Previously in our group, we also adopted the idea of combined task and motion planning for various robotic manipulation tasks. Chen et al. [11] developed a planner that automatically finds an optimal assembly sequence for a dual-arm robot to build a woodblock structure while considering various constraints and supporting grasps from a second hand. Wan et al. [12] presented a combined planner to solve a test tube arrangement problem.

In this work, we use similar combined task and motion planning methods to get bending sequence considering constraints of the bending machine, reason the grasp poses for metal wire, and generate manipulation motion as Chen et al. [11]. We pay special attention to the geometric constraints of the bending machine, the changing shapes of the metal wire, and the kinematic constraints of the robot. We determine tentative bending sequences and prune them considering the mutual influences of the constraints.

III. PRELIMINARIES

A. Structure of the Bending Machine

We designed a bending machine shown in Fig. 2 by following the mechanism of commercial metal plate benders or folders. The folding machine comprises a 3Nm stepping motor, a 1:80 gearbox to increase the output torque, an optical sensor as a photo-interrupter for calibrating a zero position, and several rollers for pressing metal wires. A metal wire is assumed to be placed between the center roller and the other two rollers. The punch roller can rotate clockwise and counterclockwise by starting from the calibrated zero
position. Thus the machine allows bending from two directions without reversing the plate. The effective work range of the machine is shown in Fig. 3(a). Its maximum bending angle for a zero-thickness metal wire will be the green zone of the figure.

![Image](image_url)

**Fig. 3:** (a) Effective work range of the bending machine assuming zero-thickness metal wire. (b) One wire shape pressed by the punch roller. (c) If the punch roller cannot touch the metal wire, the bending action is reported as infeasible. For simplicity, the figures are illustrated in a 2D top view. They are actually three dimensional as shown in Fig. 4 and 5.

The actual bending angle depends on the shape and thickness of the metal wire. It is essentially smaller than the maximum bending angle. In an ideal case where the metal wire is straight, the bending angle can be computed directly considering the punch roller’s rotating angle and wire thickness. It is difficult to figure out an explicit equation to formulate the bending in more common cases. Thus instead of modeling the bending process, we develop a kinematic simulator to examine the collision between the wire and the environment (including robots, other parts of the bending machine, work table, etc.) and determine the next curve shape after the current bending action.

We have the following assumptions for the kinematic simulator. First, we assume the bending force from the bending machine is large enough to cause inelastic deformation. No kinetic energy will be accumulated and released to cause unexpected results. Second, the wire begins to bend when the punch roller starts to collide with it and stops bending when the punch roller reaches a limit, or the wire collides with the environment. Third, we assume the bending arc is clamping to the center roller during bending. The wire will wind around the center roller to form a circular arc. Fourth, if two bending actions overlapped, we assume the newer circular arc overrides the former arc sections. Under these assumptions, we can compute the next shapes after each bending by replacing related wire sections with circular arcs determined by collisions. Note that although we use this design to present our combined task and motion planning implementation, our method is not limited to this specific machine. It can be applied to other bending mechanisms [13][14].

B. Representing a 3D Curve Using a Bending Set

We formulate the bending problem considering the bending machine as a searching problem. We simplify a given desired 3D curve using polygonal approximation as a first step. We assume a desired 3D curve is initially given as a list of dense points. This representation includes many delicate curvatures that our bending machine cannot bend. Instead of directly using these dense points, we approximate them by line segments using the Ramer-Douglas-Peucker algorithm [15], while considering an error tolerance ε.

Fig. 4 illustrates a example of polygonal representation. Here, the smooth black curve is the originally desired one. The orange polygonal curve is the approximated result. The approximation results into several pivoting points named \( p_{i-1}, p_i, p_{i+1}, \ldots \) and their normals \( n_{i-1}, n_i, n_{i+1}, \ldots \), since the curve is a spatial one. By further considering the radius of the center roller \( r_c \), we can compute a bending set \( B = \{ B_i \} \) where each element in it indicates a bending candidate represented as \( B_i = \{ q_i, \theta_i, \alpha_i, \beta_i \} \). Here, \( q_i \), \( \theta_i \), \( \alpha_i \), and \( \beta_i \) are respectively the starting point of the bending, the wire’s bending angle, the wire’s twisting angle, and the wire’s lifting angle at this point. These parameters are described in the local coordinate system of its preceding bending point \( B_{i-1} \), as shown by Fig. 5. The definition of \( q_i \) is graphically illustrated in the dashed box of Fig. 5. The metal wire begins to wind around the center roller starting from this point. The three angles are illustrated in both the dashed box of Fig. 4 and Fig. 5. They represent how to rotate the metal wire to reach a desired local shape.

