The Dubins car and other arm-like mobile robots

Devin Balkcom Andrei Furtuna Weifu Wang

***Abstract*—This paper investigates the connection between the kinematics of robots arms and the shortest paths for mobile robots. Lagrange multipliers are used to show that the shortest paths are equivalent to arms in configurations that balance an external force, while applying equal torques and forces at each joint. Analysis of the arm Jacobian yields a further geometric interpretations of optimal paths, constraining the locations of rotation centers and the directions of translations that may occur along optimal paths.**

I. Introduction

The kinematics of serial robot arms are taught in university robotics courses and textbooks using standard notation. Mobile robots driven using constant controls are much like arms: controls rotate or translate the robot, creating an arm-like path that reaches different locations based on the durations of each control, as shown in Figures 1 and 2. This paper explores the connection between kinematics of robot arms and particularly, the shortest or fastest paths for simple models of mobile robots.

The primary contribution of this paper is in providing insights into the geometry of mobile robot systems, and bringing the notation in line with that for robot arms. The insights of the paper do not directly enable theorems or algorithms that could not be derived some other way; we nonetheless hope that this interpretation will be as useful for others studying efficient motion of mobile robots as it has been for us.

We will show how to write a generic parameterized representation of the kinematics of mobile robot paths, much as the Denavit-Hartenberg conventions and homogeneous transform matrices provide kinematics for robot arms.

We then turn to shortest or fastest paths for mobile robots. Although a motion planner may optimize any of several objective functions, a natural first step is to understand shortest paths for a vehicle. We study time-optimal trajectories, as a generalization of shortest paths – the shortest paths are time-optimal if each segment is followed with equal speed. We show that the time-optimal paths for the Dubins car and similar systems are entirely analogous to robot arms in geometric configurations in static balance with an external force applied to the arm, and unit forces and torques applied at the joints.

We focus on planar systems in this paper, but most of the principles are easily extensible. We show how the derived geometry applies to a 3D Dubins airplane or submarine model, a system which has been of some interest to the robotics optimal control community, but which is not yet well understood.

II. Model and assumptions

Detailed models of mobile robots involve dynamics described as differential equations that must be integrated. However, simpler kinematic models are often useful for gaining a basic understanding of robot behavior, and have been used in motion planning and design planning for systems ranging from wheeled or humanoid robots, to parts pushed by arms [24], to steerable medical needles [2].

The Dubins car, the Reeds-Shepp car, the differential drives, and the omni-directional robots for which the time-optimal trajectories are known analytically are all *kinematic* models. The Dubins car, for example, may be thought of as a particle at a location (x, y) in the plane, with a heading of θ, and a unit forwards velocity $v = 1$; the control is the angular velocity of the vehicle, which is bounded to be in some symmetric range, $\omega \in [-1, 1]$. The obvious kinematic controls for the differential drive and three-wheeled omnidirectional robots are the speeds of the wheels, which are independently bounded.

To unify the disparate models, the thesis of Furtuna [17] shows that any of these four systems can be modeled as a rigid body in the plane, with generalized velocity controls (v_x, v_y, ω), where v_x is the forwards velocity in the frame of the robot, v_y the sideways velocity, and ω the angular velocity. We have

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix}. \tag{1}$$

The trajectory of the body is then given by Lebesgue integration of the right hand side of the equation, with the controls Lebesgue-integrable functions of time.

Constraints on the controls can be mapped into this space, and in each of the four cases described, are linear. Other systems, such as rigid objects under frictional pushing manipulation by an arm, are also described by this model, with linear, but perhaps asymmetric, bounds on the generalized velocity controls.

Some motion planners, including Rapidly exploring Random Trees (RRTs) [23], require the system to be described using a small set of motion primitives. For the Dubins car, for example, one might select a left-arc of a circle, a right-arc of a circle, and a forward translation action, each of fixed duration, as the primitives. Very conveniently, the work of Dubins [16] showed that the corresponding controls are sufficient to provide optimality, if there are no obstacles.

Both the fixed duration of the primitives, and the presence of obstacles, may prevent planners from achieving optimality or even an approximation. For example, consider a space in which a single circular arc of radius greater than 1 is cut out of a rectangular obstacle. The Dubins car can follow the curve, and there is only one path, so it is optimal – but the chosen motion primitives are not sufficient. Nonetheless, the common use of motion primitives motivates the study of optimal paths for a given finite set of controls; we would at least like motion planners to do as well as possible with the primitives they are given. Further, a technique for computing the optimal paths in an obstacle-free space may serve as the *steering method* required for other planners that provide probabilistic approximate optimality (PRM* [22]), or guarantees of approximate optimality [3].

