Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Towards Finding the Shortest-Paths for 3D Rigid Bodies

Weifu Wang, Ping Li

Cognitive Computing Lab
Baidu Research
No.10 Xibeiwang East Road, Beijing 100193, China
10900 NE 8th St. Bellevue, Washington 98004, USA
{wangweifu, liping11}@baidu.com

***Abstract*—In this work, we analyze and present an algorithm to find shortest-paths for generic rigid bodies. We derived the necessary conditions for optimality using Lagrange multipliers, and compared it to the conditions derived from Pontraygin's Maximum Principle. We derived the equations of the necessary conditions using geometric Jacobian, drawing inspiration from the similarity between the rigid-body systems and the arm-like systems. In the previous work [30], the analysis focused on finding shortest-paths to reach goals in positions only. This work extends the analysis to find the shortest-path to reach a goal with complete configuration in 3D. We show that the algorithm is resolution complete even when the orientations are included. To overcome the complexity of 3D orientations, we describe the system using three points in the robot frame, and show that this parameter system is redundant but can derive the same necessary conditions as those derived using the minimum parameters (configuration). We used a 3D Dubins system to demonstrate the correctness of the analysis and the algorithm.**

I. Introduction

In this work, we study the shortest-path problem for 2D and 3D generic rigid bodies reaching arbitrary configurations. The shortest-path is time-optimal when each segment is followed with the same speed, which corresponds to a sequence of constant controls. The early studies of the shortest-path problem date back to the late 1950s when Dubins presented the shortest geodesics for what we now know as the Dubins' car [13]. The model is a simplified version of a jet-airplane cruising at the same altitude. Further studies of richer kinematic models have been conducted, leading to known solutions to time-optimal control structures for Reeds-Shepp car [22], Differential Drive [2, 9], and Omni-directional vehicle [3, 28]. The study of such shortest-paths can further extend to 2D generic rigid bodies, but the extension to 3D has met many challenges.

There have been well-known theorems that give constraints for shortest-paths. Many of the solutions derived above were built upon the Pontryagin's Maximum Principle (PMP) [21]. At the same time, the Hamilton-Jacobi-Bell (HJB) equations also give strong conditions for optimality, which is less suited for analysis compared to PMP, but often is the basis for numerical solutions for optimal-control problems. PMP gave strong necessary conditions for optimality, introducing an adjoint function along the trajectory. To find the adjoint function, however, requires the integration of complex functions, which may not always be feasible.

In this work, we study the optimization problem using the Lagrange multipliers, and show that the Lagrange multipliers are similar to the adjoint functions introduced in PMP. The reason we attempt to use the Lagrange multipliers to derive the necessary conditions is that this process involves mostly derivation, versus integration when deriving the conditions using PMP. In 2D, we prove that the necessary conditions derived from Lagrange multipliers and PMP are in exactly the same form for full configuration constraints. What is more, in 2D, if we change the representation of the parameters from configuration (x, y, θ) to two fixed points in the robot frame, the derived necessary conditions hold. In 3D, however, no matter what parameterization we choose, the adjoint function from PMP cannot be easily integrated. We therefore derive the necessary conditions in 3D using three fixed points in the robot frame with Lagrange multipliers. Even though the conditions introduced by the Lagrange multipliers are weaker than that derived from PMP, combined with geometric reasoning, the conditions from both derivations can be mostly equivalent .

Based on the derived necessary conditions, we show the resolution complete algorithm in [30] can be extended to find shortest-paths reaching arbitrary *configurations* for generic rigid bodies. Specifically, we demonstrate the correctness of the algorithm with a 3D Dubins model and show that the shortest-path of a 3D rigid body can be found using a three-dimensional search when the control set is discrete. Such a 3D Dubins model may resemble a submarine more than an airplane.

In Figure 1, we show the comparison of the paths found with full configuration constraint (left panel, this work), and with just position constraint (right panel, previous work [30]). By approximately reaching the goal, we mean the error between the actual end-configuration and the goal is small, with respect to the given search resolution. In the figure, the yellow axes are the world frame, the blue and green frames are the target and actual reached goal configurations. The arrows of other colors represent different rotation axes along the path. On the right panel, though the path is very short and the goal position is reached, the final orientation is quite different from what is

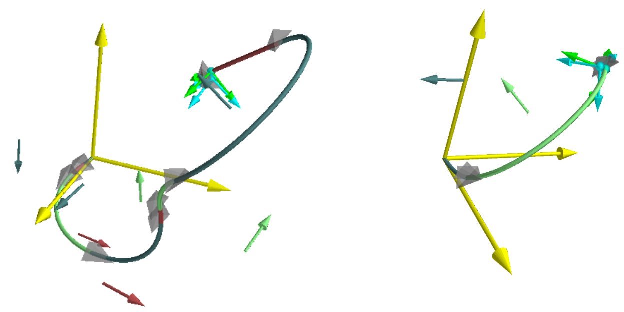

Figure 1. Sample extremal paths, (approximately) reaching the same goal (light blue). On the left panel, the path reaches the goal in full configuration; on the right panel, the path reaches goal in position only.

desired, with x-axis of the robot frame aligned with the z axis of the desired orientation.

The presented work also has a few weaknesses. First, we skip the proof of the existence of the shortest-paths for generic 3D rigid bodies. Specifically, we do not rule out the possibility of chatter. We can go around the issue by adding switch costs to limit the number of controls used in optimal sequence, though practically, our experiments with many random configurations did not find chatter behavior. Second, we assume the control set is discrete. Such assumption is valid for many known systems where the optimal control set is proven to be the (finite) vertices of the control region in the control space. However, the assumption has not been proven to be sufficient for 3D generic rigid bodies. The main contribution of this work is the inclusion of orientations in the analysis and the algorithm, and the equivalence of derivations from using PMP and Lagrange multipliers. However, the resulting algorithm is derivative from that presented in [30].

