Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Conjugate shape simplification via precise algebraic planar sweeps toward gear design

Jinesh Machchhar[a,*], Henry Segerman[b], Gershon Elber[c]

[a] Indian Institute of Technology Mandi, Himachal Pradesh, India
[b] Oklahoma State University, Stillwater, OK, USA
[c] Technion - Israel Institute of Technology, Haifa, Israel

ARTICLE INFO

Article history:
Received May 3, 2020

Keywords: Swept volumes, B-splines, algebraic computation, freeform shapes, constraint solving

ABSTRACT

Gears play a pivotal role in machine design. This paper proposes an algorithm to simplify the shapes of planar gears. This is achieved via iterative conjugation, using precise algebraic sweeps. The notion of shape simplification is introduced in a mathematically rigorous manner and it is shown that the conjugation process converges, yielding a pair of meshing gears that follow the desired motion. Simplified gear shapes may lead to improved mechanical characteristics and reduction in manufacturing costs.

The generality of algebraic sweeps allows precise design of gears with freeform shapes and non-uniform motion transmission. Moreover, the computational framework proposed in this paper is versatile, with applications beyond gear design. A variety of examples from an implementation of our algorithm, that offers topological guarantees, are presented, which demonstrate the robustness and efficacy of our approach.

1. Introduction

This paper studies conjugation through the problem of design of a pair, G, H, of meshing gears in $\mathbb{R}^2$. Here conjugation refers to swept volume computation followed by Boolean-negation operation. An example is shown in Figure 1 illustrating a cam-follower mechanism. A cam is a non-circular disk, which rotates about a pivot, and transmits motion to the follower via tangential contact. The follower is restricted to undergo reciprocating motion about a line. The shape of H is obtained as the conjugate of G. We reverse the roles of G and H and repeat. This leads to simplified shapes of G and H. The method of conjugation also allows for design of non-circular gears which have non-uniform motion transmission [1]. Moreover, the conjugation operation has a wide field of application in design of kinematic pairs wherein two bodies are in motion while maintaining tangential contact. For instance, cam-follower mechanisms [2], or replacement parts for joints in human body [3], to name a few. In order to perform conjugation, a robust computational framework for sweeps is proposed. Sweeping is a fundamental geometric primitive with diverse applications such as machining verification [4] and collision detection [5].

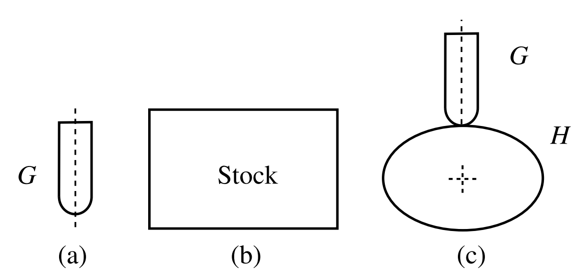

Fig. 1: Design of cam-follower mechanism using conjugation. The follower, H, shown in (a) is swept along an elliptical path, while orienting it so that its axis stays aligned with the normal to the ellipse. The swept volume thus obtained is Boolean-subtracted from the stock shown in (b) to obtain the elliptical cam, G, shown in (c) along with the follower, H. The follower's motion is restricted to be along its axis. As the cam rotates about its center, it displaces the follower, which stays in tangential contact.

Gears transmit motion via tangential contact. Typically, one

*Corresponding author:
e-mail: jinesh@iitmandi.ac.in (Jinesh Machchhar)

of the pair, say G, acts as the driving gear, whose uniform circular motion is translated into the desired motion for the driven gear, H. Such a pair is illustrated in Figure 2(a) and (b). The overall shapes as well as the geometry of the teeth of G, H dictate the motion profile of H. In other words, together, the shape of G and the relative motion M between G and H, implicitly encode the shape of H. Such a pair of meshing gears is an example of a conjugate pair. In contrast to traditional methods for design of gears and other conjugate shapes, which terminate after a single step of conjugation, our approach involves multiple iterations of conjugation. The shape of H obtained in the previous step is employed to compute a refined version of G. This process is repeated. We show that the iterations converge after at most two steps. As a result, the shapes of G, H are simplified, without altering the relative motion M. This is illustrated by an example shown in Figure 2. The shape H_0 shown in (b) is obtained as the conjugate of G_0 shown in (a), along M. H_0 is further conjugated along M^{-1} to obtain G_1, shown in (c). Likewise, G_1 is used to compute H_1 shown in (d). G_0, G_1 and H_0, H_1 are shown in overlapping positions in (e) and (f). As can be seen, G_1 is a simplified version of G_0. Since H_1 is identical to H_0, the conjugation process has converged. These notions are formalized in Section 3.

One way to approximate H as the conjugate of G would be via a series of Boolean operations. A more precise approach is using swept volumes. Under this scheme, the swept volume of G under M is computed and Boolean-subtracted from a block to obtain H. A number of approaches for swept volume computation have been proposed previously, but they all lack one or the other key ingredient, preventing a general and practical implementation. In particular, the requirements of tight numerical tolerances and topological completeness render the previous approaches unsuitable for our purpose. We propose a swept volumes framework based on algebraic computation, upon which the conjugation algorithm is built. Use of B-spline functions for constructing algebraic equations aids in precise modeling of the envelope condition, only to be fed to robust constraint solvers which return the solution with the prescribed numerical tolerance and topological guarantee.

The contribution of this work is threefold. To our knowledge, this is the first attempt at shape simplification for conjugate geometries. A shape is simplified so that any region on its boundary, which does not come in contact with the conjugate, is excised. In the context of gears, this leads to tighter meshing. Secondly, our framework accommodates design of gears with freeform shapes, which allow non-uniform motion transmission. Unlike previous works on gear-design, our algorithm provides a complete computational framework with numerical guarantees. Handling of local and global self-intersections ensures correctness of output in difficult cases, for instance, when the geometry of all teeth is not identical. Finally, we propose a robust computer implementation for sweeps in 2D. Our algebraic approach is based on B-spline functions and use of state-of-the-art numerical solvers to provide guarantees on the numerical precision and topological completeness, lacking in previous works.

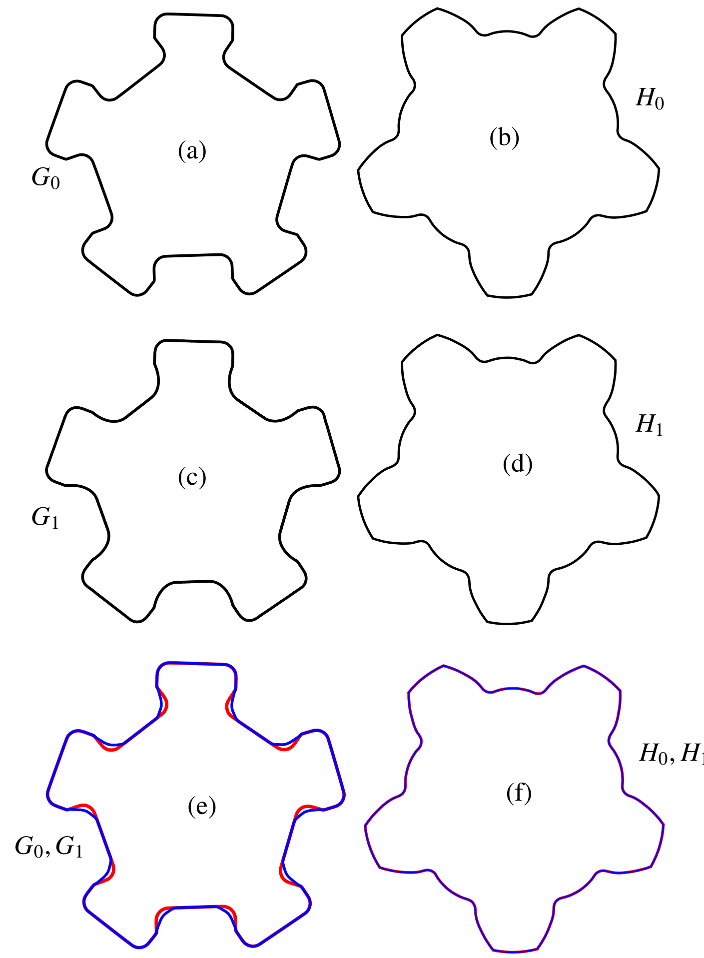

Fig. 2: (a) The input gear G_0 which has five block-like teeth. It is being swept along motion M which consists of rotation by an angle of 2π around the center of G_0 and translation along a circle of radius twice the radius of G_0 about center of H_0, with uniform speed. (b) The gear H_0 obtained as the conjugate of G_0 shown in (a), along M. (c) The gear G_1 obtained as the conjugate of H_0 shown in (b), along M^{-1}. (d) The gear H_1 obtained as the conjugate of G_1 shown in (c) along M. (e) Gears G_0 and G_1 shown in overlapping positions, in red and blue respectively. Clearly, they are not congruent. (f) Gears H_0 and H_1 are shown in overlapping positions, in red and blue respectively, and are identical. Thus the conjugation process has converged.

