Using monodromy to recover symmetries of polynomial systems

Timothy Duff*
timduff@uw.edu
University of Washington
Seattle, Washington, USA

Viktor Korotynskiy
viktor.korotynskiy@cvut.cz
CIIRC, CTU in Prague
Prague, Czech Republic

Tomas Pajdla
pajdla@cvut.cz
CIIRC, CTU in Prague
Prague, Czech Republic

Margaret Regan
mregan@math.duke.edu
Duke University
Durham, North Carolina, USA

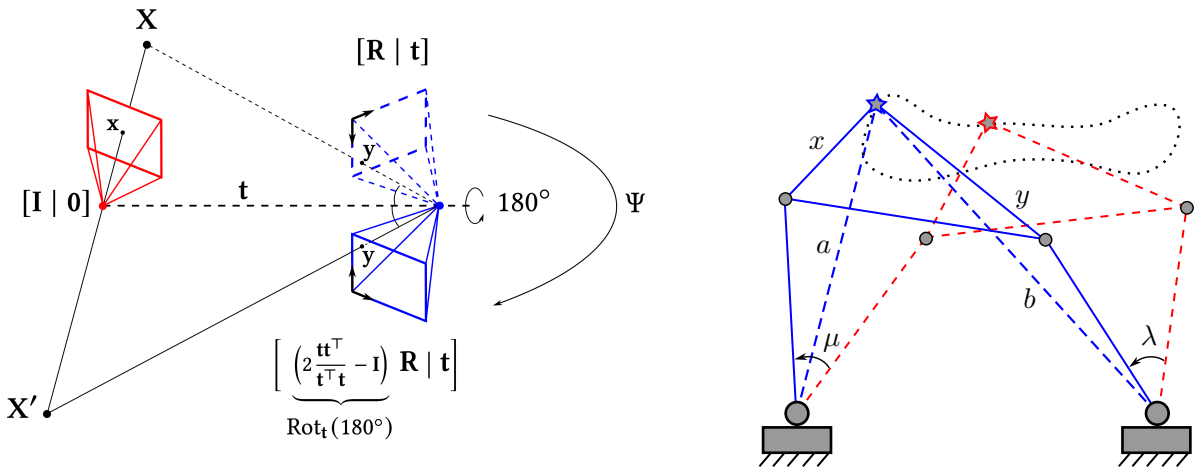

Figure 1: (Left) Twisted pair symmetry for five-point relative pose. (Right) Nine-point four-bar mechanism synthesis.

ABSTRACT

Galois/monodromy groups attached to parametric systems of polynomial equations provide a method for detecting the existence of symmetries in solution sets. Beyond the question of existence, one would like to compute formulas for these symmetries, towards the eventual goal of solving the systems more efficiently. We describe and implement one possible approach to this task using numerical homotopy continuation and multivariate rational function interpolation. We illustrate our methods on several examples, including two cases with nonlinear symmetries which appear in applications from computer vision and robotics.

CCS CONCEPTS

• **Mathematics of computing** → *Solvers*; *Nonlinear equations.*

ISSAC 2023, July 24–27, 2023, Tromsø, Norway

ACM ISBN 979-8-4007-0039-2/23/07.
https://doi.org/10.1145/3597066.3597106

KEYWORDS

Galois group, monodromy, polynomial system, interpolation

ACM Reference Format:
Timothy Duff, Viktor Korotynskiy, Tomas Pajdla, and Margaret Regan. 2023. Using monodromy to recover symmetries of polynomial systems. In *International Symposium on Symbolic and Algebraic Computation 2023 (ISSAC 2023), July 24–27, 2023, Tromsø, Norway.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3597066.3597106

1 INTRODUCTION

Structured systems of nonlinear equations appear frequently in applications like computer vision and robotics. Although the word "structure" can be interpreted in many ways, one of its aspects that is strongly connected to the complexity of solving is the *algebraic degree* of the problem to be solved. In many contexts, this may simply refer to the number of solutions of a system (usually counted over the complex numbers). However, if we adopt this definition without scrutiny, we may fail in certain special cases to detect additional structure such as symmetry.

To answer more refined questions involving structure, one can often consider a *Galois/monodromy* group naturally associated to the problem of interest. In this case, "problem" refers to a *parametric family* of problem instances which must be solved for different

sets of parameter values. In our work, we are primarily interested in *geometric* Galois groups arising from algebraic extensions of functions fields of varieties defined over the complex numbers.[1]

Currently, a number of heuristic methods for computing Galois/monodromy groups using numerical homotopy continuation methods have been proposed and implemented, eg. [9, 15]. It is also fairly well-understood how Galois/monodromy groups encode important structural properties such as *decomposability*, or the existence of problem symmetries which may be expressed as rational functions known as *deck transformations*. Thus, Galois/monodromy group computation provides us a useful toolkit for detecting the *existence* of special structure. However, one key challenge remains: once we know that our problem *does* have such special structure, can we use this information to solve systems more efficiently?

Our work focuses on a natural first step towards addressing this challenge: *given the data of a numerical Galois/monodromy group computation, can we recover formulas for the rational maps realizing the underlying symmetry or decomposability?*

In this paper, we describe and implement a solution to this first step that combines the previous techniques for numerically computing Galois/monodromy groups with floating-point interpolation of multivariate rational functions. Although both components are well-established within the domain of symbolic-numeric computation, we are unaware of any previous work which combines them in this novel way.

In Section 2, we provide some context for our approach by considering related previous works. In Section 3, we establish terminology and useful background facts. In Section 4, we describe our main algorithm for interpolating deck transformations and illustrate it on simple examples. In Section 5, we describe experiments performed with our accompanying software package `DecomposingPolynomialSystems` for the Julia programming language [3]. The source code for this package may be obtained at the url below:
https://github.com/vviktorrK/DecomposingPolynomialSystems.jl

2 RELATED WORK

Galois/monodromy groups have long had a presence in algebraic computation, used as a tool in the study of algebraic curves, polynomial factorization, and numerical irreducible decomposition [8, 12, 26]. In recent years, monodromy-based methods have become a popular heuristic for computing the isolated solutions of parametric polynomial systems [9, 19]. One appealing aspect of these methods is that they are useful for constructing efficient *start systems* to be used in parameter homotopies, particularly in cases where more traditional start systems (total degree, polyhedral) fail to capture the full structure. Another appealing feature is that symmetry or decomposability can be naturally incorporated in both the offline monodromy and online parameter homotopy phases. This is the main idea behind several recent, closely-related works which use *a priori* knowledge of symmetries to speed up solving [1, 5]. In contrast to these works, our approach recovers symmetries with no such knowledge, and with limited assumptions on the system to be solved. Our work is also a natural continuation of the paper [10], where Galois/monodromy groups were used to infer decompositions and symmetries that were not previously known for some novel problems in computer vision. Here, we instead describe a novel *method*, illustrated with familiar examples.

Interpolation is a well-studied problem in symbolic-numeric computation and an important ingredient for solving our recovery problem. In our work, we are faced with the difficult task of interpolating an *exact* rational function (as opposed to some low-degree approximation) from *inexact inputs* in double-precision floating-point arithmetic. For this reason, we employ many heuristics, and make no attempt to match state-of-the-art interpolation techniques. On the other hand, we hope that experts on interpolation will view our particular application as a potential use case for their own methods. Some relevant references for the specific problem of multivariate rational function interpolation include [7, 18, 30].

Our focus on inexact inputs is due to the fact that interpolation occurs downstream of numerical homotopy continuation in our framework. This is also why we cannot pick inputs for the interpolation problem arbitrarily. With that said, we point out that assuming exact inputs could also be relevant if, say, certified homotopy continuation (see [2, 14, 29, 33]) is used, augmented by some additional postprocessing.

3 BACKGROUND

In this work, we are interested in solving polynomial systems whose solutions correspond to points in a generic fiber of a branched cover of complex algebraic varieties. Here we collect some definitions and theoretical facts that we need to work within this framework. The section concludes with Proposition 3.7 and Corollary 4, which we use to justify the correctness of our interpolation setup.

Definition 1. Let X and Z be irreducible algebraic varieties of dimension m over the complex numbers. A *branched cover* is a dominant, rational map $f : X \dashrightarrow Z$. The varieties X and Z are called the total space and the base space of the cover, respectively. The number of (reduced) points in the preimage over a generic $z \in Z$ is called the degree of f, denoted $\deg f$.

