mpcMech: Multi-Point Conjugation Mechanisms

KE CHEN, University of Science and Technology of China, China
SIQI LI, Singapore University of Technology and Design, Singapore
PENG SONG, Singapore University of Technology and Design, Singapore
JIANMIN ZHENG, Nanyang Technological University, Singapore
LIGANG LIU, University of Science and Technology of China, China

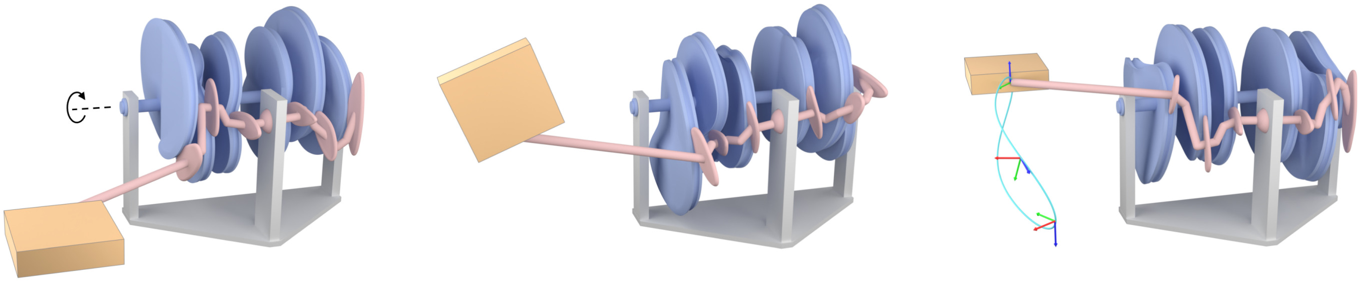


Fig. 1. We present a computational approach for modeling a multi-point conjugation mechanism capable of generating a user-specified 3D motion, e.g., for manipulating a box, driven by a single actuator. Our mechanism features a simple topology comprising only two moving parts, modeled as a pair of conjugate surfaces with multiple conjugation points.

A mechanism is an assembly of moving parts interconnected by joints to transfer an input motion to a desired output motion. Traditionally, to generate a complex motion, mechanisms are modeled by selecting and combining a number of mechanical parts with simple shapes such as links, gears, and cams. Combining multiple mechanical parts results in a mechanism with an intricate topology, which not only complicates assembly and maintenance but also deteriorates the functionality of generating motions due to accumulation of manufacturing imprecisions. To get rid of these limitations, we study mechanisms with *a single pair* of moving parts for generating complex motions. We model the pair of moving parts as a pair of conjugate surfaces with multiple conjugation points, forming a *multi-point conjugation mechanism*.

To study this new mechanism, we establish a connection between conjugate surface pairs and form-closure grasps to formulate a *dynamic form closure* condition under which one conjugate surface is able to continuously transfer the motion to the other conjugate surface by utilizing multiple conjugation points. Guided by the condition, we propose an optimization-based approach to model the geometry of a multi-point conjugation mechanism for exactly generating a user-specified motion, in 1-, 2-, or 3-DOF motion space. The core of our approach is to model multiple conjugate curve pairs that satisfy various requirements in multi-point conjugation, dynamic form closure, and surface fabricability. We demonstrate the effectiveness of our approach by modeling different classes of multi-point conjugation mechanisms to generate various motions, evaluating the mechanisms' kinematic performance with 3D printed prototypes, and presenting three applications of these mechanisms.

CCS Concepts: • **Computing methodologies** → *Shape modeling*.

Additional Key Words and Phrases: mechanism modeling, motion generation, conjugate surface pair, form closure

ACM Reference Format:
Ke Chen, Siqi Li, Peng Song, Jianmin Zheng, and Ligang Liu. 2024. mpcMech: Multi-Point Conjugation Mechanisms. *ACM Trans. Graph.* 43, 6, Article 211 (December 2024), 14 pages. https://doi.org/10.1145/3687770

1 INTRODUCTION

A mechanism is an assembly of moving parts interconnected by joints to transfer an input motion, usually a rotary motion from an actuator, to a desired output motion. Elemental mechanisms can only generate simple motions; e.g., a cam-follower mechanism generates a linear motion. To generate a complex motion, a conventional approach is to model a composite mechanism by connecting multiple elemental mechanisms including linkages, cam-followers, and gear pairs [Gatti and Mundo 2007; Mckinley et al. 2005; Takahashi and Okuno 2018]. These composite mechanisms usually have an intricate topology, complicating their assembly and maintenance. Moreover, it is reasonable to assume that the mechanism's intricate topology would deteriorate its functionality of generating motions due to accumulation of manufacturing imprecision of each component part.

Authors' addresses: Ke Chen, University of Science and Technology of China, China, ckck@mail.ustc.edu.cn; Siqi Li, Singapore University of Technology and Design, Singapore, siqi_li@mymail.sutd.edu.sg; Peng Song, Singapore University of Technology and Design, Singapore, peng_song@sutd.edu.sg; Jianmin Zheng, Nanyang Technological University, Singapore, asjmzheng@ntu.edu.sg; Ligang Liu, University of Science and Technology of China, China, lgliu@ustc.edu.cn.

0730-0301/2024/12-ART211 $15.00
https://doi.org/10.1145/3687770

To overcome the above limitations of conventional mechanisms, an emerging approach is to model a mechanism with a *simple topology* yet still able to transfer *complex motions*, where the core idea is to encode the complex motion into *freeform geometry* of the mechanism. Thanks to the recent advancement in additive manufacturing, mechanisms with freeform geometry can be fabricated with 3D printing conveniently and inexpensively. Yet, modeling mechanisms with freeform geometry to generate a user-specified motion remains a challenging task. Recently, a few research works attempted to address this challenge by generalizing geometry of 3D cam-follower mechanisms [Cheng et al. 2021] and gear pairs [Hou and Lin 2020; Hu et al. 2021]. However, these mechanisms can only generate a motion with limited complexity such as motions in 2-DOF motion space for realizing a path on a spherical surface.

In this paper, we study the modeling of mechanisms with freeform geometry for 3D motion generation from a new perspective in terms of *conjugate surface pairs*. According to the theory of conjugate surfaces [Chen 1978], a pair of mechanical parts is a pair of conjugate surfaces that always contact each other to transfer relative motion between them. The contact entity between a pair of conjugate surfaces may be a point, a line segment, or a surface patch. Conventional mechanisms principally rely on line conjugation (e.g., gear pair) and surface conjugation (e.g., revolute joint) to transfer motions. Very little attention is paid to the point conjugation for modeling man-made mechanisms although it has been encountered in human/animal skeletons (i.e., bio-mechanisms) [Chen and Chen 2003]. To transfer a complex motion, our idea is to use *multiple conjugation points* between a conjugate surface pair, forming a *multi-point conjugation mechanisms* (mpcMech). By coordinating the instantaneous contact positions and normals of the multiple conjugation points, one conjugate surface is able to continuously transfer its motion to the other conjugate surface to achieve a desired output motion.

However, it is non-trivial to model multi-point conjugation mechanisms to generate a user-specified motion. First, the existing theory of conjugate surfaces [Chen 1978] is not sufficient to guide the modeling of mpcMechs since it only provides fundamental requirements of point conjugation but not the condition under which one conjugate surface is able to continuously transfer the motion to the other conjugate surface. Second, a conjugate surface pair should satisfy various requirements in order to form a working 2-moving-part mechanism, including multi-point conjugation, continuous motion transfer, and fabricable surface. To address these challenges, we make the following contributions:

- We develop a *dynamic form closure* [1] condition under which one conjugate surface is able to continuously transfer the motion to the other conjugate surface by utilizing multiple conjugation points. This new condition is achieved by establishing a connection between conjugate surface pairs and form-closure grasps, laying a foundation for modeling multi-point conjugation mechanisms.
- We propose an optimization-based approach to model the geometry of a multi-point conjugation mechanism for exactly generating a user-specified motion in N-DOF, $N \in [1, 3]$ motion space. The core of the approach is to model multiple pairs of conjugate curves that satisfy various requirements in multi-point conjugation, dynamic form closure, surface fabricability, and motion generation.

We show that our mpcMech is able to exactly generate a wide variety of motions specified by users, including those in 3-DOF motion space. To the best of our knowledge, our mpcMech is the first 2-moving-part mechanism that is able to generate motions in motion space beyond 2-DOF. We have evaluated the kinematic performance of our modeled mpcMechs with 3D printed prototypes, and demonstrated the usefulness of these mechanisms with three applications.

2 RELATED WORK

Mechanism design. The goal of mechanism design is to transfer an input motion from a single actuator to a desired target motion, represented by a prescribed set of sequential rigid-body poses or a prescribed trajectory. To generate complex motions, the conventional approach is to select and combine multiple mechanical parts with simple shapes, e.g., a linkage with a number of rigid [Bächer et al. 2015; Megaro et al. 2014; Thomaszewski et al. 2014] or compliant joints [Megaro et al. 2017; Tang et al. 2020; Xu et al. 2018], a combination of linkages with planar cams [Gatti and Mundo 2007; Mundo et al. 2006; Takahashi and Okuno 2018], a combination of linkages with circular [Ceylan et al. 2013; Roussel et al. 2018] or non-circular [Coros et al. 2013; Mundo et al. 2009] gears, and a combination of linkages with belts [Liu and McCarthy 2017]. The combination of multiple mechanical parts with simple shapes usually results in an intricate topology of the mechanism, which has a negative effect on the assembly, usage, and maintenance of the mechanism.

Emerging mechanisms. An emerging approach to overcoming the above limitations of conventional mechanisms is to encode the target motion into the freeform geometry of a mechanism with 2 or 3 moving parts. Hou and Lin [2020] studied oval non-circular bevel gears that generate a compound motion of rotation and axial translation. Hu et al. [2021] studied the curve-face gear pair for generating a spatial finite helical motion with a variable transmission ratio. Abe et al. [2021] proposed spherical gears in which the 3-DOF rotational motion of a cross spherical gear is driven by two monopole gears' 2-DOF rotational motions. Cheng et al. [2021] proposed a 3D cam-follower mechanism for exact 2D path generation (e.g., on a spherical surface), where the target path is encoded into the freeform geometry of a ball-move-in-groove joint on the 3D cam. This mechanism is later combined with a 5-bar spatial linkage for exact 3D path generation [Cheng et al. 2022] .

The geometry of the above 2-moving-part mechanisms [Cheng et al. 2021; Hou and Lin 2020; Hu et al. 2021] is specifically modeled to generate motions in 2-DOF motion space. To the best of our knowledge, we do not find any existing 2-moving-part mechanism that is able to generate motions in motion space beyond 2 DOFs. In

[1] We use the word "dynamic" to highlight the dynamic change of contact points between a pair of conjugate surfaces for transferring motion over time and to differentiate our condition from the well-known (static) form closure condition in grasping.

this paper, we propose a new 2-moving-part mechanism called multi-point conjugation mechanism that is able to generate a variety of motions, including those in 1-, 2-, 3-DOF motion space; see Figure 10. We model the two moving parts in the mechanism as a conjugate surface pair with multiple conjugation points.