We choose a model in which the motions are a finite set of planar rotations and translations, or for 3D problems, twists. Each of these motions is generated by applying a constant control in Equation 1. The problem of finding optimal paths is then to find the correct sequence of motions, and a duration for each motion. We focus primarily on time optimality in this paper, as a generalization of the shortest paths studied by Dubins [16] and Reeds and Shepp [29].

A *twist* is a rotation around a rotation axis combined with a translation parallel to that axis; Chasle's theorem indicates that every 3D rigid body displacement is a twist [27]. A *differential twist* has an axis and a *pitch*, the ratio of translational to rotational velocity. We define a *constant-velocity arm-like system* to be a system whose kinematics are described by a sequence of constant-velocity differential twists connected by rigid links. Further, the twists must be selected from a finite set, such that each twist axis is fixed with respect to the previous link, with known pitch.

A planar 2R arm for which the joint angles θ_1 and θ_2 are driven with constant velocity of ± 1 is a simple arm-like system; the two zero-pitch twists are described by the two joint axes, which are perpendicular to the plane and move with the joints. In a 3D arm with revolute and prismatic joints, the joint axes locations with respect to the prior joint frames are described by Denavit-Hartenberg parameters [13].

The primary generalization of arm-like systems (with respect to arms) is that we permit each following joint to be selected from some finite set, rather than being of known type. For example, a shortest path for the Dubins car may be thought of as an $\{R_l, P, R_r\}\{R_l, P, R_r\}\{R_l, P, R_r\}$ arm-like system, where left-revolute, prismatic, or right-revolute joints must be selected from each of the three sets to describe the sequence of actions for the particular path. Once actions have been selected giving the *structure* of the trajectory, the trajectory is now a simple arm.

For the Dubins car, a three-link arm is sufficient to describe all shortest paths; for the Reeds-Shepp car, four links may be needed. For other vehicles, more links may be needed. The goal of finding an optimal trajectory is to both select a suitable structure, and a duration for each control.

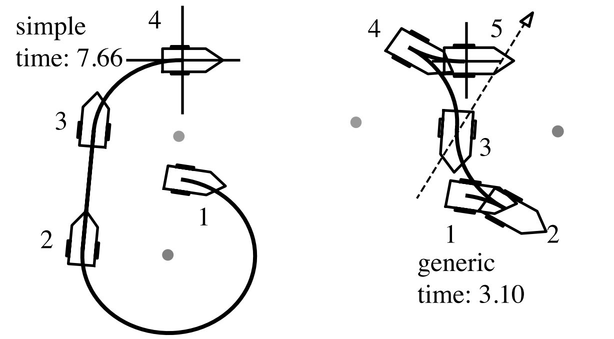

Fig. 1. A three-action and a four-action trajectory from a start (1) to a goal (4 or 5), for a Reeds-Shepp car. The 'simple' trajectory is slower than the optimal trajectory, 'generic'.

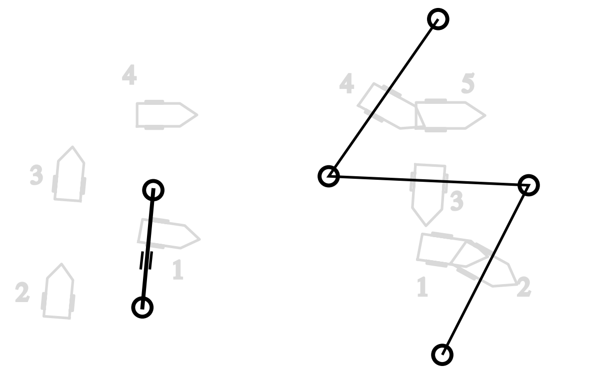

Fig. 2. Arms with kinematic structures (RPR and RRRR) corresponding to the trajectories in Figure 1.

III. Related work

Much current work on optimal motion for robots is built on either the Hamilton-Jacobi-Bellman (HJB) equations [6], or Pontryagin's Maximum Principle (PMP) [28]. The HJB equations provide sufficient conditions for optimality, suitable for numerical solution. On the other hand, PMP provides a strong necessary condition on optimality and the local structures of control functions, leading to analytical results for some systems.