A. Related work

Most of the existing work on finding optimal trajectories is built either on Hamilton-Jacobi-Bell (HJB) equations [5] or Pontryagin's Maximum Principle (PMP) [21]. The HJB equations are often used to find numerical solutions to complex systems directly as it provides sufficient conditions for optimality. PMP, on the other hand, provides a strong necessary condition and even local structures of control functions, which can lead to even analytical solutions for some systems.

Our work follows directly from the long line of investigation of the shortest geodesics problem, started back in 1957 by Dubins [13]. These shortest geodesics describe the control strategy of an airplane cruising at a fixed altitude, or a planar with a fixed turning radius. The work was further extended by Reeds and Shepp to allow backward motion [22], and then further extended in [7, 26]. Control synthesis for this model was later presented by Soueres and Lamound [23]. Studies on robotics models such as Differential Drives [2, 9] and Omni-directional vehicles [3, 28] were derived using PMP. The work was further extended to generic planar rigid bodies with arbitrary translations and rotations [17, 16, 15, 29].

Although these kinematic models can lead to nice theoretical results, they may not model the reality very well. Dynamic effects and bounded accelerations can dramatically change the control strategy. It was proved that no analytical solutions of time-optimal trajectories exist for bounded-acceleration vehicles [24, 25]. Different variations have been introduced in the past, including underwater vehicles [10] and vehicles under constant velocity fields [12].

When the acceleration is not bounded, like in the kinematic models we study in this work, there is a chance that chattering behavior may appear. In some systems, though, it can be proved that there always exists non-chattering trajectories which are equivalent [14], but the result is not general. By applying Blatt's Indifference Principle (BIP) [6], Lyu showed that some cost associated with switches can be introduced to simulate bounded accelerations [20, 19]. Those results only apply when no obstacle is present. Though it is possible to integrate obstacles into the conditions presented by PMP, the integration would make the already complex problem even more challenging. There exists work on measuring distances between a car-like robot with obstacles [27, 18] and on planning among simple obstacles for car-like systems [1, 11].

In 3D, the time-optimal trajectories are also more challenging to find, even for seemingly simple systems. A simplified example of Dubins car with altitude control was presented by Chitsaz and Lavalle [8], and the optimal control problem for Dubins car was also studied in [31]. In recent work, we have found an approach to find time-optimal trajectories for generic 3D rigid-bodies, but only for reaching goal positions rather than full orientations [30]. In this work, we extend the result to 3D configurations and show a simplified search in 3D can still find time-optimal trajectories when controls are given from a discrete set.

II. Necessary conditions for optimality

In recent work [4], the authors showed that the models of mobile vehicles are similar to that of arm-like systems, but the analysis and comparison were primarily in 2D. One of the weaknesses of the previous work in [4] and [30] is that the Lagrange equations based derivations did not include orientations. Whether the derived necessary conditions are equivalent to that derived from PMP when the orientation is included was not shown.

In this section, we first show that in 2D, the *necessary conditions derived using Lagrange equations and PMP for full configuration constraints are equivalent*. We further show that when we *change the parameterization in 2D from the configuration ((x, y, θ)) to two points with fixed distance in the robot frame, the derived conditions are still in the same form*.

We extend the notations to 3D and show the derived necessary conditions in 3D that include the full configuration. We will use the derived conditions to show that the algorithm from [30] can be updated with minimum changes to find the shortest path between two arbitrary configurations.

We briefly restate the results from [4], which is the basis of the following analysis. The configuration of a mobile robot after the sequence of *controls* can be computed using forward kinematics, just like how we compute the configuration of the end effector of a robot arm. By control, we mean a constant velocity motion, either a translation or rotation for simplicity. Such simplification of controls enables the analysis of paths with geometrical tools, leading to shortest geodesics, or referred by some literature, time-optimal trajectories. Under the assumption of constant velocity controls, for a given a goal g and sequence of n controls with duration $t_1, t_2, \ldots, t_n$, we can form the following constrained optimization problem to find the shortest-paths,

$$\min f(t) = \sum_{i=1}^{n} t_i \tag{1}$$

$$s.t. \quad T_f = g \tag{2}$$

where T_f is the result of multiplication of sequence of transformation matrices in the given order. Readers can refer to [4] for more details on T_f and how it is derived. Let us denote $h(t) = T_f - g$. With the introduction of the Lagrange multipliers, we would like to find a vector $\mathbf{t}$ such that

$$\boldsymbol{\nabla}_t f(t) = \lambda \boldsymbol{\nabla}_t h(t), \tag{3}$$

at points where $h(t) = 0$.

In this problem, $\boldsymbol{\nabla}_t f(t)$ is simply an n-vector of ones. Writing the Lagrange condition out in matrix form, each constraint is a column in the matrix, and the matrix is in the same form as an analytical Jacobian for a serial arm. For Cartesian coordinates, the analytical Jacobian is identical to its geometric Jacobian. For rotational control, the geometric Jacobian is computed as a cross-product between the rotation axis and the vector pointing from the rotation center to g. For translation, the geometric Jacobian is just the linear velocity vector.

The necessary condition from [4] derived using the cross-product rule with position constraints (Eq. (6)) is almost identical to the condition derived using PMP [14], which is in the form of

$$k_1\dot{x} + k_2\dot{y} + \omega(k_1 y - k_2 x + k_3) = H \tag{4}$$

where k_i and H are constants, $\sqrt{k_1^2 + k_2^2} = 1$, $\dot{x}$ and $\dot{y}$ are world-frame velocities for the robot, and ω is the angular velocity. The missing term is associated with orientation, which was not considered in previous work. We show below that when we do consider the orientation, Eq. (4) can be derived using the Lagrange multipliers.

A. Equivalent necessary conditions in 2D

Let the goal be $g = (x_g, y_g, \theta_g)$, and the configuration of the robot be $q(t) = (x(t), y(t), \theta(t))$. Let the rotation center r_c be (u_x^r, u_y^r) in the robot frame.