Conjugate curves/surfaces. A pair of conjugate curves/surfaces consists of two smooth curves/surfaces that always keep continuous and tangent contact with each other under motion law [Chen 1978]; see Section 4.1 for a review. Conjugation is a necessary condition for motion transmission between contact surfaces such as a pair of gears. In recent years, researchers in mechanical engineering modeled a single pair of conjugate curves to explore gear tooth profiles with varying shapes, including spur gears [Chen et al. 2014a] and spiral bevel gears [Chen et al. 2014b] with circular arc tooth profiles, spur gears with 2D freeform tooth profiles [Yu and Ting 2013], and spiral bevel gears with spatial freeform tooth profiles [Tan et al. 2015]. Researchers also investigated modeling gear tooth profiles with two [Tan et al. 2017] or three [Gao et al. 2014; Zhang et al. 2019] pairs of conjugate curves for high-performance gear transmission, and found that increasing the number of contact points (i.e., conjugation points) helps to reduce the maximum contact stress and to increase the load capacity.

All the above works explore varying conjugations between a conjugate curve/surface pair to model gears with new tooth profiles, assuming that the gears still perform the conventional motion transmission task (i.e., transferring 1-DOF rotation to another 1-DOF rotation). Compared with these works, our work makes use of multi-point conjugation between a conjugate surface pair to transfer complex motions, i.e., transferring 1-DOF rotation to a motion in N-DOF, $N \in [1, 3]$ motion space.

3 PROBLEM STATEMENT AND OVERVIEW

Our problem is to model a 2-moving-part mechanism for exactly generating a user-specified motion, driven by a single actuator. The user-specified motion is a periodic 3D motion, represented by a set of sequential rigid-body poses. To achieve the goal of motion generation, the 2-moving-part mechanism has to satisfy the following two requirements:

(1) *Working mechanism.* The 2-moving-part mechanism should be a working mechanism, where the driver is able to transfer its motion to the follower continuously, without collision or losing contacts.
(2) *Fabricable mechanism.* The geometry of each component part in the mechanism should be fabricable, e.g., by 3D printing. Thin or sharp geometric features should be avoided for each part.

Multi-point conjugation mechanisms. We model the 2-moving-part mechanism as a multi-point conjugation mechanism; see Figure 2 for an example. In a mpcMech, there are three mechanical parts in total:

- *Driver.* The driver part performs the input motion from a hand crank or a motor, which is always a periodic 1-DOF rotational motion.

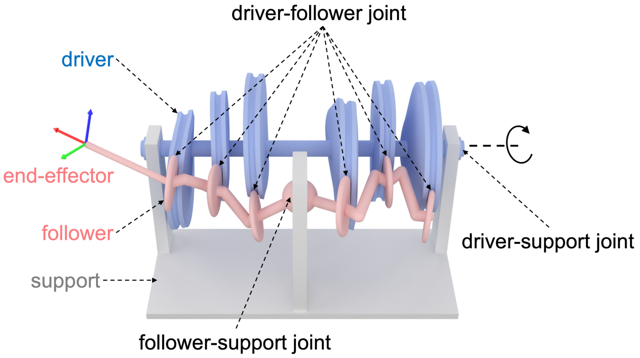


Fig. 2. An example multi-point conjugation mechanism (mpcMech_3R), where the follower-support joint is a spherical joint, allowing 3-DOF rotation of the follower.

- *Follower.* The follower part performs the motion specified by users. An end-effector is attached with the follower to output the motion, e.g., for accomplishing some tasks.
- *Support.* The support part is static and holds both the driver and the follower.

There are three mechanical joints in the mechanism:

- *Driver-support joint.* The driver-support joint is a revolute joint that enables the 1-DOF rotational motion of the driver.
- *Follower-support joint.* The follower-support joint defines the motion space of the follower. For example, if the follower-support joint is a spherical joint, the motion space of the follower is the 3-DOF rotational motion space.
- *Driver-follower joint.* The driver-follower joint is a *multi-point conjugation joint*, maintaining multiple conjugation points between the driver and the follower during any instant of motion.

Among the parts and joints in the mechanism, the driver-follower joint or the multi-point conjugation joint is the core component since it determines how the input motion of the driver is transferred to the output motion of the follower. We explain how to model the multi-point conjugation joint in Section 5.

Classification of multi-point conjugation mechanisms. We classify multi-point conjugation mechanisms according to the follower motion type allowed by the follower-support joint. We name each class of mpcMech as mpcMech_N_RRN_TT, where N_R and N_T are the number of DOFs for rotation and translation of the follower motion space, respectively. For example, mpcMech_3R represents mpcMechs whose follower motion is in 3-DOF rotational motion space. In this paper, we focus on modeling three classes of mpcMechs, i.e., mpcMech_1R, mpcMech_1R1T, and mpcMech_3R, whose follower-support joint is a revolute joint, cylindrical joint, and spherical joint, respectively; see Figure 10.

Overview of our approach. It should be pointed out that multiple conjugation points between a pair of conjugate surfaces are necessary but not sufficient to form a working multi-point conjugation mechanism. Thus in Section 4 we further develop a dynamic form closure condition, under which one conjugate surface (the driver) is able to continuously transfer the motion to the other conjugate surface (the follower) via utilizing multiple conjugation points. This is achieved by establishing a connection between conjugate surface pairs and form-closure grasps. In Section 5, we propose an optimization-based approach to model the geometry of a multi-point conjugation mechanism to generate a user-specified motion while ensuring that the mechanism is workable and fabricable.

4 THEORIES OF MULTI-POINT CONJUGATION MECHANISMS

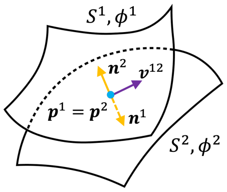
In our 2-moving-part mechanism, it is necessary that the two moving parts always maintain point contacts for transferring relative motion between them. However, maintaining point contacts does not guarantee that one moving part is able to transfer the motion to the other moving part continuously, without relying on friction; see the middle example in Figure 3. Existing theory of conjugate surfaces provides fundamental requirements for maintaining point contacts (i.e., point conjugation) between a pair of surfaces. By making a connection with form-closure grasps in robotics, we propose a new condition under which one conjugate surface is able to continuously transfer the motion to the other conjugate surface. In this section, we first review the existing theory of conjugate surfaces (Section 4.1) and the existing theory of form-closure grasps (Section 4.2), and then we present our theory of dynamic form closure (Section 4.3).

4.1 Theory of Conjugate Surfaces

The theory of conjugate surfaces [Chen 1978] deals with a common problem in mechanical processing and mechanical transmissions, i.e., the problem of mutual transformation between conjugate geometry and conjugate motion. We review the theory of conjugate surfaces below.

Assume that there are two surfaces $\mathbf{S}^1$ and $\mathbf{S}^2$, and a reference space R. Let the motions of surfaces $\mathbf{S}^1$ and $\mathbf{S}^2$ relative to the space R be represented by ϕ^1 and ϕ^2 respectively. If surface $\mathbf{S}^1$ (with motion ϕ^1) pushes $\mathbf{S}^2$ to perform motion ϕ^2, then surfaces $\mathbf{S}^1$ and $\mathbf{S}^2$ are called "conjugate surfaces" under "conjugate motions" ϕ^1, ϕ^2. The entire set of instantaneous contacting points within space R forms the locus surface/curve $\mathbf{p}$. Theory of conjugate surfaces deals with the intrinsic relationships among $\mathbf{S}^1$, $\mathbf{S}^2$, $\mathbf{p}$, ϕ^1, ϕ^2 [Chen 1978, 1985].

The contact entity between a pair of conjugate surfaces may be a point, a line segment, or a surface patch [Chen and Chen 1994]. Accordingly, conjugate pairs are classified into 3 types, i.e. point conjugation type, line conjugation type, and surface conjugation type. Traditionally, the former two types are called higher pairs, while the latter type is called the lower pair. However, in fact, the fundamental type is the point conjugation type, while the line and surface conjugation types are its degenerate cases. For the point conjugation type (see the inset), four fundamental conditions [Chen 1978] should be satisfied:



(1) *Coincide contact point.* The contact point (i.e., conjugation point) $\mathbf{p}^1$ on surface $\mathbf{S}^1$ and the contact point $\mathbf{p}^2$ on surface $\mathbf{S}^2$ must coincide with each other:

$$\mathbf{p}^1 = \mathbf{p}^2. \tag{1}$$

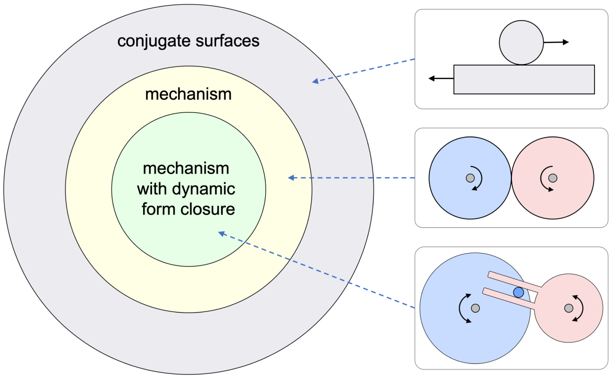


Fig. 3. Diagram that shows the relation among conjugate surfaces (grey region), mechanisms that maintain continuous contact (yellow region), and mechanisms that satisfy the dynamic form closure condition (green region). We here show a 2D example for each of them.

(2) *Coincide contact normal.* Two conjugate surfaces must be tangent with each other at the contact point $\mathbf{p}^1$ in order to avoid mutual interference at the point, i.e., coincident surface normal but opposite in direction:

$$\mathbf{n}^1 = -\mathbf{n}^2. \tag{2}$$

(3) *Relative velocity.* Relative velocity $\mathbf{v}^{12}$ at the contact point $\mathbf{p}^1$ must be perpendicular to the common normal $\mathbf{n}^1$ to ensure continuous contact between the two conjugate surfaces:

$$\mathbf{n}^1 \cdot \mathbf{v}^{12} = 0, \tag{3}$$

where the relative velocity $\mathbf{v}^{12} = \mathbf{v}^1 - \mathbf{v}^2$, and $\mathbf{v}^1$ and $\mathbf{v}^2$ are the velocity of point $\mathbf{p}^1$ and $\mathbf{p}^2$ in the space R, respectively.

(4) *Induced normal curvature.* Induced normal curvature $\overline{K}_{\boldsymbol{\tau}}$ of the two conjugate surfaces, along any tangential direction $\boldsymbol{\tau}$ at the contact point $\mathbf{p}^1$, should be positive or zero, to avoid interference between the two surfaces in differential neighborhood around the contact point $\mathbf{p}^1$:

$$\overline{K}_{\boldsymbol{\tau}} \geqslant 0. \tag{4}$$

The point conjugation type can be further classified as single-point and multi-point conjugation. For single-point conjugation, two conjugate surfaces keep in contact with each other only at *a single point* at any instant of motion. Single-point conjugation allows a relative motion with high degrees of freedom between a pair of conjugate surfaces [Chen and Chen 1994]. Multi-point conjugation is able to reduce the DOFs of relative motion between a pair of conjugate surfaces by using multiple contact points at any instant of the motion, thus serving potentially as a key element in the design of innovative precision mechanisms [Chen 1997].

Multi-point conjugation is a necessary but not sufficient condition for modeling a pair of conjugate surfaces as a 2-moving-part mechanism. This is because in a 2-moving-part mechanism, the input motion of the driver should always drive the output motion of the follower at any time. In other words, at any configuration of the conjugate pair, if we fix the pose of the driver surface, the follower surface should be fully immobilized by the contact points between the two surfaces. This instantaneous configuration of the driver and the follower is similar to a form-closure grasp in robotics, where a gripper fully immobilizes an object based on multiple contact points between them.

4.2 Theory of Form-Closure Grasps

In robotics, the most fundamental requirements for grasping and dexterous manipulation are the abilities to hold an object and control its position and orientation relative to the palm of the hand. *Form closure* and *force closure* are two most useful characterizations of grasp restraint [Prattichizzo and Trinkle 2016]. Form closure refers to a theoretical stable state where a set of *stationary* contacts fully immobilize a rigid body, without relying on friction. Form-closure grasps are widely used in robotics since they are very secure.