The rest of the paper is organized as follows. In Section 2, we survey related previous work on gear design and swept volumes. In Section 3, our approach of iterative conjugation is discussed, assuming that the initial shape G and the relative motion M are given. Section 4 discusses the design of non-circular gears. Some examples from an implementation of our algorithm in the IRIT [6] kernel are presented, in Section 6. The paper is concluded, in Section 7, with remarks on possible extensions of this work.

2. Previous work

Designing of gears is well-studied [1, 7, 8, 9, 10]. One of the earliest attempts at computing conjugate gears using sweeps was by Litvin [11], wherein, the necessary condition for meshing of gears is characterized. This condition is well-known as the *envelope condition* in the literature on sweeps [12] and involves the dot-product of the normal and the velocity at a point under consideration. Puccio et al. [13] propose an alternative description of the condition of meshing based on vectorial notation. The above methods are restricted to circular gears. Here circle refers to the overall shape of the gear, without teeth. Design of non-circular gears is considered by Litvin et al. [14] by computing the *centrodes* of the gears from the motion profiles. Centrodes of a pair of meshing gears are the curves which represent the overall shape of the gears, without teeth, along which they make contact without slipping. Zarebski et al. [15] propose a method for designing non-circular gears wherein the envelope computation is done approximately without any error-bounds. Bendefy et al. [16] design gears with varying gear-ratio and center-distance. In their method, different portions of teeth such as the flank, top and bottom land are constructed separately. A common limitation of all the previous methods is a lack of computational framework by which to construct the shape of the conjugate gear with the prescribed numerical precision and topological guarantee. In particular, solving the envelope condition has been one of the primary computational bottlenecks in previous approaches. The envelope condition poses an underconstrained system of algebraic constraints whose solution remains a challenging task. Bonandrini et al. [17, 18] propose a computational framework which is limited to design of circular gears.

Johann et al. [19] study the geometry of the conjugate flanks of gear teeth of arbitrary shape. The *flank* of a tooth is the leading edge of the tooth which is in contact. In their work, the focus is on local contact while the arrangement of teeth and global self-intersections resulting therefrom are not studied. Likewise, Litvin et al. [20] characterize the singularities on the envelopes of gear tooth surfaces. This approach handles local self-intersections but not global self-intersections. Resolving global self-intersections is especially important when the gears are not circular and the geometry of all the teeth may not be identical [21].

Swept volume generation, which forms the backbone of our algorithm, is a classical problem in solid modeling [22, 23, 24]. We survey some of the prominent works in this area. Blackmore et al. [4, 25] formulate the boundary of the swept volume as the solution of a differential equation. This requires the input shape to be in implicit form and the output surface is constructed by interpolating sampled points. Such an approach lacks bounds on the approximation error. Abdel-Malek and Yeh [26] propose a swept volume method based on rank deficiency condition of the Jacobian of the sweep map. This method readily generalizes to arbitrary dimensions. However, their approach for finding solutions restricts the input to analytic shapes. Erdim and Ilies [27, 5] give a membership test for a candidate point to belong inside, outside or on the boundary of the swept volume. Singularities on the envelope in 2D case are also identified. Their approach requires performing curve-curve intersection for each query point. Such an approach yields a complete characterization of the boundary of the swept volume, but is computationally expensive. Rossignac et al. [28] compute the boundary of the swept volume by restricting the input motion to be a screw motion. While this leads to an efficient algorithm, the limitation is clear, namely, the class of admissible motions. Zhang et al. [29] give a method for fast computation of swept volumes but which is restricted to polygonal input solids. Peternell et al. [30] obtain a set of sampled points on the boundary of the swept volume. They derive a formula for the evolution of curves of contact, which helps bound the distance between any two consecutive curves of contact. This approach, however, does not give a guarantee on the completeness of the output. Wallner et al. [31] propose a method for swept volume computation along motion specified by a set of discrete pose cloud, however, their approach is limited to polyhedral shapes. Adsul et al. [32] propose a computational framework for swept volumes in parametric boundary representation format with analysis of local and global self-intersections. However, no topological guarantee is given on the completeness of the output. In summary, a robust implementation of sweeps with a high degree of numerical precision is missing. A method either restricts the class of inputs, or approximates the output without bounds on error. Moreover, to our knowledge, no previous approach addresses the issue of solving the envelope condition - which is central to swept volumes - with numerical and topological guarantees. Our approach alleviates these issues.

3. Iterative conjugation

The gears G and H undergo one parameter family of rigid motions M_G and M_H, respectively, while making tangential contact with each other. The system (G, H, M_G, M_H) is equivalent, up to some rigid transform, to the system $(G, H, M_H^{-1} \circ M_G, I)$, wherein the relative motion between G and H is applied to G, while H stays stationary. We will denote the relative motion $M_H^{-1} \circ M_G$ by M. Similarly, another equivalent representation would be $(G, H, I, M_G^{-1} \circ M_H)$. All the discussion in this paper is with respect to some fixed, global coordinate system. The input to the iterative conjugation algorithm is an initial design for G, and the relative motion M. We employ the boundary representation (B-rep) for G.

We will denote the boundary and the interior of a shape G by ∂G and G^o, respectively. It is assumed that ∂G is free of self-intersections. For example, G is represented by its boundary in Figure 2(a). G moves along a one parameter family of rigid motions M, defined as follows:

Definition 1. A **one parameter family of rigid motions**, M, in $\mathbb{R}^2$, is a map $M : [0, 1] \rightarrow (SO(2), \mathbb{R}^2)$, such that $M(t) = (A(t), b(t))$, where $A(t)$ is a 2×2 rotation matrix and $b(t)$ is a translation vector in $\mathbb{R}^2$. The parameter t in this definition represents time.

The action of M on a point $p \in \mathbb{R}^2$ at time $t \in [0, 1]$ is given by $A(t)p + b(t)$ and is denoted by $p_M(t)$. Likewise, the action of M on G at time t is denoted by $G_M(t)$ and obtained as $\{A(t)p + b(t) | p \in G\}$. The velocity at $p_M(t)$ will be denoted by $p'_M(t)$ and is computed as $A'(t)p + b'(t)$, where $'$ denotes the derivative with respect to t. If $n^G(p)$ is the outward normal to ∂G at $p \in \partial G$, then the outward normal to $G_M(t)$ at $p_M(t)$ is given by $A(t)n^G(p)$ and denoted by $n^G_M(p, t)$.

The computation of the geometry of H is performed by computing the swept volume of G along M, defined as follows:

Definition 2. The **swept volume** of G along a one parameter family of rigid motions M is defined as $\bigcup_{t\in[0,1]} G_M(t)$ and denoted by $V(G, M)$.

Thus, $V(G, M)$ is the infinite union of all the transforms of G along M. Like G, we also use the B-rep for $V(G, M)$. Thus, it suffices to compute $\partial V(G, M)$ in order to obtain a complete representation of $V(G, M)$. We make the following assumption for the ease of exposition, which holds from now on.

Assumption 3. *It is assumed that $G_M(0) = G_M(1)$, i.e., the initial and final position of G under M, coincide. Further, $V(G, M)$ is assumed to be homeomorphic to an annulus.*

Conjugation involves computing the swept volume $V(G, M)$, which is *carved*, i.e., Boolean-subtracted, from the *stock* of H, to obtain H. The actual shape of the stock is of little relevance as long as it is large enough. Whenever carving, we assume the stock to be the Euclidean plane, $\mathbb{R}^2$. Since $V(G, M)$ is homeomorphic to an annulus, $\mathbb{R}^2 \setminus V(G, M)$ has two components, one finite and another infinite. Only the finite component is of interest, the other one is discarded. We will denote the finite component of $\mathbb{R}^2 \setminus V(G, M)$ by $\overline{V(G, M)}$.

Our approach involves multiple iterations of conjugation. Each iteration involves computing the swept volume of G (H) and carving it from $\mathbb{R}^2$ to obtain the next version of H (G). The input to the algorithm is the initial version, G_0 for G, and the relative motion M. In the first iteration, $V(G_0, M)$ is conjugated to obtain H_0, i.e. $H_0 = \overline{V(G_0, M)}$. In the next iteration, G_1 is obtained as $\overline{V(H_0, M^{-1})}$. This process is repeated until $G_i = G_{i+1}$ and $H_i = H_{i+1}$ for all $i > m$ for some $m \geq 0$. We later show, in Proposition 12, that no more than two iterations are necessary. The iterative conjugation process is summarized below.

$$H_i = \overline{V(G_i, M)}, \qquad i \geq 0, \tag{1}$$

$$G_i = \overline{V(H_{i-1}, M^{-1})}, \qquad i \geq 1. \tag{2}$$

By construction, $\partial H_i = \partial\overline{V(G_i, M)}, i \geq 0$ and $\partial G_i = \partial\overline{V(H_{i-1}, M^{-1})}, i \geq 1$. An example of iterative conjugation is shown in Figure 2 wherein G_0, H_0, G_1 and H_1 are shown in (a), (b), (c) and (d).