Essentially, the base space Z in Definition 1 can be thought of as a space of parameters or observations. The fiber $f^{-1}(z)$ over some particular $z \in Z$ should usually be understood as the solutions of a particular problem instance specified by z. Oftentimes, Z may be assumed to be an affine space $\mathbb{C}^m$, and in this case we write $\mathbf{p} \in \mathbb{C}^m$ for parameter values. The assumptions that f is dominant and $\dim X = m$ imply that there is a finite, nonzero number of solutions for almost all parameters. Counting solutions over $\mathbb{C}$, that number is $\deg f$. Additionally, the total space X is often either

(1) an irreducible variety consisting of *problem-solution pairs*,

$$X = \{(\mathbf{x}, \mathbf{p}) \in \mathbb{C}^{n+m} \mid f_1(\mathbf{x}, \mathbf{p}) = \cdots = f_k(\mathbf{x}, \mathbf{p}) = 0\} \tag{1}$$

for some system of polynomials $f_1, \ldots, f_k \in \mathbb{C}[\mathbf{x}, \mathbf{p}]$, with projection $f : X \to \mathbb{C}^m$ given by $f(\mathbf{x}, \mathbf{p}) = \mathbf{p}$, or

(2) an affine space of unknowns $X = \mathbb{C}^m$, and $f : \mathbb{C}^m \dashrightarrow \mathbb{C}^m$.

Cases (1) and (2) for the total space X given above are closely related. Indeed, (2) reduces to (1) if we take X to be the graph of f. Conversely, it can often be the case that the variety X has a unirational parametrization $p : \mathbb{C}^m \dashrightarrow X$. In this case, (1) reduces

[1] See eg. [27, §1.2] for a discussion of how other fields of definition relate to this setup.

to (2) by replacing f with the branched cover $f \circ p : \mathbb{C}^m \dashrightarrow \mathbb{C}^m$. When $\deg f$ and $\deg p$ are both greater than 1, the composite map $f \circ p$ is an example of a decomposable branched cover.

Definition 2. A branched cover $f : X \dashrightarrow Z$ is said to be *decomposable* if there exist two branched covers $g : X \dashrightarrow Y$ and $h : Y \dashrightarrow Z$ with $\deg g, \deg h < \deg f$ such that $f(x) = h \circ g(x)$ for all x in a nonempty Zariski-open subset of X. The maps g and h are said to give a *decomposition* of f.

Example 3.1. Let $X = \mathbf{V}(ax^6+bx^5+cx^4+dx^3+cx^2+bx+a) \subset \mathbb{C}^5$, $Z = \mathbb{C}^4$, and $f : X \to Z$ given by $f(a,b,c,d,x) = (a,b,c,d)$. The projection f is a decomposable branched cover in the sense of Definition 2. To see this, take $Y = \mathbf{V}(a(y^3-3y)+b(y^2-2)+cy+d) \subset \mathbb{C}^5$, and define $g : X \dashrightarrow Y$ by $g(a,b,c,d,x) = (a,b,c,d,\frac{x^2+1}{x})$, and $h : Y \to Z$ by $h(a,b,c,d,y) = (a,b,c,d)$. The degrees of the various maps satisfy $6 = \deg f = \deg(h\circ g) = \deg(h)\cdot\deg(g) = 3\cdot 2$.

Example 3.2. The following example is based on [5, §2.3.2], and belongs to a general class of examples where decomposability can be detected via equations' Newton polytopes. Let $Z = \mathbb{C}^{23}$, and $X \subset \mathbb{C}^{26}$ be the vanishing locus of the three equations below:

$$a\,x^3y\,z^4 + b\,x^2y^2z^4 + c\,x^2y\,z^3 + d\,x\,y^2z^3 + e\,x^2z^2 + f\,x\,y\,z^2 + g\,x\,z + h,$$
$$i\,x^3y\,z^4 + j\,x^2y^2z^4 + k\,x^2y\,z^3 + l\,x\,y^2z^3 + m\,x^2z^2 + n\,x\,y\,z^2 + o\,x\,z + p,$$
$$q\,x\,y\,z^4 + r\,y\,z^5 + s\,x\,z^3 + t\,z^4 + u\,z^3 + v\,z^2 + w.$$

The projection $f : X \to \mathbb{C}^{23}$ given by $f(a,\ldots,z) \mapsto (a,\ldots,w)$ is a branched cover of degree 32. If we let Y be the set of all $(a,\ldots,w,\hat{x},\hat{y}) \in \mathbb{C}^{25}$ such that

$$a\,\hat{x}^3\hat{y} + b\,\hat{x}^2\hat{y}^2 + c\,\hat{x}^2\hat{y} + d\,\hat{x}\,\hat{y}^2 + e\,\hat{x}^2 + f\,\hat{x}\,\hat{y} + g\,\hat{x} + h =$$
$$i\,\hat{x}^3\hat{y} + j\,\hat{x}^2\hat{y}^2 + k\,\hat{x}^2\hat{y} + l\,\hat{x}\,\hat{y}^2 + m\,\hat{x}^2 + n\,\hat{x}\,\hat{y} + o\,\hat{x} + p = 0,$$

then $g : X \to Y$ given by $g(a,\ldots,w,x,y,z) = (a,\ldots,w,xz,yz)$ and $h : Y \to Z$ given by $h(a,\ldots,w,\hat{x},\hat{y}) = (a,\ldots,w)$ show that f is a decomposable branched cover in the sense of Definition 2. Here we have $\deg h = 8$ and $\deg g = 4$.

The Galois/monodromy group is an invariant that allows us to decide whether or not a branched cover is decomposable, without actually exhibiting a decomposition. We recall the basic definitions here. For a branched cover $f : X \dashrightarrow Z$, fix a dense Zariski-open subset $U \subset Z$ such that $f^{-1}(z)$ consists of $d = \deg(f)$ points. Over a regular locus, the branched cover f restricts to a d-sheeted covering map in the usual sense given by $f^{-1}(U) \to U$. For any basepoint $z \in U$, we may construct via *path-lifting* a group homomorphism from the fundamental group $\pi_1(U;z)$ to the symmetric group S_d.

More precisely, if $\gamma : [0,1] \to U$ is any map that is continuous with resepct to the Euclidean topology, then the *unique lifting property* [13, Prop. 1.34] implies that there are precisely d continuous lifts $\tilde{\gamma}_1,\ldots,\tilde{\gamma}_d : [0,1] \to \pi^{-1}(U)$ satisfying $f \circ \tilde{\gamma}_i(t) = \gamma_i(t)$ for all $i = 1,\ldots,d$ and $t \in [0,1]$. In particular, $\tilde{\gamma}_i(0), \tilde{\gamma}_i(1) \in f^{-1}(z)$, and there is a permutation σ_γ that sends each $\tilde{\gamma}_i(1)$ to $\tilde{\gamma}_i(0)$. One may check that this permutation is independent of the chosen representative γ of the homotopy class $[\gamma] \in \pi_1(U;z)$. Thus, for our chosen U and z we may define the *monodromy representation*,

$$\begin{aligned} \rho_{u,Z} : \pi_1(U;z) &\to S_d \\ [\gamma] &\mapsto \sigma_\gamma. \end{aligned} \tag{2}$$

This gives a group homomorphism, whose image is a subgroup of S_d, which turns out to be independent of the choice of U and z.

Definition 3. The *Galois/monodromy group* of a branched cover f is the subgroup of S_d given by the image of the map (2).

The abstract structure of the Galois/monodromy group, although interesting, is not our main focus. Instead, we will be mainly interested in the action of this group given by (2). Since X is irreducible, this action is transitive (see eg. [20, Lemma 4.4, p87].)

The monodromy action also provides a clean characterization of decomposable branched covers. Recall that the action of a group G on a finite set B is said to be *imprimitive* if there exists a nontrivial partition $B = B_1 \sqcup B_2 \sqcup \cdots \sqcup B_k$ such that for any $g \in G$ and B_i there exists a B_j with $g \cdot B_i = B_j$. If B has d elements and G is a finite, transitive subgroup of S_d, it follows that the subsets B_i must all have the same size. The sets $B_1,\ldots,B_k$ are called *blocks* of the imprimitive action, and are said to form a *block system*.

Proposition 3.3. (See eg. Brysiewicz et al. [5, Proposition 1].) A branched cover is decomposable if and only if its Galois/monodromy group is imprimitive.