Specifically, first-order form closure can be established based only on the contact locations and the contact normals, without considering the contact curvature. First-order form closure requires at least 4 point contacts for a 2D rigid body (see Figure 4 (left) for an example) and at least 7 point contacts for a 3D rigid body [Somov 1897a,b]. Lakshminarayan [1978] generalized this to prove that n_v+1 contacts are necessary to form close an object with n_v DOFs. For example, a pinned 2D rigid body has $n_v = 1$ DOF; to fully immobilize the body, at least $n_v+1 = 2$ contact points are necessary; see Figure 4 (right) for an illustration.

Assume that there is a 3D rigid body with infinitesimal rigid motion. We express the infinitesimal rigid motion of the body using the generalized velocity $\mathbf{u} = [\mathbf{v}^\top, \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^6$, where $\mathbf{v}$ is the linear velocity and $\boldsymbol{\omega}$ is the angular velocity. Without loss of generality, we assume that the generalized velocity has a unit norm, i.e., $\|\mathbf{u}\| = 1$. Assume that there are K contact points $\{\mathbf{p}_k\}$, $k \in [1, K]$, on the surface of the body applied by a gripper. Denote the unit surface normal of the body (with inward direction) at point $\mathbf{p}_k$ by $\mathbf{n}_k$; see Figure 4. Both the contact point $\mathbf{p}_k$ and the contact normal $\mathbf{n}_k$ are defined in the local frame of the rigid body. The set of infinitesimal rigid motion of the body restrained by contact point $\mathbf{p}_k$ with normal $\mathbf{n}_k$ is represented as:

$$U_k = \{\mathbf{u} \mid \hat{\mathbf{n}}_k \cdot \mathbf{u} < 0\}, \quad k \in [1, K]$$
$$\hat{\mathbf{n}}_k = \begin{bmatrix} \mathbf{n}_k \\ \mathbf{p}_k \times \mathbf{n}_k \end{bmatrix}, \tag{5}$$

where $\hat{\mathbf{n}}_k$ is the generalized normal of contact point $\mathbf{p}_k$ with normal $\mathbf{n}_k$. To fully immobilize the rigid body, all the possible motions of the body should be restrained by the K contact points.

Hence, the condition of form closing a rigid body with n_v DOFs using K contact points is that the set of contact points and normals $\{(\mathbf{p}_k, \mathbf{n}_k)\}$ should satisfy

$$U \subset \bigcup_{k=1}^{K} U_k, \tag{6}$$

where U is the infinitesimal n_v-DOF motion space of the rigid body, and U_k is the motion space restrained by contact point $\mathbf{p}_k$ with normal $\mathbf{n}_k$ represented by Equation (5). According to [Lakshminarayan 1978], a necessary condition to fulfill Equation (6) is that the number of contact points, K, satisfies

$$K \geq n_v + 1. \tag{7}$$

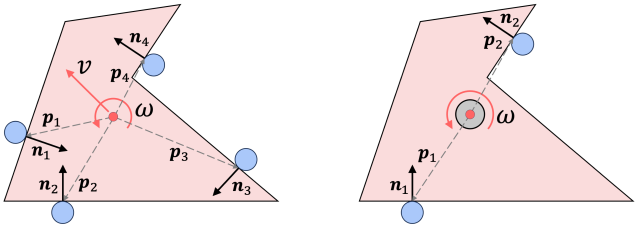
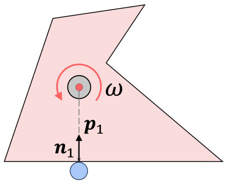


Fig. 4. Left: form closure of a 2D rigid body with 3 DOFs (2-DOF translation and 1-DOF rotation) using 4 contact points. Right: form closure of a 2D rigid body with 1-DOF rotation using 2 contact points, where the center of the support (in gray) is the center of rotation (in red).

We observe that $U \cap U_k = \emptyset$ if and only if $\hat{\mathbf{n}}_k \perp U$, which means that the contact point $\mathbf{p}_k$ with normal $\mathbf{n}_k$ does not contribute to restraining the rigid body's motion. To avoid this case, we require

$$\hat{\mathbf{n}}_k \not\perp U. \tag{8}$$

The inset shows a counter example. The pinned rigid body can rotate clockwise or counterclockwise along the z-axis. Hence, its motion space is $U = \{(0, 0, 0, 0, 0, -1)^\top, \ (0, 0, 0, 0, 0, 1)^\top\}$. The point $\mathbf{p}_1$ and its normal $\mathbf{n}_1$ are colinear. Hence, the generalized normal is $\hat{\mathbf{n}}_1 = [\mathbf{n}_1^\top, \mathbf{0}]^\top$. It is obvious that $\hat{\mathbf{n}}_1 \perp U$ in this counter example. Figure 4 shows two examples of form-closure grasps that meet the form closure condition, where each contact point satisfies Equation (8).



4.3 Theory of Dynamic Form Closure

To continuously transfer motion from one conjugate surface (driver surface) to the other (follower surface), the driver surface should *dynamically form close* the follower surface based on the multiple conjugation points between them at any instant of motion; see the bottom example in Figure 3. We call this condition *dynamic form closure*, and formulate it by extending the (static) form closure condition in Section 4.2.

Assume that the driver and follower form a conjugate surface pair that performs periodic motions with motion period T. The driver surface and the follower surface always maintain K conjugation points at any time $t \in [0, T)$, where each conjugation point satisfies the four conditions given in Section 4.1. The position and normal of each conjugation point at time t are denoted by $\mathbf{p}_k(t)$ and $\mathbf{n}_k(t)$, $k \in [1, K]$, respectively. According to Equation (5), the restrained motion set by the conjugation point $\mathbf{p}_k(t)$ with normal $\mathbf{n}_k(t)$ at time t is

$$U_k(t) = \{\mathbf{u} \mid \hat{\mathbf{n}}_k(t) \cdot \mathbf{u} < 0\}, \quad k \in [1, K], \ t \in [0, T)$$
$$\hat{\mathbf{n}}_k(t) = \begin{bmatrix} \mathbf{n}_k(t) \\ \mathbf{p}_k(t) \times \mathbf{n}_k(t) \end{bmatrix}, \tag{9}$$

where the restrained motion set $U_k(t)$ is a 5-dimensional semi-hypersphere in $\mathbb{R}^6$.

Theory of dynamic form closure. In order for a driver surface to form close a follower surface at any time $t \in [0, T)$, the set of K conjugation points with normals denoted as $\{(\mathbf{p}_k(t), \mathbf{n}_k(t))\}$ should satisfy:

$$U \subset \bigcup_{k=1}^{K} U_k(t), \quad \forall t \in [0, T), \tag{10}$$

where U is the infinitesimal N-DOF motion space of the follower surface, and $U_k(t)$ is the motion space restrained by conjugation point $\mathbf{p}_k(t)$ with normal $\mathbf{n}_k(t)$ at time t represented by Equation (9).

Similar to Equation (7), the number of conjugation points, K, should satisfy

$$K \geq N + 1, \tag{11}$$

where N is the number of DOFs of the follower-support joint.

Similar to Equation (8), the generalized normal $\hat{\mathbf{n}}_k(t)$ of each conjugation point should satisfy

$$\hat{\mathbf{n}}_k(t) \not\perp U, \quad \forall t \in [0, T), \tag{12}$$

such that each conjugation point will contribute to form closing the follower surface at any time t.

The follower's motion space U is defined by the follower-support joint. For a follower-support joint with N-DOF, it is clear that the motion space U of the follower allowed by the joint is a $(N-1)$-dimensional hypersphere in $\mathbb{R}^6$. For example, the motion space U for a spherical joint is a 2-sphere, and the motion space for a revolute joint is a 0-sphere. The dynamic form closure condition in Equation (10) can be understood as a requirement of covering a $(N-1)$-dimensional hypersphere U by K 5-dimensional semi-hypersphere $\{U_k(t)\}$. Under Equation (12), Equation (10) can be reformulated as

$$\mathbf{0} \in \text{interior}\left(\text{conv}(\{\tilde{\mathbf{n}}_k(t)\}_{1 \leqslant k \leqslant K})\right), \quad \forall t \in [0, T), \tag{13}$$

where interior$(\cdot)$ is the interior operator, conv$(\cdot)$ is a function that computes the convex hull of a set of points, and $\tilde{\mathbf{n}}_k(t)$ is a normalized vector that represents a projection of the generalized normal $\hat{\mathbf{n}}_k(t)$ onto the linear subspace spanned by U. For example, the motion space U for a spherical joint is $U = \{(\mathbf{0}, \boldsymbol{\omega}^\top)^\top \mid \|\boldsymbol{\omega}\| = 1\}$, and $\tilde{\mathbf{n}}_k(t) = \dfrac{\mathbf{p}_k(t) \times \mathbf{n}_k(t)}{\|\mathbf{p}_k(t) \times \mathbf{n}_k(t)\|}$. This reformulation is a generalization of the form closure condition in [Mishra et al. 1987] and it will be used for optimization-based modeling in Section 5.2. The supplementary material provides a proof showing that Equation (10) is equivalent to Equation (13).

5 MODELING OF MULTI-POINT CONJUGATION MECHANISMS

In this section, we introduce a bottom-up approach to model the geometry of a multi-point conjugation mechanism for exactly generating a user-specified motion, starting from a single pair of conjugate curves (Section 5.1), multiple pairs of conjugate curves (Section 5.2), until a multi-point conjugation joint and the whole mechanism (Section 5.3). In particular, we model the multi-point conjugation joint as a pair of conjugate surfaces, $\mathbf{S}^1$ and $\mathbf{S}^2$, with K conjugation points. The driver surface $\mathbf{S}^1$ performs periodic 1-DOF rotation $\mathbf{M}^1(t)$, and the follower surface $\mathbf{S}^2$ performs the user-specified motion $\mathbf{M}^2(t)$. The motion $\mathbf{M}^\alpha(t)$, $\alpha \in \{1, 2\}$ is represented by a rigid transformation of surface $\mathbf{S}^\alpha$ relative to its local frame.

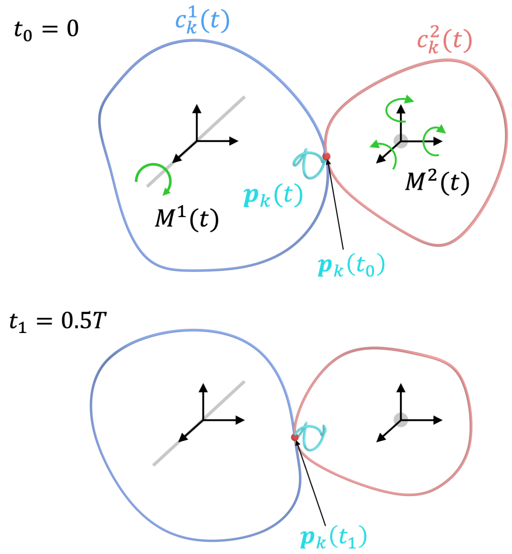


Fig. 5. Modeling a conjugate curve pair $\mathbf{c}_k^1(t)$ and $\mathbf{c}_k^2(t)$ with conjugate motions $\mathbf{M}^1(t)$ and $\mathbf{M}^2(t)$. We show the conjugate curve pair with two different configurations at time (top) $t_0 = 0$ and (bottom) $t_1 = 0.5T$. The small cyan curve in the middle is the locus curve $\mathbf{p}_k(t)$.

