Proceedings of the American Control Conference San Francisco, California • June 1993

Algorithms for steering on the group of rotations*

Gregory Walsh, Augusto Sarti and Shankar Sastry

Electronic Research Laboratory
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720
Fax: 510-642-1341.

Abstract

The paper focuses on the problem of explicitly generating open loop strategies for steering control systems with left-invariant vector fields on the Lie group of rigid rotations $SO(3)$. Both systems with and without drift are considered as well as systems with three, two or one input(s). For each of these cases, if possible, we present a constructive solution to the steering problem.

The most interesting cases are those of systems with drift and either only or two inputs. Having two inputs gives us the freedom to choose the steering time. In the case of only one input our algorithm will drive the system to the desired orientation in a finite time. There are, however, limitations on the choice of the arrival time.

Simulations have been developed and the results animated on a Silicon Graphics Iris workstation. In particular, videotaped animations of the algorithms mentioned above will be presented at the conference.

1 Introduction and Problem Statement

Noether's theorem [1, 4] identifies conserved quantities associated with invariant actions of a Lie group on the Lagrangian of a system. The Lie group (for a good review, see [7]) associated with the conservation of angular momentum is $SO(3)$, the space of orthogonal matrices of determinant 1. The conserved quantities induce constraints on the tangent bundle of the configuration space; these constraint equations [8] can be converted to control systems. To this end, we will study left-invariant control systems on $SO(3)$. The Lie algebra $so(3)$ associated to $SO(3)$ is the set of all 3×3 skew-symmetric matrices, with the Lie bracket being the matrix commutator. The differential equation describing the evolution of g, with $g \in SO(3)$, is as follows:

$$\dot{g} = A_0(g) + \sum_{i=1}^{m} A_i(g) u_i \qquad g \in SO(3) \qquad (1)$$

where each vector field $A_i(g)$ may be written as:

$$A_i(g) = g\,(b_i \times) \qquad (b_i \times) \in so(3)$$

with the $(b_i\times)$'s constant and linearly independent members of the Lie algebra. We will often map a skew-symmetric matrix $(b\times)$ to the vector b with $b \in \mathbb{R}^3$. Thus given $b \in \mathbb{R}^3$, the skew-symmetric matrix is then:

$$(b\times) = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}$$

*Research supported in part by NSF under grant IRI90-14490

To avoid confusion between the identity element in the group and the exponential map, we will use $\text{Exp}(\cdot)$ for the latter and e for the former. The development of this paper may also be carried out for right-invariant systems.

The problem that we approach is to explicitly generate open-loop strategies for solving the steering problem, that is, given some initial point g_i with $g_i \in SO(3)$ and some final point g_f with $g_f \in SO(3)$, find a time T and a control $u(\cdot)$ piecewise continuous, defined on the interval $[0, T]$, such that the system (1), starting at the initial condition of g_i at time 0, will at time T arrive at g_f. We note that least-squares optimal control of systems of the form (1) was studied by [2]. In this paper, we focus on explict steering laws given the initial and final points in $SO(3)$.

Suppose the system had an input constraint, say, $u(t) \in U$ for all t. If the set U contained a neighborhood of the origin, we would be able to rescale any bounded solution both in time and magnitude to obtain an alternate solution which obeys the constraint U, provided there is no drift. In many cases with drift, this is possible as well.

We will consider six cases. We will set $A_0(g) \equiv 0$ in the first three and we will vary the number of input vector fields from three to one (*drift-free cases*). In the next three cases the drift term $A(g)$ will be nontrivial and the input vector fields will be varied in the same manner. In all cases we will assume the input vector fields are not redundant (meaning they are not linearly dependent).

2 Left-Invariant Control Systems

A satellite with two or three rotors at rest, that is, with zero total angular momentum, may be modeled as a drift-free system on $SO(3)$. The kinematic equations for such a system are given by (1) (for details, see [10]). The vector b_0 is zero and all other vectors depend on the physical parameters of the system.

As there is no drift to this system, Chow's theorem [3] may be applied in order to check controllability. Some simplifications in the case of left-invariant systems on Lie groups will apply. The Lie bracket reduces to the matrix commutator on the Lie algebra, and in the case of $so(3)$, $[b_i, b_j] = ((b_j \times b_i) \times)$.