Example 3.4. For the branched cover f from Example 3.1, the Galois/monodromy group acts transitively on the set of roots, which we replace with a set of labels $B = \{1,\ldots,6\}$. Up to relabeling, there is a block decomposition for this action given by $B = \{1,2,3\} \sqcup \{4,5,6\}$. There are $48 = 2^3 \cdot 3!$ permutations in S_6 that preserve this block decomposition. These permutations form a group called the *wreath product* $S_2 \wr S_3$. This group can be presented by three permutation generators, for instance

$$\langle (12)(45), (123)(456), (14)(25)(36) \rangle. \tag{3}$$

Computing the Galois/group monodromy group numerically, we find that every element of $S_2 \wr S_3$ arises as σ_γ for some loop γ.

Similarly, for the branched cover from Example 3.2, we find by numerical computation that its Galois/monodromy group is the wreath product $S_4 \wr S_8$, a group of order $(4!)^8 \cdot 8!$

In general, a transitive, imprimitive permutation group has a block system $B_1,\ldots,B_k$ whose blocks all have the same size l, and is thus permutation-isomorphic to a subgroup of the wreath product $S_l \wr S_k$. Unlike the previous example, there are a number of surprising cases of decomposable branched covers where the Galois/monodromy group is a *proper* subgroup of the associated wreath product: for instance, the five-point problem of Section 5.1.

We point out that Proposition 3.3 dates back, at least in some form, to work of Ritt on polynomial decompositions [25]. This work is directly related to *decomposition problems* for polynomials and rational functions studied in computer algebra (see eg. [11, 31]).

However, the main focus in this paper is not decomposability *per se*. Rather, we are interested in a property that is usually stronger: the existence of symmetries. A natural, and general, notion of symmetry can be obtained by studying the embedding of function fields $f^* : \mathbb{C}(Z) \dashrightarrow \mathbb{C}(X)$ induced by a branched cover. The field extension $\mathbb{C}(X)/\mathbb{C}(Z)$, although not usually a Galois extension, may nevertheless a have a nontrivial group of automorphisms. These automorphisms correspond to rational maps $\Psi : X \dashrightarrow X$ with $f \circ \Psi = f$. In topological terms, these comprise the group of *deck transformations* of f.

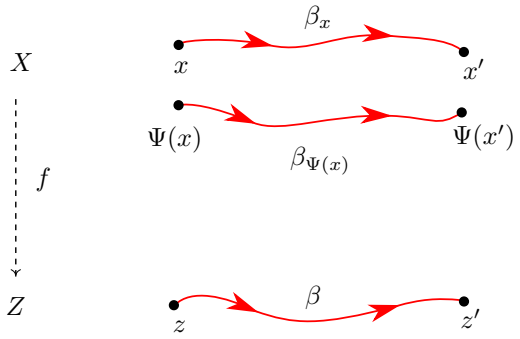

Figure 2: Illustration of Proposition 3.7.

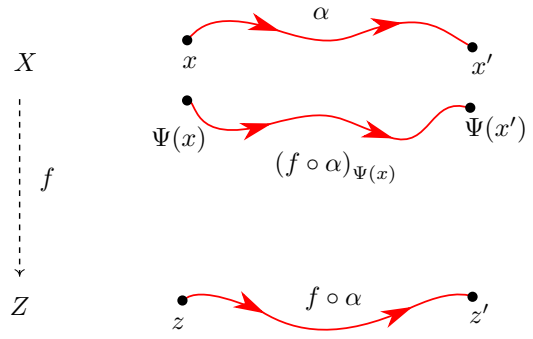

Figure 3: Illustration of Corollary 4.

Proposition 3.5 below explains the relationship between deck transformations and decomposability, and provides an analogue of Proposition 3.3 for detecting the existence of deck transformations. Proofs may be found in [10, §2.1].

Proposition 3.5. Let $f : X \dashrightarrow Z$ be a branched cover of degree d.

(1) If f has a nontrivial deck transformation group, then its Galois/monodromy group is either decomposable or cylic of order d. (Both are true when d is composite.)
(2) Restricting the deck transformations to the fiber $f^{-1}(z)$ defines another permutation group which is the centralizer of the Galois/monodromy group in S_d. In particular, there exists a nontrivial deck transformation if and only if this centralizer is nontrivial.

Example 3.6. For the branched cover f from Example 3.1, the centralizer in S_6 of the Galois/monodromy group presented as in eq. (3) is a cyclic group of order 2, namely $\langle(14)(25)(36)\rangle$. Correspondingly, there is a nontrivial deck transformation $\Psi : X \dashrightarrow X$ defined by $\Psi(a, b, c, d, x) = (a, b, c, d, 1/x)$.

For the branched cover f of Example 3.2, the centralizer of its Galois/monodromy group $S_4 \wr S_8$ in S_{32} is trivial. Thus, this decomposable branched cover has no nontrivial deck transformations.

In the final results of this section, Proposition 3.7 and Corollary 4, we use the terminology *generic path* for a given branched cover $f : X \dashrightarrow Z$. This means a path $\alpha : [0, 1] \to U$ where U is some suitably small set, either a regular locus in Z or its preimage in X. In the former case, we write $\widetilde{\alpha}_x$ for the unique lift of a path α through f starting at $x \in f^{-1}(\alpha(0))$.

Proposition 3.7. Let $f : X \dashrightarrow Z$ be a branched cover with a fixed generic point $x \in X$. Then the value of a deck transformation $\Psi \in \mathrm{Deck}(X/Z)$ at a generic point $x' \in X$ is completely determined via path-lifting by where it sends x. Explicitly,

$$\Psi(x') = \widetilde{(f \circ \alpha)}_{\Psi(x)}(1), \tag{4}$$

where α is a generic path in X from x to x' (see Figure 2).

Proof. We refer to the proof of [13, Prop. 1.33] and the general definition of a *lift* given on [13, p. 60]. The deck transformation Ψ is a lift of f to X in the sense of this definition. This means the proof of Proposition 1.33 can be applied to construct a deck transformation Ψ' with $\Psi'(x) = \Psi(x)$. This construction uses lifts of a generic path α to construct Ψ', with the additional property that $\Psi'(x') = \widetilde{(f \circ \alpha)}_{\Psi(x)}(1)$. The unique path-lifting property then implies that $\Psi(x') = \Psi'(x')$. □

A consequence of Proposition 3.7 is that the correspondence between solutions for fixed set of parameters under a fixed deck transformation Ψ is preserved under path-lifting.

Corollary 4. *Let $f : X \dashrightarrow Z$ be a branched cover and $\Psi \in \mathrm{Deck}(X/Z)$. Let $z \in Z$ be a generic point and β be a generic path in Z starting at z (see Figure 3). Then for $x \in X_z$ we have*

$$\Psi(\widetilde{\beta}_x(1)) = \widetilde{\beta}_{\Psi(x)}(1)$$

In other words, the points in the 2 lifts of β starting at x and $\Psi(x)$ are conjugate under Ψ (see Figure 3.)

Proof. By Proposition 3.7, $\Psi(\widetilde{\beta}_x(1)) = \widetilde{\left(f \circ \widetilde{\beta}_x\right)}_{\Psi(x)}(1)$, which, in turn, is equal to $\widetilde{\beta}_{\Psi(x)}(1)$, since $f \circ \widetilde{\beta}_x = \beta$. □

4 INTERPOLATING SYMMETRIES

Consider a branched cover encoding problem-solution pairs $(\mathbf{x}, \mathbf{p})$,

$$\begin{aligned} f : X &\to \mathbb{C}^m \\ (\mathbf{x}, \mathbf{p}) &\mapsto \mathbf{p} \end{aligned} \tag{5}$$

with $\deg f = d$, which has a nontrivial deck transformation

$$\Psi(\mathbf{x}, \mathbf{p}) = \begin{bmatrix} \psi_1(\mathbf{x}, \mathbf{p}) & \dots & \psi_n(\mathbf{x}, \mathbf{p}) & \mathbf{p}^\top \end{bmatrix}^\top. \tag{6}$$

As mentioned in the introduction, we may compute the Galois/monodromy group of f using numerical homotopy continuation. This is possible provided that we make the following assumptions about how our branched cover is given as input.