5.1 Modeling One Conjugate Curve Pair

In this section, our task is to model a pair of 3D conjugate curves with augmented normals that satisfy the point conjugation conditions in Section 4.1. We first reformulate the point conjugation conditions to facilitate geometry modeling, and then present an optimization-based approach to model a conjugate curve pair that facilitates fabrication.

Coincident contact point. The conjugate surface pair, $\mathbf{S}^1$ and $\mathbf{S}^2$, always maintains K conjugation points at any time $t \in [0, T)$. Now, we only consider a single conjugation point, say the k-th conjugation point, where $k \in [1, K]$. During the conjugate motions of $\mathbf{S}^1$ and $\mathbf{S}^2$, the k-th conjugation point forms a trajectory in the local frame of $\mathbf{S}^2$, denoted by curve $\mathbf{p}_k(t)$. At the same time, the k-th conjugation point forms a trajectory on surface $\mathbf{S}^\alpha$, $\alpha \in \{1, 2\}$, denoted by curve $\mathbf{c}_k^\alpha(t)$. The two curves $\mathbf{c}_k^1(t)$ and $\mathbf{c}_k^2(t)$ form a pair of *conjugate curves*, with $\mathbf{p}_k(t)$ as the locus curve; see Figure 5. The conjugate curve pair $\mathbf{c}_k^1(t)$ and $\mathbf{c}_k^2(t)$ should satisfy the coincident contact point condition (see Equation (1)) for any $t \in [0, T)$:

$$\mathbf{F}_1^2 \mathbf{M}^1(t) \mathbf{c}_k^1(t) = \mathbf{p}_k(t) = \mathbf{M}^2(t) \mathbf{c}_k^2(t), \tag{14}$$

where $\mathbf{F}_1^2$ is the transformation (i.e., a 3D translation) on surface $\mathbf{S}^1$'s local frame to make it align with surface $\mathbf{S}^2$'s local frame.

We assume that the geometry of each conjugate curve $\mathbf{c}_k^\alpha(t)$ is continuous, closed, and simple to facilitate modeling of the conjugate surface pair $\mathbf{S}^1$ and $\mathbf{S}^2$. To this end, we model the conjugate curve $\mathbf{c}_k^2(t)$ as a 3D closed cubic B-spline with n control points denoted

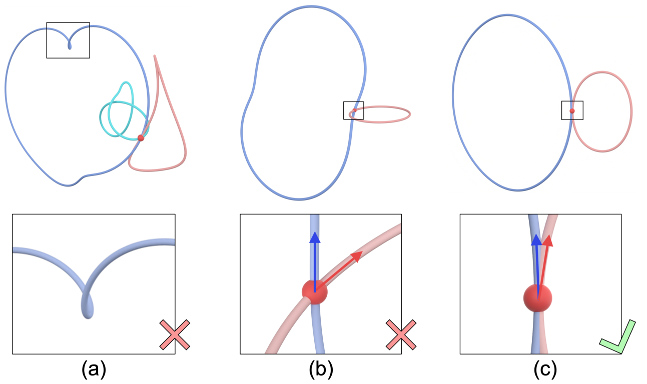


Fig. 6. Modeling the geometry of a conjugate curve pair. (a) Conjugate curves that are highly curved. (b) Conjugate curve pair whose tangents have a large angle at the conjugation point. (c) A good example of a conjugate curve pair.

as $\{\mathbf{e}_{\mathbf{c}_k^2}\}$, where n is set to 8 in our experiments. Given the known conjugate motions $\mathbf{M}^1(t)$ and $\mathbf{M}^2(t)$, the geometry of curve $\mathbf{c}_k^1(t)$ and curve $\mathbf{p}_k(t)$ can be easily derived from Equation (14), and is modeled as a polyline, respectively; see Figure 6 for examples of conjugate curve pairs.

Coincident contact normal. We augment each point on the conjugate curve pair $\mathbf{c}_k^1(t)$ and $\mathbf{c}_k^2(t)$ with a unit normal, denoted as $\mathbf{n}_k^1(t)$ and $\mathbf{n}_k^2(t)$, respectively. According to the coincident contact normal condition (see Equation (2)), we have the following equation for the conjugation point $\mathbf{p}_k(t)$ at any $t \in [0, T)$:

$$-\mathbf{F}_1^2\mathbf{M}^1(t)\mathbf{n}_k^1(t) = \mathbf{n}_k(t) = \mathbf{M}^2(t)\mathbf{n}_k^2(t), \tag{15}$$

where $\mathbf{n}_k(t)$ is the contact normal of the conjugation point $\mathbf{p}_k(t)$ in the local frame of surface S^2.

For each of the two conjugate curves $\mathbf{c}_k^1(t)$ and $\mathbf{c}_k^2(t)$, it is obvious that the contact normal should be perpendicular to the curve tangent for any point on the curve:

$$\mathbf{n}_k^1(t) \cdot \mathbf{t}_k^1(t) = 0, \quad \mathbf{n}_k^2(t) \cdot \mathbf{t}_k^2(t) = 0, \tag{16}$$

where $\mathbf{t}_k^\alpha(t)$, $\alpha \in \{1, 2\}$ is the tangent vector of conjugate curve $\mathbf{c}_k^\alpha(t)$. Based on Equation (15), Equation (16) can be rewritten as

$$\mathbf{n}_k(t) \cdot \mathbf{T}_k^1(t) = 0, \quad \mathbf{n}_k(t) \cdot \mathbf{T}_k^2(t) = 0, \tag{17}$$

where $\mathbf{T}_k^1(t) = \mathbf{F}_1^2\mathbf{M}^1(t)\mathbf{t}_k^1(t)$ is the tangent vector of conjugate curve $\mathbf{c}_k^1(t)$ in the local frame of surface S^2 and $\mathbf{T}_k^2(t) = \mathbf{M}^2(t)\mathbf{t}_k^2(t)$ is the tangent vector of conjugate curve $\mathbf{c}_k^2(t)$ in its local frame. According to Equation (17), the contact normal $\mathbf{n}_k(t)$ can be chosen as the cross product of $\mathbf{T}_k^1(t)$ and $\mathbf{T}_k^2(t)$. However, this simple strategy cannot guarantee continuity of contact normal $\mathbf{n}_k^\alpha(t)$ along curve $\mathbf{c}_k^\alpha(t)$, which is necessary to model a conjugate surface pair that is fabricable; see Figure 7(a) for a counter example.

To satisfy Equation (17), we assume that the contact normal $\mathbf{n}_k(t)$ is always perpendicular to the tangent $\mathbf{T}_k^2(t)$, i.e., $\mathbf{n}_k(t) \perp \mathbf{T}_k^2(t)$. At any time t, we establish a 2-dimensional frame $\{\mathbf{N}_k(t), \mathbf{B}_k(t)\}$ in the normal plane of $\mathbf{T}_k^2(t)$, and then $\mathbf{n}_k(t)$ can be expressed as

$$\mathbf{n}_k(t) = \mathbf{N}_k(t)\cos\theta_k(t) + \mathbf{B}_k(t)\sin\theta_k(t). \tag{18}$$

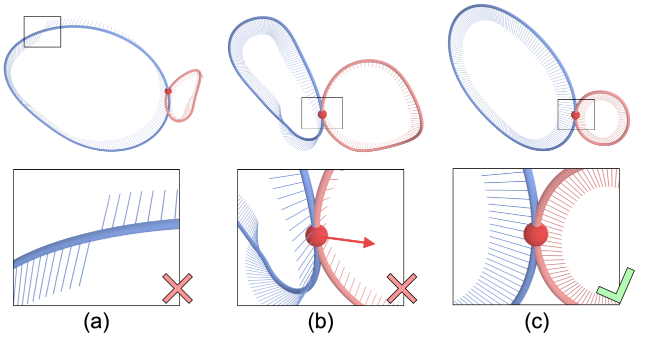


Fig. 7. Modeling the normals of a conjugate curve pair. (a) The normals of one conjugate curve change suddenly at one point. (b) The normal and the curvature vector (in red color) have a large angle at the conjugation point. (c) A good example of curve normals.

We model the set of angles $\{\theta_k(t)\}$ using a 1D closed cubic B-spline to ensure a smooth change of the normals $\{\mathbf{n}_k(t)\}$ with respect to time t. The control points of the 1D B-spline are denoted as $\{\mathbf{e}_{\theta_k}\}$ and we use 10 control points in our experiments. We satisfy $\mathbf{n}_k(t) \perp \mathbf{T}_k^1(t)$ by minimizing:

$$E_{\text{norm}} = \sum_{i=0}^{N-1}\left(\mathbf{n}_k(t_i) \cdot \frac{\mathbf{T}_k^1(t_i)}{\|\mathbf{T}_k^1(t_i)\|}\right)^2, \tag{19}$$

where N is the number of samples for the motion period T and $t_i = \frac{i}{N}T$, $i \in [0, N)$. In all our experiments, N is set to 360. Figure 7(c) shows a good example of curve normals modeled by this approach.

Relative velocity. Denote the relative velocity at the conjugation point $\mathbf{p}_k(t)$ by $\mathbf{v}_k^{12}(t)$. We have the following kinematic relation [Litvin and Fuentes 2004]:

$$\mathbf{v}_k^{12}(t) = \mathbf{T}_k^2(t) - \mathbf{T}_k^1(t), \tag{20}$$

by taking derivatives of Equation (14); see the supplementary material for the derivation. Based on Equations (17) and (20), the relative velocity condition (see Equation (3)) of point conjugation is always satisfied.

Induced normal curvature. The induced normal curvature condition (see Equation (4)) is about surface curvature. Hence, it will be satisfied when modeling the conjugate surface pair in Section 5.3.

Optimization-based modeling. Besides the above conditions for point conjugation, we propose three additional energy terms to model a conjugate surface pair that is fabricable and compact. First, we require that each conjugate curve $\mathbf{c}_k^\alpha(t)$, $\alpha \in \{1, 2\}$ is smooth:

$$E_{\text{smth}} = \sum_\alpha\sum_{i=0}^{N-1}\left\|\frac{\mathbf{t}_k^\alpha(t_{i+1})}{\|\mathbf{t}_k^\alpha(t_{i+1})\|} - \frac{\mathbf{t}_k^\alpha(t_i)}{\|\mathbf{t}_k^\alpha(t_i)\|}\right\|^2; \tag{21}$$

see Figure 6(a) for a counter example. Second, we require that the tangents of the two conjugate curves are close to colinear:

$$E_{\text{tang}} = \sum_{i=0}^{N-1}\left\|\frac{\mathbf{T}_k^1(t_i)}{\|\mathbf{T}_k^1(t_i)\|} - \frac{\mathbf{T}_k^2(t_i)}{\|\mathbf{T}_k^2(t_i)\|}\right\|^2; \tag{22}$$

see Figure 6(b) for a counter example. Third, we require that the two conjugate curves are close to co-planar:

$$E_{\text{plan}} = \sum_{i=0}^{N-1} \left\| \mathbf{n}_k(t_i) - \frac{\boldsymbol{\kappa}_k^2(t_i)}{\|\boldsymbol{\kappa}_k^2(t_i)\|} \right\|^2 , \tag{23}$$

where $\boldsymbol{\kappa}_k^2(t)$ is the curvature vector of conjugate curve $\mathbf{c}_k^2(t)$; see Figure 7(b) for a counter example.