![Image](image_url)

**Fig. 4:** Polygonal approximation of a desired curve. The polygonal approximation leads to a set of bending candidates defined as \( \{ q_i, \theta_i, \alpha_i, \beta_i \} \). They are illustrated in the dashed box. Please also see Fig. 5 for details of the \( \theta_i, \alpha_i, \beta_i \) angles.

Note that since the metal wire is a cylinder and central symmetric, a local shape essentially has two degrees of freedom and can be defined by two parameters. Thus it may seem unnecessary to use both the \( \alpha \) and \( \beta \) angles to represent the \( B_i \). This understanding is true from the viewpoint of the final shape, but choosing \( \alpha \) or \( \beta \) will lead to different wire motions during bending. The different wire motions could be helpful to avoid collisions. Thus, instead of using a single angle parameter, we use both of them \( B_i \)’s definition. Considering environmental constraints, the combined task
Three angles used to represent a bending state. They are defined in the local coordinate system of a previous bend. Since the metal wire is central symmetric, using both $\alpha$ and $\beta$ are redundant. The combined task and motion planner will determine which one to use considering environmental constraints.

and motion planner will determine which one to use.

Using the radius of the center roller $r_c$, the values obtained from polygonal approximation, i.e. $p_0$, $p_1$, ..., and their normals $n_0$, $n_1$, ..., the parameters of $B_i$ can be computed as follows.

$$
\theta_i = \arccos \left( \frac{1}{|\vec{p}_{i+1} - \vec{p}_i|} \cdot \vec{p}_{i+1} \times \vec{n}_{i+1} \right) 
$$

$$
q_i = p_i - r_c \cdot \tan \left( \frac{\theta_i}{2} \right).
$$

$$
\alpha_i = \arccos \left( \frac{n_{i+1} \times \vec{p}_{i+1} - \vec{p}_{i+1} \times \vec{p}_i}{|\vec{p}_{i+1} - \vec{p}_i|} \cdot \vec{p}_{i+1} \times \vec{n}_{i+1} \right) 
$$

$$
\beta_i = \arccos \left( \frac{n_{i+1} \times \vec{p}_{i+1} - \vec{p}_{i+1} \times \vec{p}_i}{|\vec{p}_{i+1} - \vec{p}_i|} \cdot \vec{p}_{i+1} \times \vec{n}_{i+1} \right) 
$$

Here, $B_i$ means projecting $B$ onto $A$. The $A$ is replaced by $\Sigma_{B_{i-1}}$, $\Sigma_{B_{i+1}}$, $\Sigma_{B_{i-1}}$, in equations (1), (4), and (6) respectively. They indicate projecting on to the $xy$, $yz$, $xz$ planes of the $\Sigma_{B_{i-1}}$ coordinate system.

The overview workflow of our planner, with a particular focus on the combined task and motion planning part, is shown in Fig. 6. First, it takes the desired shape as input and down-sample it to many key points. Then it generates a bending sequence considering the collision constraints by incrementally permuting the bending order and pruning the searching tree. Finally, together with the pre-annotated grasp poses of metal wire primitive, the workflow will produce a sequence of kinematic metal wire poses and send them to the motion planner to determine grasp poses and plan manipulation motion. It will iterate through the pre-annotated grasp poses to find the candidates that can finish all the motions. The essential combined task and motion planning component is illustrated in the gray region. Its details will be presented in the following subsections.

