

An Analytic Solution to the 3D CSC Dubins Path Problem

Victor M. Baez¹, Nikhil Navkar², Aaron T. Becker¹

Abstract—We present an analytic solution to the 3D Dubins path problem for paths composed of an initial circular arc, a straight component, and a final circular arc. These are commonly called CSC paths. By modeling the start and goal configurations of the path as the base frame and final frame of an RRRR manipulator, we treat this as an inverse kinematics problem. The kinematic features of the 3D Dubins path are built into the constraints of our manipulator model. Furthermore, we show that the number of solutions is not constant, with up to seven valid CSC path solutions even in non-singular regions. An implementation of solution is available at <https://github.com/aabecker/dubins3D>.

I. INTRODUCTION AND RELATED WORK

In 1957, Lester Eli Dubins proved that the shortest path between two (x, y, θ) coordinates for a forward-moving vehicle with a minimum turning radius r is composed entirely of no more than three components, where each component is either a circular arc at minimum turning radius or a straight line [1]. In a common shorthand notation, ‘C’ is a circular arc at the maximum curvature and ‘S’ is a straight-line component. The shortest curvature-constrained path is fully solved in 2D, with four to six candidate solutions of the form CSC or CCC everywhere. The 3D extension has applications in drone control, underwater vehicles, tendon transmission paths, directional well trajectory planning, and manipulator design [2]–[23]. In exciting recent work, Chen et al. in [3] used the path computation from [24] to plan 3D Dubins paths for robot models.

Hota and Ghose provided a numerical method to efficiently find a 3D CSC path [2], and further expanded this method to find up to four different CSC paths in [24]. Some works split the 3D problem into a 2D Dubins path in (x, y) and a separate pitch command for Z , incorporating an iterative procedure as in [4], or adding a separate helical trajectory for altitude adjustment as in [5]. Exhaustive search methods based on optimal control were formed and tested on the 3D Dubins path in [25]. However, these methods are only resolution complete and the results are only close to optimal over a search space. In later work [26], authors of [25] describe their search as usually slow. In contrast, this paper provides an analytic method to determine all candidate

This work was initiated at the Dagstuhl Seminar 23091, “Algorithmic Foundations of Programmable Matter” in 2023. We thank Cynthia Sung for presenting this open problem and for discussions on practical applications. Support by the National Priority Research Program (NPRP) award (NPRP13S-0116-200084) from the Qatar National Research Fund (a member of The Qatar Foundation), the Alexander von Humboldt Foundation, and the National Science Foundation under [IIS-1553063, 1849303, 2130793].

¹Electrical Engineering, University of Houston, TX USA
{vjmontan, atbecker}@uh.edu

²Department of Surgery, Hamad Medical Corporation, Doha, Qatar
NNavkar@hamad.qa

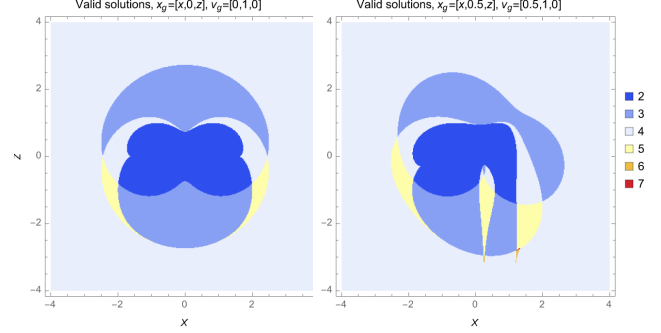


Fig. 1. 2D slices of the 5-DOF configuration space, showing that the number of valid solutions to the CSC Dubins path ranges from 2 to 7 for the configurations shown. See video overview at <https://youtu.be/aWfmgsl0JU>.

CSC paths in 3D. Given these paths, it is then trivial to choose the shortest path, or to choose between the candidate options to satisfy other control objectives.

A key advantage of the analytic solution over a numeric approach is that it provides all possible solutions. As shown in Fig. 1, the number of valid CSC solutions varies over the workspace. Another benefit is consistent run time. The numerical approach in [24] took 11.0 ± 11.6 seconds to find all seven solutions in Fig. 6, but our Mathematica implementation solves for all seven in 0.02499 ± 0.00097 seconds. All tests were run on a Macbook Pro with an Apple M2 Pro processor and 32 Gb unified memory. Converting the code to a compiled language is left for future work, but should increase the speed.

A. Problem statement

The CSC 3D path with a minimum curvature radius r has five inputs as shown in Fig. 2: the bending angle of the first C path ψ_1 , the orientation ϕ_1 of this angle with regard to the (x_0, y_0) plane, the S extension length $d \geq 0$, the bending angle of the second C path ψ_2 and the orientation ϕ_2 of this angle with regard to the (x_1, y_1) plane. We define our coordinate plane so that the path starts at $\mathbf{x}_0 = [0, 0, 0]^\top$ with orientation $\mathbf{v}_0 = [0, 0, 1]^\top$. We adopt the shorthand $\cos(\psi) = c_\psi$, and $\sin(\psi) = s_\psi$. Then the final orientation $\mathbf{v}_g$ and position $\mathbf{x}_g$ are given as follows.