Assumption 4.1. For the branched cover defined in eq. (5), assume that n rational functions $f_1, \dots, f_n$ vanishing on X are known, and that we have access to a sampling oracle that produces generic $(\mathbf{x}^*, \mathbf{p}^*) \in X$ such that the $n \times n$ Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^*, \mathbf{p}^*)$ has rank n.

Assumption 4.1 is often satisfied in practice, including cases where even a set-theoretic description of X is not known. Additionally, we assume that homotopy continuation—specifically, coefficient parameter homotopy—can be used to track d known solutions

for fixed, generic parameter values $\mathbf{p}^*$ (corresponding to $f^{-1}(\mathbf{p}^*)$) to d solutions for some other parameter values $\mathbf{p} \in \mathbb{C}^m$ (corresponding to $f^{-1}(\mathbf{p})$). These parameter homotopies are the basis of the unspecified subroutines in lines 1 and 8 of Algorithm 1.

An important observation is that we can interpolate each of the coordinate functions $\psi_j(\mathbf{x}, \mathbf{p})$ in eq. (6) independently. We assume that the rational function ψ_j contains only monomials up to total degree D. Since these monomials may or may not involve the parameters $\mathbf{p}$, we distinguish the *parameter-dependent* and *parameter-independent* settings, in which we take the number of monomials t to be either

$$t = \binom{n+m+D}{D}, \text{ or} \qquad \text{(for parameter-dependent } \psi_j(\mathbf{x}, \mathbf{p}))$$

$$t = \binom{n+D}{D}. \qquad \text{(for parameter-independent } \psi_j(\mathbf{x}))$$

Our task is then to recover two vectors of unknown coefficients

$$\mathbf{a} = \begin{bmatrix} a_1 & \dots & a_t \end{bmatrix}^\top, \ \mathbf{b} = \begin{bmatrix} b_1 & \dots & b_t \end{bmatrix}^\top \in \mathbb{C}^t,$$

such that ψ_j can be represented on X as

$$\psi_{\mathbf{a},\mathbf{b}}(\mathbf{x}, \mathbf{p}) = \frac{\sum_{k=1}^t a_k \cdot (\mathbf{x}, \mathbf{p})^{\boldsymbol{\alpha}_k}}{\sum_{k=1}^t b_k \cdot (\mathbf{x}, \mathbf{p})^{\boldsymbol{\beta}_k}}. \tag{7}$$

In Equation (7), the vectors $\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k \in \mathbb{Z}^{n+m}_{\geq 0}$ range over a suitable set of multidegrees, depending on whether we are in the parameter-dependent or parameter-independent setting. If we know that $(\mathbf{x}_i', \mathbf{p}_i) = \Psi(\mathbf{x}_i, \mathbf{p}_i)$ for points $(\mathbf{x}_i, \mathbf{p}_i), (\mathbf{x}_i', \mathbf{p}_i) \in X$, then this gives us one homogeneous linear constraint on $\mathbf{a}$ and $\mathbf{b}$,

$$\sum_{k=1}^t a_k \cdot (\mathbf{x}_i, \mathbf{p}_i)^{\boldsymbol{\alpha}_k} - x_{ij}' \cdot \left(\sum_{k=1}^t b_k \cdot (\mathbf{x}_i, \mathbf{p}_i)^{\boldsymbol{\beta}_k} \right) = 0. \tag{8}$$

Suppose we have already computed permutations generating the monodromy group based at parameter values $\mathbf{p}_1 \in \mathbb{C}^m$, and let $\mathbf{x}_1, \mathbf{x}_1'$ be two solutions with $\Psi(\mathbf{x}_1, \mathbf{p}_1) = (\mathbf{x}_1', \mathbf{p}_1)$. Proposition 3.5 implies that $\sigma \cdot (\mathbf{x}_1, \mathbf{p}_1) = (\mathbf{x}_1', \mathbf{p}_1')$ for some element of the centralizer $\sigma \in \mathrm{Cent}_{S_d}(\mathrm{Mon}(f, \mathbf{p}_1))$ corresponding to Ψ. Now, Corollary 4 implies that we may obtain additional sample points satisfying (8) by tracking parameter homotopies using the system $f_1, \dots, f_n$. Specifically, we may track the solution curves with initial values $\mathbf{x}_1, \mathbf{x}_1'$ from $\mathbf{p}_1$ to generic $\mathbf{p}_i \in \mathbb{C}^m$ for $i = 1, \dots, 2t$, which then allows us recover the coordinate functions of Ψ.

Proposition 4.1 (Correctness of Algorithm 1)**.** Suppose that ψ_j in (6) can be represented as the quotient of polynomials with degree $\leq D$ and t monomials each. For a sufficiently generic sample

$$(\mathbf{x}_1, \mathbf{p}_1), \dots, (\mathbf{x}_{2t}, \mathbf{p}_{2t}), (\mathbf{x}_1', \mathbf{p}_1), \dots, (\mathbf{x}_{2t}', \mathbf{p}_{2t}) \in X,$$

with $(\mathbf{x}_i', \mathbf{p}_i) = \Psi(\mathbf{x}_i, \mathbf{p}_i)$ for all i, suppose $\begin{bmatrix} \mathbf{a}^\top & \mathbf{b}^\top \end{bmatrix}$ is a solution to the $2t$ linear equations given by (8) for $i = 1, \dots, 2t$, which lies outside the span of all solutions with $\mathbf{a} = \mathbf{0}$ or $\mathbf{b} = \mathbf{0}$. Then the rational function obtained by restricting $\psi_{\mathbf{a},\mathbf{b}}(\mathbf{x}, \mathbf{p})$ to X equals ψ_j.

Proof. The assumption that $\begin{bmatrix} \mathbf{a}^\top & \mathbf{b}^\top \end{bmatrix}$ is a nontrivial linear combination of solutions with $\mathbf{a}, \mathbf{b} \neq \mathbf{0}$ ensures that $\psi_{\mathbf{a},\mathbf{b}}$ is a well-defined, nonzero rational function on X. Such a function of the form (7) is determined by its values on $2t$ generic points of X. Since ψ_j, by assumption, is also such a function, the $2t$ linear constraints (8) force ψ_j and $\psi_{\mathbf{a},\mathbf{b}}$ to agree on X. □

Thus, to interpolate ψ_j, we may determine from the linear equations (8) a $2t \times 2t$ Vandermonde-type coefficient matrix $\mathbf{A}$. We represent the nullspace of $\mathbf{A}$ by the column-span of a matrix $\mathbf{N}$ with $2t$ rows. Although Proposition 4.1 can be viewed as a uniqueness statement, the matrix $\mathbf{N}$ will generally have more than one column, even for generic samples $(\mathbf{x}_1, \mathbf{p}_1), \dots, (\mathbf{x}_{2t}, \mathbf{p}_{2t}) \in X$. The "extra" columns of $\mathbf{N}$ appear for two reasons:

(1) There may exist different representatives of ψ_j on X of the form (7), whose coefficient vectors are linearly independent.
(2) The nullspace of $\mathbf{A}$ may contain *spurious solutions* not satisfying the hypothesis $\mathbf{a} = \mathbf{0}$ or $\mathbf{b} = \mathbf{0}$ in Proposition 4.1. For instance, fixing $\mathbf{b} = \mathbf{0}$ we may interpolate polynomial functions of the form $\sum_{k=1}^t a_k \cdot (\mathbf{x}, \mathbf{p})^{\boldsymbol{\alpha}_k}$ vanishing on X. In the same way, fixing $\mathbf{a} = \mathbf{0}$ we interpolate polynomial functions of the form $\sum_{k=1}^t b_k \cdot (\mathbf{x}, \mathbf{p})^{\boldsymbol{\beta}_k}$ vanishing on X.

For some applications it may be necessary to pick a sparse representative from the nullspace of $\mathbf{A}$. In general, finding the sparsest vector in the nullspace of a matrix is NP-hard [6]. Nevertheless, in many cases we may find a relatively good sparse representative by looking at the reduced row echelon form of $\mathbf{N}^\top$ for some particular ordering of its columns and picking one with the fewest zeros subject to the additional constraints $\mathbf{a}, \mathbf{b} \neq \mathbf{0}$. We illustrate some of the choices involved on two simple examples.

