Routing Functions for Robot Paths that Avoid Singularities and Algebraic Obstacles

Jonathan D. Hauenstein[0000−0002−9252−8210],
Emma L. Schmidt[0009−0004−5731−5457], and
Charles W. Wampler[0000−0002−9790−1600]

University of Notre Dame, Notre Dame, IN 46556 USA,
{hauenstein,eschmi23,cwample1}@nd.edu,
WWW home page: http://hauenstein.phd, http://www.cwampler.com

Abstract. Routing functions provide a provably complete method of analyzing the smooth connectivity of a real algebraic variety minus an algebraic hypersurface. When applied to robot path planning, the real variety in question is the solution of the kinematic loop equations, and the algebraic hypersurface to be avoided can be singularity conditions or algebraic obstacles or some combination thereof. This paper provides an introduction to routing functions and illustrates their use for studying the connectivity of the singularity-free, collision-free workspace of a 2DOF planar five-bar robot with a line obstacle. After an initial computation of routing points and the corresponding routing road map, the routing function provides a way of quickly computing a feasible path between any two points in the same smoothly connected component.

Keywords: kinematics, singularity, robot path connectivity, real algebraic geometry, routing functions

1 Introduction

Planning robot paths that avoid obstacles, including avoiding singularities, is a longstanding problem in robotics with a rich history. We discuss the application of a recent development from real algebraic geometry called *routing functions* to robot path planning [2, 5]. This new approach has features in common with two of the main genres of path-planning algorithms. Like Probabilistic Road Maps (PRM) [6, 8, 9], it builds a road map that can be queried to retrieve a path between any two points in the same connected component of the robot's collision-free configuration space. Like a navigation function [7], its paths are determined by gradient flow lines of a continuous function.

There are also differences. PRMs are probabilistically complete, which means that if any path exists, they will find one given enough samples. However, the number of samples required to terminate grows if the configuration space has narrow passages; and if the number of samples is constrained, failure to find a path does not imply that no path exists. In contrast, routing functions give a

definitive answer as to whether two points lie in the same connected component, and the complexity of computing the associated routing road map is only weakly affected by narrow passages. Meanwhile, in contrast to a navigation function, which is an artificial potential function with a single local minimum at a preset target point, a routing function is much simpler to construct and supports multiple queries to start-point/target-point pairs. Gradient descent connects any given point to the routing road map, so by connecting both the starting point and the target point to the road map, one obtains a path between any two points in the same connected component.

As a potential downside, routing functions require the configuration space and the obstacles to be representable as algebraic sets, and the complexity of computing routing points is directly dependent on the degrees of those sets. This approach is highly effective for singularity avoidance, as the singularity loci of most robot mechanisms are naturally algebraic. Simple obstacles, such as joint limits or a wall modeled as a linear half-space, can also be treated effectively, but routing functions may not be well-suited to planning in cluttered environments.

To illustrate the method, we consider a 2DOF planar robot consisting of a five-bar loop with one link extended to define the robot's end-effector point. We analyze the connectivity of the robot's configuration space after the locus of input singularities is excluded and also consider scenarios where the configuration space is further reduced by subtracting a buffer zone around the singularities and by introducing a barrier wall that limits the travel of the end-effector. Since planar linkages are most efficiently modeled using isotropic coordinates, we use them in our analysis. The adjustments necessary for adapting routing functions to isotropic coordinates, where the definition of "real" is different than in Cartesian coordinates, is an additional contribution herein.

The rest of the paper is structured as follows. Section 2 provides an overview of routing functions in real affine space, gives an example with a narrow passage in Section 2.1, and translates the method into isotropic coordinates in Section 2.2. Section 3 applies routing functions to a 2DOF planar five-bar robot. A short conclusion is provided in Section 4.

2 Connectivity and routing functions

We wish to find paths between points in a real algebraic configuration space $V_{\mathbb{R}}(F) = \{x \in \mathbb{R}^n \mid F(x) = 0\}$, where $F = \{f_1, \ldots, f_k\}$ is a system of k polynomials describing the kinematics of a robot mechanism. We further require the path to avoid the loci where $b_j(x) = 0$ for any $b_j \in B = \{b_1, \ldots, b_m\}$ and to obey real polynomial inequalities $p_j(x) > 0$ for all $p_j \in P = \{p_1, \ldots, p_\ell\}$. The polynomial systems B and P may represent singularities to avoid, joint limits to obey, or obstacles in the robot's workspace. Taken together, we wish to compute the connected components of the constrained configuration manifold $X \subset \mathbb{R}^n$ where

$$X = V_{\mathbb{R}}(F) \cap \bigcap_{\alpha=1}^{m} \{b_\alpha \neq 0\} \cap \bigcap_{\beta=1}^{\ell} \{p_\beta > 0\}. \tag{1}$$

Moreover, given any two points in X, we wish to determine if they are in the same connected component, and if so, to find a path in X between them.