We formulate the problem of modeling a conjugate curve pair $\mathbf{c}_k^1(t)$ and $\mathbf{c}_k^2(t)$ with augmented normals $\mathbf{n}_k^1(t)$ and $\mathbf{n}_k^2(t)$ as an optimization problem:

$$\begin{aligned} \min_{\{\mathbf{e}_{\mathbf{c}_k^2}\},\{\mathbf{e}_{\theta_k}\}} \quad & E_{\text{k}} = \omega_1 E_{\text{smth}} + \omega_2 E_{\text{tang}} + \omega_3 E_{\text{norm}} + \omega_4 E_{\text{plan}}, \\ \text{s.t.} \quad & L_1 \leqslant \text{Length}(\mathbf{c}_k^2(t)) \leqslant L_2, \end{aligned} \tag{24}$$

where the conjugate curve pair's shape is parameterized by $\{\mathbf{e}_{\mathbf{c}_k^2}\}$ and the conjugate curve pair's normals are parameterized by $\{\mathbf{e}_{\theta_k}\}$. The constraint on the length of the conjugate curve $\mathbf{c}_k^2(t)$ controls the size of the follower surface, and we set $L_1 = 4\pi$ and $L_2 = 6\pi$ in our experiments. The weights are typically set as $\omega_1 = 50$, $\omega_2 = 1.0$, $\omega_3 = 100$, and $\omega_4 = 1.0$. We solve this constrained optimization problem using the sequential least squares programming algorithm (SLSQP) [Kraft 1988] implemented in the NLopt package [Johnson 2020], where the conjugate curve $\mathbf{c}_k^2(t)$ is initialized as a circle-like curve due to its simplicity.

5.2 Modeling K Conjugate Curve Pairs

In this section, our task is to model K 3D conjugate curve pairs $\{\mathbf{c}_k^1(t),\ \mathbf{c}_k^2(t)\}$ with augmented normals $\{\mathbf{n}_k^1(t),\ \mathbf{n}_k^2(t)\}$, $k \in [1, K]$. The conjugate curves $\{\mathbf{c}_k^1(t)\}$ perform the input motion $\mathbf{M}^1(t)$, and will be used to model the driver surface S^1. The conjugate curves $\{\mathbf{c}_k^2(t)\}$ perform the output motion $\mathbf{M}^2(t)$, and will be used to model the follower surface S^2. To ensure a working mechanism, the K conjugate curve pairs with augmented normals should satisfy the dynamic form closure condition in Section 4.3. To ensure a fabricable mechanism, the K conjugate curve pairs should be laid out in the 3D space without intersection. We formulate this modeling problem as an optimization and introduce its search space, optimization formulation, and solver.

5.2.1 Search Space. The search space of modeling K conjugate curve pairs includes:

(1) *Position of driver surface* S^1 *relative to follower surface* S^2. This relative position is described by the 3D translation represented by $\mathbf{F}_1^2$. In particular, we assume that driver surface S^1 and follower surface S^2 are aligned in the y−axis and z−axis of the local frame of S^2; see Figure 5. By this, we describe the relative position using a single parameter d_x, which is the distance between the driver axis and the follower-support joint center.

(2) *Number of conjugate curve pairs* K. We assume that each pair of conjugate curves has a single conjugation point. Hence, the number of conjugate curve pairs is the same as the number of conjugation points. According to Equation (11), the number of conjugation points, K, should satisfy $K \geq N + 1$, where N is the number of DOFs of the follower-support joint.

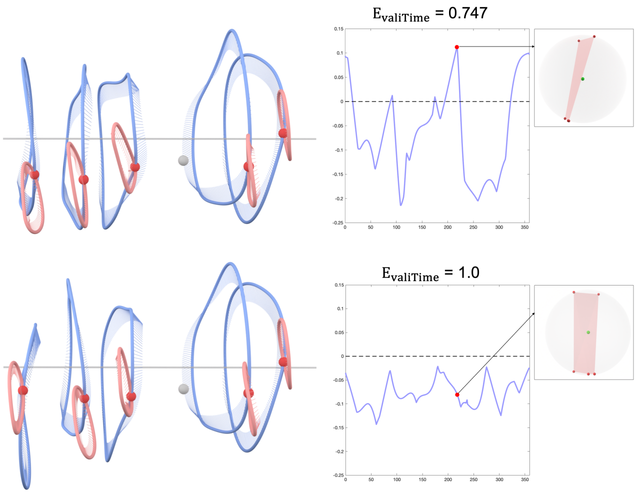


Fig. 8. Two examples of K conjugate curve pairs modeled by our approach that (top) does not satisfy and (bottom) satisfies the dynamic form closure condition. (Left) One configuration of the K conjugate curve pairs at time t_c. (Middle) Visualization of dist(t), where the red dot shows dist(t_c). (Right) Visualization of the convex hull conv($\{\tilde{\mathbf{n}}_k(t_c)\}_{1\leqslant k\leqslant K}$) (in red) and the origin (in green) for the configuration in the projection space.

(3) *Geometry of* K *conjugate curve pairs* $\{\mathbf{e}_{\mathbf{c}_k^2}\}$. The geometry of each conjugate curve pair is parameterized by control points $\mathbf{e}_{\mathbf{c}_k^2}$ of the conjugate curve $\mathbf{c}_k^2(t)$, $k \in [1, K]$; see Section 5.1.

(4) *Normals of* K *conjugate curve pairs* $\{\mathbf{e}_{\theta_k}\}$. The normals of each conjugate curve pair are parameterized by control points $\mathbf{e}_{\theta_k}$ of the set of angles $\{\theta_k(t)\}$, $k \in [1, K]$; see Section 5.1.

5.2.2 Optimization Formulation. We introduce the objective function and constraints of modeling K conjugate curve pairs, where the K conjugation points with normals are $\{(\mathbf{p}_k(t), \mathbf{n}_k(t))\}$.

Objective function. According to the theory of dynamic form closure in Section 4.3, the set of K conjugation points with normals $\{(\mathbf{p}_k(t), \mathbf{n}_k(t))\}$ has to meet Equation (13) such that the follower surface can be fully immobilized by the driver surface at any time t. Denote the signed distance from the origin $\mathbf{0}$ to the surface of convex hull conv($\{\tilde{\mathbf{n}}_k\}_{1\leqslant k\leqslant K}$) at time t by:

$$\text{dist}(t) = \text{dist}(\mathbf{0},\ \text{conv}(\{\tilde{\mathbf{n}}_k(t)\}_{1\leqslant k\leqslant K})). \tag{25}$$

We consider that the K pairs of conjugate curves satisfy the dynamic form closure condition at time t if the signed distance dist(t) is less than a tolerance $-\epsilon$:

$$E(t) = \begin{cases} 1, & \text{if dist}(t) < -\epsilon. \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

where we set $\epsilon = 0.02$ in our experiments.

Our objective function evaluates how well the dynamic form closure condition is satisfied using two terms. The first term evaluates if the dynamic form closure condition is satisfied for the whole

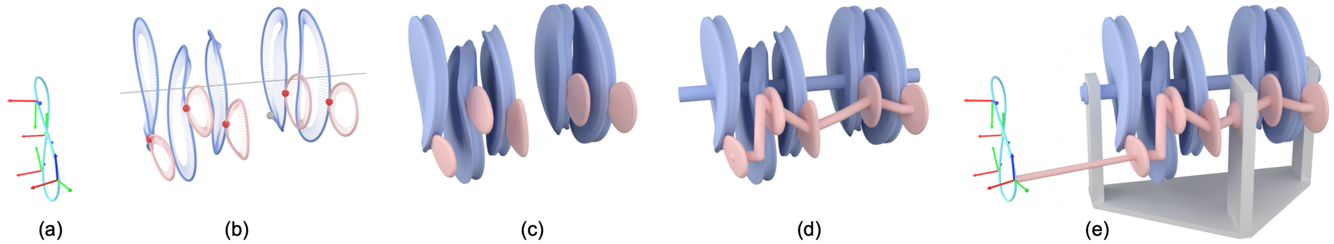


Fig. 9. Modeling a multi-point conjugation mechanism. (a) Given the user-specified motion $\mathbf{M}^2(t)$, (b) we first model K conjugate curve pairs with augmented normals to generate the motion. Next, we (c) model a conjugate surface pair that passes through the conjugate curve pairs as a multi-point conjugation joint and (d) ensure both the driver surface and follower surface form a piece of connected geometry, respectively, for fabrication. (e) Lastly, we model the support, the driver-support joint, and the follower-support joint.

motion period T:

$$E_{\text{valiTime}} = \frac{1}{N}\sum_{i=0}^{N-1} E(t_i). \tag{27}$$

The second term evaluates the worst case of satisfying the dynamic form closure condition over the motion period T:

$$E_{\text{maxDist}} = \max_i \ \text{dist}(t_i). \tag{28}$$

This term is inspired by the grasp quality measure in [Cornellà and Suárez 2009]. Figure 8 shows two examples of K conjugate curve pairs as well as their evaluation.

In our formulation, E_{valiTime} and E_{maxDist} serve a redundant purpose for the satisfaction of the dynamic form closure condition. In particular, $E_{\text{valiTime}} = 1$ and $E_{\text{maxDist}} \leq 0$ when the condition is satisfied. Although this formulation works well in practice, it is not the only way to formulate the problem.

Constraints. Our optimization has two constraints. First, according to Equation (12), the generalized normal should not be perpendicular to the motion space U for each conjugate curve pair. Hence, we introduce the constraint:

$$C_{\text{closure}} = C(\mathbf{n}_k(t), \mathbf{p}_k(t), U) \leqslant 0, \quad k \in [1, K]. \tag{29}$$

For mpcMech_3R, the constraint is equivalent to avoiding $\mathbf{n}_k \times \mathbf{p}_k = \mathbf{0}$. Hence, the constraint expression can be written as:

$$C_{\text{closure}} = \left(\mathbf{n}_k(t) \cdot \frac{\mathbf{p}_k(t)}{\|\mathbf{p}_k(t)\|}\right)^2 - \cos^2(\gamma_0 + \frac{\pi}{2}), \tag{30}$$

where γ_0 is a threshold set as 10° in our experiments. We apply a similar treatment of C_{closure} for mpcMech_1R and mpcMech_1R1T; see the supplementary material for details.

The second constraint is to avoid intersection among the K conjugate curve pairs $\{\mathbf{c}_k^1(t),\ \mathbf{c}_k^2(t)\}$ such that there is no interference among the K conjugation points at any time t. This constraint is formulated as:

$$z_k^{\min} \leqslant \mathbf{c}_k^2(t).z \leqslant z_k^{\max}, \quad k \in [1, K],\ \forall t \in [0, T), \tag{31}$$

where $\mathbf{c}_k^2(t).z$ is the z-coordinate of the conjugate curve $\mathbf{c}_k^2(t)$ in its local frame, and the ranges $\{[z_k^{\min}, z_k^{\max}]\}$ are disjoint equal-length segments along the z axis for $k \in [1, K]$.

5.2.3 *Optimization Solver.* To solve the optimization problem, we first solve for the geometry $\{\mathbf{e}_{\mathbf{c}_k^2}\}$ and normals $\{\mathbf{e}_{\theta_k}\}$ of K conjugate curve pairs, assuming the relative position (d_x) of the driver surface and the number (K) of conjugate curves are given. Next, we explain how to solve K and d_x. The details of our optimization solver are provided in the supplementary material.