In [4], the necessary condition derived from the Lagrange multipliers are as follows. There exist constants k_1, k_2, and H such that for any segment number $i \in \{1, 2, \ldots, n\}$,

$$k_1(-\omega_i(y - r_{iy}) + v_i p_{ix}) + k_2(\omega_i(x - r_{ix}) + v_i p_{iy}) = H, \tag{5}$$

where ω_i is the angular velocity and v_i the linear velocity of control i; and p_{ix} and p_{iy} are the projections of orientation of the robot frame to the world x and y frame. Note that $(-\omega_i(y - r_{iy}) + v_i p_{ix})$ gives the x velocity of a point on the robot due to control i, and $(\omega_i(x - r_{ix}) + v_i p_{iy})$ gives the y-velocity. We therefore have

$$k_1\dot{x} + k_2\dot{y} = H. \tag{6}$$

This condition is identical to the condition derived from PMP without the k_3 term [14]. In the above result, the rotation centers r_i are represented in the world frame.

Let us further assume that the consecutive controls can be the same, i.e. any time t can be a potential switch in the trajectory. At first glance, this seems to increase the complexity of the trajectory structure. However, the analysis will show that the results of derivation are the same. What is more, the previous analysis in [4] though is simple, only applies at the time of switch rather than any time along the trajectory. By considering the control at any time as a potential switch, we can extend the necessary condition for the entire trajectory, which is the condition derived from PMP.

Given a shortest-path starting from the origin, we know that at any time t, the segment between $q(t)$ and g must be optimal. Let us first consider the case where the control at time t is a rotation. We therefore have,

$$r_c^w(t) = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix} \cdot \begin{bmatrix} u_x^r \\ u_y^r \end{bmatrix} \quad + \quad \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

In the world frame, let the trajectory before time t and after time t are of fixed structure, so that if the rotation at time t were to last $t + \Delta t$, the location of the endpoint would move. Therefore, the derivative of this control's contribution to the movement of the endpoint with respect to t should maintain a constant dot product with Lagrange multipliers ($\vec{\lambda}$), which is consistent with the use of geometric Jacobian shown in [4]. The derivative can be computed as the cross product between the rotation axis with the vector pointing from the rotation axis to the goal/endpoint. We, therefore, have in the world frame,

$$\omega([0, 0, 1] \times (g - r_c^w(t))) \cdot \lambda \quad = \quad 1 \tag{7}$$

At the same time, the orientation of the endpoint should be equal to the summation of the rotation control duration. The contribution of the angular velocity to the final orientation with respect to t should be equal to the angular velocity. Let the angular velocity at time t be ω, we therefore can extend Eq. (7) as,

$$\begin{bmatrix} \omega(\sin(\theta(t))u_x^r + \cos(\theta(t))u_y^r - y_g + y(t)) \\ \omega(x_g - \cos(\theta(t))u_x^r + \sin(\theta(t))u_y^r - x(t)) \\ \dot{\theta} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = 1$$

$$\Rightarrow \begin{aligned} &\omega(\sin(\theta(t))u_x^r + \cos(\theta(t))u_y^r - y_g + y(t))\lambda_1 + \\ &\omega(x_g - \cos(\theta(t))u_x^r + \sin(\theta(t))u_y^r - x(t))\lambda_2 + \\ &\omega \cdot \lambda_3 = 1 \end{aligned} \tag{8}$$

During the same rotational control, the rotation center does not move. Therefore, if we apply the same rotational control for Δt longer, we have $\theta(t+\Delta t) = \theta(t) + \omega\Delta t$ and

$$\begin{aligned}
& \sin(\theta(t+\Delta t))u_x^r + \cos(\theta(t+\Delta t))u_y^r + y(t+\Delta t) \\
= \ & \sin(\theta(t)) \cdot u_x^r + \cos(\theta(t)) \cdot u_y^r + y(t) \qquad (9) \\
\Rightarrow \ & \frac{y(t+\Delta t) - y(t)}{\Delta t} \\
= \ & -u_x^r \frac{\sin(\theta(t)+\omega\Delta t) - \sin\theta(t)}{\Delta t} \\
- \ & u_y^r \frac{\cos(\theta(t)+\omega\Delta t) - \cos\theta(t)}{\Delta t} \qquad (10) \\
\Rightarrow \ & \dot{y} = \omega(u_y^r \sin(\theta(t)) - u_x^r \cos(\theta(t))) \qquad (11)
\end{aligned}$$

Similarly, we can derive $\dot{x} = \omega(\sin(\theta_t) \cdot u_x^r + \cos(\theta_t) \cdot u_y^r)$. Define a constant $c = \lambda_3 + \lambda_2 x_g - \lambda_1 y_g$, as x_g, y_g, λ_1, λ_2, and λ_3 are all constants throughout the trajectory. Rewrite Eq. (8) as,

$$\dot{x}\lambda_1 + \dot{y}\lambda_2 + \omega(\lambda_1 y(t) - \lambda_2 x(t) + c) = 1$$

Scale λ_1 to k_1 and λ_2 to k_2 so that $\sqrt{k_1^2 + k_2^2} = 1$, and scale c accordingly to k_3, we have

$$k_1\dot{x} + k_2\dot{y} + \omega(k_1 y - k_2 x + k_3) = H, \qquad (12)$$

which is exactly the same equation derived from Pontryagin's Maximum Principle (PMP) for the 2D cases [14]. This condition holds for any moment along the trajectory.

The necessary condition for the time-optimal trajectory derived using Lagrange multipliers is in the exact same form as the Hamiltonian derived from PMP. However, the condition derived using Lagrange multipliers lacks the maximization condition for the Hamiltonian.

On the 2D plane, we know that two points uniquely determine the configuration of a rigid body. Therefore, finding the time-optimal trajectory for a rigid body should be equivalent to finding a coordinated time-optimal trajectory for two points of fixed distance on the same rigid body. We can derive the same necessary condition using the parameterization with two points, where the two points are being described by their Cartesian coordinates. The parameterization would allow the use of Geometric Jacobian to replace the Analytical Jacobian in the derivatives of the constraints.