Example 4.2. Let $X = \mathbf{V}(x^2 + px + 1)$, $f(x, p) = p$. The Galois/monodromy group and deck transformation group are both S_2. When interpolating a nontrivial deck transformations of degree $D = 1$, we obtain the reduced row echelon form for $\mathbf{N}^\top$ below.

$$\widetilde{\mathbf{N}} = \begin{array}{c} \begin{matrix} 1 & x & p & 1 & x & p \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \frac{1}{x} \\ -x-p \end{matrix}$$

We see that $\Psi(x, p)$ has 2 different representatives $\frac{1}{x}$ and $-x - p$, which both agree on X. There is no clear choice of "best representative". In terms of sparsity, the representative $\frac{1}{x}$ is superior. However, one might instead prefer $-x - p$ since it is a polynomial.

Example 4.3. Consider the branched cover

$$\begin{aligned} f \colon \mathbf{V}(x^2 + x + p, x + y + p) &\to \mathbb{C} \\ (x, y, p) &\mapsto p, \end{aligned}$$

which has a unique non-identity deck transformation $\Psi = (\psi_1, \psi_2)$. If we interpolate parameter-dependent deck transformations, we may find matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ representing Ψ which are 8×8. The reduced row echelon forms of the transposed nullspaces are

$$\widetilde{\mathbf{N}_1} = \begin{array}{c} \begin{matrix} 1 & x & y & p & 1 & x & y & p \end{matrix} \\ \begin{bmatrix} 1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} \frac{1-y}{-1-y-p} \\ \frac{x+y}{y+p} \\ \frac{p}{-y-p} \\ \text{spurious,} \end{matrix}$$

and for $\psi_2(x, y, p)$ we have

$$\widetilde{\mathbf{N}_2} = \begin{array}{c} \begin{matrix} 1 & x & y & p & 1 & x & y & p \end{matrix} \\ \begin{bmatrix} 1 & 0 & -1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \begin{array}{l} 1-y-2p \\ \text{spurious} \\ \text{spurious.} \end{array}$$

If we are not interested in the sparsest representative, then we may take $\psi_1 = \frac{p}{-y-p}$ and $\psi_2 = 1 - y - 2p$.

In this example, it is possible to find the sparsest polynomial representative for ψ_1 by solving an auxiliary linear system. In other words, we compute a linear combination of rows $\begin{bmatrix}\mathbf{a}^\top & \mathbf{b}^\top\end{bmatrix} = \mathbf{r}^\top\widetilde{\mathbf{N}_1}$ such that $\mathbf{b}^\top = \begin{bmatrix}1 & \mathbf{0}^\top\end{bmatrix}$ and $\mathbf{a}^\top$ contains the minimum number of zeros. First, to obtain $\mathbf{b}^\top = \begin{bmatrix}1 & \mathbf{0}^\top\end{bmatrix}$, we solve a linear system obtained from the right 4×4 block of $\widetilde{\mathbf{N}_1}$,

$$\left(\mathbf{r}^\top\widetilde{\mathbf{N}_1}\right)_{:,5:8} = \begin{bmatrix}1 & \mathbf{0}^\top\end{bmatrix}.$$

The general solution of this system is given by

$$\mathbf{r}^\top = \begin{bmatrix}-1 & r & r+1 & 0\end{bmatrix}, \quad r \in \mathbb{C}.$$

Using $\mathbf{r}$ to form a linear combination of rows now from the *left* 4×4 block of $\widetilde{\mathbf{N}_1}$, we obtain

$$\mathbf{a}^\top = \begin{bmatrix}-1 & r & r+1 & r+1\end{bmatrix}.$$

To maximize the sparsity, we may set $r = -1$ to obtain

$$\begin{bmatrix}\mathbf{a}^\top & \mathbf{b}^\top\end{bmatrix} = \begin{bmatrix}-1 & -1 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

which encodes the function

$$\psi_1(x, y, p) = -x - 1.$$

Our pseudocode in Algorithm 1 outlines a degree-by-degree procedure for interpolating the full set of deck transformations up to a given degree D^*. To implement such a procedure, there are many design choices that could improve performance or meet the needs of a particular task. Among the design choices, we note that the monodromy, parameter homotopy, and `get_representative` subroutines on respective lines 1, 8, and 16 are left unspecified. Our implementation relies on `HomotopyContinuation.jl` for the first two of these subroutines. For `get_representative`, our implementation chooses the sparsest row in the rref matrix $\widetilde{\mathbf{N}_j}$. For the final output of line 18, we heuristically truncate "small" entries of $\widetilde{\mathbf{N}_j}$ of size $< 10^{-5}$.

Finally, we note the following improvements to the pseudocode in Algorithm 1, which we have used in our implementation.

(1) Computing the monodromy group and centralizer in lines 1–2 is an offline task which only needs to be performed once for a given family of systems.
(2) In practice, we might only need to recover generators of the deck transformation group. The needed modifications are trivial, since deck transformations are interpolated independently.
(3) To restart the computation at a higher degree limit D^*, one can use previously-computed samples from X. In principle, one can also draw $> 2t$ samples and compute the nullspace of the resulting rectangular matrices $\mathbf{A}_j$.

Algorithm 1: Interpolating deck transformations

Input: $F = (f_1, \ldots, f_n)$ and $(\mathbf{x}^*, \mathbf{p}^*)$ as in Assumption 4.1, representing f as in (5); an upper bound for the total degree D^* of monomials in each interpolant
Output: Partially-specified rational maps representing the group of deck transformations, $\{\Psi_1, \ldots, \Psi_q\} = \mathrm{Deck}(f)$, with all coordinate functions representable in degree $\leq D^*$ specified

1 $(x^{(1)}, \ldots, x^{(d)}), \mathrm{Mon}(f, \mathbf{p}^*) \leftarrow$ run_monodromy$(F, \mathbf{x}^*, \mathbf{p}^*)$
2 $\{\sigma_1, \ldots, \sigma_q\} \leftarrow \mathrm{Cent}_{S_d}(\mathrm{Mon}(f, \mathbf{p}^*))$
3 $\Psi_1 \leftarrow \mathbf{x}$
4 **for (** $i \leftarrow 2$; $i \leq q$; $i \leftarrow i+1$ **)**
5 $\quad \Psi_i \leftarrow \begin{bmatrix}\text{nothing} & \ldots & \text{nothing}\end{bmatrix}^\top$
6 **for (** $D \leftarrow 1$; $D \leq D^*$; $D \leftarrow D+1$ **)**
7 $\quad t \leftarrow \binom{n+m+D}{D}$, or $\binom{n+D}{D}$ if parameter-independent
8 $\quad$ Track parameter homotopies for $\geq 2t$ samples from X.
9 $\quad$ **for (** $i \leftarrow 2$; $i \leq q$; $i \leftarrow i+1$ **)**
10 $\quad\quad$ **for (** $j \leftarrow 1$; $j \leq n$; $j \leftarrow j+1$ **)**
11 $\quad\quad\quad$ **if** Ψ_{i_j} *is nothing* **then**
12 $\quad\quad\quad\quad \mathbf{A}_j \leftarrow 2t \times 2t$ Vandermonde matrix from (8),
13 $\quad\quad\quad\quad\quad \mathbf{x}'_k = \sigma_i \cdot \mathbf{x}_k$ for $k = 1, \ldots, 2t$
14 $\quad\quad\quad\quad \mathbf{N}_j \leftarrow \mathrm{nullspace}(\mathbf{A}_j)$
15 $\quad\quad\quad\quad \widetilde{\mathbf{N}_j} \leftarrow \mathrm{rref}(\mathbf{N}_j^\top)$
16 $\quad\quad\quad\quad \begin{bmatrix}\mathbf{a}^\top & \mathbf{b}^\top\end{bmatrix}^\top \leftarrow$ get_representative$(\widetilde{\mathbf{N}_j})$
17 $\quad\quad\quad\quad$ **if** $\begin{bmatrix}\mathbf{a}^\top & \mathbf{b}^\top\end{bmatrix}^\top$ *is not nothing* **then**
18 $\quad\quad\quad\quad\quad \Psi_{i_j} \leftarrow \frac{\sum_{k=1}^t a_k \cdot (\mathbf{x},\mathbf{p})^{\boldsymbol{\alpha}_k}}{\sum_{k=1}^t b_k \cdot (\mathbf{x},\mathbf{p})^{\boldsymbol{\beta}_k}}$
19 $\quad$ **if** *all* Ψ_i *are interpolated* **then**
20 $\quad\quad$ **return** $\{\Psi_1, \ldots, \Psi_q\}$
21 **return** $\{\Psi_1, \ldots, \Psi_q\}$

(4) To minimize the number of calls to the parameter homotopy subroutine, one can attempt to track samples in "batches": since every fiber consists of d points and each point gives 1 constraint on ψ_j, then we need to obtain a complete set of d solutions for r different sets of parameters (including $\mathbf{p}$) such that

$$rd \geq 2t \Rightarrow r \geq \left\lceil \frac{2t}{d} \right\rceil. \tag{9}$$

In our experience, this strategy can work well, but comes with the additional caveat that the samples need not satisfy the genericity conditions of Proposition 4.1, since multiple parameter values are duplicated. We encoutered one (ultimately benign) instance of this phenomenon in our study of Alt's problem Section 5.2. In this example, we had $d = 8652 \geq 2t = 650$, and this strategy resulted in many more spurious rows in $\widetilde{\mathbf{N}}$ due to all samples using the same parameter values.