Solving for $\{\mathbf{e}_{\mathbf{c}_k^2}\}$ and $\{\mathbf{e}_{\theta_k}\}$. The key challenge of solving for the geometry $\{\mathbf{e}_{\mathbf{c}_k^2}\}$ and normals $\{\mathbf{e}_{\theta_k}\}$ of K conjugate curve pairs is that they have to satisfy various requirements in Sections 5.1 and 5.2. To address the problem, our idea is to first generate a variety of candidates for each of the K conjugate curve pairs by solving the optimization in Equation (24) augmented with two constraints of Equations (29) and (31), and then use a genetic algorithm to combine these candidates guided by the two objective terms in Equations (27) and (28) to find a solution of K conjugate curve pairs that satisfy the dynamic form closure condition. To ensure diversity of the generated candidates of conjugate curve pairs, we add one more energy term E_{vary} to the objective function in Equation (24); see the supplementary material for details.

Solving for K and d_x. According to our theory of mpcMechs in Section 4.3, at least $N + 1$ conjugate curve pairs are necessary to generate a motion in N-DOF motion space. To obtain a compact mechanism, the number (K) of conjugate curve pairs should be as small as possible. Hence, we start by trying to find a solution with $K = N + 1$ conjugate curve pairs using the above algorithm. If we cannot find such a solution, we increase K by one until we find a feasible solution. d_x is used to determine a suitable size of the driver surface, given that the follower surface size has been constrained by requiring the length of the conjugate curve $\mathbf{c}_k^2$ within a user-specified range $[L_1, L_2]$ in Equation (24). On the one hand, the driver-follower distance (d_x) should be large enough to model the driver surface for encoding the prescribed motion. On the other hand, d_x should not be too large to keep the resulting mechanism compact. Hence, we search for a suitable value of d_x by starting from a small value (set empirically) and gradually increasing it until finding a d_x that results in a working and compact mechanism. Alternatively, the value of d_x can be interactively adjusted [Hutama et al. 2011; Song et al. 2014] by users until producing a desirable mechanism.

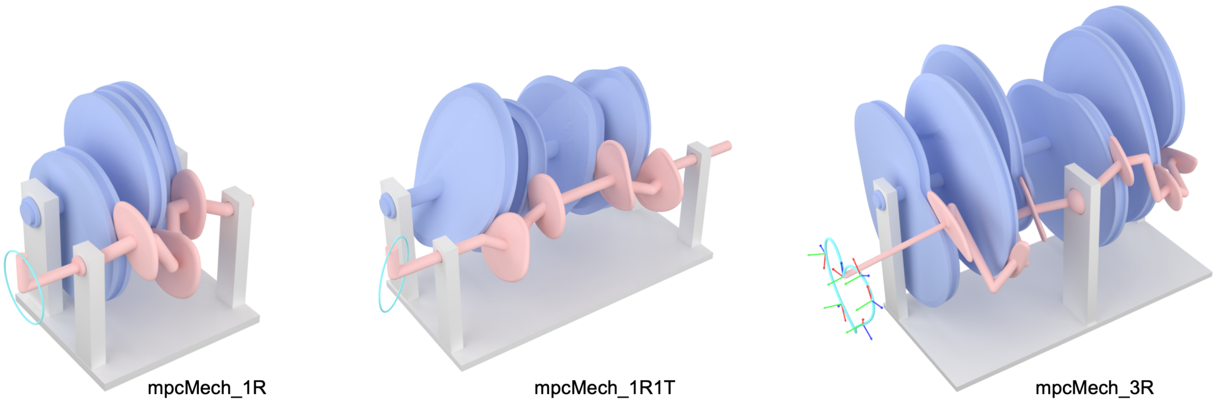


Fig. 10. mpcMechs modeled by our approach to generate motions in different kinds of motion space: (a) 1-DOF rotation, (b) 1-DOF rotation and 1-DOF translation, and (c) 3-DOF rotation space.

Table 1. Statistics and timings. For each result shown in the paper, we report the mpcMech class, number (K) of conjugate curve pairs, relative position (d_x) of the driver surface, energy term E_{valiTime}, energy term E_{maxDist}, and optimization time for modeling K conjugate curve pairs.

Fig	mpcMech Class	# conjugate curve pair (K)	Relative position (d_x)	E_{valiTime}	E_{maxDist}	Optim. Time (min)
1	3R	6	7.97	1.0	-0.039	23.98
9	3R	5	7.00	1.0	-0.021	12.27
10	1R	3	6.00	1.0	-1.000	4.88
	1R1T	4	6.00	1.0	-0.244	9.89
	3R	6	9.73	1.0	-0.052	18.94
11	1R	3	6.00	1.0	-1.000	8.88
12	3R	6	9.00	1.0	-0.073	18.52
13	3R	6	6.00	1.0	-0.081	18.58
14	3R	6	8.33	1.0	-0.048	21.67
15	3R	6	4.44	1.0	-0.056	18.03

5.3 Modeling Multi-Point Conjugation Mechanisms

Modeling a multi-point conjugation joint. After obtaining the K conjugate curves $\{\mathbf{c}_k^2(t)\}$, we model the follower surface $\mathbf{S}^2$ by sweeping a cross section shape along each conjugate curve $\mathbf{c}_k^2(t)$. We select a half ellipse curve as the cross section shape and align the ellipse's major axis with the curve normal $\mathbf{n}_k^2(t)$ during the sweeping. The sweeping along each conjugate curve results in an annular surface, which is further made watertight by filling the hole in the middle. The driver surface $\mathbf{S}^1$ is initially modeled as a swept surface using the same approach, where the cross section is selected as a rectangle. To avoid collision between the driver and the follower, we progressively carve the driver's initial shape using the follower shape by simulating their conjugate motions for a whole motion period. By this, we obtain the driver surface with a groove-like joint; see Figure 9(b&c). This carving method guarantees that the conjugate surface pair, $\mathbf{S}^1$ and $\mathbf{S}^2$, satisfies the induced normal curvature condition (see Equation (4)) of point conjugation. The modeled conjugate surface pair forms a multi-point conjugation joint.

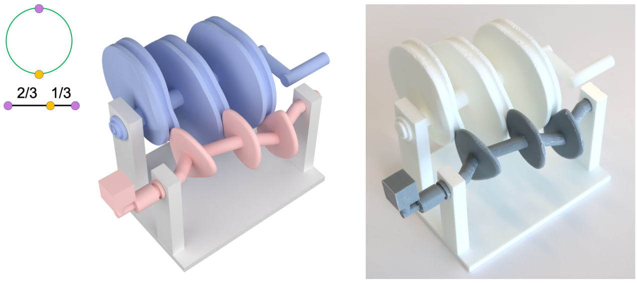


Fig. 11. Evaluating the kinematic performance of a mpcMech_1R for generating a 1-DOF non-uniform rotational motion, where the user specification on the timing is shown at the top left corner. (Left) Our modeled mpcMech. (Right) The 3D printed prototype.

Finalizing geometry modeling of the mpcMech. Until now, both the driver surface and the follower surface are a piece of geometry with disconnected parts, respectively, which cannot be directly fabricated. Hence, we make the driver surface connected by connecting its parts using a cylinder centered at the driver's rotation axis. We make the follower surface connected by connecting each pair of its consecutive parts using a thickened polyline with three linear segments, while ensuring that the thickened polyline does not collide with the driver surface during the conjugate motion; see Figure 9(d). We finalize the mechanism modeling by adding the driver-support joint to the driver surface, adding the follower-support joint to the follower surface, and modeling the support part as a few pillars (connected at the bottom) that hold the driver and follower; see Figure 9(e).

6 RESULTS

We implemented our mpcMech modeling approach in C++ and libigl [Jacobson et al. 2018] on a MacBook with a 3.2GHz CPU and 16GB memory. We show that our approach is able to model mpcMechs with different kinds of motions (Figure 10), evaluate kinematic performance of our modeled mpcMechs with 3D printed prototypes (Figures 11 and 12), demonstrate the usefulness of our mpcMechs with three applications (Figures 1, 13, and 14), and compare our mpcMech with a conventional mechanism to show its

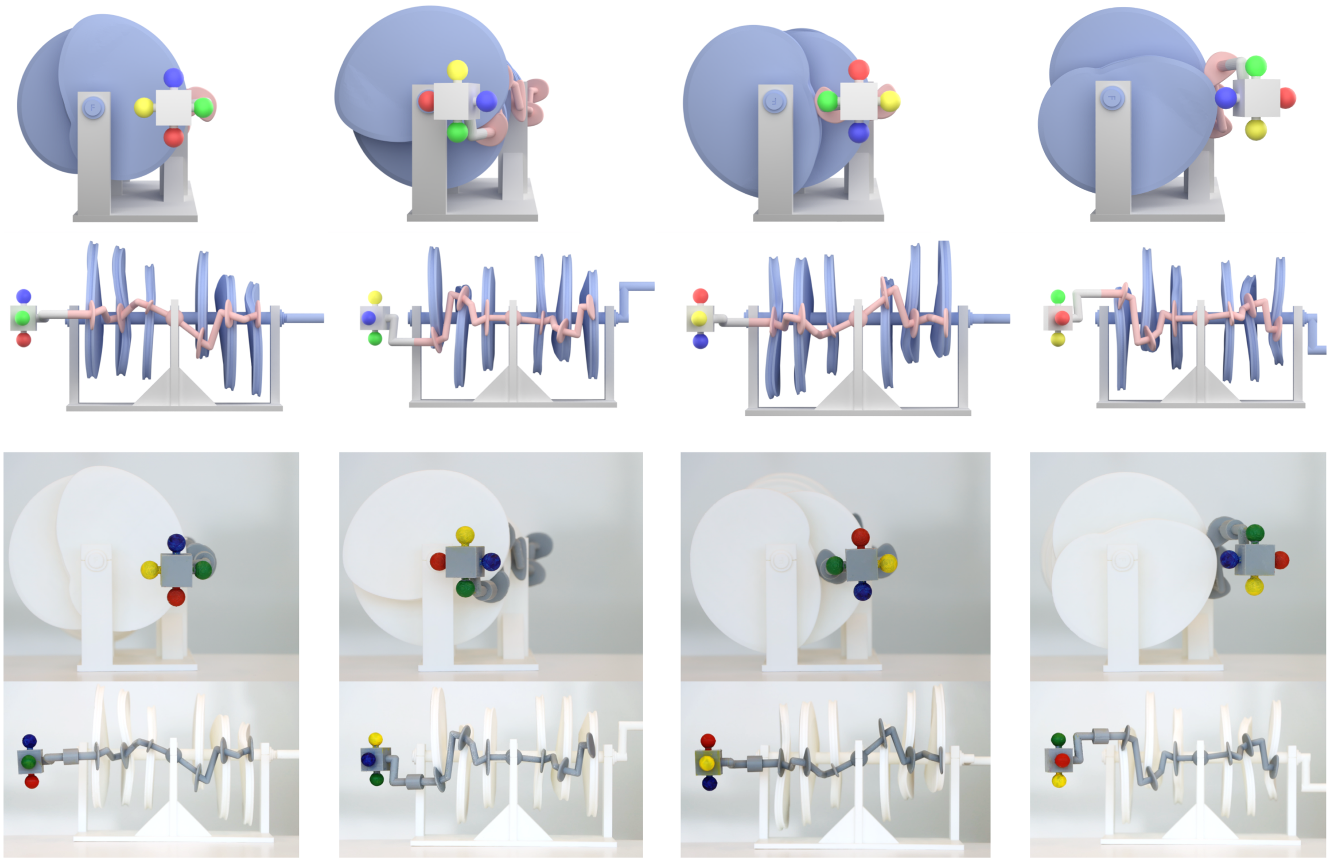


Fig. 12. Evaluating the kinematic performance of a mpcMech_3R for generating a 3D motion (see the inset in the next page). (Top) Four key poses generated by our modeled mpcMech from the front and side views, where the leftmost one shows the first key pose. (Bottom) The 3D printed prototype in the corresponding configurations and views.

characteristics in terms of motion generation (Figures 15). Please watch the accompanying video for demos. Table 1 presents statistics of all the results shown in the paper. Overall, modeling a mpcMech takes from 5 mins to 24 mins.