$$\mathbf{v}_g = \begin{bmatrix} s_{\psi_2}(c_{\psi_1}c_{\phi_1}c_{\phi_2} - s_{\phi_1}s_{\phi_2}) + s_{\psi_1}c_{\psi_2}c_{\phi_1} \\ s_{\psi_2}(c_{\psi_1}s_{\phi_1}c_{\phi_2} + c_{\phi_1}s_{\phi_2}) + s_{\psi_1}c_{\psi_2}s_{\phi_1} \\ c_{\psi_1}c_{\psi_2} - s_{\psi_1}s_{\psi_2}c_{\phi_2} \end{bmatrix} \quad (1)$$

$$\mathbf{x}_g = \begin{bmatrix} c_{\phi_1}(s_{\psi_1}(d + rs_{\psi_2}) + rc_{\psi_1}(c_{\phi_2}(1 - c_{\psi_2}) - 1) + r) + r(c_{\psi_2} - 1)s_{\phi_1}s_{\phi_2} \\ s_{\phi_1}(s_{\psi_1}(d + rs_{\psi_2}) + rc_{\psi_1}(c_{\phi_2}(1 - c_{\psi_2}) - 1) + r) - r(c_{\psi_2} - 1)c_{\phi_1}s_{\phi_2} \\ c_{\psi_1}(d + rs_{\psi_2}) + rs_{\psi_1}(c_{\phi_2}(c_{\psi_2} - 1) + 1) \end{bmatrix} \quad (2)$$

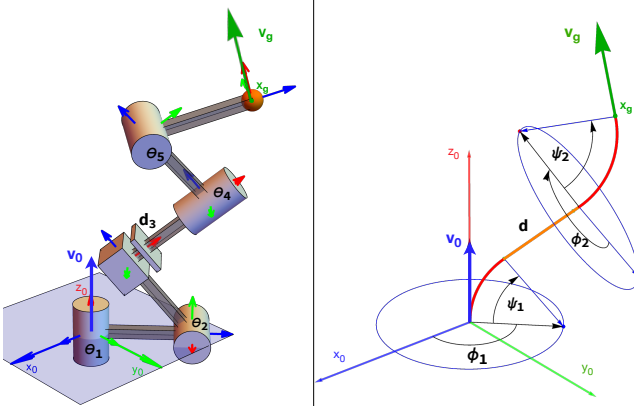


Fig. 2. Figure showing our RRPRR analogue of the 3D CSC Dubins path. The left shows the RRPRR manipulator that we solve the inverse kinematics for, the right shows the equivalent 3D CSC Dubins path.

The inverse kinematics problem is to solve for the $[\phi_1, \psi_1, d, \phi_2, \psi_2]$ that produce the orientation specified by a unit vector $\mathbf{v}_g$ and position specified by $\mathbf{x}_g$.

II. MODEL

Balkom, Furtuna, and Wang point out a relationship between kinematic descriptions of mobile robots driven by constant controls and serial manipulators in [27]; they conclude this work by suggesting that there may be an arm-like model for 3D Dubins paths. We model the 3D CSC Dubins path as an RRPRR manipulator, defining θ_i as the curved components and the prismatic joint as the d_3 straight path component. An illustration of our manipulator Dubins path analogue is shown in Fig. 2.

TABLE I: THE CSC DUBINS PATH AS A 5-DOF RRPRR ARM.

Denavit-Hartenberg (DH) Parameters					
Joint	Type	r	α	d	θ_i
1	revolute	1	$\pi/2$	0	θ_1
2	revolute	1	$\pi/2$	0	θ_2
3	prismatic	0	0	d_3	0
4	revolute	1	$\pi/2$	0	θ_4
5	revolute	1	$\pi/2$	0	θ_5

Table I lists the Denavit-Hartenberg (DH) parameters corresponding to our RRPRR manipulator model shown in Figure 2. DH parameters specify the transformation between consecutive frames as a composition of a rotation about the z -axis and a rotation about the rotated x -axis [28]. The conversion from DH parameters to Dubins parameters is

$$\begin{bmatrix} \phi_1 \\ \psi_1 \\ d \\ \phi_2 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \pi - \theta_2 \\ d_3 \\ \theta_4 - \pi \\ \pi - \theta_5 \end{bmatrix}. \quad (3)$$

The relationship between frames i and $i - 1$ is

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & r_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & r_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

and the transformation matrix relating the final and first frame is

$$A_1 A_2 A_3 A_4 A_5 = A_{hand}, \quad (5)$$

where

$$A_{hand} = \begin{bmatrix} l_x & m_x & v_{gx} & x_{gx} \\ l_y & m_y & v_{gy} & x_{gy} \\ l_z & m_z & v_{gz} & x_{gz} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

The l and m terms are ignored because $\mathbf{x}_g$ and $\mathbf{v}_g$ only specify the final z -axis and position. See Section III-A for details.

III. SOLUTION

To solve the inverse kinematics problem for our manipulator, we follow closely the inverse kinematics solution of a 6R manipulator developed by Raghavan and Roth in [29]. The solution involves forming and solving a system of multivariate polynomials in a way that is similar to Gauss elimination but for nonlinear equations – techniques deeply rooted in resultant-based elimination methods pioneered by Newton and Euler, and later popularized mostly by Sylvester. The goal is to eliminate all but one variable resulting in a polynomial whose roots can be substituted back into the system to solve for the remaining variables. We use the same notation as [29]; any deviation is intended for clarity.

A. Initial setup

We substitute our DH parameters from Table I and begin our solution by first moving the terms in eq. (5) so that

$$A_3 A_4 A_5 = A_2^{-1} A_1^{-1} A_{hand}. \quad (7)$$

The choice of terms on the left and right of eq. (7) is discussed in Section VI-A. From here we work with the six equations from columns three and four of the resulting expansion and refer to them as $\tilde{\mathbf{I}}$ and $\tilde{\mathbf{p}}$:

$$\tilde{\mathbf{I}} = \begin{bmatrix} c_4 s_5 \\ c_5 \\ s_4 s_5 \end{bmatrix} = \begin{bmatrix} v_{gx} c_1 c_2 + v_{gy} s_1 c_2 + v_{gz} s_2 \\ v_{gz} c_2 - v_{gx} c_1 s_2 - v_{gy} s_1 s_2 \\ -v_{gy} c_1 + v_{gx} s_1 \end{bmatrix} \quad (8a)$$

$$\tilde{\mathbf{p}} = \begin{bmatrix} c_4(1 + c_5) + 1 \\ -d_3 - s_5 \\ s_4(1 + c_5) \end{bmatrix} = \begin{bmatrix} -c_2(x_{gx} c_1 + x_{gy} s_1 - 1) + x_{gz} s_2 \\ s_2(1 - x_{gx} c_1 - x_{gy} s_1) + x_{gz} c_2 \\ x_{gx} s_1 - x_{gy} c_1 \end{bmatrix} \quad (8b)$$