Routing functions [2, 5] provide a means of answering these questions. A routing function is a rational function, and the form we consider here is

$$r(x) = \frac{N(x)}{q(x)^d} = \frac{\prod_{\alpha=1}^{m} b_\alpha(x) \cdot \prod_{\beta=1}^{\ell} p_\beta(x)}{q(x)^d}, \qquad q(x) = 1 + \|x - c\|^2, \quad (2)$$

where $c \in \mathbb{R}^n$ is randomly selected, $\|x - c\|^2 = (x_1 - c_1)^2 + \cdots + (x_n - c_n)^2$, and $d \in \mathbb{Z}_{>0}$ such that

$$2d > \sum_{\alpha=1}^{m} \deg(b_\alpha) + \sum_{\beta=1}^{\ell} \deg(p_\beta). \quad (3)$$

The randomness of c together with the inequality on d ensure that r is a Morse function that vanishes at infinity. See [2, Thm. 3.4] for more details.

For any connected component of X, points on the boundary satisfy $r(x) = 0$, and the sign of $r(x)$ is the same everywhere inside. Let us assume the sign is negative. (For the positive case, just flip the sign of r.) Gradient descent from any point in the component must terminate at a critical point of $r(x)$, that is, at a point $x \in X$ satisfying the criticality conditions

$$q(x)\,\nabla N(x) - dN(x)\,\nabla q(x) + \sum_{j=1}^{k} \lambda_j \nabla f_j(x) = 0, \qquad F(x) = 0, \quad (4)$$

where $\lambda_1, \ldots, \lambda_k$ are Lagrange multipliers. We call these points *routing points*.

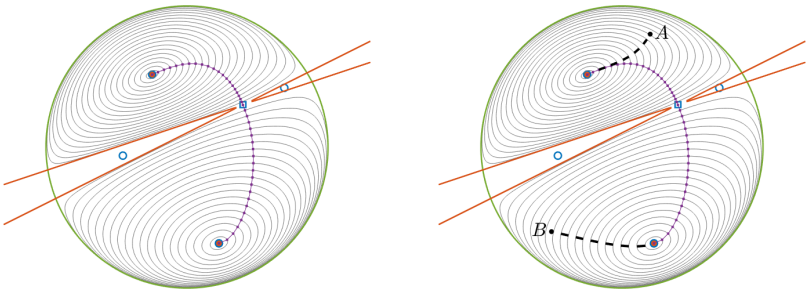
The theory of routing functions guarantees that all the routing points are nonsingular. The signs of the eigenvalues of the Hessian matrix of r evaluated at a routing point reveal whether it is a local minimum or not. At each routing point that is not a local minimum, we initiate gradient descent by stepping from the routing point in both directions along each unstable eigenvector. These descent paths connect all of the routing points that lie in the same connected component, providing a road map for the component.

Once this process is complete, the network of connected routing points can be analyzed to determine the number of connected components. Gradient flow from any point in X terminates at a routing point of a road map, and if two points belong to the same component, they connect to the same road map, which provides a path between them. If desired, post-processing can be used to deform the path to smooth it and optimize it, similar to what is done with Probabilistic Road Maps (PRM) and Rapidly-Exploring Random Trees (RRT) [9].

2.1 Narrow passage example

Consider a point robot (x, y) that must navigate in a domain bounded by polynomials representing the unit circle and a hyperbola:

$$f(x, y) = 1 - (x^2 + y^2), \qquad g(x, y) = (x - 3y + 0.5)(x - 2y + 0.2) + e, \quad (5)$$



(a) (b)

Fig. 1. Routing through a narrow passage: (a) road map, (b) path query

where $e = 0.0001$. First, consider $B = \{f, g\}$ so that $X = \mathbb{R}^2 \cap \{f \neq 0\} \cap \{g \neq 0\}$. As seen in Figure 1, X consists of 7 connected components, 4 outside the unit circle and 3 inside, one of which contains a narrow passage.

The critical system (4) is a pair of degree 5 polynomials. After choosing $c = (0.23, -0.79)$ for the constants in q, we used Bertini [1] to find all solutions, of which 13 were real, all nonsingular. Four of these, the intersections of the circle and hyperbola, are not in X, and four outside the unit circle are not shown. One routing point is a saddle point that lies near the narrow passage. Gradient flow from there connects it to two local extrema, forming the road map for the region. Figure 1(a) shows the contours of r in the region along with the resulting routing road map. When queried to connect points A and B in Figure 1(b), the method returns the path shown. If desired, postprocessing could shorten and smooth this feasible path.