2.1 The Three Input Control System

For the case of three independent inputs, the input vector fields span the tangent space at every point therefore controllability is assured. The system has the form:

$$\dot{g} = g\,(b_1\times)\,u_1 + g\,(b_2\times)\,u_2 + g\,(b_3\times)\,u_3$$

Thus given that g_i, with $g_i \in SO(3)$, is the initial state of the system, the configuration which results from the action of a combination of the constant inputs (u_1, u_2, u_3) for one second[1] is

$$g_f = g_i \,\mathrm{Exp}((b_1\times)\,u_1 + (b_2\times)\,u_2 + (b_3\times)\,u_3)$$

We can thus consider the desired net movement $g_d = g_i^{-1} g_f$ with[2] $g_d \in SO(3)$, find the controls which will steer the system from the identity to g_d and apply these to our system to obtain the movement from g_i to g_f. Thus we may solve this equation:

$$g_i^{-1} g_f = \mathrm{Exp}((b_1\times)\,u_1 + (b_2\times)\,u_2 + (b_3\times)\,u_3)$$

While the exponential map does not in general cover every group, it does for $SO(3)$ and some others [9]. Euler's theorem [7], in the case of $SO(3)$, guarantees the existence of an element $(a\times)$ of the Lie algebra $so(3)$ such that $g = \mathrm{Exp}((a\times))$, for any $g \in SO(3)$. Once the element $(a\times) \in so(3)$ is found, we only need to find numbers (u_1, u_2, u_3) such that $\sum_{i=1}^{m} u_i\,(b_i\times) = (a\times)$. As the $(b_i\times)$'s form a basis for $so(3)$, such constants are uniquely specified.

For the special case of $SO(3)$ and its Lie algebra $so(3)$ there exists a formula called Cayley's formula [5],

$$(a\times) = (g - e)(g + e)^{-1}$$

which allows us to efficiently compute the matrix logarithm by means of a simple matrix inversion when $(g + e)$ is nonsingular.

2.2 The Two Input Control System

Given that b_1 and b_2 are independent, Chow's theorem assures controllability because $b_1 \times b_2$ is perpendicular to b_1 and b_2. A more constructive argument for controllability follows from the various parameterizations of $SO(3)$. Besides the classic *roll–pitch–yaw* parameterization, there exist others like the *roll–pitch–roll* parameterization. One can think of this coordinate chart as a recipe for steering to some configuration from the identity while using only two left–invariant vector fields, $g\,(e_1\times)$ and $g\,(e_2\times)$ with e_1 and e_2 in $\mathbb{R}^3$ being the standard basis elements $(1,0,0)^T$ and $(0,1,0)^T$. Of course, in general the system will not be at the identity and have $g\,(e_1\times)$ and $g\,(e_2\times)$ as input vector fields; however, with a little work this can be put right. First, a linear transformation is needed to decouple the inputs by orthogonalizing their action. Secondly, the random disposition of these now orthonormal vector fields may be made to appear as the canonical ones with the appropriate conjugate transformation. In this way the critical formula, that is the *roll–pitch–roll* inversion, must be computed only once for any systems in this class.

Proposition 1
Given a control system on SO(3) whose evolution is described by $\dot g = g\,(b_1\times)\,u_1 + g\,(b_2\times)\,u_2$, *with* b_1 *and* b_2 *linearly independent,* g_i *and* g_f *both in* $SO(3)$ *and a time* $T > 0$,
Then there exists a $u(\cdot)$ *defined on* $[0, T]$, *piecewise constant, which will steer the system from* g_i *to* g_f *in the interval* $[0, T]$.

Proof: The proof will be given in algorithmic form.

[1]This is the same as applying the constant inputs $(\frac{u_1}{T}, \frac{u_2}{T}, \frac{u_3}{T})$ for T seconds. This consideration is valid also in what follows, except where specified.

[2]Recall that g^{-1} is the transpose of g when g is in $SO(3)$

Step 1: Decoupling the inputs We assume the *roll* motion to correspond to the action of the first input, and the *pitch* motion to be a linear combination of the two inputs[3]. If we call v_1 the *roll*, and v_2 the *pitch*, the input transform is:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} \\ 0 & \beta_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad (2)$$

with

$$\begin{aligned} \beta_{11} &= (\|b_1\|)^{-1} \\ \beta_{22} &= \left(\|b_2 - b_2^T b_1 \beta_{11}^2 b_1\|\right)^{-1} \\ \beta_{12} &= -b_2^T b_1 \beta_{11}^2 \beta_{22} \end{aligned}$$