We form eight new equations from $\tilde{\mathbf{I}}$ and $\tilde{\mathbf{p}}$: two equations from $\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}}$ and $\tilde{\mathbf{p}} \cdot \tilde{\mathbf{I}}$; three equations from $\tilde{\mathbf{p}} \times \tilde{\mathbf{I}}$; and three more equations from $(\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}})\tilde{\mathbf{I}} - (2\tilde{\mathbf{p}} \cdot \tilde{\mathbf{I}})\tilde{\mathbf{p}}$. For reference, see the

section, “The Ideal of $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{I}}$ ” in [29] or [30] on elimination theory in general and proofs regarding the independence of generated ideals.

These 14 equations are treated as linear combinations of s_4 and c_4 , which [29] calls the “suppressed” variables. Our choice in having the s_4 and c_4 terms as our suppressed variables and the tradeoffs between choosing these and not the other terms as our suppressed variables is discussed in Section VI-B. The 14 equations are written as

$$\mathbf{P}(s_4, c_4) \begin{bmatrix} d_3^2 s_5 \\ d_3^2 c_5 \\ d_3^2 \\ d_3 s_5 \\ d_3 c_5 \\ d_3 \\ s_5 \\ c_5 \\ 1 \end{bmatrix} = \mathbf{Q}(v_{g_i}, x_{g_i}) \begin{bmatrix} s_1 s_2 \\ s_1 c_2 \\ c_1 s_2 \\ c_1 c_2 \\ s_1 \\ c_1 \\ s_2 \\ c_2 \end{bmatrix}. \quad (9)$$

B. Eliminating θ_1 and θ_2

Eight out of 14 equations in (9) are used to solve for the eight terms on the right of (9) in terms of the eight terms on the left. These solutions are substituted into the remaining six equations resulting in a system

$$\Sigma(s_4, c_4, v_{g_i}, x_{g_i}) \begin{bmatrix} d_3^2 s_5 \\ d_3^2 c_5 \\ d_3^2 \\ d_3 s_5 \\ d_3 c_5 \\ d_3 \\ s_5 \\ c_5 \\ 1 \end{bmatrix} = 0_{6 \times 1}, \quad (10)$$

where $\Sigma(s_4, c_4, v_{g_i}, x_{g_i})$ is a 6×9 matrix.

C. Eliminating d_3 and θ_5

Using the tangent half-angle identity, the two terms, s_4 and c_4 in $\Sigma(s_4, c_4, v_{g_i}, x_{g_i})$ can be treated as one variable, x_4 with the trigonometric substitution

$$s_4 \leftarrow \frac{2x_4}{1+x_4^2} \quad \text{and} \quad c_4 \leftarrow \frac{1-x_4^2}{1+x_4^2},$$

so that $\Sigma(s_4, c_4, v_{g_i}, x_{g_i})$ becomes $\Sigma(x_4, v_{g_i}, x_{g_i})$. This x_4 can be mapped to a unique θ_4 value. After multiplying out the denominators, we then multiply each equation in $\Sigma(x_4, v_{g_i}, x_{g_i})$ by d_3 to create 6 additional equations (see [31] and [32] on *resultants* and *dialytic elimination*). Although this creates three additional terms ($d_3^3 s_5, d_3^3 c_5, d_3^3$),

we arrive at a 12×12 system

$$\begin{bmatrix} \Sigma(x_4, v_{g_i}, x_{g_i}) & 0_{6 \times 3} \\ 0_{6 \times 3} & \Sigma(x_4, v_{g_i}, x_{g_i}) \end{bmatrix} \begin{bmatrix} d_3^3 s_5 \\ d_3^3 c_5 \\ d_3^3 \\ d_3^2 s_5 \\ d_3^2 c_5 \\ d_3^2 \\ d_3 s_5 \\ d_3 c_5 \\ d_3 \\ s_5 \\ c_5 \\ 1 \end{bmatrix} = 0_{12 \times 1}. \quad (11)$$

To make (11) have a non-zero solution, set the determinant

$$\begin{vmatrix} \Sigma(x_4, v_{g_i}, x_{g_i}) & 0_{6 \times 3} \\ 0_{6 \times 3} & \Sigma(x_4, v_{g_i}, x_{g_i}) \end{vmatrix} = 0, \quad (12)$$

and solve (12) for a single polynomial in terms of x_4 [31]. After factoring and removing roots that are always imaginary, we are left with a 12th order polynomial. [29] calls this the *characteristic polynomial*, because it characterises the maximum number of possible solutions. Finally we substitute the v_{g_i} and x_{g_i} values and solve for the roots.

D. Solving for $\theta_4, d_3, \theta_5, \theta_1$, and θ_2

For each real root x_4 , θ_4 is calculated as $\theta_4 = 2 \arctan(x_4)$ and then we proceed with the following two steps to get up to 12 solution sets. We discard solution sets with values of d_3 that are negative, because these do not apply to the 3D CSC Dubins path.

1) *solving for d_3 and θ_5* : Substituting the solution for x_4 and v_{g_i} and x_{g_i} into eq. (11) and solving the system yields the solution for d_3 , and the solutions for the terms s_5 and c_5 . Then $\theta_5 = \arctan2(c_5, s_5)$.