Modeling mpcMechs of different classes. Figure 10 shows 3 classes of mpcMechs modeled by our approach, including mpcMech_1R, mpcMech_1R1T, and mpcMech_3R. We observe that there is a larger number of conjugate curve pairs in the modeled mpcMech when the follower's motion space has a higher DOF; see also Table 1. Moreover, the number (K) of conjugate curve pairs computed by our approach is slightly larger than the minimum number of conjugate curve pairs required by the dynamic form closure condition (Equation (11)). For example, the modeled mpcMech_1R has 3 conjugate curve pairs, given that the minimum number is 2; the modeled mpcMech_1R1T has 4 conjugate curve pairs, given that the minimum number is 3.

Evaluation of kinematic performance. We conducted two physical experiments to evaluate the kinematic performance of two different classes of mpcMechs (i.e., mpcMech_1R and mpcMech_3R). In each experiment, we model a mpcMech to generate the specified motion and fabricate the mpcMech with 3D printing using an Ultimaker S5 printer with tough PLA material; see Figures 11 and 12. We have two goals in these experiments. The first goal is to validate if our modeled mpcMech is a working mechanism in which the driver is able to continuously transfer the motion to the follower via the multi-point conjugation joint. The second goal is to evaluate if the motion generated by the physical mechanism is identical to the virtual counterpart. Please watch the accompanying video for demos of the two experiments.

In the first experiment, we design a mpcMech_1R to transfer a 1-DOF uniform rotation to a 1-DOF non-uniform rotation; see Figure 11. The specified motion is a 1-DOF non-uniform rotation, where the follower rotates slowly in the first half circle ($\theta \in [0^\circ, 180^\circ]$) and fast in the second half circle ($\theta \in [180^\circ, 360^\circ]$). In particular, the timing of the first half circle is two times of the timing of the second half circle. We model a mpcMech_1R that exactly generates the prescribed motion. In our physical test, we found that the driver is able to drive the motion of the follower continuously. Moreover, while making the driver rotate uniformly, we record the timings that the follower rotates within the two half circles, respectively. The timing ratio between the first half circle and the second half circle is close to 2, matching the user specification.

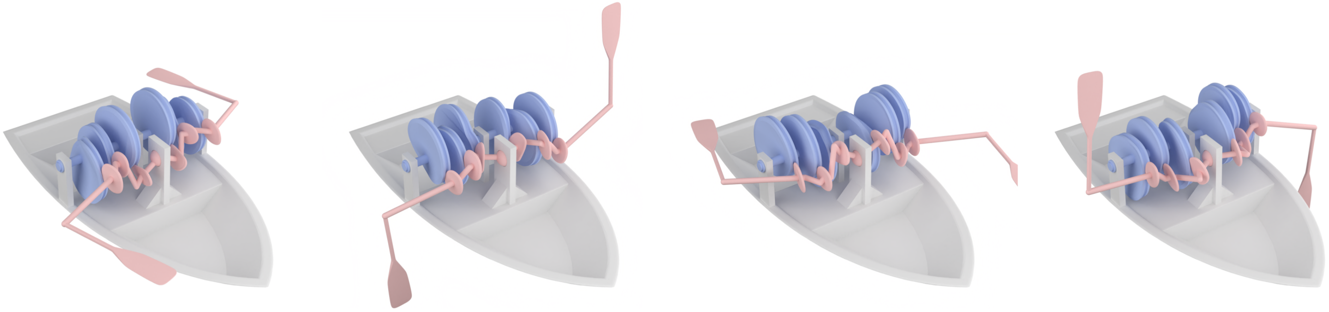


Fig. 13. A paddling mechanism modeled as a mpcMech using our approach for driving a boat.

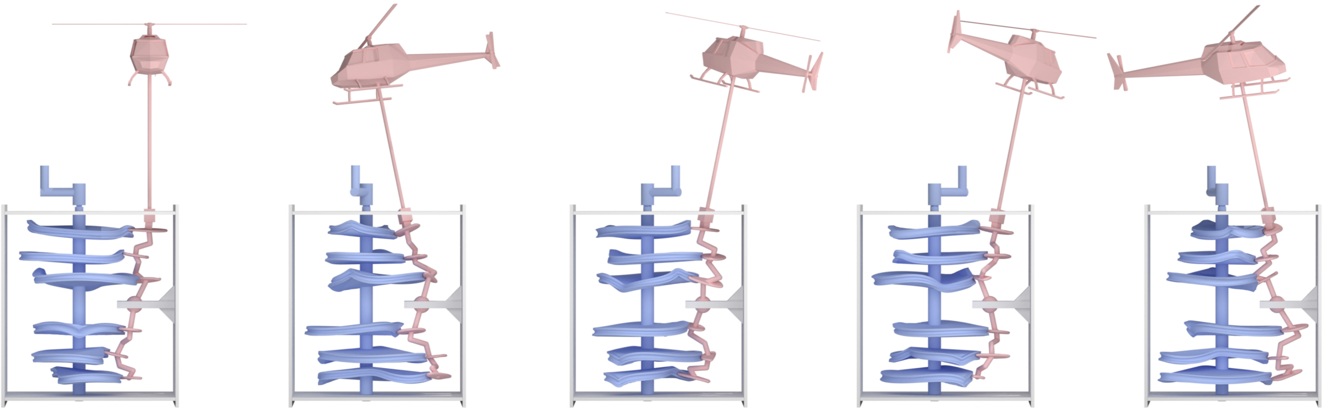
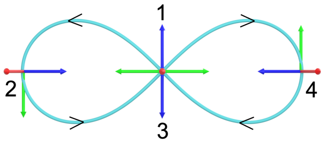


Fig. 14. A mechanical toy in which a mpcMech is modeled by our approach to control the flying motion of a helicopter.

In the second experiment, we model a mpcMech_3R to generate a user-specified 3D motion; see Figure 12. The target motion is specified as an eight-like curve on a spherical surface with four key poses at the two ends and middle of the curve; see the inset. To visualize the output motion, the end-effector is modeled as a box with four attached spheres of different colors. In our physical test, we found that the driver is able to continuously transfer motion to the follower, despite that the output motion is 3D and complex. We compare the four key poses generated by the virtual and physical mpcMechs in two different views, and find that they are identical to each other. For example, for both the virtual and physical results, we only see the front face of the end-effector cube in the 1st and 3rd key poses, and see the front face and one side face on the left (right) of the end-effector cube in the 2nd (4th) key pose; compare the 1st row and the 3rd row in Figure 12.



In these two experiments, the two physical prototypes function well for transferring motions, validating our dynamic form closure condition that transforms a conjugate surface pair into a working 2-moving-part mechanism. Moreover, the two physical prototypes are able to generate user-specified motions, demonstrating that these mechanisms can be used in practice. Note that the two fabricated examples are relatively simple and do not operate smoothly which may be caused by fabrication tolerances and friction. We consider fabrication of well-functioning complex mpcMechs as future work.

Applications. We show that our mpcMech can be used for three different applications, i.e., low-cost manipulator, paddling mechanism, and mechanical toy. In all these applications, we model a mpcMech_3R to generate a user-specified 3D motion. First, our mpcMech can be used as a low-cost manipulator driven by a single actuator to perform a pick-and-place task repetitively; see Figure 1. In this application, the task is to pick a box from the floor, manipulate the box, and place it on a table. The prescribed motion is more than a simple 3D translation since the box has to be flipped during the pick-and-place; see Figure 1 (right) for the motion trajectory and three key poses. Second, our mpcMech can be used as a paddling mechanism, where two paddles are connected to the two ends of the follower, respectively; see Figure 13. Simply by actuating the driver, the follower is able to perform a repetitive paddling motion to drive a boat. Third, our mpcMech can be used for designing mechanical toys. We show an example mechanical toy modeled by our approach, where a helicopter is connected to the follower and performs an intriguing flying motion; see Figure 14. In this example, the mpcMech is put in an orientation where the driver rotation axis is perpendicular to instead of parallel with the ground. Thanks to

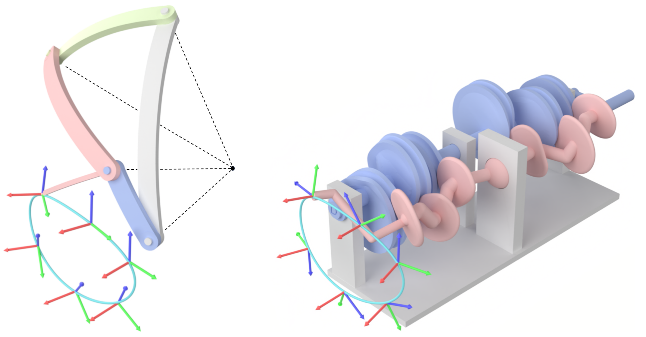


Fig. 15. Comparing (right) our mpcMech with (left) a spherical four-bar linkage [Sun et al. 2016] for 3D motion generation.

its dynamic form closure property, our mpcMech can work under different orientations.

Comparison with a spherical 4-bar linkage. Spherical motions are motions in 3-DOF rotational motion space. Spherical motions occur frequently in mechanical systems such as orienting mechanisms and robotic wrists. Spherical 4-bar linkage is one of the most widely used mechanisms for generating spherical motions [Bai et al. 2019]. Sun et al. [2016] designed a spherical 4-bar linkage to generate a spherical motion; see Figure 15 (left). We modeled a mpcMech that generates the same path as the spherical linkage but different rotational motions along the path; see Figure 15 (right). In particular, the end-effector poses are similar along the path for the spherical linkage while the end-effector poses have significant changes along the path for the mpcMech; see the sampled poses in Figure 15. Due to this difference, our mpcMech complements spherical linkages in terms of 3D motion generation.

Discussion. Compared with traditional mechanisms like spatial linkages, one strength of our mpcMech is that it is able to exactly generate a prescribed motion. However, this does not mean that our approach is able to model a mpcMech to exactly generate an arbitrary prescribed motion. In particular, when the prescribed motion is too complex, our approach may fail to model such a mpcMech. This is because the conjugate curve pairs likely have complex geometry due to the complex prescribed motion, which may make modeling of a fabricable conjugate surface pair fail.

7 CONCLUSION

We have presented a new class of mechanisms called multi-point conjugation mechanisms (mpcMechs). Our mpcMech consists of two moving parts, a driver and a follower, and it relies on a multi-point conjugation joint to transfer motion from the driver to the follower. Our aim is to understand and model mpcMechs for 3D motion generation. We solve this problem via proposing a dynamic form closure condition that transforms a conjugate surface pair into a working mpcMech. We also propose an optimization-based approach to model a mpcMech for generating a user-specified 3D motion, particularly in 3-DOF motion space. As evidenced by the wide range of mpcMechs that we modeled, our method achieves this goal. Our work advances the state-of-the-art in mechanism design by showing a successful attempt of encoding complex motions into the free-form geometry of novel mechanisms modeled as a conjugate surface pair.