Let us describe the system using the position of two fixed points in the robot frame: $q = (x_o, y_o, x_1, y_1)$, where $p_o = (x_o, y_o)$ and $p_1 = (x_1, y_1)$ are the world-frame coordinates of $(0, 0)$ and $(1, 0)$ in the robot frame, respectively. Let g and g_1 be the goal for p_o and p_1. Consider a rotational control i, we have

$$\begin{pmatrix} k_1 \\ k_2 \\ 0 \end{pmatrix} \cdot (\hat{w}_i \times \vec{r_i g}) + \begin{pmatrix} k_3 \\ k_4 \\ 0 \end{pmatrix} \cdot (\hat{w}_i \times \vec{r_i g_1}) = 1 \ . \qquad (13)$$

The distance between the two points is fixed along the trajectory. Now, to compare Eq (13) to the equation derived from PMP, we need the rotational velocity, denoted as $\omega = \dot{\theta}$. For simplicity, let the rotation axis always points along the positive z axis, making ω a signed value.

We can rewrite $\hat{w}_i \times \vec{r_i g}$ as $\hat{w}_i \times (\vec{r_i p_o} + \vec{p_o g})$, so that the first part of the cross product becomes the velocity $\dot{p_o} = (\dot{x}_o, \dot{y}_o)$, and the second part becomes the $\hat{w}_i \times g - \hat{w}_i \times p$. We know that $\hat{w}_i$ is fixed and points along the positive z axis in 2D problems, and g is fixed. We can rewrite the equations above as follows,

$$\begin{aligned}
& k_1\dot{x}_o + k_2\dot{y}_o + \omega \cdot \hat{w} \cdot (\begin{pmatrix} k_1 \\ k_2 \end{pmatrix} \times \vec{p} + \vec{g} \times \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} + \\
& k_3\dot{x}_o + k_4\dot{y}_o + \omega \cdot \hat{w} \cdot (\begin{pmatrix} k_3 \\ k_4 \end{pmatrix} \times \vec{p} + \vec{g_1} \times \begin{pmatrix} k_3 \\ k_4 \end{pmatrix}) = H
\end{aligned}$$

Let $\vec{k} = (k_1, k_2)^T + (k_3, k_4)^T = (k_x, k_y)^T$, we have,

$$k_x \cdot \dot{x}_o + k_y \cdot \dot{y}_o + \omega \cdot (k_x \cdot y_o - k_y \cdot x_o + c) = H \ , \qquad (14)$$

where $c = g^x k_2 - g^y k_1 + g_1^x k_4 - g_1^y k_3$, which is a constant in 2D cases. We can see that this equation is also equivalent to that derived from PMP, provided that proper scaling is performed to get k_x, k_y, c, and H.

The above derivation is also consistent with the geometric interpretation we concluded from the PMP-derived equations: the location of the control line $k_1 y - k_2 x + k_3 = 0$ is important for determining the path length. Some non-shortest-paths can share the same control line directions, i.e. $(k_1, k_2)^T$, but have different k_3 values. This derivation also shows one additional feature we did not analyze in the previous interpretation of the control line: the relations between the offset of the control line and the orientation of the control line. From Eq. (14), we can see that the offset of the control line can be computed from the orientation of the control line and the goal.

B. Extending to 3D

We have shown that using the Lagrange multipliers and using PMP can lead to the same necessary condition. We have also shown that in 2D, using the two-points parameterization and the configuration (x, y, θ) can also derive the same necessary optimality condition. In 3D, the orientation parameterization becomes much more complex, and different representations have different advantages and flaws. The angle-based parameterization can be hard to take derivatives or to integrate. We will describe the 3D system using three fixed points in the robot frame. Based on the analysis in 2D, the two parameterizations can lead to the same necessary condition.

Let us denote the configuration of a 3D rigid body as a collection of three points, $q = (p_o, p_x, p_y)$, where $p_i \in \mathbb{R}^3$ and $i = \{o, x, y\}$. For simplicity, let p_o, p_x and p_y be $(0, 0, 0)$, $(1, 0, 0)$, and $(0, 1, 0)$ from the robot frame respectively. Then, the problem of finding the optimal trajectory becomes finding the path that can lead all three points to their respected goal positions simultaneously in the shortest amount of time. As the three points are fixed in the robot frame, their distances are maintained throughout the trajectory.

Given a sequence of n controls, we need the three points to reach the goal simultaneously. Using forward kinematics, we can find matrices T_f^o, T_f^x, T_f^y, and find the shortest path using the following constrained optimization:

$$\min f(t) = \sum_{i=1}^{n} t_i \quad (15)$$

$$s.t.\ \ h_1(t) = T_f^o - g_o = 0 \quad (16)$$

$$h_x(t) = T_f^x - g_x = 0 \quad (17)$$

$$h_y(t) = T_f^y - g_y = 0 \quad (18)$$

Using the Lagrange multipliers, we have $\nabla_t f(t) = \lambda_o \nabla_t h_1(t) + \lambda_x \nabla_t h_2(t) + \lambda_y \nabla_t h_3(t)$. Denote $\lambda_o = (\lambda_1, \lambda_2, \lambda_3)$, $\lambda_x = (\lambda_4, \lambda_5, \lambda_6)$, and $\lambda_y = (\lambda_7, \lambda_8, \lambda_9)$, we can rewrite the optimal condition as

$$(\lambda_1, \lambda_2, \lambda_3)^T \cdot (\omega \cdot \hat{w}_i \times r_i\vec{g}_o) \quad (19)$$

$$+ \quad (\lambda_4, \lambda_5, \lambda_6)^T \cdot (\omega \cdot \hat{w}_i \times r_i\vec{g}_x) \quad (20)$$

$$+ \quad (\lambda_7, \lambda_8, \lambda_9)^T \cdot (\omega \cdot \hat{w}_i \times r_i\vec{g}_y) = 1\ , \quad (21)$$

when the control i is a rotation around $\hat{w}_i$ at angular velocity ω, or

$$(\lambda_1, \lambda_2, \lambda_3)^T \cdot (\dot{x}(t), \dot{y}(t), \dot{z}(t)) \quad (22)$$

$$+ \quad (\lambda_4, \lambda_5, \lambda_6)^T \cdot (\dot{x}(t), \dot{y}(t), \dot{z}(t)) \quad (23)$$

$$+ \quad (\lambda_7, \lambda_8, \lambda_9)^T \cdot (\dot{x}(t), \dot{y}(t), \dot{z}(t)) = 1\ , \quad (24)$$

when the control i is a translation with linear velocity $v = (\dot{x}(t), \dot{y}(t), \dot{z}(t))$. For simplicity, in the text below, we assume the rotations all have the same angular velocity, and let $\hat{\omega}_i = \omega\hat{w}_i$.