2) *solving for θ_1 and θ_2* : v_{g_i}, x_{g_i} and the solutions for d_3, θ_4 , and θ_5 are substituted into the system, eq. (9). Next, eight of the 14 equations are used to solve for the terms in the left column of eq. (9). Then $\theta_1 = \arctan2(c_1, s_1)$ and $\theta_2 = \arctan2(c_2, s_2)$.

We note here that in Sections III-B and III-C, when these systems are solved, the multivariate terms, $d_3^2 s_5, d_3^2 c_5, \dots, s_1 s_2, s_1 c_2, \dots$ are treated as single-term variables; this is standard in the treatment of multivariate polynomials using resultant-based elimination methods. However, here in step two, the single-term variables for the d_3 -terms and the θ_5 -terms are converted back to multivariate terms when substituting in the solutions for d_3, θ_4 , and θ_5 , while the four multivariate θ_1 -and- θ_2 -terms are left as single-term variables.

IV. SPECIAL CASES

“... but as always with geometric algorithms, special cases arise that can be handled separately.”

– David Eberly [33]

There are configurations of $\mathbf{x}_g$ and $\mathbf{v}_g$ that complicate the system (11). As [32] points out, methods that extract

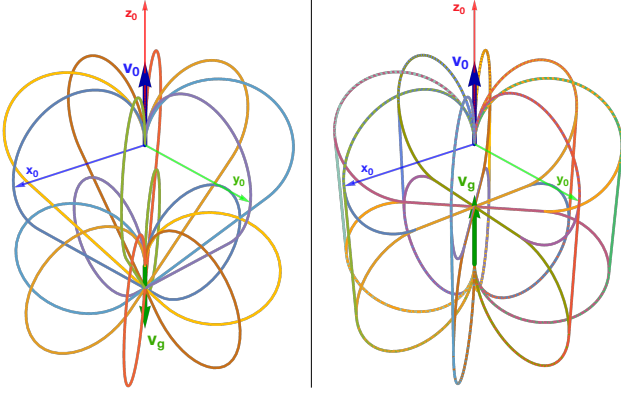


Fig. 3. An example of configurations with infinite valid CSC Dubins paths shown with representative paths. The goal orientation $\mathbf{v}_g$ and the goal position $\mathbf{x}_g$ are colinear with $\mathbf{v}_0$ and $\mathbf{x}_0$. On the left, $\mathbf{v}_g = [0, 0, -1]$ and on the right, $\mathbf{v}_g = [0, 0, 1]$.

resultants from determinants fail for singular matrices. For configurations that make the coefficient matrix in eq. (12) singular, we cannot generate a characteristic polynomial because the determinant is always zero. Fortunately, a few algebraic and geometric observations allow us to find solutions for these configurations.

A. Configurations with one solution

If $\mathbf{v}_g$, $\mathbf{x}_g - \mathbf{x}_0$, and $\mathbf{v}_0$ are parallel, then the goal configuration is reached with a straight line path.

B. Configurations with infinite solutions

If both $\mathbf{v}_g$ and $\mathbf{x}_g$ are aligned with $\mathbf{v}_0$, then there will be infinite valid paths. The first arc begins in any direction, θ_1 , the length of the straight segment will depend on the distance $\mathbf{x}_g - \mathbf{x}_0$, and the second arc will be on the same plane as the first so that the final orientation is colinear with the first. Since we know where the second arc will be, we can supply θ_1 and θ_4 pairs to equations that have not vanished in system (9) and solve for θ_2 , d_3 , and θ_5 . If there is sufficient distance, $\mathbf{x}_g - \mathbf{x}_0 > 2r$, and $\mathbf{v}_g$ is in the same direction as $\mathbf{v}_0$, then there will be two θ_4 -solutions for any θ_1 direction ($\theta_4 = \{0, \pi\}$), but when $\mathbf{v}_g$ and $\mathbf{v}_0$ are in the opposite directions, and $\mathbf{x}_g - \mathbf{x}_0 \leq 2r$, then there will only be one θ_4 -solution for any θ_1 direction ($\theta_4 = 0$). This is illustrated in Figure 3 for both directions of $\mathbf{v}_g$ at the distance threshold, $\mathbf{x}_g - \mathbf{x}_0 = 2r$. Both images show sample paths with eight supplied θ_1 values. The image on the right has 16 corresponding paths when $\mathbf{v}_g$ and $\mathbf{v}_0$ are in the same direction, while the image on the left has eight paths because $\mathbf{v}_g$ and $\mathbf{v}_0$ are in opposite directions.

C. Planar paths in 3D

If $\mathbf{v}_g$ is in the plane that contains the initial orientation $[0, 0, 1]^\top$ and $\mathbf{x}_g$, then the problem is 2D, and can be solved using the methods of [1]. In these cases, we can show that our model suggests planar paths even in 3D space.

In Section III-B, when forming $\Sigma(s_4, c_4, v_{g_i}, x_{g_i})$, many of the terms contain a constant denominator, $x_{g_x} v_{g_y} - x_{g_y} v_{g_x}$.

This quantity is the z -component of $\mathbf{x}_g \times \mathbf{v}_g$. Naturally we multiply this quantity out of the denominators when forming the systems (10), (11), and eq. (12) for the general solution, but the multiples are still present in the system. When $\mathbf{x}_g \times \mathbf{v}_g$ is on the $x_0 y_0$ -plane, $x_{g_x} v_{g_y} - x_{g_y} v_{g_x} = 0$, and system (11) loses many equations, so we cannot use it to generate a characteristic polynomial or solve for d_3 and θ_5 .