In 1957, Dubins solved the problem of finding the shortest planar curve of bounded curvature that connects to given tangent vectors [16] using what he called *R-geodesics*. The control strategy following these curves corresponds to the kinematic time-optimal control strategy for a cruising airplane at a given altitude, or for a car with bounded steering radius. Later in 1990, Reeds and Shepp extended the approach to find time-optimal trajectories for a planar robotic cart that can move both forwards and backwards, and turns at a bounded angle [29]. The optimal control problem was then further invested by robotics community for application to mobile robot planning. The solution proposed by Reeds and Shepp was further generalized for the given model [8, 34], and control synthesis was later presented by Soueres and Lamound [31]. PMP was the basis for much of this work.

Other kinematic models of mobile agents have also been studied, including differential-drive [5, 11] and omni-directional vehicles [4, 36]. The PMP-based approach was further extended to derive an algorithm to find time-optimal trajectories for arbitrary planar rigid bodies with arbitrary translation and rotation controls [20, 19, 18].

Dynamic models of mobile robots may be more accurate than simple kinematic models, but analysis of optimal motion for dynamic systems appears difficult, and analytical solutions can be hard to find. In fact, it has been shown that optimal trajectories do not even exist for apparently obvious models of motion for bounded-acceleration vehicles [32, 33]. Chyba *et al.* have studied the motion of motion of simple underwater vehicles [12], and found some interesting characterizations of trajectories. Ships are affected by the wind, and optimal trajectories under a constant velocity field have been studied by Dolinskaya [15].

Recently, Lyu *et. al.* [26, 25] studied the implications of introducing a cost for control switches, to simulate the effect of acceleration, or to penalize jerky trajectories with many switches, using Blatt's Indifference Principle (BIP) [7].

Much of the existing work on optimal control focuses on scenarios where there are no obstacles, as the introduction of obstacles makes the already complex problem even more challenging. However, for simple systems, there has been study on shortest paths among obstacles, including work by Vendittelli *et. al.* on how to measure the distance between a car-like robot and obstacles [35, 21]. For simple car-like systems, planning among simple obstacles has also been studied [1, 14].

Chitsaz and Lavalle considered a simple extension of Dubin's car to allow motion in 3D [10], and modeled the airplane as a Dubin's car with altitude control. Building upon the extensive results on Dubin's car models, analytical and numerical approaches have been used to study optimal control problems in 3D [37].

IV. Forward kinematics for simple models of mobile robots

Forward kinematics relations for an arm are described using homogeneous transform matrices, parameterized using Denavit-Hartenberg conventions that provide a standard way to measure the geometry of a robot arm. In this section, we describe how transform matrices may similarly describe kinematic motion of mobile robots.

The kinematics of a 3R arm might be given in the form

$$ {}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3, \tag{2} $$

where transform matrix ${}^iT_j$ expresses the location and orientation of frame j with respect to frame i; each new link in the arm can be analyzed in isolation, and then the homogeneous transform matrices are simply multiplied together to describe the relationship between the world frame and a frame attached to the final link or end-effector [13].

The structure of each homogeneous transform matrix is determined first by whether the joint is a revolute or prismatic joint, and second by the geometric relationship between successive joints, measured by DH parameters. For an arm-like system, we would like to write a single equation for the kinematics, but the structure of the arm is not known in advance: a Dubins curve might be either an RPR arm, or 3R arm, depending on the particular path.

There are exponentially many structures for an arm-like system, with the base determined by the number of primitives available, and the exponent determined by the number of actions in the path; it is exactly this exponential tree that an RRT explores during a search. In order to avoid writing out exponentially many kinematics equations, we need to somehow unify translation and rotation actions, so that a single transform matrix can express either a translation or rotation.

A well-known trick for unifying rotational and translational rigid-body motions in the plane is to describe any rigid body motion using a rotation center. A translation is represented using a rotation about a point "at infinity" in a particular direction, using an idea from projective geometry. This aproach is used, for example, in Reuleaux's geometric method for analyzing whether an object is immobilized by point fingers, as described in [27].

We can write out a transform matrix in terms of the location of the rotation center. Selig [30] gives us the homogeneous transform matrix for planar rotation around a point $\mathbf{c}$:

$$ \begin{bmatrix} R & (I_2 - R)\mathbf{c} \\ 0 & 1 \end{bmatrix}, \tag{3} $$

where R is a $2X2$ rotation matrix and I_2 is the identity matrix.

The challenge is that as the rotation center gets further and further away, and the motion becomes more translation-like, the action becomes a shorter and shorter duration rotation around a larger and larger circle – computing the matrix in the obvious way becomes more and more numerically unstable, and the matrix for a pure translation, with the rotation center at infinity, has zero over zero terms.