We can rewrite the above equations into a single necessary condition, by integrating them together. By replacing $\hat{\omega}_i \times \vec{r_i g}$ with $\hat{\omega}_i \times (r_i\vec{p}_o + \vec{g} - \vec{p_o})$, we have the first part of the cross-product equal to the linear velocity. Thus, we have

$$\vec{\lambda}(\dot{x}(t), \dot{y}(t), \dot{z}(t)) + \hat{\omega}_i(\vec{\lambda} \times \vec{p} + \vec{g_o} \times \vec{\lambda} + \vec{c^{\alpha}}) = 1 \quad (25)$$

where $\vec{\lambda} = (\lambda_1, \lambda_2, \lambda_3) + (\lambda_4, \lambda_5, \lambda_6) + (\lambda_7, \lambda_8, \lambda_9)$, and $\vec{c^{\alpha}} = g_o\vec{g}_x \times (\lambda_4, \lambda_5, \lambda_6)^T + g_o\vec{g}_y \times (\lambda_7, \lambda_8, \lambda_9)^T$. As all λ are constant Lagrange multipliers, and the goal positions are fixed, $\vec{c}$ is a constant vector for a given goal. It is easy to show that if the system state are described as $q = (p_o, d_x, d_y)$ where $d_x = p_o\vec{p}_x$ and $d_y = p_o\vec{p}_y$, the resulting condition can be reorganized into the same form. Let $\vec{\lambda} = \alpha \cdot \vec{k}$, and $\vec{c^{\alpha}} = \alpha \cdot \vec{c}$, we then have

$$\vec{k} \cdot (\dot{x}(t), \dot{y}(t), \dot{z}(t)) + \hat{\omega}_i \cdot (\vec{p_o g} \times \vec{k} + \vec{c}) = H \quad (26)$$

$$\vec{k} \cdot (\hat{\omega}_i \times \vec{r_i g}) + \hat{\omega}_i \cdot \vec{c} = H \quad (27)$$

Eq. (25) through Eq. (27) are equivalent. There are two parts in Eq. (27), one is the dot product between $\vec{k}$ and control moment, which is $\hat{\omega}_i \times \vec{r_i g}$ for rotation and $(\dot{x}, \dot{y}, \dot{z})^T$ for translation. This control moment is the geometric Jacobian for Cartesian components with respect to rotation and translation. There is another term in Eq (27): the dot product between the rotation axis $\hat{\omega}_i$ and a constant vector $\vec{c}$. Note that the geometric Jacobian for an orientation is the corresponding rotation axis in world frame.

The time-optimal necessary condition can be interpreted as the two components of the geometric Jacobian dot product with two constant vectors, the sum of which need to remain a constant for all control segments. This observation is interesting when studying robot arms of more than six degrees of freedom, and may help to solve inverse kinematics faster for complex systems.

The derived necessary condition (Eq. (26)) is an extension of Eq. (4) in 2D. The dot product between the Lagrange multipliers and the velocity of the system appears in both equations, and the remaining term is governed by rotational components. In 2D, all rotation axes are parallel to z-axis, simplifying the Eq. (27) into $(\lambda_o + \lambda_x) \cdot (\dot{x_o}, \dot{y_o})^T + \omega(g \times \lambda_o + g_x \times \lambda_x - p_o \times (\lambda_o + \lambda_x)) = 1$, which can be written as $\lambda \cdot (\dot{x_o}, \dot{y_o})^T + \omega(\lambda \times p_o + c)$ where $\lambda = \lambda_o + \lambda_x$. This is the same condition derived from PMP in our previous work [29]. However, in 3D, the rotational axes no longer point along the same direction. Similar simplification of the necessary condition cannot be performed.

Let us look at the same problem from the PMP's perspective. First, we need to represent the Hamiltonian $H(t)$, which is the dot product between the adjoint function $\lambda(t)$ and the velocity of the system $\dot{q}(q, u)$. The velocity depends on the configuration and the control for the robot. Using the three points parameterization, we have $\dot{q} = (\dot{p_o}, \dot{p_x}, \dot{p_y})$, which is the transformation of the control in the robot frame to the world frame. In the robot frame, we have $u(t) = (\dot{p_o^R}, \dot{p_x^R}, \dot{p_y^R})$, the velocities of the three corresponding points in the robot frame. Denote the transformation matrix of $u(t)$ to $\dot{q}(t)$ as $\mathcal{R}$, which is a tridiagonal matrix of dimension nine by nine. The tridiagonal elements are the displacement of the same three-by-three matrix R, which transforms a vector in the robot frame at time t to the world frame. The matrix R can be written as,

$$R = \begin{bmatrix} d_x^T & d_y^T & d_z^T \end{bmatrix}\ , \quad (28)$$

where $d_z = d_x \times d_y$. Now, if we take derivatives of $H(t)$ with respect to $q = (p_o, p_x, p_y)$, we will have non-zero matrix for all $\frac{\partial H}{\partial q_i}$, as $d_x = p_o\vec{p}_x$ and $d_y = p_o\vec{p}_y$. It is possible that we can use another parameterization for the configuration, $q^{\Delta} = (p_o, d_x, d_y)$. Then, we have the derivatives of the adjoint function as follows:

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial q} \quad (29)$$

$$\Rightarrow \dot{\lambda} = -\frac{\partial}{\partial q}\langle \lambda, \dot{q}(q, u)\rangle \quad (30)$$

The first three terms are 0 because the matrix $\mathcal{R}$ is now independent of p_o, with elements from d_x and d_y. Therefore, we know that the first three dimensions of the adjoint function λ are constants, agreeing to the results derived from Lagrange multipliers. The remaining dimensions of the adjoint function, however, are not so simple. For example, the derivatives of R with respect to d_x^x is

$$\frac{\partial}{\partial d_x^x} R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -d_z^y \\ 0 & 0 & d_y^y \end{pmatrix} \quad (31)$$

This matrix appears in $\partial\mathcal{R}(t)/\partial q_4^{\Delta}$, which computes the derivative of the fourth element of the adjoint function (denoted as λ:

$$\begin{aligned}\dot{\lambda_4} &= \lambda_1 u(p_o^x) + \lambda_2(-d_z^y u(p_o^y)) + \lambda_3 d_y^y u(p_o^z) \\ &+ \lambda_4 u(d_x^x) + \lambda_5(-d_z^y u(d_y^x)) + \lambda_6 d_y^y u(d_z^x) \\ &+ \lambda_7 u(d_x^y) + \lambda_8(-d_z^y u(d_y^y)) + \lambda_9 d_y^y u(d_z^y)) \ , \quad (32)\end{aligned}$$

where $u(d_x^x)$ is the x component of $u(d_x)$.

This shows that the derivatives of the remaining dimensions of the adjoint functions are not straight 0s. In other words, the necessary condition derived from PMP is not obviously equivalent to that derived using the Lagrange multipliers. The equivalence only holds when Eq. (32) and other similar equations would all equal to 0 for *feasible* controls. This is a reasonable condition to satisfy. Essentially, we are requiring the intersection of the null space of a matrix function $A(t)$ to be non-empty, for all candidates of optimal controls at configurations on the optimal trajectory. We know that there are limitations of what controls at what configurations can be optimal. Therefore, what we need is to find the set of controls at given configurations that forces $\dot{\lambda}$ to be 0. What is more, the constraint is independent of position. Therefore, for any given system, the constraints for each orientation may be pre-computed. Such information can be used to study control synthesis directly. This analysis is not the main focus of this work but will be studied in future work.

The main source of complication in the derivation and analysis of the necessary condition in 3D is the varying rotational axes. In Eq. (27), for each given $\hat{\omega}$ of fixed direction, the corresponding segment of a path can be treated as part of a planar path, satisfying geometric constraints for any planar shortest-paths. The geometric constraints on a plane can be described in association with a line. For all the possible rotation axes directions, all the associated lines is most likely to be bounded within a cylinder associated with $\vec{k}$ and $\vec{c}$. However, we are not yet able to prove the existence and derive the configuration of this cylinder. If such a cylinder can be proved to exist, then based on our previous experience, the search for the cylinder can be more efficient in assist of the search for the shortest paths.

C. Simplifying the search

Given a start s and a goal g, if a trajectory reaches the goal g in full configuration, it will satisfy the condition specified in Eq. (27). In the previous work [30], if one only require the agent to reach the goal in position, we only need to satisfy the same equation without the component associated with $\vec{c}$. In other words, reaching goal position is a *necessary condition* for reaching goal with full orientation. Therefore, there would also exist a vector λ, which maybe different from $\vec{k}$ that satisfy the following equations:

$$\begin{aligned}\lambda \cdot (\dot{x}, \dot{y}, \dot{z})^T &= 1 \text{ (Translation control)} \quad (33) \\ \lambda \cdot (\hat{\omega} \times \vec{rg}) &= 1 \text{ (Rotational control)} \quad (34)\end{aligned}$$

We know that the path reaching a goal with full configuration must also be a path reaching the goal in position, satisfying all the geometric constraints. Such paths usually are not the shortest path reaching the goal in position only. On the other hand, if a path reaches the goal position satisfying the conditions in Equations 33 and 34, and happens to reach the goal in the desired orientation, this path can be a candidate of the shortest-path reaching the goal in full configuration. The following lemma shows that these paths satisfy the necessary conditions to reach the goal in full configuration.

Lemma 1: Given a path following a sequence of controls reaching goal position g_p in orientation $\hat{\Omega}$ and a corresponding vector λ satisfying necessary condition specified in Equations 33 and 34. Then there exist constants $\vec{k}$, $\vec{c}$, and H so that Eq. (27) can be satisfied using the same control sequence with $\vec{k}$, $\vec{c}$, and H to reach goal $g = (g_p, \hat{\Omega})$.

Proof: There always exists a vector $\vec{k}$ that is parallel to λ and a zero vector $\vec{c}$ that satisfies both necessary conditions at the same time. When λ^x is parallel to $g_o g_x$, and λ^y is parallel to $g_o g_y$, $\vec{c}$ is zero. This would produce non-trivial Lagrange multipliers, and still, both necessary conditions can be satisfied at the same time. Thus, the given trajectory is a candidate for the shortest-path reaching goal $g = (g_p, \hat{\Omega})$. This does not rule out other possibilities for vector $\vec{k}$ and $\vec{c}$. ■

Note that, for any path with translations that are parallel to λ or with more than three translation controls that are not co-linear, λ vector and $\vec{k}$ must be parallel, and $\vec{c}$ must be a 0 vector to maintain both necessary conditions to be satisfied at the same time. If there are more than one translation control on the same trajectory, λ and k must be on the same plane.

Following the Lemma, if we can find all trajectories reaching the goal position with the specified orientation, one of them must be the shortest path. Therefore, it is sufficient to just search for paths satisfying the necessary conditions associated with λ (3 dimensional search), and enforce the goal orientation geometrically. The direct search for $\vec{k}$ and $\vec{c}$ is a 6-dimensional search. In this work, as we assume the controls are selected from a discrete set, the last control that may satisfy the goal configuration can be computed analytically, thus simplifying the search for the shortest path.