In Section III-B, system (10) is in the form $\mathbf{A} \mathbf{b} = 0$ where $\mathbf{A}$ is a 6×8 matrix and $\mathbf{b}$ is an 8×1 column vector. After applying the condition that $x_{g_x} v_{g_y} = x_{g_y} v_{g_x}$ to system (10), $\mathbf{A}$ is a $n \times m$ matrix and $\mathbf{b}$ is an $m \times 1$ column vector where $n < 6$ and $m < 8$. This system can be expanded dialytically to generate the singular $m \times n$ matrix $\mathbf{A}$, where $m = n < 12$. The resultant characteristic polynomial's factors reveal that the roots are all either imaginary or zero.

Now that we know x_4 can only equal zero for these configurations, we can use its geometric implications to find θ_1 , and our original system to solve for θ_2 , d_3 , and θ_5 . The remainder of this subsection refers to Fig. 4, which shows an example of a CSC configuration with planar-path solutions, and we use Dubins-path notation mapped from DH parameters through eq. (2).

To start, we know that $x_4 = 0$ when θ_4 (or ϕ_2) is either 0 or multiples of π , defining one possible plane for the $r\psi_2$ arc. If we know that the $r\psi_2$ arc is on the plane, $\phi_2 = \{0, \pi\}$, then we can also conclude that arc $r\psi_1$ must be on the same plane, because that is the only way that it can produce a trajectory that will arrive on this ϕ_2 -plane. Although a single ϕ_1 -plane can define infinite ϕ_2 -planes in the forward kinematics problem, a fixed ϕ_2 -plane governs only one possible ϕ_1 -plane in the inverse kinematics problem. Using this logic, we can compute ϕ_1 from the projection of the position vector $\mathbf{x}_g$ onto the xy plane, $\phi_1 = \arctan2(x_{g_x}, x_{g_y})$. But this is also the same plane as $\phi_1 = \arctan2(x_{g_x}, x_{g_y}) + \pi$. So for both $\phi_2 = \{0, \pi\}$, and for both $\phi_1 = \{\arctan2(x_{g_x}, x_{g_y}), \arctan2(x_{g_x}, x_{g_y}) + \pi\}$ (or $\phi_1 = \{\arctan2(v_{g_x}, v_{g_y}), \arctan2(v_{g_x}, v_{g_y}) + \pi\}$ if $x_{g_y} = x_{g_x} = 0$) mapped to θ_1 and θ_4 , we can use parts of the original system (9) to solve for θ_2 , d_3 , and θ_5 for a total of four solution sets; this is the order of the repeating zero-roots of the characteristic polynomial. Two equations from $\tilde{\mathbf{p}}$, and two equations from $\tilde{\mathbf{I}}$ are used to solve for a unique θ_2 and θ_5 in terms of d_3 , and then substituted into $\tilde{\mathbf{p}} \cdot \tilde{\mathbf{p}}$ to solve for d_3 . Substituting θ_2 and θ_5 as functions of d_3 into any of the remaining equations from system (9) that have not vanished from the condition $x_{g_x} v_{g_y} = x_{g_y} v_{g_x}$ result in a d_3 -polynomial of order $n \in \{2, 3, 4\}$, and the real roots of these d_3 -polynomials are all equal.

There are additional singular cases that need to be handled separately, but due to page limitations we will document them in future work.

V. RESULTS

To examine the variation in the number of CSC solutions over the configuration space, we generated one million sample points uniformly in the $[-4, 4]$ cube and $\mathbf{v}_g$ uniformly sampled on the unit sphere [34]. We then solved the

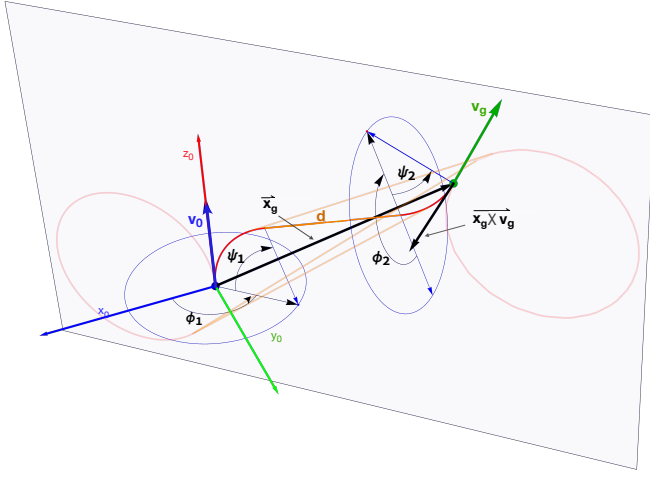


Fig. 4. Planar paths in 3D occur when $\mathbf{x}_g \times \mathbf{v}_g$ is only in the x_0y_0 plane. The four valid CSC paths in this figure are all on the light blue plane defined by $\mathbf{v}_0$, $\mathbf{v}_g$, and $\mathbf{x}_g$. The blue circles are orthogonal to this plane and show the reachable orientations of the ψ_1 and ψ_2 arcs.

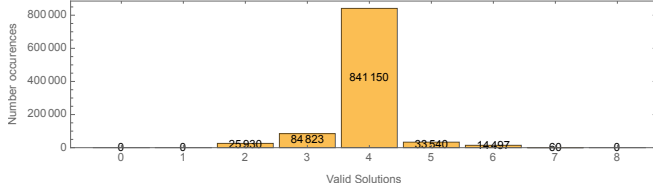


Fig. 5. Number of valid CSC paths for one million trials, with $\mathbf{x}_g$ uniformly sampled in $[-4, 4]^3$ and $\mathbf{v}_g$ uniformly sampled on the unit sphere.

inverse kinematics. A histogram of the number of solutions found are shown in Fig. 5, and in Table II. Each inverse kinematics query required on average 0.025 seconds. For 84% of configurations four solutions were found. The inverse kinematics always found at least two solutions. In 88.9% of the configurations at least four solutions are found, and seven valid solutions were found 60 times. One example of a configuration with seven valid CSC paths is shown in Fig. 6.