However, these particular zero over zero terms are well-behaved. If we make use of the well-known cardinal sine function,

$$ \text{sinc}\ (x) = \begin{cases} \frac{\sin x}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \tag{4} $$

and by analogy, define a differentiable "cardinal versine" function,

$$ \text{verc}\ (x) = \begin{cases} \frac{1-\cos x}{x}, & x \neq 0 \\ 0, & x = 0 \end{cases}, \tag{5} $$

we may obtain the following transformation matrix that corresponds to the application of control $u = (\dot{x}, \dot{y}, \dot{\theta})$ for time t:

$$ T(u,t) = \begin{bmatrix} \cos\dot{\theta}t & -\sin\dot{\theta}t & \dot{x}t\ \text{sinc}\ \dot{\theta}t - \dot{y}t\ \text{verc}\ \dot{\theta}t \\ \sin\dot{\theta}t & \cos\dot{\theta}t & \dot{x}t\ \text{verc}\ \dot{\theta}t + \dot{y}t\ \text{sinc}\ \dot{\theta}t \\ 0 & 0 & 1 \end{bmatrix}. \tag{6} $$

A trajectory with a piecewise constant control law can be given as a sequence of (u_i, t_i) pairs, where each consecutive control $u_i = (\dot{x}_i, \dot{y}_i, \dot{\theta}_i)$ is applied for time t_i. Given such a sequence, we first assemble the $T(u_i, t_i)$ integration matrices as above. These matrices compose by post-multiplication. Thus, the final state of trajectory $[(u_1, t_1), (u_2, t_2), \ldots (u_n, t_n)]$, starting from state q_0 (equivalently specified by the robot frame to world frame transform matrix T_0) is:

$$T_f = T_0 T(u_1, t_1) T(u_2, t_2) \dots T(u_n, t_n). \quad (7)$$

The active control at time $t \le \sum_{j=0}^{n} t_j$ is the control corresponding to the largest index k such that $t \ge \sum_{j=0}^{k} t_j$. The state at time t is thus:

$$T(t) = T_0 T(u_1, t_1) T(u_2, t_2) \dots T(u_k, t_k) T(u_{k+1}, t'), \quad (8)$$

where k is the largest index such that $t' = t - \sum_{j=0}^{k} t_j \ge 0$.

Equation 6 is quite useful from a practical perspective, and makes implementing a unified simulator for Dubins and Reeds-Shepp cars, differential drives, omnidirectional vehicles, and variations, as easy as computing the forward kinematics for a planar robot arm. The inputs are the structure of the trajectory, given by indices into the control set for the particular vehicle, and durations for each control; the output is a transform matrix describing the final configuration of the vehicle. Our implementation is about 30 lines of simple Python code; the only if-statement is in the implementation of the cardinal versine, for which we used a Taylor series approximation of $\operatorname{verc}(x) \approx x/2 + x^3/24 + x^5/720$ near zero. There is an additional benefit – Equation 6 is numerically stable even for systems that include primitives that are nearly translations, such as a differential drive with slightly mismatched wheel sizes or motors.

Equation 6 is in fact quite useful not just for mobile robots, but for arms themselves, unifying the kinematic equations for all arms composed of revolute and prismatic joints. Although we focus on the planar case in this paper, extension to three dimensions appears straight-forward.

V. Lagrange multipliers and time-optimal trajectories

The modern approach to studying optimal trajectories for Dubins-like systems makes use of Pontryagin's Maximum Principle (PMP) [28, 34]. Based on the control equations, a differential equation is derived. If possible, that differential equation is integrated, yielding the *Hamiltonian* for the system. PMP indicates that if optimal trajectories exist, it is necessary that the control at each instant of time maximizes the Hamiltonian (the maximization condition). Furthermore, for time-optimal trajectories, the Hamiltonian is constant (the transversality condition) [28].

Applying this technique to rigid bodies in the plane, [18] showed that for these systems, the Hamiltonian is:

$$H = k_1 \dot{x} + k_2 \dot{y} + \dot{\theta}(k_1 y - k_2 x + k_3), \quad (9)$$

where k_1, k_2, and k_3 are constants of integration that must be selected so that the trajectory reaches a given goal, x, y and θ give the configuration of the robot at a particular time. This necessary condition is quite strong, and with sufficient geometric analysis of a particular system, permits the synthesis of optimal trajectories over the configuration space to be found.

One challenge in extending this approach to other systems, including 3D, is that for arbitrary control inputs, the motion of the vehicle will not typically be an analytically described curve such as an arc of circle or straight line. This suggests that for most systems, we will not be able to integrate the adjoint equations to find the Hamiltonian. Another challenge is that PMP is somewhat opaque in its results. Why should this particular Hamiltonian describe optimal trajectories for a rigid body in the plane, and what are the implications?