5 EXPERIMENTS

5.1 Five-point relative pose

One of the most well-known minimal problems in computer vision is that of the classical five-point problem. While many solvers exist for this problem [21], and the symmetry is well-known, this section aims to show how the methods in this paper can recover this symmetry without any *a priori* knowledge.

The general set-up is as follows. There are 5 correspondences between 2D image points $\mathbf{x}_1 \leftrightarrow \mathbf{y}_1, \ldots, \mathbf{x}_5 \leftrightarrow \mathbf{y}_5$. These 2D data points are 3×1 vectors whose third coordinates equal 1, and are assumed to be images of 5 world points under two calibrated cameras, where the two camera frames differ by a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. This relative orientation $[\mathbf{R} \mid \mathbf{t}] \in \mathrm{SE}_{\mathbb{R}}(3)$ between the two cameras is what this problem aims to solve for, in addition to each of the five points in 3D space, as measured by their depths with respect to the first and second camera frames.

Writing $\alpha_1, \ldots, \alpha_5$ for the depths with respect to the first camera and $\beta_1, \ldots, \beta_5$ for the depths with respect to the second camera, solutions to the five-point problem must satisfy a system of polynomial equations and inequations:

$$\begin{aligned} \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \quad \det \mathbf{R} = 1, \\ \beta_i \mathbf{y}_i = \mathbf{R}\alpha_i \mathbf{x}_i + \mathbf{t}, \quad \alpha_i, \beta_i \neq 0, \quad \forall\, i = 1, \ldots, 5. \end{aligned} \tag{10}$$

The parameters of the depths, along with $\mathbf{t}$, are defined in projective space, meaning $\mathbf{t}, \alpha_1, \ldots, \alpha_5, \beta_1, \ldots, \beta_5$ can only be recovered up to a common scale factor. One option to remove this ambiguity is to treat these unknowns as homogeneous coordinates on a 12-dimensional projective space, then for generic data $\mathbf{x}_1, \ldots, \mathbf{x}_5, \mathbf{y}_1, \ldots, \mathbf{y}_5$, there are at most finitely many solutions in $(\mathbf{R}, \mathbf{t}, \alpha_1, \ldots, \alpha_5, \beta_1, \ldots, \beta_5) \in \mathrm{SO}_{\mathbb{C}}(3) \times \mathbb{P}^{12}_{\mathbb{C}}$ to the system (10). This finite-ness is what creates the minimal problem structure. In practice, these solutions may be computed by working in a fixed affine patch of $\mathbb{P} = \mathbb{P}^{12}_{\mathbb{C}}$ (eg. $\alpha_1 = 1$.)

There are exactly 20 solutions over the complex numbers for generic data in $Z = \left(\mathbb{C}^2 \times \{1\}\right)^5 \times \left(\mathbb{C}^2 \times \{1\}\right)^5$. The solutions to (10) are naturally identified with the fibers of a branched cover $f : X \to Z$, where X is the incidence correspondence

$$\begin{aligned} X = \{(\mathbf{R}, (\mathbf{t}, \alpha_1, \ldots, \alpha_5, \beta_1, \ldots, \beta_5), (\mathbf{x}_1, \ldots, \mathbf{x}_5, \mathbf{y}_1, \ldots, \mathbf{y}_5)) \\ \in \mathrm{SO}_{\mathbb{C}}(3) \times \mathbb{P}^{12}_{\mathbb{C}} \times Z \mid (10) \text{ holds } \}. \end{aligned}$$

Alternatively, the problem may be formulated using a branched cover between affine spaces of the same dimension $\mathbb{C}^{20} \dashrightarrow \mathbb{C}^{20}$, eg. using *Cayley's parametrization* $\mathbb{C}^3 \dashrightarrow \mathrm{SO}_{\mathbb{C}}(3)$.

With our chosen formulation, the branched cover f has a single deck transformation Ψ known as the *twisted pair*. We refer to [10, §1], and (11) below for explicit formulas for Ψ, which show that Ψ consists of component functions of total degree at most 3. The effect of this deck transformation on solutions to the five-point problem is illustrated on the left in Figure 1. The component functions $\Psi(\mathbf{R}, \mathbf{t})$ are parameter-independent, whereas the components of $\Psi(\alpha_1, \ldots, \beta_5)$ are parameter-*dependent.*

We ran Algorithm 1 on the formulation (10) with the upper bound for the total degree $D^* = 3$. However, when running Algorithm 1, we considered only the parameter-independent setting, for which $t = \binom{22+3}{3} = 2300$. In the parameter-dependent setup, we would have $2t = 2\binom{22+20+3}{3} = 28380$ coefficients to interpolate. This seemed to exceed the capacity of the machine we used.[2]

The computation described above succeeded in revealing known formulas for the twisted pair on $\mathbf{R}$ and $\mathbf{t}$, namely

$$\psi(\mathbf{R}, \mathbf{t}, \alpha_1, \ldots, \beta_5) = \left(2\frac{\mathbf{t}\mathbf{t}^\top}{\mathbf{t}^\top \mathbf{t}} - \mathbf{I}\right)\mathbf{R}, \quad \psi(\mathbf{R}, \mathbf{t}, \alpha_1, \ldots, \beta_5) = \mathbf{t}. \tag{11}$$

For the coordinate functions corresponding to α_i or β_i, no reasonable representative was found—all rows of $\widetilde{\mathbf{N}}$ were such that $\mathbf{a} \approx \mathbf{0}$ or $\mathbf{b} \approx \mathbf{0}$. These coordinate functions remain "nothing" in Algorithm 1. This is expected, because the twisted pair is parameter-dependent for the depths $\alpha_1, \ldots, \beta_5$.

The experiment described above took approximately 20 minutes on our machine. Due to the low number of solutions and relatively small value of t, computing monodromy and tracking the solutions to the additional parameter values (lines 1 and 8) took only seconds. The bottleneck of the algorithm in the case was computing the nullspaces of $\mathbf{A}_j \in \mathbb{C}^{4600 \times 4600}$ for $j = 1, \ldots, 22$. In total, we sampled the solutions for $r = \left\lceil \frac{2t}{d} \right\rceil = \left\lceil \frac{4600}{20} \right\rceil = 230$ random instances using the "batch" strategy described at the end of Section 4.

5.2 Nine-point Four-bar path generation

We now turn our attention to *Alt's problem.* This is a classic problem of kinematic synthesis which was first solved using homotopy continuation in work of Morgan, Sommese, and Wampler [32]. Several more recent works have used monodromy to verify their result, eg. [16, 23, 24]. Here we explain how this problem can be modeled using a branched cover, and show how its well-known symmetry group can be recovered in our approach.

The formulation we use follows [32], employing the standard convention of *isotropic coordinates.* A vector in the plane is represented by two variables $x, \overline{x} \in \mathbb{C}$. For the purpose of solving polynomial systems, x and $\overline{x}$ are treated as *independent* complex variables; for any physically meaningful solutions, these coordinates will be related by complex conjugation. With this convention, angles $T = e^{i\theta}$ are modeled by points on the hyperbola $T\overline{T} = 1$.

In Figure 1, the vectors x and y point from the coupler point p_0 to the upper joints of the four bar, and vectors a and b point from P_0 towards the ground pivots. The four-bar mechanism has four revolute joints: two connecting the left "crank" and right "rocker" bars to the ground pivots, and another two connecting these bars to the base of the coupler triangle. The motion of the mechanism is induced by rotating the crank bar about its ground pivot. Atop the coupler triangle sits the coupler point, which traces out a curve as the mechanism moves. Without loss of generality, we may assume $(0, 0)$ is a point on this curve.