Moreover, if we change $e = 0.0001$ to $e = 0$, the hyperbola degenerates into a pair of lines, cutting off the narrow passage. Unlike a random sampling method, the complexity of the computation does not change as $e \to 0$. For tiny values of $e > 0$, the critical point calculation directly pinpoints the spot where the boundary must be examined for the existence of a narrow passage.

2.2 Isotropic coordinates

Planar robots are often treated more simply in isotropic coordinates, where points in the Cartesian (x, y)-plane are mapped into the complex plane $\mathbb{C}$ as $z = x + iy$ with $i = \sqrt{-1}$. The squared distance of (x, y) from the origin is zz^*, where z^* is the complex conjugate of z. Since complex conjugation is not algebraic, we introduce a conjugate variable $\bar{z} = x - iy$, and represent the Cartesian point (x, y) by the isotropic point $(z, \bar{z})$. This changes the definition of "real" in isotropic coordinates: the point is real if and only if $z^* = \bar{z}$. Rotation of vector $v \in \mathbb{C}$ by angle Θ is $e^{i\Theta}v$. To model this in isotropic coordinates, we introduce $(\theta, \bar{\theta})$ with the algebraic constraint $\theta\bar{\theta} = 1$, so that $(v, \bar{v})$ rotates to $(\theta v, \bar{\theta}\bar{v})$.

It is easy to adapt routing functions to isotropic coordinates. Suppose that $(z, \bar{z}) \in (\mathbb{C} \times \mathbb{C})^n$ is an array of n isotropic coordinate pairs. Define the con-

jugate $\bar{f}$ of a polynomial f as the result of swapping z with $\bar{z}$ and replacing each coefficient with its complex conjugate. We require that every polynomial in systems F, B, and P be either self-conjugate, i.e., equal to itself under conjugation, or a member of a self-conjugate pair, e.g., $(\bar{f}, \bar{g}) = (g, f)$. After changing $V_{\mathbb{R}}(F)$ to mean isotropic real instead of Cartesian real, the only other modification needed to proceed is to change the denominator polynomial $q(x)$ in (2) to $1 + (z - \gamma)(\bar{z} - \gamma^*)$ for random complex $\gamma \in \mathbb{C}^n$. (If the problem has a mix of real and isotropic variables, the conjugate of a real variable x is itself, and the corresponding term in q keeps the form $(x - c)^2$.) With these modifications, the function r takes real values for isotropically real arguments, and it retains all the properties required of a routing function.

Gradients and Hessians of r in isotropic coordinates are computed exactly as in Cartesian coordinates. However, when computing the gradient and Hessian of r restricted to X, which is used for gradient flow and determining unstable eigenvector directions, respectively, one must use a basis for the isotropically real tangent space of X, that is, one must account for the fact that variations on X are isotropically real.

3 Planar 5-bar

Figure 2 shows a schematic of a planar five-bar robot, which has input angles (Θ_1, Θ_2) and output position $R = (x, y)$. We model its kinematics using isotropic coordinates, so that the output position is $(r, \bar{r})$, $r = x + iy$ and rotations of links 1-4 are given by $\theta_1, \theta_2, \theta_3, \theta_4$ and their conjugates. Link lengths $\ell_1, \ell_2, \ell_3, \ell_4$ are real, while p designates the output point in the local coordinates of link 3. Constants a and b locate the ground pivots in world coordinates.

The diagram in Figure 2 has two loops: one around the five-bar loop and one from the origin O to endpoint R and back. For each loop, summing the vectors around it gives an equation in $r, \theta_1, \theta_2, \theta_3, \theta_4$ along with its conjugate equation in $\bar{r}, \bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3, \bar{\theta}_4$. We may solve these for $\theta_3, \theta_4, \bar{\theta}_3, \bar{\theta}_4$ and substitute the results