By limiting ourselves to the study of a finite set of integrable motion primitives, we may hope to avoid the difficulty of integrating the adjoint, and find some results using no more than undergraduate calculus. We will see that those results have a nice geometric interpretation; in fact, one that has already been studied in the context of robot arms.

Given a known sequence of motion primitives, the goal is to find durations $t_1, \dots t_k$ that so the vehicle reaches a given goal with minimum time. The objective function to maximize is $f(\mathbf{t}) = t_1 + t_2 + t_3 + \cdots + t_k$. There are also equality constraints $h(\mathbf{t})$ that enforce the constraint that the forward kinematics, as described by Equation 7, must carry the vehicle to the goal. Using the method of Lagrange multipliers, we want to find a vector $\mathbf{t}$ such that

$$\boldsymbol{\nabla}_t f(t) = \lambda \boldsymbol{\nabla}_t h(t), \quad (10)$$

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

Let us consider a case where the goal is to get some point on the robot to a final configuration (x_g, y_g). The functions $x(\mathbf{t})$ and $y(\mathbf{t})$ give the final location of the point after applying the primitives. For this problem, $\boldsymbol{\nabla}_t f(t)$ is simply a k-vector of ones. Writing the Langrange condition out in matrix form, each constraint gets a column in a matrix. For an example trajectory structure composed of four actions,

$$\begin{pmatrix} \partial x/\partial t_1 & \partial y/\partial t_1 \\ \partial x/\partial t_2 & \partial y/\partial t_2 \\ \partial x/\partial t_3 & \partial y/\partial t_3 \\ \partial x/\partial t_4 & \partial y/\partial t_4 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \quad (11)$$

A. Force-torque balances and the principle of virtual work

For each particular trajectory structure, we could write out the functions h_i, and then compute the partial derivatives. But the expressions for h_i are lengthy. Instead, we will now explore the geometry of Equation 11. The matrix of partials on the left-hand side is clearly a Jacobian matrix of the constraints. In fact, since the constraints express the forward kinematics of this arm-like system, the matrix is in fact the *transpose of the Jacobian for a corresponding planar arm*, which we would more typically see with partials w.r.t. joint angle variables θ_i. We have

$$J^T \lambda = \mathbf{1} \quad (12)$$

Differential kinematics for an arm are written using a Jacobian matrix. But also, the transpose of the Jacobian matrix appears in the solution for a standard manipulation problem. Specifically, the relationship between between forces at the end effector $\mathbf{f}$ and torques at the joints τ when the arm is in static equilibrium is given by

$$J^T f = \tau. \quad (13)$$

Therefore, the shape of a time-optimal trajectory for a Dubins car or any other arm-like system is equivalent to an arm in static force-torque balance with some external force f, in a configuration such that all of the torques at the joints (or forces at prismatic joints) are unit.

The static force-torque balance relationship given by 13 is typically justified using an argument based on the principle of virtual work [13], and the transposed Jacobian converts the external force into torques at the joints. To find optimal trajectories, we must find both a configuration of the 'arm' and an external force such that unit forces/torques at the joints balance the force – in some sense, each action in a Dubins curve must make an equal contribution.

The careful reader might note that the goal for a typical Dubins vehicle is to arrive at a particular orientation as well as location; for simplicity we have neglected orientation. To complete the analysis, a third column may be added to the Jacobian transpose, and a third λ value; this is equivalent to adding a revolute joint to the end-effector of the robot arm, and applying an external wrench (a force and torque) and not simply an external force. The final column of the Jacobian transpose will then be composed of constant 1s (for rotations) and 0s (for translations).

VI. The cross-product rule for the Jacobian

Textbooks typically present two ways of computing the Jacobian for a robot arm. First, one may compute the partial derivatives directly from the forward kinematics equation. Second, one may observe that for a particular configuration, columns of the Jacobian corresponding to revolute joints are given by the cross product of the corresponding joint axis with the vector from the axis to the end effector, yielding the contribution to the x, y motion of the end effector due to an infinitesimal motion of that joint.