Alt's problem can be stated as follows: given nine task positions $p_0 = 0, p_1, \ldots, p_8 \in \mathbb{C}$, determine the mechanism parameters x, y, a, b and angles Q_j, T_j, S_j such that the coupler point moves from p_0 to p_i for $i = 1, \ldots, 8$. Here $T_j = e^{i\lambda_j}, S_j = e^{i\mu_j}$ as in Figure 1 (right), and $Q_j = e^{i\theta_j}$ gives the rotation of $\star$ as the coupler point moves from p_0 to p_j.

[2] All timings reported were obtained with a 2022 Mac M1 with 8GB RAM.

Referring to Figure 1, we may write down for each $j = 1, \ldots, 8$ four loop-closure equations,

$$\begin{aligned} Q_j(x-a) &= T_j x + p_j - a, \\ S_j(y-b) &= T_j y + p_j - b, \end{aligned} \tag{12}$$

and their conjugates. Consequently, the orientation of the coupler point may be written as a rational function in the mechanism parameters and the other angles,

$$T_j(a,b,x,y,Q_j,S_j) = (y-x)^{-1}(S_j(y-b)+Q_j(a-x)+b-a). \tag{13}$$

The rocker angle S_j is an algebraic function of degree 2 in the quantities $\mathbf{x} = (x, \bar{x}, y, \bar{y}, a, \bar{a}, b, \bar{b})$ and the crank angle Q_j. That is,

$$A(\mathbf{x}, Q_j)\, S_j^2 + B(\mathbf{x}, Q_j) S_j + C(\mathbf{x}, Q_j) = 0 \tag{14}$$

for some $A, B, C \in \mathbb{Q}[\mathbf{x}, Q_j]$. We note that for generic, fixed values of mechanism parameters $\mathbf{x}$, this equation defines an elliptic curve in the affine plane of $(Q_j, S_j) \in \mathbb{C}^2$. Since the discriminant of the quadratic (14) is square-free, we may define an irreducible variety

$$X' = \{(\mathbf{x}, Q_1, \ldots, S_8) \in \mathbb{C}^{24} \mid (14),\ A(\mathbf{x}, Q_j) \neq 0 \text{ hold},\ j = 1, \ldots, 8\}.$$

Using (12), each coupler point can now be expressed in terms of rational functions on X', say $p_j(\mathbf{x}, Q_j, S_j)$, and $\bar{p}_j(\mathbf{x}, \bar{Q}_j, \bar{S}_j)$ for the conjugate. We may then take as an irreducible variety of problem-solution pairs $X \subset \mathbb{C}^8 \times \mathbb{C}^{16}$ be the closed image of X' under the map $(\mathbf{x}, \mathbf{D}, \mathbf{Q}) \mapsto (\mathbf{x}, \mathbf{p}(\mathbf{x}, \mathbf{Q}, \mathbf{S})), \bar{\mathbf{p}}(\mathbf{x}, \mathbf{Q}, \mathbf{S}))$. This gives a branched cover $f : X \to \mathbb{C}^{16}$. Although not yet formally proved, there is strong evidence that $\deg f = 8652$. Following the elimination strategy described in [32], we obtain a system of 8 equations

$$f_1(\mathbf{x}; \mathbf{p}, \bar{\mathbf{p}}) = \ldots = f_8(\mathbf{x}; \mathbf{p}, \bar{\mathbf{p}}) = 0 \tag{15}$$

that vanishes on X and satisfies Assumption 4.1. With this formulation, we have two parameter-independent deck transformations: a *label-swapping* that exchanges the crank and rocker bars

$$\Psi_{\text{swap}}(\mathbf{x}) = (y, \bar{y}, x, \bar{x}, b, \bar{b}, a, \bar{a}), \tag{16}$$

(we omit the dependence of Ψ on $\mathbf{p}, \bar{\mathbf{p}}$), and the *Roberts cognate* map

$$\Psi_{\text{Rob}}(\mathbf{x}) = \left(\frac{(x-a)y}{x-y}, \frac{(\bar{x}-\bar{a})\bar{y}}{\bar{x}-\bar{y}}, \frac{bx-ay}{x-y}, \frac{\bar{b}\bar{x}-\bar{a}\bar{y}}{\bar{x}-\bar{y}}, a-x, \bar{a}-\bar{x}, a, \bar{a} \right). \tag{17}$$

Despite its simplicity in these variables, we note that extending Ψ_{Rob} to the eliminated variables $\{Q_j, S_j, T_j\}$ yields parameter-dependent coordinate functions.

We ran Algorithm 1 on the formulation (15) with the upper bound for the total degree $D^* = 2$. As with the previous experiment, we assumed parameter-independent deck transformations, so that $t = \binom{8+2}{2} = 45$. In total, we sampled $r = \lceil \frac{2t}{d} \rceil = \lceil \frac{90}{8652} \rceil = 1$ instance. In this case, due to the relatively large number of solutions, *monodromy* was the bottleneck, taking approximately 15 minutes. The subsequent interpolation tasks took approximately 3 minutes in total. We were able to interpolate both label-swapping (16) and Roberts cognates (17), as well as the other 3 nontrivial deck transformations they generate. The bottleneck during the interpolation phase, as expected, is again nullspace computation for $\mathbf{A}_{i_j} \in \mathbb{C}^{8652 \times 90}$, $i = 2, \ldots, 6$, $j = 1, \ldots, 8$.

The first numerical evidence that $\deg f = 8652$ was given in [32]. Later on, the lower bound $\deg f \geq 8652$ was certified by Hauenstein and Sottile using Smale's α-theory [17]. A rigorous proof that this bound is tight remains an open problem. More recently, Sottile and Yahl have posed the problem of determining the Galois/monodromy group of the branched cover f [27, §7.3]. From equations (15), we heuristically computed permutations in $\text{Mon}(f)$ using the software package `HomotopyContinuation.jl` [4]. This produced 4 permutations using default settings. Using Proposition 3.5, we determine that the deck transformation group is isomorphic to S_3, generated by a transposition and 3-cycle corresponding to (16) and (17), respectively. This confirms that these symmetries generate the full deck transformation group of f. Our attempts to determine the order of the full Galois/monodromy group using the Julia interface to GAP [22, 28] did not succeed after 2 days of computation. However, we were able to easily determine that image of the homomorphism $\text{Mon}(f) \to S_{1442}$ given by the action on the maximal block system was the full-symmetric group. [3] Based on what we know, it seems plausible that these permutations generate the group $S_3 \wr S_{1442}$, where $S_3 \to S_6$ via the regular representation. Is this really the case, and do they generate $\text{Mon}(f)$?

6 CONCLUSION

In summary, we have proposed a novel method for recovering hidden symmetries of commonly-occuring parametric polynomial systems. Despite its heuristic nature, our experiments demonstrate that the method is capable of delivering results on examples with a relatively larger number of solutions (namely Section 5.2) or with relatively large numerical interpolation subproblems (Section 5.1).

One obvious avenue for further research is to test more examples and develop better heuristics. There is also potential for fruitful contact with more traditional methods of symbolic computation. In addition to our comments in Section 2, we point out that some hybrid symbolic-numerical methods may be useful in practice. For instance, it seems plausible that one could (1) run Algorithm 1 until recovering coordinate functions for the deck transformations on some subset of variables $\mathbf{y} \subset \mathbf{x}$, then (2) eliminate the remaining variables $\mathbf{x} \setminus \mathbf{y}$ and use parametric Gröbner bases to solve for their coordinate functions using the interpolated expressions from step (1). Such hybrid methods might also be useful for recovering deck transformations when a decomposition as in Definition 2 is already known, or vice-versa. Development of numerical methods for determining the maps and intermediate variety appearing in such a decomposition is also an appealing next step.

ACKNOWLEDGMENTS

T. Duff acknowledges support from NSF DMS-2103310. V. Korotynskiy and T. Pajdla acknowledge support from EU RDF IMPACT No. CZ.02.1.01/0.0/0.0/15 003/0000468 and EU H2020 No. 871245 SPRING projects. V. Korotynskiy was partially supported by the Grant Agency of CTU in Prague project SGS23/056/OHK3/1T/13. We thank Taylor Brysiewicz for helpful conversations that got us up to speed on Julia package development.