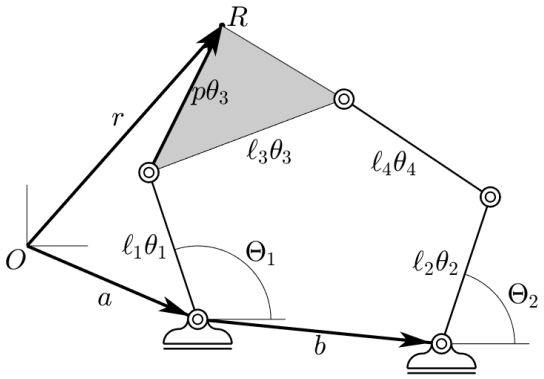


Fig. 2. Five-bar robot schematic

into $\theta_3\bar{\theta}_3 - 1$ and $\theta_4\bar{\theta}_4 - 1$ to write the kinematics as $F = \{f_1, f_2, f_3, f_4\}$ with

$$\begin{aligned}
f_1 &= (r - a - \ell_1\theta_1)(\bar{r} - a^* - \ell_1\bar{\theta}_1) - pp^*,\\
f_2 &= [\ell_1\theta_1 + \mu(r - a - \ell_1\theta_1) - \ell_2\theta_2 - b][\ell_1\bar{\theta}_1 + \mu^*(\bar{r} - a^* - \ell_1\bar{\theta}_1) - \ell_2\bar{\theta}_2 - b^*] - \ell_4^2,\\
f_3 &= \theta_1\bar{\theta}_1 - 1,\\
f_4 &= \theta_2\bar{\theta}_2 - 1,
\end{aligned}$$

where $\mu = \ell_3/p$ is a complex constant.

Following [3], an input singularity (also called a type II singularity and DKP singularity) is a singularity of the forward kinematics problem and should be avoided. Letting $f_{12} = [f_1 \;\; f_2]^T$, input singularities are defined by the 2×2 determinant $D = \det\left[\partial f_{12}/\partial r \; \partial f_{12}/\partial \bar{r}\right]$. Although D is not self-conjugate, iD is self-conjugate and can be used in the routing function framework.

The following considers three scenarios using parameters from [3, Ex. 1] after adjusting to the different labeling scheme:

$$\begin{array}{cccc}
a = 0.259 + 0.586i, & b = -0.199 + 0.004i, & p = 0.049 + 0.328i,\\
\ell_1 = 0.465, & \ell_2 = 0.249, \quad \ell_3 = 0.349, & \ell_4 = 0.411.
\end{array}$$

To form the routing function denominator, we order the variables as (r, θ_1, θ_2) and randomly select $\gamma = (0.67 - 0.82i, 0.95 - 0.28i, 0.73 - 0.15i)$. We then solve the critical system (4) using regeneration [4] in Bertini [1] running on an Intel Xeon E5-2680 v3 (2.50 GHz) with 24 threads and 256 GB RAM. For the three scenarios, this takes 0.4, 2.1, and 5.4 minutes, respectively.

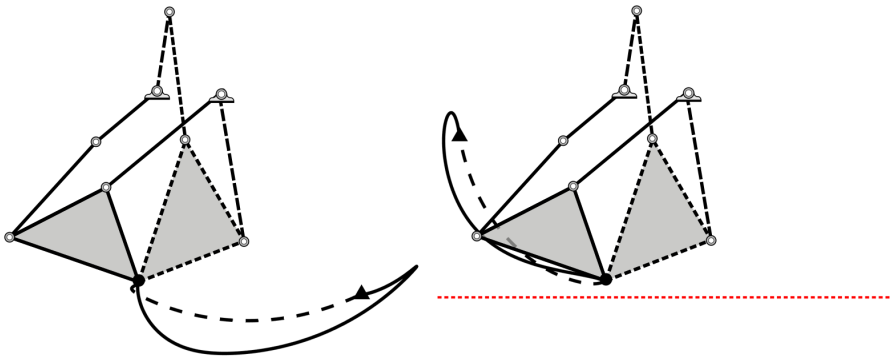
3.1 Remove input singularities

Using affine coordinates, [2, § 6.4] determined that the configuration space with the input singularities removed consisted of two connected components. Repeating this computation with isotropic coordinates confirms that result. In particular, $V_{\mathbb{R}}(F) \cap \{iD > 0\}$ and $V_{\mathbb{R}}(F) \cap \{iD < 0\}$ are each connected with each having a routing road map consisting of one local extrema and three saddles.