The computation of $\partial V(G_i, M)$ is done via the well-known envelope condition [33] which states that a point $p_M(t) \in \partial G_{i_M}(t)$ belongs to $\partial V(G_i, M)$ only if the velocity at $p_M(t)$ is tangent to $\partial G_{i_M}(t)$.

Definition 4. A point $p_M(t) \in \partial G_M(t)$ is said to satisfy the **envelope condition** if $\mathcal{E}(p, t) := \left\langle p'_M(t), n^G_M(p, t)\right\rangle = 0$. The set of points satisfying the envelope condition will be referred to as the **envelope**.

Figure 3 illustrates an example wherein a circular disc is being swept along a parabolic path, shown as a dotted curve. The envelope curves are shown in black, green and red. Note that this example is given only to clearly illustrate the concept of swept volumes and does not satisfy Assumption 3, i.e., $G_M(0) \neq G_M(1)$.

Not all points in the envelope belong to the boundary of the swept volume. The envelope may contain self-intersections [24] which need to be trimmed away in order to obtain the boundary of the swept volume.

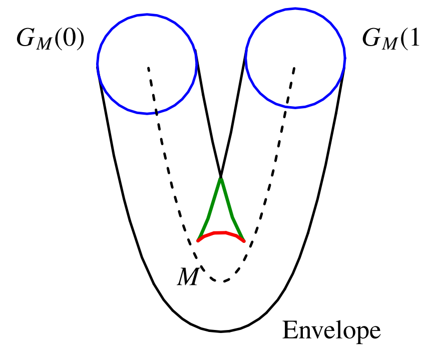

Fig. 3: Swept volume computation for a circular disc undergoing translation along a parabola, shown as a dotted curve. The disc is shown at initial and final positions in blue. The envelope curves are shown in black, green and red. The boundary of the swept volume, which is a subset of the envelope curves, is shown in black. Portions of envelope with local and global self-intersection are shown in red and green.

Definition 5. While sweeping G_i along M, the envelope is said to have **self-intersection** at a point p if there exists $t \in [0, 1]$ such that $p_{M^{-1}}(t) \in G^o_i$.

The above definition is equivalent to saying that $p \in G^o_M(t)$ for some $t \in [0, 1]$. The following lemma states that such points do not belong to the boundary of the swept volume. In the example shown in Figure 3, the portion of envelope shown in red and green has self-intersection.

Lemma 6. *While sweeping G_i along M, if the envelope has a self-intersection at a point q, then $q \notin \partial V(G_i, M)$.*

Proof. If $p := q_{M^{-1}}(t) \in G^o_i$, then $p_M(t) = q$. Since $p \in G^o_i$, $q \in V(G_i, M)$ but $q \notin \partial V(G_i, M)$. The point q being obscured by an interior point of $G_{i_M}(t)$, is not on the boundary of the swept volume. □

The set $\partial V(G_i, M)$ is obtained after trimming away points of self-intersection from the envelope. In the example of Figure 3, the set $\partial V(G, M)$ is shown in black. It follows that if a point, q, satisfies the envelope condition and is free from self-intersection, then $q \in \partial V(G_i, M)$.

There may exist points in ∂G_0 which do not give rise to any point in $\partial V(G_0, M)$, i.e., for some $p \in \partial G_0$, $p_M(t) \notin \partial V(G_0, M)$, for all $t \in [0, 1]$. The reason being, p may either fail to satisfy the envelope condition in Definition 4 or may get obscured by self-intersections as in Definition 5. Interestingly, Lemma 8 states that all the points in ∂H_i give rise to some point in $\partial V(H_i, M^{-1})$.

Lemma 7. *If $q = p_M(t_0)$ for some $t_0 \in [0, 1]$, then $q'_{M^{-1}}(t_0) = -p'_M(t_0)$.*

For a proof refer to Appendix A.

Lemma 8. *For every $q \in \partial H_i, i \geq 0$, there exists $t \in [0, 1]$ such that $q_{M^{-1}}(t) \in \partial V(H_i, M^{-1})$.*

Proof. Fix a point $q \in \partial H_i$. Since $q \in \partial V(G_i, M)$, there exists $p \in \partial G_i$ and $t_1 \in [0, 1]$ such that $p_M(t_1) = q$. In other words, $p_M(t_1)$ satisfies the envelope condition, i.e.,

$$\left\langle p'_M(t_1), n^{G_i}_M(p, t_1)\right\rangle = 0. \tag{3}$$

We first show that while sweeping H_i along M^{-1}, the point $q_{M^{-1}}(t_1)$ satisfies the envelope condition. From Lemma 7 we know that $q'_{M^{-1}}(t_1) = -p'_M(t_1)$. Also, it can be shown that $n^{H_i}_{M^{-1}}(q, t_1) = -n^{G_i}_M(p, t_1)$. Hence,

$$\left\langle q'_{M^{-1}}(t_1), n^{H_i}_{M^{-1}}(q, t_1)\right\rangle = \left\langle p'_M(t_1), n^{G_i}_M(p, t_1)\right\rangle = 0. \quad (4)$$

Thus, we have shown that the point $q_{M^{-1}}(t_1) = p$ satisfies the envelope condition. We now show that the point $q_{M^{-1}}(t_1)$ is free of self-intersection. Suppose not, i.e., by Definition 5, there exists some $t_2 \in [0, 1]$ such that $p_M(t_2) \in H^o_i$. In other words, $p_M(t_2) \notin V(G_i, M)$. However, since $p \in \partial G_i$, we have, for all $t \in [0, 1]$, $p_M(t) \in V(G_i, M)$. This leads to a contradiction. Thus, it is proved that $q_{M^{-1}}(t_1)$ satisfies the envelope condition and is free of self-intersection, i.e., $q_{M^{-1}}(t_1) \in \partial V(H_i, M^{-1})$. □

Lemma 8 states that all points of $\partial H_i, i \geq 0$, lead to some point on the boundary of the swept volume $\partial V(H_i, M^{-1}) = \partial G_{i+1}$. This happens because each point of $\partial H_i, i \geq 0$ itself is resulting from the sweep of G_{i-1}. The following corollary makes a similar claim for $G_i, i \geq 1$.

Corollary 9. *Every point in ∂G_i leads to some point in ∂H_i for $i \geq 1$.*

The proof is symmetric to that of Lemma 8, stated for $\partial H_i, i \geq 1$. It is important to note that Corollary 9 may not hold for $i = 0$, as explained in the paragraph before Lemma 7.

Proposition 10. *For all $i \geq 0$, if $p \in \partial G_i$ produces a point $p_M(t_0) \in \partial H_i$ for some $t_0 \in [0, 1]$, then $p \in \partial G_{i+1}$.*

Proof. From the proof of Lemma 8, it follows that the point $q := p_M(t_0) \in H_i$ produces the point $q_{M^{-1}}(t_0) \in \partial G_{i+1}$. Since $q_{M^{-1}}(t_0) = p$, we have that $p \in \partial G_{i+1}$. □

Proposition 10 precisely identifies the set of points in G_i that remain invariant under the conjugation process described by Equations (1) and (2), i.e., the set $G_i \cap G_{i+1}$. This is illustrated in the example of Figure 2 by showing G_0 and G_1 in overlapping positions in (e). This provides useful guidelines about how to go about refining the input, G_0, in order to obtain the desired pair of gears.

Corollary 11. *For all $i \geq 0$, if $q \in \partial H_i$ produces a point $q_{M^{-1}}(t_0) \in \partial G_{i+1}$ for some $t_0 \in [0, 1]$, then $q \in \partial H_{i+1}$.*

Proposition 12. *$G_i = G_{i+1}$, $i \geq 1$ and $H_i = H_{i+1}$, $i \geq 0$.*

Proof. That $G_i = G_{i+1}$, $i \geq 1$ follows from Corollary 9 and Proposition 10. That $H_i = H_{i+1}$, $i \geq 0$ follows from Lemma 8 and Corollary 11 . □

Proposition 12 says that the iterative conjugation process converges after at most two steps. In the example of Figure 2, H_0 and H_1 are shown in overlapping positions in (f), in red and blue. Since H_1 is identical to H_0, the conjugation process has converged.

4. Non-circular gears for non-uniform circular motion

In the previous section, the method of iterative conjugation is exemplified using circular gears. We now focus on non-circular gears. Such gears enable conversion of uniform circular motion of the driving gear, say G, into the desired non-uniform circular motion of the driven gear, H. For the ease of discussion, we again assume that the relative motion, M, between G and H is given, so that H remains stationary. The design now proceeds in two steps. In the first step, the overall shape of H, i.e., the shape of H without teeth, is computed. This shape is referred to as the *centrode* of H in previous literature [1]. In the second step, teeth are arranged along the centrode of H, and G is obtained as the conjugate of H.