A. Bending Sequence Planning

A feasible bending sequence should fulfill the following constraints: (i) The wire does not collide with the machine or any other obstacles before and after bending. (ii) The flange is long enough to contact the die roller and punch roller before and after bending. If the metal wire is unable to touch either of the rollers, the bending candidate $B_j$ pertaining to the current shape of the metal wire will be reported as infeasible as shown in Fig. 3(c). Determining such a sequence is difficult as a set of $n$ bends can be arranged in $n!$ ways. Thus, instead of brutally evaluating all permutations, we leverage pruning to accelerate searching speed. The algorithm is shown in Algorithm 1. The sequence is initialized with $s=0, 1, ..., n$. The algorithm will explore the bending candidates one by one. If collision or infeasible bending appears at $B_j$, the remaining actions in the sequence no longer need to be explored. The failed sequence $s[j]$ will be stored in a tree structure $t$. A new sequence not in $t$ will be generated by Depth First Search (DFS), and the next iteration will be examined, and failures will be stored similarly until a feasible sequence is found. The time efficiency of the pruned search will be detailed in Section V.
Algorithm 1: Prune Search

Input: \( B = B_0, B_1, ..., B_n \), a bending set; 
\( t \), invalid sequence tree; 
\( s = [0, 1, ..., n] \), a bending sequence;

1 \hspace{1em} \text{begin} \\
2 \hspace{1em} \text{while} \ s \ \text{is not empty} \ \text{do} \\
3 \hspace{2em} \text{if} \ s \ \text{is feasible} \ \text{then} \\
4 \hspace{3em} \text{Grasp reasoning and motion planning} \\
5 \hspace{3em} \text{if} \ success \ \text{then} \ \text{return} \ s \\
6 \hspace{2em} \text{update}(t, s) \\
7 \hspace{2em} s \leftarrow \text{Find a new sequence not in} \ t \\
8 \hspace{1em} \text{return} \ False \\
\text{end}

Fig. 7: Workflow of grasp reasoning and motion planning. The inputs are shown in dashed boxes on the left. The results are in the solid boxes on the right. (a) Pre-annotated grasps. (b) Planned bending sequence. (c) Example of grasp reasoning. (d) Example of motion planning result.

B. Grasp Reasoning and Motion Planning

This subsection uses a simple example with two bends to present the grasp reasoning and motion planning. The input of this step is: (1) Pre-annotated grasp poses for the metal wire; (2) Wire poses generated by sequence planning. The input will be used to reason IK-feasible and collision-free grasps. Fig. 7(a) shows an example of grasp reasoning. The collided grasps are shown in red, the IK-infeasible grasps are shown in yellow, and the available grasps are shown in green. The grasps available for all bends are named “common grasps”. A robot can hold the wire using a “common grasp” at one pose and move it to another without regrasp. After obtaining the common grasps, the motion planning will iterate through them to plan the joint space motion for moving a wire to the pre-defined poses. Fig. 7(b) exemplifies a planned motion.

V. EXPERIMENTS AND ANALYSIS

Our experimental platform includes a PC with an Intel Core i7-10700 CPU and 32 GB memory, and a UR3e robotic arm equipped with a Robotiq Hand-E two-finger parallel gripper. A prototype of the bending machine was previously shown in Fig. 1. It is fixed on a work table under an UR3e collaborative robotic arm (Max payload: 3kg).

A. Sequence Planning

In this subsection, we demonstrate some planning results and examine the time efficiency of the sequence planner. The experiments were carried out by randomly generating bending sets and planning their order using the developed method. Especially, we randomized the parameters bending angle \( \theta_i \in [\pi, \pi] \), twisting angle \( \alpha_i \in [-\pi, \pi] \) and lift angle

| # of Bending Candidates | 3 | 4 | 5 | 6 | 7 |
|-------------------------|---|---|---|---|---|
| Success Rate            | 6/10 | 6/10 | 5/10 | 7/10 | 5/10 |

Fig. 8: Statistical performance of sequence planning. Upper Table: Success rates under different number of bending candidates. Lower Graphs: Time costs.

Fig. 9: One exemplary result planned using our method.
\[ \beta_i \in [-\pi/18, \pi/18] \]. The start position of each bending candidate, \( q_i \), was set at a random position that did not overlap with others. Five groups of curves with the number of bending candidates in approximated bending sets ranging from 3 to 8 were evaluated. Each group included ten random desired curves.

Fig. 8 shows the detailed statistic performance. The upper table shows the success rate of our planner with a changing number of bending candidates. The lower two graphs show the time cost of success and failure cases separately. The time cost increases with the number of bends, and this tendency is significant in failed cases.