3.2 Remove kerf around input singularities

The inverse kinematics of the five-bar has up to four real solutions, and one may desire to move between these to change the force and velocity characteristics. Since it is important to maintain a safety margin away from input singularities at all times, [3] planned paths using a sampling technique that discarded any point within a "kerf" cut out along the singularity curve. To closely mimic this, we use the fact that $\frac{1}{2}|g|/\sqrt{(\partial g/\partial r)(\partial g/\partial \bar{r})}$ is a first-order approximation of the affine distance from $(r, \bar{r})$ to $V_{\mathbb{R}}(g)$ for fixed $(\theta_1, \bar{\theta}_1, \theta_2, \bar{\theta}_2)$. Hence, for $\epsilon = 0.075$, we replace $iD \neq 0$ with the inequality

$$p_1 = (iD)^2 - 4 \cdot \epsilon^2 \cdot \frac{\partial(iD)}{\partial r} \cdot \frac{\partial(iD)}{\partial \bar{r}} > 0 \tag{6}$$



(a) (b)

Fig. 3. Routing between two solutions: (a) with kerf, (b) with kerf and floor

to remove additional points around the set of input singularities. Even with the kerf removed, we find that there are still two connected components, namely $V_{\mathbb{R}}(F) \cap \{p_1 > 0\} \cap \{iD > 0\}$ and $V_{\mathbb{R}}(F) \cap \{p_1 > 0\} \cap \{iD < 0\}$. Each component has a routing road map consisting of one local extrema and three saddles.

We query this road map to solve the problem posed in [3, Ex. 1] of connecting two configurations, namely $(\theta_1, \theta_2) \approx (-0.806 - 0.592i, -0.803 - 0.596i)$ and $(\theta_1, \theta_2) \approx (0.105 - 0.995i, 0.182 + 0.983i)$, with the same output $(r, \bar{r}) = (0, 0)$, shown as the solid and dashed configurations in Figure 3(a), respectively. Paths generated by gradient flow from these points end at the same local extreme, marked with a small triangle, thereby providing a singularity-free path for switching configurations.

3.3 Kerf around input singularities and a floor obstacle

For the final example, we consider adding an obstacle representing a floor, $y = (r - \bar{r})/(2i) > -0.05$, while keeping the same kerf around input singularities. Modifying the routing function to include this inequality along with p_1 from (6) results in a routing map that has two connected components, one contained in $\{iD > 0\}$ and the other in $\{iD < 0\}$. As in the other two cases, the connected component contained in $\{iD > 0\}$ has a route map consisting of one local extrema and three saddles. For the other connected component that is contained in $\{iD < 0\}$, the set of routing points gains a second local extreme and the number of saddles increases to four. Gradient flows from the saddles connect the local extrema into a single road map for the region.

With the floor obstacle present, the configuration-switching problem gives the path shown in Figure 3(b). Even though the floor interferes with the path found earlier, the two configurations are still in the same connected component inside of $\{iD < 0\}$, so the routing function provides a singularity-free path between them. As before, the gradient flows lead to the same local extreme, marked with a small triangle, but the presence of the floor has significantly altered the path.

4 Conclusion

Routing functions provide a complete approach for determining paths in an algebraic configuration space that avoid singularities and obstacles described by algebraic conditions. As demonstrated on a 2DOF planar five-bar robot, one first solves a polynomial system to find routing points and then connects these to form a road map for each connected component. A connectivity query can be used to determine if two points are in the same smoothly connected component and, if so, rapidly determine a collision-free path between them. This approach is demonstrated for finding paths that respect a buffer zone around input singularities and avoid a floor obstacle. Although we have presented a planar example, the method applies as well to spatial robots with open-chain or closed-loop topologies. We expect its greatest usefulness will be for analyzing singularity-free workspaces, which are naturally formulated as algebraic constraints.

Acknowledgments

JDH and ELS were supported in part by NSF CCF 2331400 and Robert and Sara Lumpkins Collegiate Professorship. CWW was supported in part by Huisking Foundation, Inc. Research Collegiate Professorship.

References

1. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: Software for Numerical Algebraic Geometry. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5.
2. Cummings, J., Hauenstein, J., Hong, H., Smyth, C.: Smooth connectivity in real algebraic varieties. Numerical Algorithms, 100:63-84, (2025).
3. Edwards, P.B., Baskar, A., Hills, C., Plecnik M., Hauenstein, J.D.: Output mode switching for parallel five-bar manipulators using a graph-based path planner. 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 9735-9741.
4. Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Regeneration homotopies for solving systems of polynomials. Math. Comp., 80(273):345-377, (2011).
5. Hong, H.: Connectivity in semi-algebraic sets. In: 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2010, pp. 4–7. IEEE Computer Society (2010).
6. Kavraki, L.E., Svestka, P., Latombe, J.-C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom., 12(4):566-580, (1996).
7. Koditschek, D.E., Rimon, E.: Robot navigation functions on manifolds with boundary. Advances Applied Math., 11(4):412-442, (1990).
8. Latombe, J.-C.. Robot motion planning. Vol. 124. Springer Science & Business Media (2012).
9. LaValle, S.M. Planning algorithms. Cambridge University Press, (2006).