Just as the Dubins 2D solution is not a smooth distance function and is not a metric, the distance returned by the shortest 3D CSC paths is also not smooth. The shortest CSC distance plotted for two slices of the configuration space is shown in Fig. 8. The plot for $\mathbf{x}_g = [x, 0, z]^T$, $\mathbf{v}_g = [0, 1, 0]^T$ is relatively simple. Outside the cardioid-shaped region at the center the shortest CSC distance increases with distance from the origin. However, CSC paths inside the cardioid-shaped region have large lengths, especially along the bottom edge. The plot for $\mathbf{x}_g = [x, 0.5, z]^T$, $\mathbf{v}_g = [0.5, 1, 0]^T$ is more complicated, with multiple regimes and a long-distance c-

solutions:	0	1	2	3	4	5	6	7	8
% found:	0	0	2.59	8.48	84.1	3.35	1.44	0.006	0

TABLE II: CSC SOLUTION DISTRIBUTION FOR ONE MILLION TRIALS

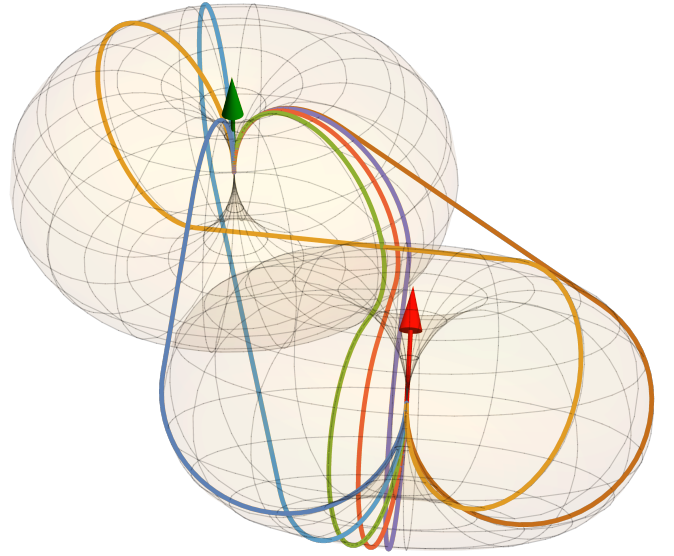


Fig. 6. A configuration with seven valid CSC Dubins paths in 3D. The initial position and orientation at $\mathbf{x}_0 = [0, 0, 0]^T$, $\mathbf{v}_0 = [0, 0, 1]^T$ is shown with a green arrow. The goal position and goal orientation $\mathbf{x}_g = [2.64101, -1.78042, -0.371051]^T$, $\mathbf{v}_g = [-0.323321, 0.729589, 0.602631]^T$ are shown with a red arrow. The tori represent the minimum turning radius $r = 1$.

shaped region at $[x, z] = [-2, -0.25]$.

Our solution is also robust to configurations where the straight line component has zero length. An example of a configuration with five solutions where one of them has a component $d = 0$ is shown in Fig. 7.

VI. DISCUSSION

Critics of dialytic elimination methods warn about the potential for explosive growth in the order of the resultant polynomial for systems where more than one multiple is needed during the expansion process to arrive at a square matrix [32], [35]. For reference, our expansion for the general solution is in Section III-C. In this section we discuss how we avoid such explosive growth and other particularities used in our solution.

A. Choice of left and right terms in the matrix model

The combination of terms on the left and right of (7) is intentional. This combination results in a system of the form:

$$\begin{bmatrix} f(\theta_{4,5}) & f(\theta_4) & f(\theta_{4,5}) & f(\theta_{4,5}) \\ f(\theta_5) & 0 & f(\theta_5) & f(d_3, \theta_5) \\ f(\theta_{4,5}) & f(\theta_4) & f(\theta_{4,5}) & f(\theta_{4,5}) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f(\theta_{1,2}, l) & f(\theta_{1,2}, m) & f(\theta_{1,2}, v_g) & f(\theta_{1,2}, x_g) \\ f(\theta_{1,2}, l) & f(\theta_{1,2}, m) & f(\theta_{1,2}, v_g) & f(\theta_{1,2}, x_g) \\ f(\theta_1, l) & f(\theta_1, m) & f(\theta_1, v_g) & f(\theta_1, x_g) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Different combinations result in different systems, some of which seem more simple because they yield equations with

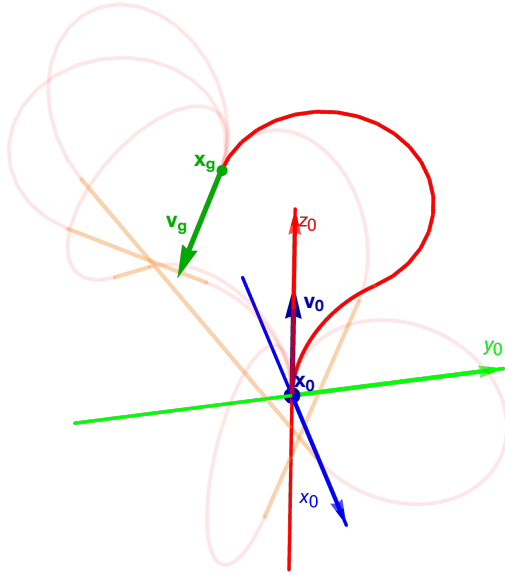


Fig. 7. A configuration with five solutions, where one of the valid paths (highlighted in red) has a zero-length straight line component.