Fig. 9 shows the result of a 3D shape with five bending candidates. The approximated bending set is illustrated in (a). One feasible bending sequence is illustrated in (b). Key poses of a robot action sequence are illustrated in (c).

B. Performance of Real-World Executions

We also carried out real-world experiments to check the final resultant bending. The goal here is to compare the desired shape and the bending result. The desired shapes include a 2D and a 3D polygon with a side length of 50mm. The metal wire used is galvanized steel wire with 1.6mm (red) and 2.6mm (yellow) diameters. Fig. 10 shows the real-world results. The goal shapes are visualized in the first column. The planned robot motion and final result are shown on the left. The details of executions are shown in a supplementary video.

VI. Conclusions and Future Work

We presented a combined task and motion planning-based planner for a robot arm bend metal wire by collaborating with a bending machine. It enables a low payload robot arm to curve a metal wire with high stiffness. Currently, we treat the planning problem hierarchically and solve each subproblem using a specially designed planner. In the future, we are interested in leveraging reinforcement learning to solve them together in a single shot to improve efficiency.

REFERENCES

[1] V. E. Arriola-Rios, P. Guler et al., “Modeling of deformable objects for robotic manipulation: A tutorial and review,” Frontiers in Robotics and AI, vol. 7, p. 82, 2020.

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzagrou, and Y. Mezouar, “Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey,” The International Journal of Robotics Research, vol. 37, no. 7, pp. 688–716, 2018.

[3] D. Henrich, T. Ogasawara, and H. Worn, “Manipulating deformable linear objects-contact states and point contacts,” in International Symposium on Assembly and Task Planning, 1999, pp. 198–204.

[4] J. Jin-gang, Z. Yong-de, J. Ming-liang, and W. Chun-ge, “Bending process analysis and structure design of orthodontic archwire bending robot,” International Journal of Smart Home, vol. 7, no. 5, pp. 345–352, 2013.

[5] Z. Xia, H. Deng, S. Weng, Y. Gan, J. Xiong, and H. Wang, “Development of a robotic system for orthodontic archwire bending,” in IEEE International Conference on Robotics and Automation, 2016, pp. 730–735.

[6] R. Laezza and Y. Karayiannidis, “Learning shape control of elastoplastic deformable linear objects,” in IEEE International Conference on Robotics and Automation, 2021, pp. 4438–4444.

[7] Y.-H. Lu, S.-Y. Wang, Y.-T. Sheng, C.-W. Lin, Y.-H. Pang, and W.-T. Hung, “Transient materialization–robotic metal curving,” 2020.

[8] J. Zhang, B. Shi, G. Feng, G. Zuo, and Y. Liang, “Design, development and control of a forming robot for an internally fixed titanium alloy strip,” Machines, vol. 10, no. 2, p. 68, 2022.

[9] R. Holladay, T. Lozano-Pérez, and A. Rodriguez, “Planning for multistage forceful manipulation,” in IEEE International Conference on Robotics and Automation, 2021, pp. 6556–6562.

[10] T. Silver, R. Chittnis, J. Tenenbaum, L. P. Kaelbling, and T. Lozano-Pérez, “Learning symbolic operators for task and motion planning,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2021, pp. 3182–3189.

[11] H. Chen, W. Wan et al., “Planning to build block structures with unstable intermediate states using two manipulators,” IEEE Transactions on Automation Science and Engineering, pp. 1–17, 2021.

[12] W. Wan, T. Kotaka, and K. Harada, “Arranging test tubes in racks using combined task and motion planning,” Robotics and Autonomous Systems, vol. 147, p. 103918, 2022.

[13] T. Ito, R. A. Hamid, and T. Ichikawa, “Collaborative design and manufacturing of prosthetics wire clasp,” in Transdisciplinary Engineering: Crossing Boundaries. IOS Press, 2016, pp. 421–428.

[14] R. A. Hamid and T. Ito, “3d prosthetics wire bending mechanism with a linear segmentation algorithm,” Journal of Advanced Manufacturing Technology, pp. 33–46, 2016.

[15] U. Ramer, “An iterative procedure for the polygonal approximation of plane curves,” Computer graphics and image processing, vol. 1, no. 3, pp. 244–256, 1972.