In order to determine the centrode of H, we use the fact that as the centrode of G moves along M, it makes tangential contact with the centrode of H, without slipping. In other words, at any time instant t_0, the velocity of the point on the centrode of G, which is in contact with that of H, is zero [1]. We call such a point, a *stationary point* at t_0. If we trace the stationary point as the function of time t, we obtain the curve along which G, H make tangential contact, with H stationary, i.e., we obtain the centrode of H. This curve may be obtained as the solution to the following algebraic equality, which is already well-known [1].

$$w : \mathbb{R}^2 \times [0, 1] \to \mathbb{R}^2, w(p, t) := A'(t)p + b'(t) = 0 \quad (5)$$

Equation (5) is a system of two algebraic equalities in three variables, viz, $p := (x, y)$ and t. We again employ the constraint solvers [34] to obtain its solution curve. It is easy to see that the solution set is non-empty if the rotation $A(t)$ is not constant. Further, if motion M is periodic, then the centrode is a closed curve. Note that periodicity of M is stronger than Assumption 3 and further requires the derivatives at the end-points to coincide.

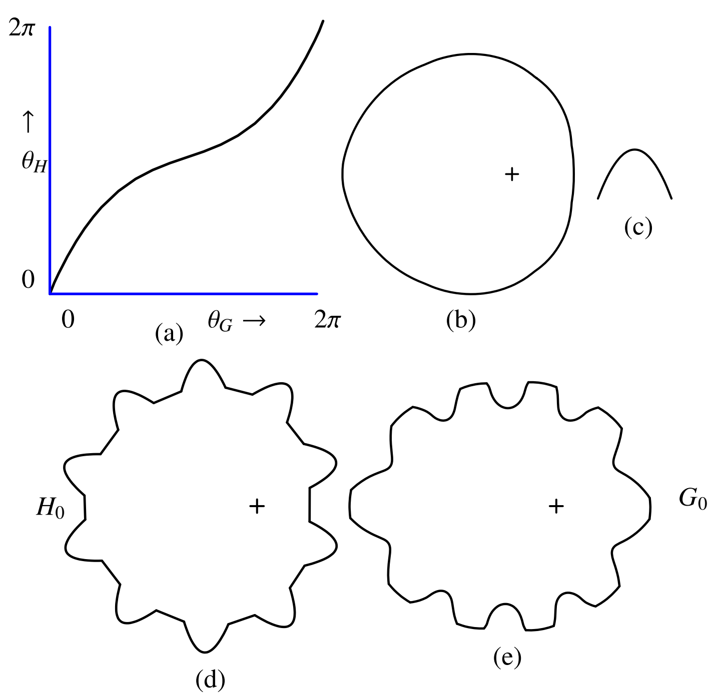

Fig. 4: Design of non-circular gears. (a) Angular displacement of H as a function of that of G. (b) The centrode for H obtained by solving Equation (5). (c) Input tooth profile for H_0, modeled as a freeform B-spline curve. (d) Gear H_0 obtained by arranging 10 copies of tooth shown in (c) along the centrode in (b). (e) The gear G_0 obtained as the conjugate of H_0.

Once the centrode of H is computed, teeth are arranged along this curve. The geometry and the number of teeth are specified by the user. This gives us the initial shape, H_0 for H, which may be conjugated along M^{-1} to obtain G_0. The iterations of conjugation continue till convergence, as discussed in Section 3. An example is illustrated in Figure 4. The relative motion between G, H is specified as a map between the angular displacement of G and that of H in (a). The centrode of H obtained as the solution of Equation (5) is shown in (b). The copies of the freeform tooth profile shown in (c) are arranged along the centrode of H to obtain the gear H_0 shown in (d). The gear G_0 obtained as the conjugate of H_0 along M^{-1} is shown in (e). Note that, if the tooth profile is being arranged along a circular centrode, as is the case in the example shown in Figure 2, one may exploit the circular symmetry and reduce the amount of computation. However, this does not hold for centrodes which do not have such symmetry, for instance, the example shown in Figure 4. In this case the tooth profile in the conjugate gear varies, as can be seen in Figure 4(e).

5. Algebraic swept volume computation

The conjugation operation described in previous sections is based upon precise computation of swept volumes. In this section, we describe the computation of $V(G, M)$, as in Definition 2. B-spline functions are used for representing ∂G. The translation $b(t)$ as well as the entries of the rotation matrix $A(t)$ of Definition 1 are also represented using B-spline functions. The entries of $A(t)$ involve trigonometric functions which we approximate with B-splines by employing the method of [35]. While rational splines may be able to precisely represent circles, the parametrization is non-uniform and does not serve the purpose in our case. The output, $\partial V(G, M)$, is again represented using B-spline functions, within prescribed tolerance. Our algebraic approach ensures numerical precision and topological guarantee and consists of four major steps. The first step involves obtaining the solution set of the envelope condition given in Definition 4. This is described in Section 5.1. The solution thus obtained is oriented so that the enclosed swept volume is on the left side of the curve. This is described in Section 5.3. Next, local self-intersections in the envelope are excised, which is explained in Section 5.4. Finally, Section 5.5 explains the construction of $\partial V(G, M)$ which involves resolving global self-intersections.

5.1. Solution to the envelope equation

Recall from Section 3 that the necessary condition for a point on ∂G to belong to the boundary of $V(G, M)$ is to satisfy the envelope condition given by Definition 4, which we now write in parametric form.

Definition 13. The gear G is represented by its boundary, which is a closed, regular, C^1-continuous parametric curve, $g : [0, 1] \to \mathbb{R}^2$, i.e., $g(0) = g(1)$ and $\frac{dg}{dr}(r_0) \neq (0, 0), \forall r_0 \in [0, 1]$. Here r is the parameter of the curve g.

From Definitions 1, 4 and 13, it follows that a point (r_0, t_0) in the parametric space $[0, 1] \times [0, 1]$ satisfies the envelope condition if it satisfies the following equation.

$$f(r, t) := \mathcal{E}(g(r), t) = 0. \tag{6}$$

Since Equation (6) has two variables, viz. r, t, the solution, in general, is of dimension one, i.e., a set of curves in the parameter space, (r, t). Equation (6) involves the dot-product of velocity and normal. The velocity at a point (r, t) is computed as $A'(t)g(r) + b'(t)$, which is represented using B-spline functions. Recall from Section 3 that the normal is computed as $A(t)n^G_M$, which is again represented as B-spline functions. Since B-spline functions are closed under addition and multiplication [36], Equation (6) is represented using B-spline functions, precisely.

We employ the constraint solver by Barton et al. [34] to obtain the solution to Equation (6). This solver returns solutions to under-constrained systems of algebraic equations with numerical and topological guarantee. The returned solution is in the form of a sequence of connected points, each of which satisfies the prescribed constraints up to user-given numerical tolerance. The distance between consecutive points too is governed by user-given step-size. More importantly, the solver guarantees that no portion, topologically, of the solution is left out, up to the prescribed tolerance. This, in turn, ensures that no portion of the envelope is missed out, up to prescribed tolerance. The solution points, which are in parametric space, are mapped into the object space via the map $(r, t) \mapsto A(t)g(r) + b(t)$. We fit a cubic B-spline curve to the resulting points to obtain the envelope curves, as in Definition 4.

In the simple example shown in Figure 3, the envelope curves are shown in black, green and red. A non-trivial example is illustrated in Figure 5. The input gear G shown in (a) is undergoing rigid motion M which is composed of rotation about the center of G and translation along a circular path. The set of solution curves of Equation (6), mapped from parameter space (r, t) to object space (x, y), is shown in (b).

5.2. Handling C^1-discontinuities

The envelope condition prescribed by Equation (6) in Section 5.1 requires ∂G to be C^1-continuous. In this section, we describe the computation of envelope curves arising from C^1-discontinuities in ∂G. Such points will be referred to as *sharp* points in ∂G. Unlike a regular point in ∂G, a sharp point in ∂G has a cone of normals. For instance, at the point p in the schematic shown in Figure 6, ∂G has a cone of outward normals bounded by n_1, n_2. The point p generates a set of envelope curves which may be parameterized as $t \mapsto A(t)p + b(t)$. Any such curve is bounded by end-points which satisfy $\left\langle p'_M(t), A(t)n_1 \right\rangle = 0$ or $\left\langle p'_M(t), A(t)n_2 \right\rangle = 0$.

5.3. Orientating envelope curves

The boundary $g(r)$ of input G is oriented so that the interior G^o lies on the left side of the curve. The envelope curves obtained in Section 5.1 and 5.2 must be oriented in a consistent manner. Consider a point (r_0, t_0) of the envelope curve and let $A(t_0)g(r_0) + b(t_0) = q_0$. The curve is oriented so that $G_M(t_0)$ is on the left side of the envelope curve at the point q_0. By Definition 2, this ensures that the swept volume is on the left side of its boundary curve.