We can show that the derived necessary condition not only maintain a constant dot product but also hints at the maximization of H. The dot products with λ vector in Equations 33 and 34 can be interpreted as the velocity projections on λ. It is straightforward for the translation component. For rotation controls, the dot product is how fast the current rotation is moving the goal relative to λ, analogous to a velocity projection. The shortest-path is the one with the largest dot-product with a constant λ, i.e. H, among all paths that can reach the goal satisfying Eq. (33) and Eq. (34). This additional geometric interpretation makes the necessary condition derived using Lagrange multipliers equivalent to the conditions derived from PMP in the previous work [29].

III. Algorithm and sample shortest-paths

To find shortest paths satisfying Eq. (33) and Eq. (34) while reaching the goal at the desired orientation, we can select the last control to be the ones that are aligned with the goal configuration. The first and last control provides two equations/constraints and we need one more constraint to finalize λ.

Extending from the work in [30], an almost identical procedure can be used to find the shortest paths reaching different goal configurations. We select the first and last control aligned with the start and goal configuration, then loop over the duration for the first control, and loop over the possible next control. With three equations, a corresponding λ can be computed, which enables the simulation of the corresponding path. If the goal control can be (approximately) reached (based on the search resolution), the path will be compared to the shortest found so far, until no more possible path can be found.

Algorithm 1: Find shortest-path

Input: g(oal) $T \leftarrow$ Find a path that can reach g;
for *All possible one segment or two segment paths that can reach* g **do**
    Update the upper bound T to the shortest-path;
$k \leftarrow 1$;
for *All possible combinations of first and last control satisfying configurations* **do**
    for *All possible duration for first* k *controls that do not exceed* T **do**
        if *The some of the first* k *controls are co-linear* **then**
            $k \leftarrow k + 1$;
            continue;
        Compute corresponding λ;
        Starting from the second control, simulate the remaining path;
        Run DFS until reach last control, or exceeds upper bound;
        if *Found path can reach last control shorter than* T **then**
            Update the upper bound T;
return the upper bound T to be the shortest-path;

The λ and specified last control is sufficient to find the corresponding k and c vector necessary for the condition derived in Eq. (27). This algorithm is resolution complete, follow directly from our previous proof in [30].

We have implemented the algorithm in Julia, and present solutions for a 3D Dubins car model. The algorithm applies to generic rigid bodies in 2D and 3D. In 3D, the Dubins model is one of the most well-studied and understood models, so we use this model to demonstrate our algorithm. Comparisons are made between the results for paths with goal orientation constraints and previously derived results for shortest paths with position-only constraints. In Figure 1, a path reaching goal configuration optimally and a path only reaching goal position optimally are shown. These two paths differ greatly in duration. The path that reaches the goal configuration takes more than 10 units, while the path that reaches the goal position costs less than 3.5 units.

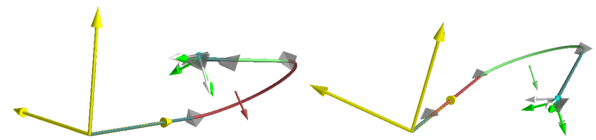

Figure 2. Sample extremal paths, (approximately) reaching the same goal drawn in light gray.

In Figure 2, two different paths reaching the same goal configuration are shown. The path structures are almost identical: translation and/or spin first, followed by a rotation, and a translation/spin is used to reach the goal. However, in the left panel of Figure 2, the first rotation is a pitch motion, while in the right panel of Figure 2, it is yaw motion. Initially, the spin motion needed to move one of the rotation axes into position is shorter if the next control is yaw, compared to pitch motion. However, the remaining orientation difference to the goal is so great that extra translation is needed before the yaw motion to admit sufficient spin motion. On the other hand, on the shortest path, despite the initial spin is longer to move the pitch rotation axis into place, the remaining orientation difference is small enough so that the spin needed is minimal after pitch. The shortest path cost 7.3 units while the other path in the same structure cost 10.7 unit.

In Figure 3, two additional time-optimal trajectories reaching different goal configurations are shown.

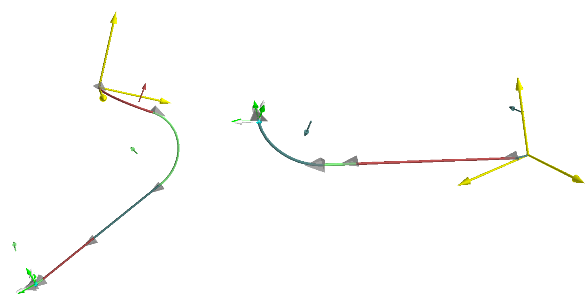

Figure 3. Sample extremal trajectories, (approximately) reaching the respective goals drawn in light gray.

When the switch of controls has no cost, the path may have many different control segments, but our experiment with a large set of random configurations did not encounter such situation. On the other hand, if there is a cost on switches, the shortest-path can be bounded much easier, and the search for different control sequence can be bounded as well. Our algorithm can find the shortest path with cost associated with control switches.

IV. Conclusions and Future Work

In this work, we present a generic approach to find the kinematic shortest path for rigid bodies, especially in 3D. We show that for a path to be time-optimal/shortest, there exists a λ vector so that necessary conditions can be satisfied. The proposed approach first finds valid λ vectors and compare the shortest path for different λs. We show that for any given goal, the proposed approach can find a path that reaches the goal within a given resolution and is approximately optimal. We have found that algebraically, the Lagrange multiplier-based approach derives a weaker condition, but integrated with geometric analysis, two can reach almost identical necessary conditions.

One future work we would like to conduct is to further analyze the PMP-derived necessary conditions in 3D, and link it with the conditions derived from the Lagrange multiplier in this work. The goal is to find alternative and possibly simpler ways to search for the shortest paths. The analysis may drive simpler synthesis analysis on the overall time-optimal control strategy for any rigid body.