For rotation actions, the partials in the corresponding row of J^T (Equation 11) can be computed using the cross-product rule, treating each rotation center as a revolute joint of a mechanical arm. Let (r_{xi}, r_{yi}) be the location of the ith joint, a rotation center. Then the motion of the final location of the body should be perpendicular to the axis of rotation (the z axis) and to the vector from the joint to the location; the magnitude should be proportional to the rotational velocity.

$$\begin{pmatrix} \partial x/\partial t_i \\ \partial y/\partial t_i \end{pmatrix} = \begin{pmatrix} -(y - r_{yi}) \\ x - r_{xi} \end{pmatrix} \tag{14}$$

This observation gives a very direct geometric constraint on the shape of optimal trajectories for arm-like systems. The dot product of each of these vectors with the λ-vector should be 1, and for now assume that the rotational velocities all have magnitude 1. Choose a vector (λ_1, λ_2). Then each rotation center must be an equal distance from a line perpendicular to this vector. Negative rotation centers will be on one side of the line, and positive rotation centers will be on the other side of the line.

If rotational velocities have different magnitudes, then there will be a set of parallel lines (which we might refer to as a *comb*, such that all rotation centers with the same angular velocity must fall on the same line.

A similar result holds for translation actions – the translations must make unit dot product with the λ_1, λ_2 vector.

VII. Deriving the Hamiltonian using Lagrange multipliers

The simplicity of using Lagrange multipliers to compute necessary conditions for trajectories derived using a finite set of motion primitives comes at a price. Equation 11 is not quite as elegant as the Hamiltonian equation for the planar arm-like systems (Equation 9). What is worse, although Equation 11 holds for any particular trajectory structure, it does not tell us how to choose good candidate trajectory structures, while the maximization condition on the Hamiltonian gives some constraints on how the next control must be chosen, given the previous.

However, some algebraic manipulation of the equation involving the Jacobian allows an equation in exactly the same form as the Hamiltonian, *without the need to integrate any differential equations*. This is quite exciting, as it gives some hints about cases where the adjoint equation of PMP may be integrated analytically, and suggests an alternate approach to computing the Hamiltonian, without requiring any particular analytical integration cleverness. Indeed, in future work, we hope to use this technique to derive an expression for the Hamiltonian for a 3D Dubins vehicle. Once the form of the Hamiltonian is known, proving that it is the solution to the adjoint equation may be accomplished by taking some derivatives.

Here is how we may derive the Hamiltonian from the Lagrange-multipliers equation. The vector (λ_1, λ_2) has some magnitude. Without loss of generality, replace the λ vector by some unit vector (k_1, k_2), and appropriately scale the vector of ones by some scalar H. In the following example, the arm is a RPRP arm: a rotation-translation-rotation-translation trajectory for the mobile robot.

$$\begin{pmatrix} -\omega_1(y - r_{1y}) & \omega_1(x - r_{1x}) \\ v_2 p_{2x} & v_2 p_{2y} \\ -\omega_3(y - r_{3y}) & \omega_3(x - r_{3x}) \\ v_4 p_{4x} & v_4 p_{4y} \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} = \begin{pmatrix} H \\ H \\ H \\ H \end{pmatrix} \tag{15}$$

Notice that for the first action, ω_1 is non-zero, and we may imagine a translational component v_1 with value zero. We may thus add a zero-velocity component to the first row with no effect. Similarly, we can add zero-angular velocity components to the second and fourth rows. Thus, we may write any row as:

$$\begin{pmatrix} -\omega_i(y - r_{iy}) + v_i p_{ix} & \omega_i(x - r_{ix}) + v_i p_{iy} \end{pmatrix} \tag{16}$$

We therefore have the following observation. There exist constants such k_1, k_2, and H such that for any i,

$$k_1(-\omega_i(y - r_{iy}) + v_i p_{ix}) + k_2(\omega_i(x - r_{ix}) + v_i p_{iy}) = H. \tag{17}$$

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 a y-velocity. We therefore have

$$k_1 \dot{x} + k_2 \dot{y} = H. \tag{18}$$

This is analogous to Equation 9, missing only the k_3 term. Considering final orientation of the robot as part of the analysis, with the third column of J^T as described above, yields the expected form. As mentioned, this result is weaker than that derived from PMP, in that it only expresses the transversality condition at switches, but was derived in a straightforward way without integration.

VIII. A 3D Dubins vehicle

Extending work on Dubins cars and similar systems to 3D is a problem of some interest in the robotics community. Airplanes are not particular well-modeled by kinematics models, but shortest curves composed of arcs of circles and straight lines may nonetheless be of practical interest. For example, steerable medical needles are spatial; current approaches to planning their paths forfeit optimality for simplicity and use Dubins curves only in particular planes to find a path the the goal. Results for spatial Dubins vehicles so far are exciting but not yet satisfactory, placing either very strong constraints on motion out of the plane [9], or achieving only analysis of particular cases or motions [12], in the case of acceleration-bounded models.