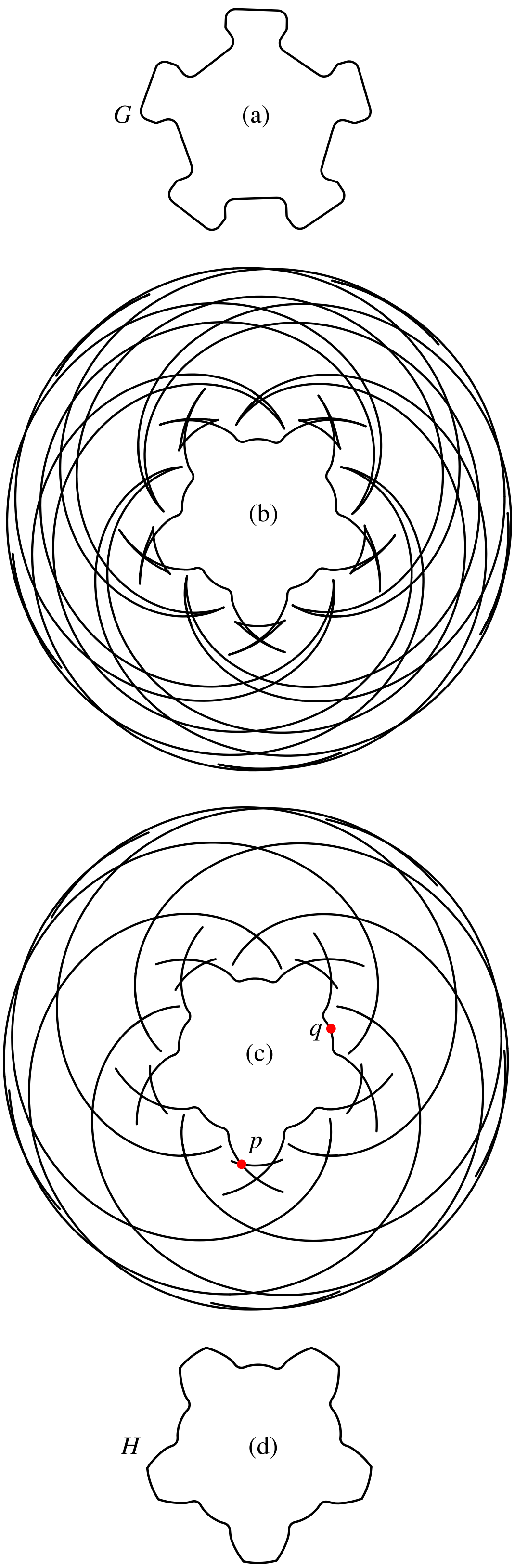

Fig. 5: Swept volume construction. (a) Input G which undergoes motion with rotation about its center and translation along a circle. (b) The envelope curves. (c) Envelope curves after trimming local self-intersections. (d) Portion of the swept volume boundary, enclosing the finite region.

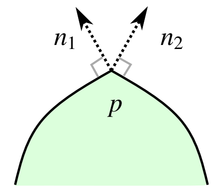

Fig. 6: A point $p \in \partial G$ with C^1-discontinuity has a cone of outward normals.

5.4. *Excising local self-intersections*

Local self-intersections are limiting cases of self-intersections as in Definition 5 and lead to singularities in the envelope. In regions with local self-intersections, the envelope loses orientability.

Definition 14. Suppose a point $p = q_M(t_0)$ in the envelope has self-intersection. We say that p has *local self-intersection* if $\forall \epsilon > 0, \exists t_1 \in [0, 1]$ such that $\|t_0 - t_1\| < \epsilon$ and $p \in G^o_M(t_1)$.

Local self-intersections in the envelope are well-understood [12, 27, 32, 37]. Unlike global self-intersections, local self-intersections may be detected by querying local data at time t_0. Local self-intersections may only arise in the envelope curves described in Section 5.1. The envelope curves arising from the sharp points of input, as described in Section 5.2 are free of local self-intersections. While a detailed exposition on local self-intersections is beyond the scope of this paper, here we briefly outline the mathematical function which detects local self-intersections. For details, the reader is referred to previous works [12, 27, 32] which discuss this issue in detail.

Let (r_0, t_0) be a point which satisfies Equation (6). Since the velocity, $\mathcal{V}(r_0, t_0) = A'(t_0)g(r_0) + b'(t_0)$, is orthogonal to the normal to ∂G at this point, we have that the velocity $\mathcal{V}$ and the tangent, $\mathcal{T}(r_0, t_0) = A(t_0)g'(r_0)$ to ∂G, are linearly dependent. In other words, there exists a non-zero real $\alpha(r_0, t_0)$ such that $\mathcal{V}(r_0, t_0) = \alpha(r_0, t_0)\mathcal{T}(r_0, t_0)$. The following lemma uses Equation (6) and furnishes a test for local self-intersections which, in effect, compares the curvature of ∂G with that of the motion M at a point on the envelope. In the following lemma, f_r and f_t refer to the derivatives of f, as in Equation (6) with respect to r and t.

Lemma 15. *A point (r_0, t_0) satisfying the envelope condition has local self-intersection if $\alpha(r_0, t_0)f_r(r_0, t_0) - f_t(r_0, t_0) < 0$.*

For proof refer to [32], Theorem 34. In the example shown in Figure 3, the portion of the envelope with local self-intersections is shown in red. The function in Lemma 15 is computed in closed form. Hence, the trimming of regions with local self-intersections is fast and numerically robust. We continue the example of Figure 5 for illustration. The envelope curves shown in (b), after trimming away local self-intersections, are shown in (c). This aids in reducing the computational complexity of the next step.

5.5. *Stitching the boundary of the swept volume*

At this stage, the envelope curves are oriented, and regions with local self-intersections discarded. It now remains to trim regions with global self-intersections and to stitch together curve segments which bound the swept volume. Recall from Assumption 3 that the swept volume is homeomorphic to an annulus which partitions $\mathbb{R}^2$ into two regions: one finite and another infinite. Here we are interested in the portion of the envelope which bounds the finite region. This is obtained as a loop, i.e., a closed sequence of adjacent curve segments. This step is similar to the computation of *lower-envelopes* [38, 39, 40] for a given set of curves. Unlike the envelope in our case, the lower-envelope in their case refers to a set of curves which lower-bound a given set of curves with respect to a given direction or distance function. However these approaches assume monotonicity of the input curves with respect to some direction, which does not hold in our case. Instead, we exploit the rich sweep structure inherent to the envelope curves.

The algorithm for stitching the loop is summarized in Algorithm 1. It takes as input the list of envelope curves, *Crv-list*, as explained above, and a starting-curve, *St-crv*, whence the construction of the loop begins. The starting-curve is obtained by shooting a ray from the center of the enclosed region and selecting the first curve segment which intersects this ray. Algorithm 1 iteratively finds the next curve segment which is adjacent to the current segment and appends the same to the loop. The next segment may be in contact with the current segment either by transversal curve-curve intersection or touching at the end-points. This is illustrated in Figure 5(c) by points p and q. The two cases are handled by the functions FindNxtCrvIntersect and FindNxtCrvAdjacent in Algorithm 1. The construction is complete when the loop is closed. This is illustrated in Figure 5(d).

Algorithm 1 ConstructLoop(*St-crv, Crv-list*)

```
Loop ← {St-crv};
Curr-crv ← St-crv;
while TRUE do
    Nxt-crv ← FindNxtCrvIntersect(Curr-crv, Crv-list);
    if Nxt-crv = ∅ then
        Nxt-crv ← FindNxtCrvAdjacent(Curr-crv, Crv-list);
    end if
    if Nxt-crv = St-crv then
        break;
    end if
    Loop ← Loop ∪ {Curr-crv};
    Curr-crv ← Nxt-crv;
end while
return Loop;
```

6. Results

Here we present examples which corroborate the observations made in the previous sections. The example shown in Figure 7 involves circular gears. The input gear G_0 is shown in (a) which is modeled by a cubic B-spline curve with 228 control points. The motion M consists of rotation by an angle of 2π and translation along a circle of radius twice that of G_0, with uniform speed. The shape of H_0, obtained as the conjugate of G_0 along M, is shown in (b). It is modeled as a cubic B-spline curve with 260 control points. The next version, G_1, obtained as the conjugate of H_0 along M^{-1} is shown in (c). It is a cubic B-spline curve with 314 control points. The shapes G_0 and G_1 are shown in overlapping position in (d), in red and blue. G_0, G_1 being identical, the conjugation process has terminated. Each iteration of conjugation took about 110 seconds on a computer with Intel i5 processor and 8 GB memory.