REFERENCES

[1] Pankaj K. Agarwal, Jean-Claude Latombe, Rajeev Motwani, and Prabhakar Raghavan. Nonholonomic path planning for pushing a disk among obstacles. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3124–3129, Albuquerque, NM, 1997.

[2] Devin J. Balkcom and Matthew T. Mason. Time optimal trajectories for bounded velocity differential drive vehicles. *International Journal of Robotics Research*, 21(3):199–218, 2002.

[3] Devin J. Balkcom, Paritosh A. Kavathekar, and Matthew T. Mason. Time-optimal trajectories for an omni-directional vehicle. *Int. J. Robotics Res.*, 25(10):985–999, 2006.

[4] Devin J. Balkcom, Andrei A. Furtuna, and Weifu Wang. The dubins car and other arm-like mobile robots. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 380–386, Brisbane, Australia, 2018.

[5] Richard Bellman and Robert Kalaba. *Dynamic Programming and Modern Control Theory*. Academic Press, 1965.

[6] John M Blatt. Optimal control with a cost of switching control. *The Journal of the Australian Mathematical Society*, 19(3):316–332, 1976.

[7] Jean-Daniel Boissonnat, André Cérézo, and Juliette Leblond. Shortest paths of bounded curvature in the plane. *J. Intell. Robotic Syst.*, 11(1-2): 5–20, 1994.

[8] Hamid Reza Chitsaz and Steven M. LaValle. Time-optimal paths for a dubins airplane. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, pages 2379–2384, New Orleans, LA, 2007.

[9] Hamid Reza Chitsaz, Steven M. LaValle, Devin J. Balkcom, and Matthew T. Mason. Minimum wheel-rotation paths for differential-drive mobile robots. *Int. J. Robotics Res.*, 28(1):66–80, 2009.

[10] Monique Chyba and T Haberkorn. Designing efficient trajectories for underwater vehicles using geometric control theory. In *Proceedings of the 24th International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*, Halkidiki, Greece, 2005.

[11] Guy Desaulniers. On shortest paths for a car-like robot maneuvering around obstacles. *Robotics Auton. Syst.*, 17(3):139–148, 1996.

[12] Irina S. Dolinskaya and Alvaro Maggiar. Time-optimal trajectories with bounded curvature in anisotropic media. *Int. J. Robotics Res.*, 31(14): 1761–1793, 2012.

[13] Lester 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*, 79(3):497–516, 1957.

[14] Andrei A. Furtuna. *Minimum time kinematic trajectories for self-propelled rigid bodies in the unobstructed plane*. PhD thesis, Dartmouth College, 2011.

[15] Andrei A. Furtuna and Devin J. Balkcom. Generalizing dubins curves: Minimum-time sequences of body-fixed rotations and translations in the plane. *Int. J. Robotics Res.*, 29(6):703–726, 2010.

[16] Andrei A. Furtuna, Devin J. Balkcom, Hamid Reza Chitsaz, and Paritosh A. Kavathekar. Generalizing the dubins and reeds-shepp cars: Fastest paths for bounded-velocity mobile robots. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2533–2539, Pasadena, CA, 2008.

[17] Andrei A. Furtuna, Wenyu Lu, Weifu Wang, and Devin J. Balkcom. Minimum-time trajectories for kinematic mobile robots and other planar rigid bodies with finite control sets. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4321–4328, San Francisco, CA, 2011.

[18] Paolo Robuffo Giordano and Marilena Vendittelli. Shortest paths to obstacles for a polygonal dubins car. *IEEE Trans. Robotics*, 25(5):1184–1191, 2009.

[19] Yu-Han Lyu and Devin J. Balkcom. Optimal trajectories for kinematic planar rigid bodies with switching costs. *Int. J. Robotics Res.*, 35(5): 454–475, 2016.

[20] Yu-Han Lyu, Andrei A. Furtuna, Weifu Wang, and Devin J. Balkcom. The bench mover's problem: Minimum-time trajectories, with cost for switching between controls. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 106–112, Hong Kong, China, 2014.

[21] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *Mathematical Theory of Optimal Process*. John Wiley, 1962.

[22] James A. Reeds and Lawrence A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145 (2):367–393, 1990.

[23] P. Soueres and J.-P. Laumond. Shortest paths synthesis for a car-like robot. *IEEE Transactions on Automatic Control*, 41(5):672–688, 1996.

[24] Philippe Souères and J-D Boissonnat. Optimal trajectories for nonholonomic mobile robots. In *Robot motion planning and control*, pages 93–170. Springer, 1998.

[25] Héctor J. Sussmann. The markov-dubins problem with angular acceleration control. In *Proceedings of the 36th IEEE Conference on Decision and Control (CDC)*, pages 2639–2643, San Diego, CA, 1997.

[26] Héctor J. Sussmann and Guoqing Tang. Shortest paths for the reeds-shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control. Technical report, Rutgers University, 1991.

[27] Marilena Vendittelli, Jean-Paul Laumond, and Carole Nissoux. Obstacle distance for car-like robots. *IEEE Trans. Robotics Autom.*, 15(4):678–691, 1999.

[28] Weifu Wang and Devin J. Balkcom. Analytical time-optimal trajectories for an omni-directional vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4519–4524, St. Paul, MN, 2012.

[29] Weifu Wang and Devin J. Balkcom. Sampling extremal trajectories for planar rigid bodies. In *Algorithmic Foundations of Robotics X - Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 331–347, Cambridge, MA, 2012.

[30] Weifu Wang and Devin J. Balkcom. Time-optimal motion of spatial dubins systems. In *Algorithmic Foundations of Robotics XIII, Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 529–544, Mérida, Mexico, 2018.

[31] Guang Yang and V. Kapila. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In *Proceedings of the 41st IEEE Conference on Decision and Control (CDC)*, pages 1301–1306 vol.2, 2002.