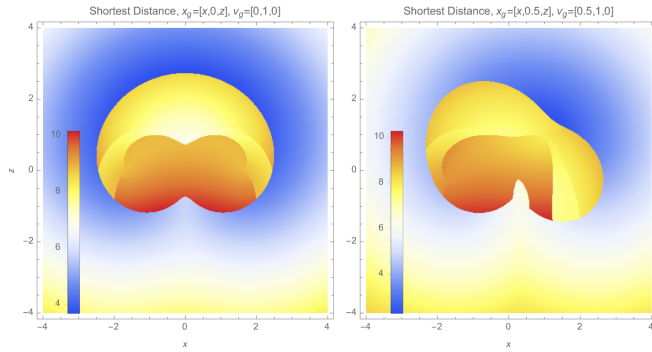


Fig. 8. 2D slices of the configuration spaces in Fig. 1, showing the length of the shortest solution to the CSC Dubins path for each configuration.

less variables to solve for. For example, instead of (7), we could formulate the system as

$$A_3 A_4 = A_2^{-1} A_1^{-1} A_{hand} A_5^{-1}. \quad (14)$$

which would expand to a system of the form:

$$\begin{bmatrix} f(\theta_4) & 0 & f(\theta_4) & f(\theta_4) \\ 0 & -1 & 0 & f(d_3) \\ f(\theta_4) & 0 & f(\theta_4) & f(\theta_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \quad (15)$$

$$\begin{bmatrix} f(\theta_{1,2,5}, l, v_g) & f(\theta_{1,2,5}, l, v_g) & f(\theta_{1,2}, m) & f(\theta_{1,2}, l, x_g) \\ f(\theta_{1,2,5}, l, v_g) & f(\theta_{1,2,5}, l, v_g) & f(\theta_{1,2}, m) & f(\theta_{1,2}, l, x_g) \\ f(\theta_{1,5}, l, v_g) & f(\theta_{1,5}, l, v_g) & f(\theta_1, m) & f(\theta_{1,5}, l, x_g) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Comparing the system in (13) to (15), the columns 3 and 4 do not contain θ_5 in (15), while they do in (13). The system (15) would require less variables to eliminate, and then θ_5 could have been solved for by substituting the 1 through 4 solutions

into

$$A_5 = A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} A_{hand}. \quad (16)$$

However, the combination in (7) accomplishes two things. 1.) For some combinations, when eliminating the first set of variables as in Section III-B, we would be left with less independent equations in system (10) requiring us to expand more than once to match the number of equations to the number of extra terms introduced to the system. 2.) This combination leaves our six equations from columns 3 and 4 in terms of only the last two columns of A_{hand} , x_{g_i} and v_{g_i} . Other combinations of eq. (7) result in a system such as (15), where all the the columns that contain the x_{g_i} and v_{g_i} constants also contain the l_i and m_i constants. Using our form, we are able to keep our solution as a function of only the goal position, x_g and goal orientation, v_g and we do not have to create compatible l and m columns of A_{hand} to solve our system.

B. Choice of suppressed variable

Our manipulator's DH parameters required the suppression of the θ_4 terms. If for example, we suppressed d_3 and the system was treated as linear combinations of the d_3 -terms instead of the θ_4 terms, the resulting 12×12 system as in eq. (11) would suffer from column dependencies that could be traced back to the coefficients of the s_4 and $s_4 c_5$ terms being equal — that and the fact that at every instance, they came in a pair. This would also require us to expand more than once, leading to a larger system.

C. Careful expansion for planar paths

Section IV-C mentions we can arrive at an $n \times m$ matrix A where $n = m \leq 12$ to begin the solution for the special case of planar paths in 3D. After applying the condition for this case, our system (9) reduces, but it still behaves in our favor and allows us to avoid explosive growth. By expanding dialytically on only the equations that remain after eliminating the first set of variables, we avoid introducing extra terms and still arrive at a square matrix without requiring more than one expansion.

VII. CONCLUSIONS

This paper described a method to determine the valid CSC paths for a Dubins car model in 3D. This model obeyed a minimum turning radius of r . The solution was generated by describing the trajectory as an RRPRR robot arm. Self-collision of the arm on this trajectory was not accounted for, but established methods for robot arms could be applied [36]. The work of Chen et al. had joint limits for the generated CSC robot arms [3]. Since our method returns all solutions, they can be easily filtered using axis joint limits to return suitable paths. These methods may also apply to closed-form inverse kinematic solutions for concentric tube robots [37], [38]. An interesting direction for future work would be to expand this method to generate CCC shortest paths, along with Sussman's helicoidal paths [39], to determine the true shortest paths.