An example of a non-circular gear is shown in Figure 8. Gear G_0 has seven block-like teeth and is modeled as a cubic B-spline curve with 280 control points. It is swept along a motion M involving rotation by an angle of $2\pi * \frac{8}{7}$ and translation along an oblong shaped curve. This curve is composed of two semi-circles connected by a pair of parallel line segments and is parametrized as follows.

$$c(t) = \begin{cases} \left(-t + \frac{\pi}{2}, 1\right) & t \in [0, \pi] \\ \left(-\frac{\pi}{2} + \cos(t - \frac{\pi}{2}), \sin(t - \frac{\pi}{2})\right) & t \in [\pi, 2\pi] \\ \left(t - \frac{5\pi}{2}, -1\right) & t \in [2\pi, 3\pi] \\ \left(\frac{\pi}{2} + \cos(t - \frac{3\pi}{2}), \sin(t - \frac{3\pi}{2})\right) & t \in [3\pi, 4\pi] \end{cases}$$

The resulting conjugate shape, H_0, as shown in (b), has eight teeth. It is a cubic B-spline curve with 350 control points. Note the variation in the geometry of the teeth of H_0. H_0 is conjugated along M^{-1} to obtain G_1, shown in (c). It is a cubic B-spline curve with 330 control points. The shape of H_1 obtained from conjugation of G_1, is identical to that of H_0 and is not shown. Each iteration of conjugation took about 120 seconds.

Design of a non-linear rack-pinion system is illustrated in Figure 9. In such an assembly, the pinion is connected to an actuator and undergoes translation along the rack as it rotates about its axis. The centrode of the rack, which determines the overall shape of the rack, is shown in (a). The gear G_0 shown in Figure 7(a) acts as the pinion. In order to compute the relative motion M, an approximate arc-length parametrization of the centrode curve is computed by a dense sampling of points. The center of G_0 undergoes translation along the offset of the centrode. The amount of rotation is determined by the length traversed along the centrode, so that there is no slipping, as explained in Section 4. The shape H_0 of the rack, thus obtained by conjugation of G_0 along M, is shown in (b). The computation of conjugation took about 170 seconds.

Three examples for design of gears with freeform tooth shapes are shown in Figure 10. The tooth profiles, modeled as B-spline curves are shown in (a). Six copies of these are arranged along a circle to obtain the initial shapes for G_0, shown in (b). The gears G_0 are swept along a relative motion M involving rotation by angle 2π about the center of G_0 and translation along a circle of radius twice that of G_0. The resulting conjugate gears H_0 are shown in (c). The gears H_0 are conjugated along M^{-1} to obtain shapes for G_1, shown in (d). The simplification is evident by overlapping the shapes of G_0, G_1 as shown in (e). The shape of H_1 is identical to that of H_0 and is not shown. Such gears may be manufactured via 3D-printing, laser cutting or 3-axis CNC-machining. It may be noted that involute gears remain the preferred choice in a majority of cases and freeform gears may be relevant only for niche applications.

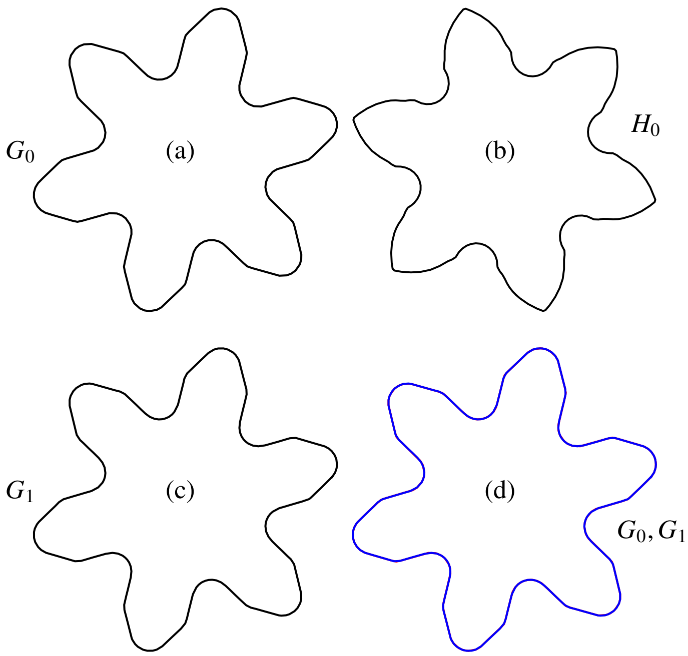

Fig. 7: (a) The input gear G_0 which has six approximately triangular teeth. It is being swept along motion M which consists of rotation by an angle of 2π and translation along a circle of radius 2, with uniform speed. (b) The gear H_0 obtained as the conjugate of G_0 along M. (c) The gear G_1 obtained as the conjugate of H_0, along M^{-1}. (d) Gears G_0 and G_1 shown in overlapping positions, in red and blue. Since G_0 and G_1 are identical, the conjugation process has converged.

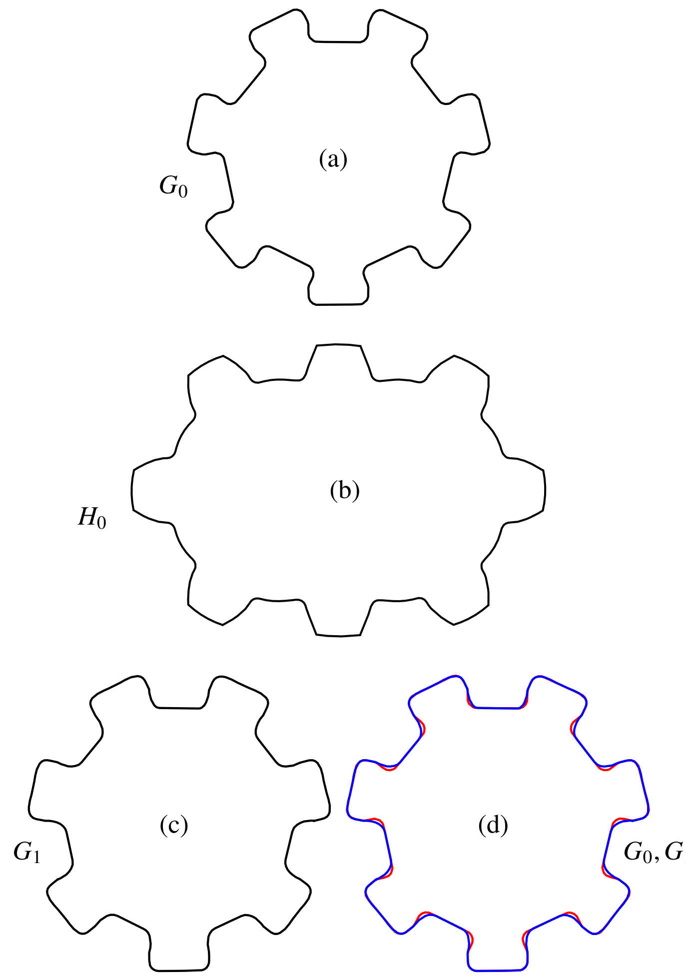

Fig. 8: (a) The input gear G_0 which has seven block-like teeth. It is being swept along motion M which consists of rotation by an angle of $2\pi * \frac{8}{7}$ and translation along an oblong shaped curve, with uniform speed. (b) The gear H_0 obtained as the conjugate of G_0 shown in (a). Thus, H_0 has eight teeth. (c) Gear G_1 obtained as the conjugate of H_0 along M^{-1}. (d) Gears G_0, G_1 are shown in overlapping position, in red and blue.

7. Conclusion

This paper proposes a general mathematical framework for design of conjugate geometries. This is achieved by an in-depth analysis of the contact between conjugate shapes. The characterization of shape invariance leads to guarantee of convergence of conjugation. The mathematical framework is complemented by a robust computational framework with numerical and topological guarantees. The resulting algorithm is employed in designing pairs of meshing gears with freeform shapes, that allow non-uniform transmission of motion. The resulting computer codes are used to generate a variety of examples in 2D.

This work may be extended along several interesting directions. Extending this framework to 3D will enlarge the field of application, for instance, in design of spiral bevel gears. Sliding between conjugate teeth profiles is an important issue related to gear design and most practical gears have non-zero sliding. While we demonstrate examples with freeform tooth geometry which allows complete freedom over tooth shape, analysis of sliding may lead to better design. The scope of this work is restricted to kinematic design. Including forces in the design process may lead to improved mechanical properties of resulting parts. Further, consideration about efficient means of manufacturing of freeform gears may lead to interesting design algorithms.

The examples shown in Figures 2, 4, 8, 9 and 10 are animated at `https://youtu.be/y1zesqut5Vs`.

8. Acknowledgment

This research was supported in part by Indian Institute of Technology Mandi, National Science Foundation grant DMS-1708239 and the ISRAEL SCIENCE FOUNDATION (grant No. 597/18).