In this section, we provide no complete solution, but do make a few geometric observations about an arm-like model that might be considered a sort of 3D Dubins vehicle. To describe the primitives for this model, attach a local frame to the airplane, such that the local x axis points forwards, and the x-y plane is used to describe yaw motion for the vehicle. Select unit-speed forward translation and left- and right- arcs in this plane as primitives. Also select up- and down- arcs to corresponding to pitch motions. Finally, permit left and right roll motions, with zero turning radius. This gives a total of seven primitives: six rotations, and one translation. Finding approximate paths for this system is well within the capabilities of an RRT algorithm or similar.

Because the system is arm-like, we may observe that time-optimal, or in this case, shortest, paths to get a single point on the vehicle to a desired location must satisfy the equation:

$$J^T \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \dots \end{pmatrix}. \tag{19}$$

Applying the cross-product interpretation of the Jacobian, we find that the arm must be in a configuration such that there must be a λ vector (analogous to an external force) such that motion of the end effector due to each joint motion must have the same projection onto the fixed λ vector.

The set of vectors that has makes a dot product of 1 with some other vector all have endpoints that lie on a plane. Thus we see that joints must be aligned in a particular geometry such that motions due to those joints lie on a plane, a strong constraint on the locations of control switches.

IX. Conclusion

This paper presented a geometric interpretation of motion for Dubins and similar vehicles as arm-like systems. There were three main results: 1) the kinematics of these vehicles may be described simply using easily computable homogeneous transform matrices computed in such a way as to unify translation and rotation actions, 2) a Lagrange multiplier method demonstrates that the time-optimal motion for these vehicles are analogous to arm-like systems in configurations of static force-torque balance, and 3) the cross-product method for computing columns of the Jacobian for arms yields interesting geometric insights into the optimal trajectories, for both planar and spatial systems.

We also showed that the results yielded by the Lagrange-multipliers method can be manipulated into the same form as the Hamiltonian equation; this connection may be of potential practical use in seeking out spatial problems for which the Hamiltonian may be found analytically. This approach may also provide a welcoming gateway for students interested in studying optimal control problems, while using only familiar tools from calculus.

A significant weakness of this paper is that it does not at all discuss the issue of existence of optimal trajectories. The results describe properties of trajectories, and those properties hold across all trajectory structures. However, unlike the Maximum Principle, no hints as to how to choose those structures are given. How do we know optimal trajectories exist, and are of finite length? Prior work by Lyu [26, 25] address this issue in part by adding fixed costs for each switch between controls, and shows that this ensures existence with only weak requirements on the set of available controls.

This paper also did not discuss practical search techniques for optimal trajectories, choosing rather to focus on geometric insights. We hope and believe that some of these insights will prove a useful starting point for reasoning about and effective search strategies for optimal 3D trajectories.

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, Albuquerque, New Mexico, USA, April 20-25, 1997*. 1997, pp. 3124–3129.

[2] Ron Alterovitz, Michael Branicky, and Ken Goldberg. "Motion Planning Under Uncertainty for Image-guided Medical Needle Steering". In: *International Journal of Robotics Research* 27.11 (2008), pp. 1361–1374.

[3] Devin J. Balkcom, Ajay Kannan, Yu-Han Lyu, Weifu Wang, and Yinan Zhang. "Metric cells: Towards complete search for optimal trajectories". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*. 2015, pp. 4941–4948.

[4] Devin Balkcom, Paritosh A. Kavathekar, and Matthew T. Mason. “Time-optimal trajectories for an omni-directional vehicle”. In: *International Journal of Robotics Research* 25.10 (2006), pp. 985–999.

[5] Devin Balkcom and Matthew T. Mason. “Time optimal trajectories for bounded velocity differential drive vehicles”. In: *International Journal of Robotics Research* 21.3 (2002), pp. 199–218.

[6] R. Bellman and R. Kalaba. *Dynamic Programming and Modern Control Theory*. Academic Press, 1965.

[7] John M Blatt. “Optimal control with a cost of switching control”. In: *The ANZIAM Journal* 19.3 (1976), pp. 316–332.

[8] Jean-Daniel Boissonnat, André Cérézo, and Juliette Leblond. “Shortest paths of bounded curvature in the plane”. In: *Journal of Intelligent and Robotic Systems* 11.1-2 (1994), pp. 5–20.

[9] H. R. Chitsaz. “Geodesic problems for mobile robots”. PhD thesis. University of Illinois at Urbana-Champaign, 2008.

[10] H. Chitsaz and S. M. LaValle. “Time-optimal paths for a Dubins airplane”. In: *2007 46th IEEE Conference on Decision and Control*. Dec. 2007, pp. 2379–2384.