REFERENCES

- [1] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [2] S. Hota and D. Ghose, "Optimal path planning for an aerial vehicle in 3d space," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 4902–4907.
- [3] W.-H. Chen, W. Yang, L. Peach, D. E. Koditschek, and C. R. Sung, "Kinigami: Algorithmic design of compliant kinematic chains from tubular origami," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1260–1280, 2023.
- [4] P. Váña, A. Alves Neto, J. Faigl, and D. G. Macharet, "Minimal 3D Dubins path with bounded curvature and pitch angle," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 8497–8503.
- [5] Y. Wang, S. Wang, M. Tan, C. Zhou, and Q. Wei, "Real-time dynamic Dubins-helix method for 3-d trajectory smoothing," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 730–736, March 2015.
- [6] R. Anderson and D. Milutinović, "A stochastic approach to dubins feedback control for target tracking," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 3917–3922.
- [7] M. Owen, R. W. Beard, and T. W. McLain, *Implementing Dubins Airplane Paths on Fixed-Wing UAVs**. Dordrecht: Springer Netherlands, 2015, pp. 1677–1701. [Online]. Available: https://doi.org/10.1007/978-90-481-9707-1_120
- [8] H. Marino, M. Bonizzato, R. Bartalucci, P. Salaris, and L. Pallottino, "Motion planning for two 3d-dubins vehicles with distance constraint," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4702–4707.
- [9] Y. Li, W. Lu, Y. Liu, D. Meng, X. Wang, and B. Liang, "Optimization design method of tendon-sheath transmission path under curvature constraint," *IEEE Transactions on Robotics*, 2023.
- [10] P. Cui, W. Yan, R. Cui, and J. Yu, "Smooth path planning for robot docking in unknown environment with obstacles," *Complexity*, vol. 2018, pp. 1–17, 2018.
- [11] H. Liu, T. B. Gjersvik, and A. Faanes, "Subsea field layout optimization (part i)—directional well trajectory planning based on 3d dubins curve," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109450, 2022.
- [12] J. Herynek, P. Váña, and J. Faigl, "Finding 3d Dubins paths with pitch angle constraint using non-linear optimization," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–6.
- [13] J. Lim, F. Achermann, R. Bähmann, N. Lawrance, and R. Siegwart, "Circling back: Dubins set classification revisited," in *Workshop on Energy Efficient Aerial Robotic Systems, International Conference on Robotics and Automation 2023*, 2023.
- [14] G. Xu, D. Zhu, J. Cao, Y. Liu, and J. Yang, "Shunted collision avoidance for multi-uav motion planning with posture constraints," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3671–3678.
- [15] B. Moon, S. Sachdev, J. Yuan, and S. Scherer, "Time-optimal path planning in a constant wind for uncrewed aerial vehicles using dubins set classification," *IEEE Robotics and Automation Letters*, 2023.
- [16] A. T. Blevins, "Real-time path optimization for 3d uas line survey operations," in *AIAA SCITECH 2023 Forum*, 2023, p. 0395.
- [17] O. I. D. Bashi, H. K. Hameed, Y. M. Al Kubaisi, and A. H. Sabry, "Developing a model for unmanned aerial vehicle with fixed-wing using 3d-map exploring rapidly random tree technique," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 473–481, 2024.
- [18] C. Consonni, M. Brugnara, P. Bevilacqua, A. Tagliaferri, and M. Frego, "A new markov–dubins hybrid solver with learned decision trees," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106166, 2023.
- [19] W. Wu, J. Xu, C. Gong, and N. Cui, "Adaptive path following control for miniature unmanned aerial vehicle confined to three-dimensional dubins path: From take-off to landing," *ISA transactions*, vol. 143, pp. 156–167, 2023.
- [20] C. Hague, A. Willis, D. Maity, and A. Wolek, "Planning visual inspection tours for a 3d dubins airplane model in an urban environment," in *AIAA SCITECH 2023 Forum*, 2023, p. 0108.
- [21] H. Liu, T. B. Gjersvik, and A. Faanes, "Practical application of 3d dubins curve method in directional well trajectory planning," in *Abu Dhabi International Petroleum Exhibition and Conference*. SPE, 2023, p. D021S037R004.
- [22] V. Patsko and A. Fedotov, "Three-dimensional reachability set for a dubins car: Reduction of the general case of rotation constraints to the canonical case," *Journal of Computer and Systems Sciences International*, pp. 1–24, 2023.
- [23] X. Tian, T. Xu, X. Luo, Y. Jia, and J. Yin, "Multi-uav reconnaissance task allocation in 3d urban environments," *IEEE Access*, 2024.
- [24] S. Hota and D. Ghose, "Optimal geometrical path in 3d with curvature constraint," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 113–118.
- [25] W. Wang and P. Li, "Towards finding the shortest-paths for 3D rigid bodies," in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [26] —, "Finding control synthesis for kinematic shortest paths," 2022.
- [27] D. Balkcom, A. Furtuna, and W. Wang, "The Dubins car and other arm-like mobile robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 380–386.
- [28] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [29] M. Raghavan and B. Roth, "Inverse Kinematics of the General 6R Manipulator and Related Linkages," *Journal of Mechanical Design*, vol. 115, no. 3, pp. 502–508, 09 1993. [Online]. Available: <https://doi.org/10.1115/1.2919218>
- [30] D. O. David A. Cox, John Little, *Ideals, Varieties, and Algorithms*. Springer, 2015.
- [31] E. T. Whittaker, "On Sylvester's dialytic method of elimination," *Proceedings of the Edinburgh Mathematical Society*, vol. 40, p. 62–63, 1921.
- [32] D. Kapur, "Algorithmic elimination methods," in *Tutorial Notes, Intl. Symp. on Symbolic and Algebraic Computation (ISSAC)*, Montreal, 1995, pp. 1–32.
- [33] D. Eberly, "Distance to circles in 3d," 2023. [Online]. Available: <https://www.geometrictools.com/Documentation/DistanceToCircle3.pdf>
- [34] G. Marsaglia, "Choosing a point from the surface of a sphere," *The Annals of Mathematical Statistics*, vol. 43, no. 2, pp. 645–646, 1972.
- [35] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, The Robotics Institute Pittsburgh, 2010.
- [36] D. Rakita, B. Mutlu, and M. Gleicher, "Relaxedik: Real-time synthesis of accurate and feasible robot arm motion," in *Robotics: Science and Systems*, vol. 14. Pittsburgh, PA, 2018, pp. 26–30.
- [37] P. E. Dupont, J. Lock, B. Izkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 209–225, 2009.
- [38] Z. Mitros, S. H. Sadati, R. Henry, L. Da Cruz, and C. Bergeles, "From theoretical work to clinical translation: Progress in concentric tube robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 335–359, 2022.
- [39] H. Sussmann, "Shortest 3-dimensional paths with a prescribed curvature bound," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 4, 1995, pp. 3306–3312 vol.4.