References

[1] Litvin, FL, Fuentes, A. Gear Geometry and Applied Theory. Cambridge University Press; 2004.
[2] Yang, SC. Determination of spherical cam profiles by envelope theory. Journal of Materials Processing Technology 2001;116(2):128 – 136. URL: http://www.sciencedirect.com/science/article/pii/S0924013601010111. doi:https://doi.org/10.1016/S0924-0136(01)01011-1.
[3] Taylor, RH, Joskowicz, L, Williamson, B, Gueziec, A, Kalvin, A, Kazanzides, P, et al. Computer-integrated revision total hip replacement surgery: concept and preliminary results. Medical Image Analysis 1999;3(3):301 – 319. URL: http://www.sciencedirect.

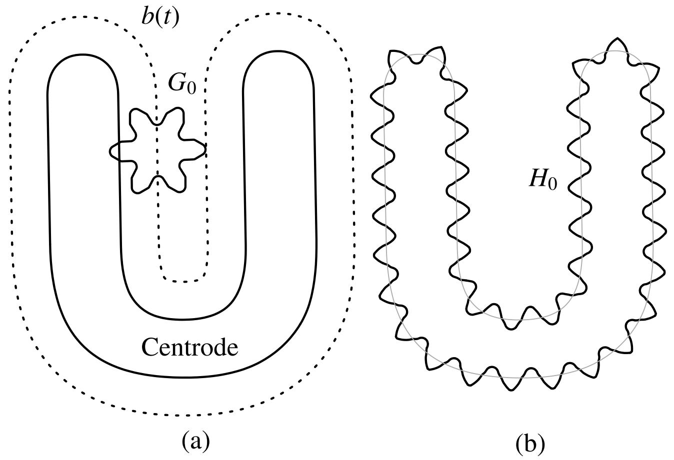

Fig. 9: Design of a non-linear rack-pinion system. (a) The gear G_0, also shown in Figure 7(a), is rolled along the centrode of H shown as a solid black curve. The translational part of motion is shown as a dotted curve. (b) The resulting non-linear rack, H_0 shown in black, along with the centrode, shown in grey.

com/science/article/pii/S1361841599800267. doi:https://doi.org/10.1016/S1361-8415(99)80026-7.

[4] Blackmore, D, Leu, M, Wang, L. The sweep-envelope differential equation algorithm and its application to nc machining verification. Computer-Aided Design 1997;29(9):629 – 637. URL: http://www.sciencedirect.com/science/article/pii/S0010448596001017. doi:https://doi.org/10.1016/S0010-4485(96)00101-7.

[5] Erdim, H, Ilies, HT. Classifying points for sweeping solids. Computer-Aided Design 2008;40(9):987 – 998. URL: http://www.sciencedirect.com/science/article/pii/S0010448508001310. doi:https://doi.org/10.1016/j.cad.2008.07.005.

[6] Elber, G. Irit modeling environment. "http://www.cs.technion.ac.il/~irit/"; 2019.

[7] Wunderlich, W. Ebene kinematik (Plane kinematics) B. I. Hochschultaschenbuercher # 447/447a, (in German). Bibliographisches Institut, Mannheim; 1970.

[8] Reuleaux, F. The constructor. Philadelphia, H.H. Suplee; 1893.

[9] Bar, GFF. Explicit calculation methods for conjugate profiles. Journal for Geometry and Graphics 2003;7(2).

[10] Dooner, DB, Seireg, AA. The Kinematic Geometry of Gearing: A Concurrent Engineering Approach. Wiley-Interscience; 1995.

[11] Litvin, FL. Determination of the enveloping lines of contact of mutually enveloped surfaces. Izv Vyssh Uchebn Zaved Mat 1975;19:40 – 43.

[12] Argyris, J, Litvin, FL, Lian, Q, Lagutin, SA. Determination of envelope to family of planar parametric curves and envelope singularities. Computer Methods in Applied Mechanics and Engineering 1999;175(1):175 – 187. URL: http://www.sciencedirect.com/science/article/pii/S0045782598003673. doi:https://doi.org/10.1016/S0045-7825(98)00367-3.

[13] Puccio, FD, Gabiccini, M, Guiggiani, M. Alternative formulation of the theory of gearing. Mechanism and Machine Theory 2005;40(5):613 – 637. URL: http://www.sciencedirect.com/science/article/pii/S0094114X04001715. doi:https://doi.org/10.1016/j.mechmachtheory.2004.10.003.

[14] Litvin, FL, Gonzalez-Perez, I, Fuentes, A, Hayasaka, K. Design and investigation of gear drives with non-circular gears applied for speed variation and generation of functions. Computer Methods in Applied Mechanics and Engineering 2008;197(45):3783 – 3802. URL: http://www.sciencedirect.com/science/article/pii/S0045782508001138. doi:https://doi.org/10.1016/j.cma.2008.03.001.

[15] Zarebski, I, Salacinski, T. Designing of non-circular gears. The Archive of Mechanical Engineering 2008;55(3):275 – 292. doi:https://DOI:10.24425/ame.2008.131628.

[16] Bendefy, AG, Horák, P. Gear pair generation with the method of transposed lines of action. In: Dorian, M, Mario, S, Neven, P, Nenad, B, Stanko, S, editors. Proceedings of the DESIGN 2016 14th International Design Conference. Dordrecht; 2016, p. 129–136.

[17] Bonandrini, G, Mimmi, G, Rottenbacher, C. Solutions of the Equation of Meshing for Planar Gear Profiles. Dordrecht: Springer Netherlands. ISBN 978-1-4020-8915-2; 2009, p. 77–85. URL: https://doi.org/10.1007/978-1-4020-8915-2_10. doi:10.1007/978-1-4020-8915-2_10.

[18] Bonandrini, G, Mimmi, G, Rottenbacher, C. Toward computerized determination of envelope to family of parametric planar curves. Computer Methods in Applied Mechanics and Engineering 2009;198(27):2218 – 2224. URL: http://www.sciencedirect.com/science/article/pii/S0045782509000747. doi:https://doi.org/10.1016/j.cma.2009.02.005.

[19] Johann, A, Scheurle, J. On the generation of conjugate flanks for arbitrary gear geometries. GAMM-Mitteilungen 2009;32(1):61–79. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.200910005. doi:10.1002/gamm.200910005.

[20] Litvin, F, Egelja, A, Donno, MD. Computerized determination of singularities and envelopes to families of contact lines on gear tooth surfaces. Computer Methods in Applied Mechanics and Engineering 1998;158(1):23 – 34. URL: http://www.sciencedirect.com/science/article/pii/S0045782597002193. doi:https://doi.org/10.1016/S0045-7825(97)00219-3.

[21] Lyashkov, AA, Panchuk, KL, Khasanova, IA. Automated geometric and computer-aided non-circular gear formation modeling. Journal of Physics: Conference Series 2018;1050:012049. doi:10.1088/1742-6596/1050/1/012049.

[22] p. Wang, W, k. Wang, K. Geometric modeling for swept volume of moving solids. IEEE Computer Graphics and Applications 1986;6(12):8–17. doi:10.1109/MCG.1986.276586.

[23] Abdel-Malek, K, Yeh, HJ. Geometric representation of the swept volume using jacobian rank-deficiency conditions. Computer-Aided Design 1997;29(6):457 – 468. URL: http://www.sciencedirect.com/science/article/pii/S0010448596000978. doi:https://doi.org/10.1016/S0010-4485(96)00097-8.

[24] Ilies, HT, Shapiro, V. The dual of sweep. Computer-Aided Design 1999;31(3):185 – 201. URL: http://www.sciencedirect.com/science/article/pii/S0010448599000159. doi:https://doi.org/10.1016/S0010-4485(99)00015-9.

[25] Leu, MC, Wang, L, Blackmore, D. A verification program for 5-axis nc machining with general apt tools. CIRP Annals 1997;46(1):419 – 424. URL: http://www.sciencedirect.com/science/article/pii/S0007850607608567. doi:https://doi.org/10.1016/S0007-8506(07)60856-7.

[26] Abdel-Malek, K, Yeh, HJ. Geometric representation of the swept volume using jacobian rank-deficiency conditions. Computer-Aided Design 1997;29(6):457 – 468. URL: http://www.sciencedirect.com/science/article/pii/S0010448596000978. doi:https://doi.org/10.1016/S0010-4485(96)00097-8.

[27] Erdim, H, Ilies, HT. Detecting and quantifying envelope singularities in the plane. Computer-Aided Design 2007;39(10):829 – 840. URL: http://www.sciencedirect.com/science/article/pii/S0010448507000802. doi:https://doi.org/10.1016/j.cad.2007.03.007.

[28] Rossignac, J, Kim, J, Song, S, Suh, K, Joung, C. Boundary of the volume swept by a free-form solid in screw motion. Computer-Aided Design 2007;39(9):745 – 755. URL: http://www.sciencedirect.com/science/article/pii/S001044850700067X. doi:https://doi.org/10.1016/j.cad.2007.02.016.