[11] Hamid Reza Chitsaz, Steven M. LaValle, Devin J. Balkcom, and Matthew T. Mason. “Minimum Wheel-Rotation Paths for Differential-Drive Mobile Robots”. In: *I. J. Robotics Res.* 28.1 (2009), pp. 66–80.

[12] M Chyba and T Haberkorn. “Designing efficient trajectories for underwater vehicles using geometric control theory”. In: *24rd International Conference on Offshore Mechanics and Artic Engineering*. 2005.

[13] John J. Craig. *Introduction to robotics - mechanics and control (3rd ed.)* Addison-Wesley, 2005.

[14] Guy Desaulniers. “On shortest paths for a car-like robot maneuvering around obstacles”. In: *Robotics and Autonomous Systems* 17.3 (1996), pp. 139–148.

[15] Irina S. Dolinskaya and Alvaro Maggiar. “Time-optimal trajectories with bounded curvature in anisotropic media”. In: *I. J. Robotics Res.* 31.14 (2012), pp. 1761–1793.

[16] L. E. Dubins. “On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents”. In: *American Journal of Mathematics* 79.3 (July 1957), pp. 497+.

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

[18] Andrei A. Furtuna and Devin Balkcom. “Generalizing Dubins curves: minimum-time sequences of body-fixed rotations and translations in the plane”. In: *International Journal of Robotics Research* 29.6 (2010), pp. 703–726.

[19] Andrei A. Furtuna, Devin Balkcom, Hamid Reza Chitsaz, and Paritosh A. Kavathekar. “Generalizing the Dubins and Reeds-Shepp cars: fastest paths for bounded-velocity mobile robots”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2008, pp. 2533–2539.

[20] Andrei A. Furtuna, Wenyu Lu, Weifu Wang, and Devin Balkcom. “Minimum-time trajectories for kinematic mobile robots and other planar rigid bodies with finite control sets”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2011, pp. 4321–4328.

[21] Paolo Robuffo Giordano and Marilena Vendittelli. “Shortest Paths to Obstacles for a Polygonal Dubins Car”. In: *IEEE Trans. Robotics* 25.5 (2009), pp. 1184–1191.

[22] Sertac Karaman and Emilio Frazzoli. “Sampling-based algorithms for optimal motion planning”. In: *I. J. Robotics Res.* 30.7 (2011), pp. 846–894.

[23] Steven M. LaValle and James J. Kuffner Jr. “Randomized Kinodynamic Planning”. In: *I. J. Robotics Res.* 20.5 (2001), pp. 378–400.

[24] Kevin M. Lynch and Matthew T. Mason. “Stable Pushing: Mechanics, Controllability, and Planning”. In: *Algorithmic Foundations of Robotics*. A. K. Peters, Boston, MA, 1995, pp. 239–262.

[25] Yu-Han Lyu and Devin Balkcom. “Optimal trajectories for planar rigid bodies with switching costs”. In: *International Journal of Robotics Research* (2015).

[26] Yu-Han Lyu, Andrei A. Furtuna, Weifu Wang, and Devin Balkcom. “The bench mover’s problem: minimum-time trajectories, with cost for switching between controls”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 106–112.

[27] Matthew T. Mason. *Mechanics of robotic manipulation*. MIT Press, 2001.

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

[29] J. A. Reeds and L. A. Shepp. “Optimal paths for a car that goes both forwards and backwards”. In: *PACIFIC JOURNAL OF MATHEMATICS* (1990).

[30] J.M. Selig. *Geometrical Methods in Robotics*. Springer-Verlag, 1996.

[31] P. Soueres and J. P. Laumond. “Shortest paths synthesis for a car-like robot”. In: *IEEE Transactions on Automatic Control* 41.5 (May 1996), pp. 672–688.

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

[33] Héctor J Sussmann. “The Markov-Dubins problem with angular acceleration control”. In: *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*. Vol. 3. IEEE. 1997, pp. 2639–2643.

[34] 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.* Tech. rep. 1991.

[35] Marilena Vendittelli, Jean-Paul Laumond, and Carole Nissoux. “Obstacle distance for car-like robots”. In: *IEEE Trans. Robotics and Automation* 15.4 (1999), pp. 678–691.

[36] Weifu Wang and Devin Balkcom. “Analytical time-optimal trajectories for an omni-directional vehicle”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2012, pp. 4519–4524.

[37] 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, 2002.* Vol. 2. Dec. 2002, 1301–1306 vol.2.