Future work. We see several exciting avenues for future work. First, our current work focused on the geometry and kinematics of mpcMechs, and ignored dynamics. One important future work is to model the dynamics of mpcMechs, which will be useful for predicting how much workload the mechanism can carry in practice. Second, we have shown that our mpcMech alone can perform complex motions to accomplish functional tasks. This capability can be further extended by combining mpcMechs with other mechanisms such as high-pair linkages [Song et al. 2017] and non-circular gears [Xu et al. 2020]. Third, we assume that the input motion of a mpcMech is 1-DOF periodic rotation. In the future, we plan to generalize mpcMechs to support input motion with multiple DOFs (i.e., multiple drivers). By this, the generalized mpcMech will be able to perform varying dexterous tasks by controlling multiple DOFs of the input motion. Last but not least, we plan to explore more applications of our mpcMechs in engineering and robotics, such as animatronic eye mechanisms, reconfigurables [Tang et al. 2019], and legged locomotion.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments, Ruohan Geng, Haixin Zhu and Yanyan Yang for discussion on theory of dynamic form closure in the paper. This work was supported by the Singapore MOE AcRF Tier 2 Grant (MOE-T2EP20222-0008), the Singapore MOE AcRF Tier 1 Grant (RG12/22), and the National Natural Science Foundation of China (62025207).

REFERENCES

Kazuki Abe, Kenjiro Tadakuma, and Riichiro Tadakuma. 2021. ABENICS: Active Ball Joint Mechanism with Three-DoF Based on Spherical Gear Meshings. *IEEE Transactions on Robotics* 37, 5 (2021), 1806–1825.

Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. 2015. LinkEdit: Interactive Linkage Editing using Symbolic Kinematics. *ACM Trans. on Graph. (SIGGRAPH)* 34, 4 (2015), 99:1–99:8.

Shaoping Bai, Xuerong Li, and Jorge Angeles. 2019. A Review of Spherical Motion Generation using Either Spherical Parallel Manipulators or Spherical Motors. *Mechanism and Machine Theory* 140 (2019), 377–388.

Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. 2013. Designing and Fabricating Mechanical Automata from Mocap Sequences. *ACM Trans. on Graph. (SIGGRAPH Asia)* 32, 6 (2013), 186:1–186:11.

Bingkui Chen, Dong Liang, and Yane Gao. 2014a. Geometry Design and Mathematical Model of a New Kind of Gear Transmission with Circular Arc Tooth Profiles Based on Curve Contact Analysis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 228, 17 (2014), 3200–3208.

Bingkui Chen, Dong Liang, and Zhaoyang Li. 2014b. A Study on Geometry Design of Spiral Bevel Gears Based on Conjugate Curves. *International Journal of Precision Engineering and Manufacturing* 15, 3 (2014), 477–482.

Chih-Hsin Chen. 1978. On Theory of Conjugate Surfaces. In *Proc. the World Symposium on Gears and Transmissions*, Vol. A. 119–132.

Chih-Hsin Chen. 1985. *Fundamentals of the Theory of Conjugate Surfaces (in Chinese)*. Beijing: Science Press.

Chih-Hsin Chen. 1997. Geometro-Kinematical Analysis of Multi-point-conjugation Joint. *Mechanism and Machine Theory* 32, 5 (1997), 597–608.

Chih-Hsin Chen and Hong-Jian Chen. 1994. D.O.F. of Equivalent Conjugate Motion Between Two Bodies in a Mechanical System. *Mechanism and Machine Theory* 29, 8 (1994), 1143–1150.

Janet HongJian Chen and Chih-Hsin Chen. 2003. Conjugate Motion Analysis and Its Application in Human and Animal's Multi-point Conjugation Joints. In *Proc. Summer Bioengineering Conference.* 0543–0544.

Yingjie Cheng, Peng Song, Yukun Lu, Wen Jie Jeremy Chew, and Ligang Liu. 2022. Exact 3D Path Generation via 3D Cam-Linkage Mechanisms. *ACM Trans. on Graph. (SIGGRAPH Asia)* 41, 6 (2022).

Yingjie Cheng, Yucheng Sun, Peng Song, and Ligang Liu. 2021. Spatial-Temporal Motion Control via Composite Cam-follower Mechanisms. *ACM Trans. on Graph. (SIGGRAPH Asia)* 40, 6 (2021), 270:1–270:15.

Jordi Cornellà and Raúl Suárez. 2009. Efficient Determination of Four-point Form-closure Optimal Constraints of Polygonal Objects. *IEEE Transactions on Automation Science and Engineering* 6, 1 (2009), 121–130.

Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational Design of Mechanical Characters. *ACM Trans. on Graph. (SIGGRAPH)* 32, 4 (2013), 83:1–83:12.

Yan'e Gao, Bingkui Chen, and Dong Liang. 2014. Mathematical Models of Hobs for Conjugate-curve Gears Having Three Contact Points. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 229, 13 (2014).

G. Gatti and D. Mundo. 2007. Optimal Synthesis of Six-bar Cammed-linkages for Exact Rigid-body Guidance. *Mechanism and Machine Theory* 42, 9 (2007), 1069–1081.

Yujie Hou and Chao Lin. 2020. Kinematic Analysis and Experimental Verification of An Oval Noncircular Bevel Gears with Rotational and Axial Translational Motions. *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42 (2020), 60:1–60:11.

Yanan Hu, Chao Lin, Shuo Li, Yongquan Yu, Chunjiang He, and Zhiqin Cai. 2021. The Mathematical Model of Curve-Face Gear and Time-Varying Meshing Characteristics of Compound Transmission. *Applied Sciences* 11, 18 (2021), 8706:1–8706:26.

William Hutama, Peng Song, Chi-Wing Fu, and Wooi Boon Goh. 2011. Distinguishing Multiple Smart-Phone Interactions on a Multi-touch Wall Display using Tilt Correlation. In *Proc. ACM CHI*. 3315–3318.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

Steven G. Johnson. 2020. The NLopt nonlinear-optimization package. http://github.com/stevengj/nlopt.

Dieter Kraft. 1988. A Software Package for Sequential Quadratic Programming. *DLR German Aerospace Center-Institute for Flight Mechanics* (1988). DFVLR.

Kannan Lakshminarayan. 1978. *Mechanics of Form Closure*. Technical Report. Indian Institute of Technology Madras. issue: 78-DET-32.

Faydor L. Litvin and Alfonso Fuentes. 2004. *Gear Geometry and Applied Theory*. Cambridge University Press.

Yang Liu and J. Michael McCarthy. 2017. Design of a Linkage System to Write in Cursive. *Journal of Computing and Information Science in Engineering* 17, 3 (2017), 031015:1–031015:8.

Javier Roldán Mckinley, David B. Dooner, Carl Crane III, and Jean-Francois Kammath. 2005. Planar Motion Generation Incorporating a 6-Link Mechanism and Non-Circular Elements. In *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 393–403.

Vittorio Megaro, Bernhard Thomaszewski, Damien Gauge, Eitan Grinspun, Stelian Coros, and Markus Gross. 2014. ChaCra: An Interactive Design System for Rapid Character Crafting. 123–130.

Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. 2017. A Computational Design Tool for Compliant Mechanisms. *ACM Trans. on Graph. (SIGGRAPH)* 36, 4 (2017), 82:1–82:12.

B. Mishra, J. T. Schwartz, and M. Sharir. 1987. On the existence and synthesis of multifinger positive grips. *Algorithmica* 2 (1987), 541–558.

D. Mundo, G. Gatti, and D.B. Dooner. 2009. Optimized Five-bar Linkages with Non-circular Gears for Exact Path Generation. *Mechanism and Machine Theory* 44, 4 (2009), 751–760.

D. Mundo, J. Y. Liu, and H. S. Yan. 2006. Optimal Synthesis of Cam-Linkage Mechanisms for Precise Path Generation. *Journal of Mechanical Design* 128, 6 (2006), 1253–1260.

Domenico Prattichizzo and Jeffrey C. Trinkle. 2016. Grasping. In *Springer Handbook of Robotics*, Bruno Siciliano and Oussama Khatib (Eds.). Springer, Chapter 38, 955–988.

Robin Roussel, Marie-Paule Cani, Jean-Claude Léon, and Niloy J. Mitra. 2018. Exploratory Design of Mechanical Devices with Motion Constraints. 74 (2018), 244–256.

P. Somov. 1897a. Über Schraubengeschwindigkeiten eines festen Körpers bei verschiedener Zahl von Stützflächen. *Z. Math. Phys* 42 (1897), 133–153.

P. Somov. 1897b. Über Schraubengeschwindigkeiten eines festen Körpers bei verschiedener Zahl von Stützflächen. *Z. Math. Phys* 42 (1897), 161–182.

Peng Song, Chi-Wing Fu, Prashant Goswami, Jianmin Zheng, Niloy J. Mitra, and Daniel Cohen-Or. 2014. An Interactive Computational Design Tool for Large Reciprocal Frame Structures. *Nexus Network Journal* 16 (2014), 109–118.

Peng Song, Xiaofei Wang, Xiao Tang, Chi-Wing Fu, Hongfei Xu, Ligang Liu, and Niloy J. Mitra. 2017. Computational Design of Wind-up Toys. *ACM Trans. on Graph. (SIGGRAPH Asia)* 36, 6 (2017), 238:1–238:13.

Jianwei Sun, Lu Chen, and Jinkui Chu. 2016. Motion Generation of Spherical Four-bar Mechanism using Harmonic Characteristic Parameters. *Mechanism and Machine Theory* 95 (2016), 76–92.

Takuto Takahashi and Hiroshi G. Okuno. 2018. Design and Implementation of Programmable Drawing Automata based on Cam Mechanisms for Representing Spatial Trajectory. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. 450–455.

Rulong Tan, Bingkui Chen, Dong Liang, and Changyan Peng. 2017. Theoretical and Experimental Analyses of Spiral Bevel Gears with Two Contact Paths. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 232, 4 (2017).

Rulong Tan, Bingkui Chen, Changyan Peng, Dong Liang, and Dongyun Xiang. 2015. The Geometric Principles of General Spiral Bevel Gears of Local Bearing Contact from Spatial Conjugate Curves. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 231, 9 (2015), 1–13.

Keke Tang, Peng Song, Xiaofei Wang, Bailin Deng, Chi-Wing Fu, and Ligang Liu. 2019. Computational Design of Steady 3D Dissection Puzzles. *Comp. Graph. Forum (Eurographics)* 38, 2 (2019), 291–303.

Pengbin Tang, Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2020. A Harmonic Balance Approach for Designing Compliant Mechanical Systems with Nonlinear Periodic Motions. *ACM Trans. on Graph. (SIGGRAPH Asia)* 39, 6 (2020), 191:1–191:14.

Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational Design of Linkage-Based Characters. *ACM Trans. on Graph. (SIGGRAPH)* 33, 4 (2014), 64:1–64:9.

Hao Xu, Tianwen Fu, Peng Song, Mingjun Zhou, Chi-Wing Fu, and Niloy J. Mitra. 2020. Computational Design and Optimization of Non-Circular Gears. *Comp. Graph. Forum (Eurographics)* 39, 2 (2020), 399–409.

Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2018. Bend-It: Design and Fabrication of Kinetic Wire Characters. *ACM Trans. on Graph. (SIGGRAPH Asia)* 37, 6 (2018), 239:1–239:15.

Bowen Yu and Kwun-Lon Ting. 2013. Free-Form Conjugation Modeling and Gear Tooth Profile Design. *Journal of Mechanisms and Robotics* 5, 1 (2013), 011001:1–011001:9.

Lu-he Zhang, Shi-xuan Li, Jian Huang, and Bingkui Chen. 2019. Theoretical and Experimental Investigations of Gear Transmission with Geometric Elements Constructed Tooth Pairs Having Triple Contact Points. *International Journal of Precision Engineering and Manufacturing* 20 (2019), 1197–1206.