[29] Zhang, X, Kim, YJ, Manocha, D. Reliable sweeps. In: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling. SPM '09; New York, NY, USA: ACM. ISBN 978-1-60558-711-0; 2009, p. 373–378. URL: http://doi.acm.org/10.1145/1629255.1629306. doi:10.1145/1629255.1629306.

[30] Peternell, M, Pottmann, H, Steiner, T, Zhao, H. Swept volumes. Computer-Aided Design and Applications 2005;2(5):599 – 608. doi:https://doi.org/10.1080/16864360.2005.10738324.

[31] Wallner, J, Yang, Q. Swept volumes of many poses. In: Proceedings of the Third Eurographics Symposium on Geometry Processing. SGP '05; Goslar, DEU: Eurographics Association. ISBN 390567324X; 2005, p. 161–es.

[32] Adsul, B, Machchhar, J, Sohoni, M. Local and global analysis

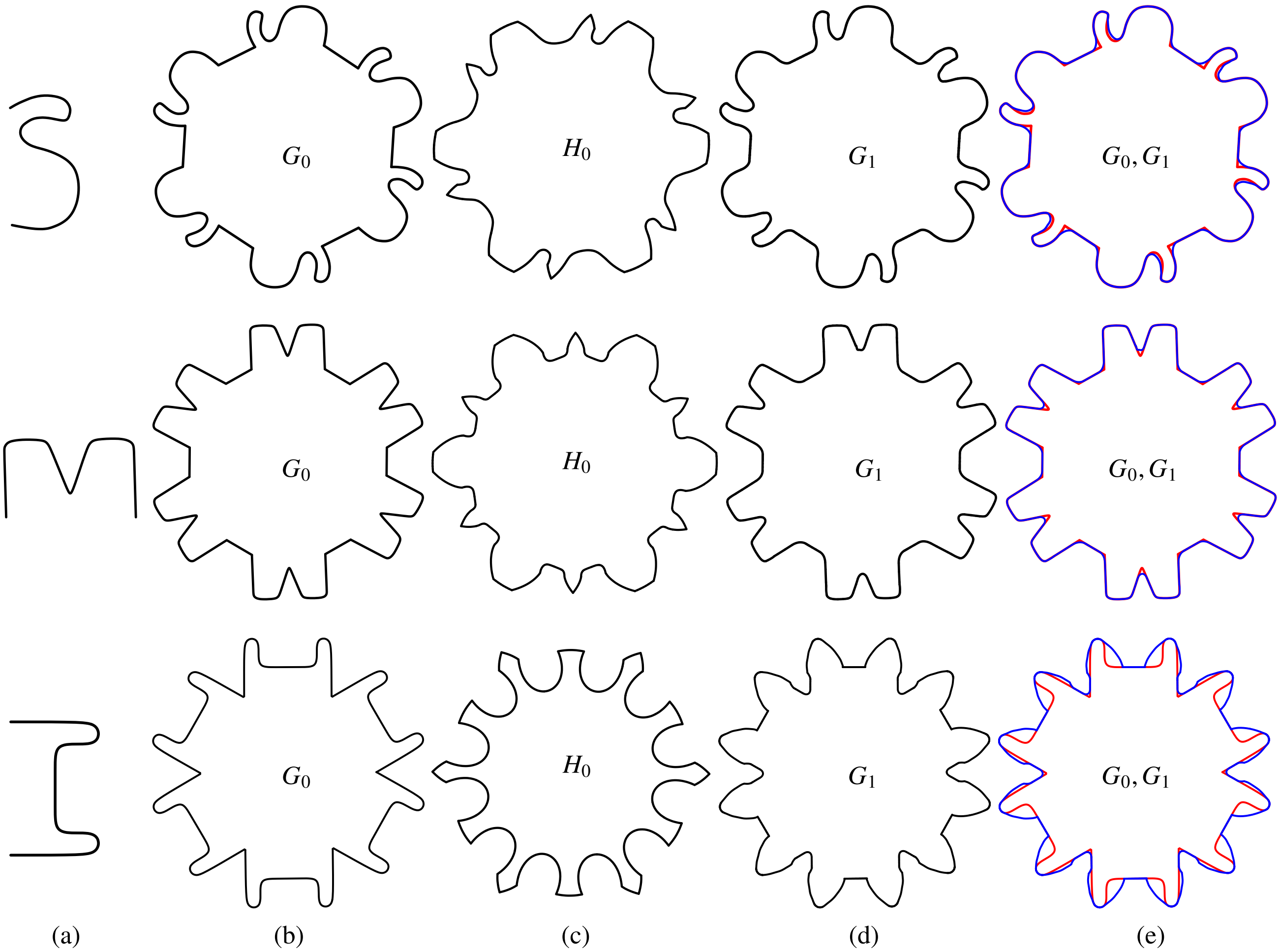

Fig. 10: Design of gears with freeform tooth shapes. (a) Tooth profiles modeled as freeform B-spline curves. (b) Initial designs for gears G_0 obtained by arranging six copies of the tooth shown in (a) around a circle. (c) The gears H_0 obtained as the conjugate of G_0 shown in (b) along motion M which involves rotation by angle 2π about center of G_0 and translation along a circle of radius twice that of G_0. (d) The gears G_1 obtained as the conjugate of H_0 shown in (c) along M^{-1}. (e) Gears G_0, G_1, shown in red and blue, in overlapping positions.

of parametric solid sweeps. Computer Aided Geometric Design 2014;31(6):294 – 316. URL: `http://www.sciencedirect.com/science/article/pii/S0167839614000582`. doi:`https://doi.org/10.1016/j.cagd.2014.05.009`.

[33] Litvin, FL. Theory of Gearing. NASA Reference Publication 1212 AVS-COM Technical Report 88-C-035; 1989.

[34] Barton, M, Elber, G, Hanniel, I. Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. Computer-Aided Design 2011;43(8):1035 – 1044. doi:`https://doi.org/10.1016/j.cad.2011.03.009`.

[35] Dokken, T, Dehlen, M, Lyche, T, Morken, K. Good approximation of circles by curvature-continuous bézier curves. Computer Aided Geometric Design 1990;7(1):33 – 41. URL: `http://www.sciencedirect.com/science/article/pii/016783969090019N`. doi:`https://doi.org/10.1016/0167-8396(90)90019-N`.

[36] Cohen, E, Riesenfeld, RF, Elber, G. Geometric Modeling with Splines, An Introduction. A K Peters; 2001.

[37] Blackmore, D, Samulyak, R, Leu, MC. Trimming swept volumes. Computer-Aided Design 1999;31(3):215 – 223. URL: `http://www.sciencedirect.com/science/article/pii/S0010448599000172`. doi:`https://doi.org/10.1016/S0010-4485(99)00017-2`.

[38] Hanniel, I, Muthuganapathy, R, Elber, G, Kim, MS. Precise voronoi cell extraction of free-form rational planar closed curves. In: Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling. SPM '05; New York, NY, USA: Association for Computing Machinery. ISBN 1595930159; 2005, p. 51–59. URL: `https://doi.org/10.1145/1060244.1060251`. doi:`10.1145/1060244.1060251`.

[39] Hanniel, I, Elber, G. Computing the voronoi cells of planes, spheres and cylinders in r3. Computer Aided Geometric Design 2009;26(6):695 – 710. URL: `http://www.sciencedirect.com/science/article/pii/S0167839608000964`. doi:`https://doi.org/10.1016/j.cagd.2008.09.010`; solid and Physical Modeling 2008.

[40] Meyerovitch, M. Robust, generic and efficient construction of envelopes of surfaces in three-dimensional spaces. In: Azar, Y, Erlebach, T, editors. Algorithms – ESA 2006. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006, p. 792–803. doi:`https://doi.org/10.1007/11841036_70`.

Appendix A. Some facts about rigid motions

Proof of Lemma 7: Assume without loss of generality that $A(t_0) = I$ and $b(t_0) = 0$. Thus, at $t = t_0$, we have,

$$p_M(t_0) = A(t_0)p + b(t_0) = p = q.$$

The matrix $A(t)$ being orthonormal, we have,

$$\begin{aligned} & A(t)A^{-1}(t) = I \\ \Rightarrow & A'(t)A^{-1}(t) + A(t)A^{-1\prime}(t) = 0 \\ \Rightarrow & A'(t_0) = -A^{-1\prime}(t_0). \end{aligned} \tag{A.1}$$

The velocity at $p_{M^{-1}}(t)$ is computed as,

$$
\begin{aligned}
q_{M^{-1}}(t) &= A^{-1}(t)(p - b(t)) \\
\Rightarrow q'_{M^{-1}}(t) &= {A^{-1}}'(t)(p - b(t)) - A^{-1}(t)b'(t).
\end{aligned}
\tag{A.2}
$$

At $t = t_0$, from Equation (A.1) and (A.2) we get,

$$
\begin{aligned}
q'_{M^{-1}}(t_0) &= -A'(t_0)p - b'(t_0) \\
&= -p'_M(t_0).
\end{aligned}
$$

$\square$