[3] The permutations we computed can be viewed here: https://github.com/vviktorrK/DecomposingPolynomialSystems.jl/blob/main/src/examples/robotics/alt/alt_monodromy.txt.

REFERENCES

[1] Carlos Améndola, Julia Lindberg, and Jose Israel Rodriguez. 2016. Solving Parameterized Polynomial Systems with Decomposable Projections. https://arxiv.org/abs/1612.08807

[2] Carlos Beltrán and Anton Leykin. 2013. Robust certified numerical homotopy tracking. *Found. Comput. Math.* 13, 2 (2013), 253–295. https://doi.org/10.1007/s10208-013-9143-2

[3] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM Review* 59, 1 (2017), 65–98. https://doi.org/10.1137/141000671

[4] Paul Breiding and Sascha Timme. 2018. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In *Mathematical Software – ICMS 2018*. Springer International Publishing, Cham, 458–465.

[5] Taylor Brysiewicz, Jose Israel Rodriguez, Frank Sottile, and Thomas Yahl. 2021. Solving decomposable sparse systems. *Numer. Algorithms* 88, 1 (2021), 453–474. https://doi.org/10.1007/s11075-020-01045-x

[6] Thomas F. Coleman and Alex Pothen. 1986. The null space problem. I. Complexity. *SIAM J. Algebraic Discrete Methods* 7, 4 (1986), 527–537. https://doi.org/10.1137/0607059

[7] Annie A. M. Cuyt and Wen-shin Lee. 2011. Sparse interpolation of multivariate rational functions. *Theor. Comput. Sci.* 412, 16 (2011), 1445–1456. https://doi.org/10.1016/j.tcs.2010.11.050

[8] Bernard Deconinck and Mark van Hoeij. 2001. Computing Riemann matrices of algebraic curves. Vol. 152/153. 28–46. https://doi.org/10.1016/S0167-2789(01)00156-7 Advances in nonlinear mathematics and science.

[9] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. 2019. Solving polynomial systems via homotopy continuation and monodromy. *IMA J. Numer. Anal.* 39, 3 (2019), 1421–1446.

[10] Timothy Duff, Viktor Korotynskiy, Tomas Pajdla, and Margaret H. Regan. 2022. Galois/Monodromy Groups for Decomposing Minimal Problems in 3D Reconstruction. *SIAM Journal on Applied Algebra and Geometry* 6, 4 (2022), 740–772. https://doi.org/10.1137/21M1422872

[11] Jean-Charles Faugère, Joachim von zur Gathen, and Ludovic Perret. 2010. Decomposition of generic multivariate polynomials. In *Symbolic and Algebraic Computation, International Symposium, ISSAC 2010, Munich, Germany, July 25-28, 2010, Proceedings*, Wolfram Koepf (Ed.). ACM, 131–137. https://doi.org/10.1145/1837934.1837963

[12] André Galligo and Adrien Poteaux. 2009. Continuations and Monodromy on Random Riemann Surfaces. In *Proceedings of the 2009 Conference on Symbolic Numeric Computation* (Kyoto, Japan) *(SNC '09)*. Association for Computing Machinery, New York, NY, USA, 115–124. https://doi.org/10.1145/1577190.1577210

[13] Allen Hatcher. 2002. *Algebraic topology*. Cambridge University Press, Cambridge.

[14] Jonathan D. Hauenstein, Ian Haywood, and Alan C. Liddell, Jr. 2014. An *a posteriori* certification algorithm for Newton homotopies. (2014), 248–255. https://doi.org/10.1145/2608628.2608651

[15] Jonathan D. Hauenstein, Jose Israel Rodriguez, and Frank Sottile. 2018. Numerical computation of Galois groups. *Found. Comput. Math.* 18, 4 (2018), 867–890. https://doi.org/10.1007/s10208-017-9356-x

[16] Jonathan D. Hauenstein and Samantha N. Sherman. 2020. Using Monodromy to Statistically Estimate the Number of Solutions. In *2nd IMA Conference on Mathematics of Robotics, Manchester, UK, 9-11 September 2020 (Springer Proceedings in Advanced Robotics, Vol. 21)*, William Holderbaum and Jon M. Selig (Eds.). Springer, 37–46. https://doi.org/10.1007/978-3-030-91352-6_4

[17] Jonathan D. Hauenstein and Frank Sottile. 2012. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Trans. Math. Software* 38, 4 (2012), Art. 28, 20. https://doi.org/10.1145/2331130.2331136

[18] Erich Kaltofen and Zhengfeng Yang. 2007. On exact and approximate interpolation of sparse rational functions. In *Symbolic and Algebraic Computation, International Symposium, ISSAC 2007, Waterloo, Ontario, Canada, July 28 - August 1, 2007, Proceedings*, Dongming Wang (Ed.). ACM, 203–210. https://doi.org/10.1145/1277548.1277577

[19] Abraham Martín del Campo and Jose Israel Rodriguez. 2017. Critical points via monodromy and local methods. *J. Symbolic Comput.* 79, part 3 (2017), 559–574. https://doi.org/10.1016/j.jsc.2016.07.019

[20] Rick Miranda. 1995. *Algebraic curves and Riemann surfaces*. Graduate Studies in Mathematics, Vol. 5. American Mathematical Society, Providence, RI. xxii+390 pages. https://doi.org/10.1090/gsm/005

[21] David Nistér. 2004. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 6 (2004), 756–777. https://doi.org/10.1109/TPAMI.2004.17

[22] OSCAR 2023. OSCAR – Open Source Computer Algebra Research system, Version 0.11.3-DEV. https://oscar.computeralgebra.de

[23] Anna Eckhardt Paul Breiding and Sascha Timme. [n. d.]. Alt's problem. https://www.JuliaHomotopyContinuation.org/examples/alts-problem/. Accessed: June 27, 2022.

[24] Mark M Plecnik and Ronald S Fearing. 2017. Finding only finite roots to large kinematic synthesis systems. *Journal of Mechanisms and Robotics* 9, 2 (2017), 021005.

[25] J. F. Ritt. 1922. Errata: "Prime and composite polynomials" [Trans. Amer. Math. Soc. **23** (1922), no. 1, 51–66; 1501189]. *Trans. Amer. Math. Soc.* 23, 4 (1922), 431. https://doi.org/10.2307/1988887

[26] Andrew J. Sommese, Jan Verschelde, and Charles W. Wampler. 2001. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM J. Numer. Anal.* 38, 6 (2001), 2022–2046. https://doi.org/10.1137/S0036142900372549

[27] Frank Sottile and Thomas Yahl. 2021. Galois Groups in Enumerative Geometry and Applications. (2021). https://doi.org/10.48550/ARXIV.2108.07905

[28] The GAP Group 2022. *GAP – Groups, Algorithms, and Programming, Version 4.12.2*. The GAP Group. https://www.gap-system.org

[29] J. van der Hoeven. 2011. *Reliable homotopy continuation*. Technical Report. HAL. http://hal.archives-ouvertes.fr/hal-00589948/fr/.

[30] Joris van der Hoeven and Grégoire Lecerf. 2021. On sparse interpolation of rational functions and gcds. *ACM Commun. Comput. Algebra* 55, 1 (2021), 1–12. https://doi.org/10.1145/3466895.3466896

[31] Joachim von zur Gathen, Jaime Gutierrez, and Rosario Rubio. 1999. On Multivariate Polynomial Decomposition. In *Proceedings of the Second Workshop on Computer Algebra in Scientific Computing, CASC 1999, Munich, Germany, May 31 - June 4, 1999*, Victor G. Ganzha, Ernst W. Mayr, and Evgenii V. Vorozhtsov (Eds.). Springer, 463–478. https://doi.org/10.1007/978-3-642-60218-4_35

[32] C. W. Wampler, A. P. Morgan, and A. J. Sommese. 1992. Complete Solution of the Nine-Point Path Synthesis Problem for Four-Bar Linkages. *Journal of Mechanical Design* 114, 1 (03 1992), 153–159. https://doi.org/10.1115/1.2916909 arXiv:https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/114/1/153/5507001/153_1.pdf

[33] Juan Xu, Michael Burr, and Chee Yap. 2018. An approach for certifying homotopy continuation paths: univariate case. In *ISSAC'18—Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*. ACM, New York, 399–406. https://doi.org/10.1145/3208976.3209010