If a_1, a_2, a_3 represent how long the inputs v_1, v_2 are applied in *roll-pitch-roll* fashion, the equation to solve becomes:

$$g_i^{-1} g_f = \mathrm{Exp}(\beta_{11}\,(b_1\times)\,a_1)\,\mathrm{Exp}((\beta_{12}\,(b_1\times) + \beta_{22}\,(b_2\times))a_2)\,\mathrm{Exp}(\beta_{11}\,(b_1\times)\,a_3) \qquad (3)$$

Step 2: Conjugate transformation Compute the rotation matrix $K \in SO(3)$, given by:

$$K = \left[\; \beta_{11} b_1 \quad (\beta_{12} b_1 + \beta_{22} b_2) \quad (\beta_{11} b_1 \times (\beta_{12} b_1 + \beta_{22} b_2)) \;\right]$$

Notice that $K^{-1}\beta_{11} b_1 = e_1$ and $K^{-1}(\beta_{12} b_1 + \beta_{22} b_2) = e_2$. Define $\bar g(t) = (g_i K)^{-1} g(t) K$. A quick calculation of the time derivative of this similarity transform will confirm the canonical representation.

$$\begin{aligned} \dot{\bar g}(t) &= (g_i K)^{-1} \dot g(t) K \\ &= K^{-1} g_i^{-1} g(t) K K^{-1} (\beta_{11}\,(b_1\times)\,v_1 + \beta_{12}\,(b_1\times) + \beta_{22}\,(b_2\times))\,K \\ &= \bar g(t)\left(K^{-1}\beta_{11}\,(b_1\times)\,K + K^{-1}\,(\beta_{12}\,(b_1\times) + \beta_{22}\,(b_2\times))\,K\right) \\ &= \bar g(t)\,((e_1\times)\,v_1 + (e_2\times)\,v_2) \end{aligned}$$

One useful fact used above is that $K\,(b\times)\,K^{-1} = (Kb\times)$.

Step 3: Computation Solve the roll-pitch-roll equation

$$(g_i K)^{-1}\, g_f K = \mathrm{Exp}((e_1\times)\,a_1)\,\mathrm{Exp}((e_2\times)\,a_2)\,\mathrm{Exp}((e_1\times)\,a_3) \qquad (4)$$

for the three coordinates (a_1, a_2, a_3).

As we will rely on the *roll-pitch-roll* inversion several times during this paper, we will compute explicitly the right hand side of equation (4) and solve it. The generic matrix g, with $g \in SO(3)$, is then:

$$\begin{bmatrix} \cos a_2 & \sin a_2 \sin a_3 & \sin a_2 \cos a_3 \\ \sin a_1 \sin a_2 & \cos a_1 \cos a_3 - \sin a_1 \cos a_2 \sin a_3 & -\cos a_1 \sin a_3 - \sin a_1 \cos a_2 \cos a_3 \\ -\cos a_1 \sin a_2 & \sin a_1 \cos a_3 + \cos a_1 \cos a_2 \sin a_3 & -\sin a_1 \sin a_3 + \cos a_1 \cos a_2 \cos a_3 \end{bmatrix}$$

Denoting the elements of g by g_{ij}, we see immediately that g_{21} and g_{31} are both zero only when $\sin a_2 = 0$, in which case g_{12} and g_{13} are zero as well and the matrix g has the following structure

$$g = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \cos(a_1 \pm a_3) & \mp \sin(a_1 \pm a_3) \\ 0 & \sin(a_1 \pm a_3) & \pm\cos(a_1 \pm a_3) \end{bmatrix}$$

[3]We have not made any assumption of orthogonality of b_1 and b_2.

Thus in this case $a_2 = 0$ or $a_2 = \pi$ means that there was either no pitch or alternatively a flip. In both cases, a single roll action is sufficient to steer the system: just solve for the quantity $a_1 \pm a_3$.

More generally, when g_{21} or g_{31} are not both zero (thus also g_{12} or g_{13} are not both zero), we can directly compute the coordinates (a_1, a_2, a_3) as follows

$$\begin{array}{rcll} a_1 & = & \text{atan2}(g_{21}, -g_{31}) & \text{if } g_{31} \neq 0 \\ & = & \text{acot2}(-g_{31}, g_{21}) & \text{else} \\ a_2 & = & \text{atan2}(g_{11}\sin(a_1), g_{21}) & \text{if } g_{21} \neq 0 \\ & = & \text{atan2}(g_{11}\cos(a_1), -g_{31}) & \text{else} \\ a_3 & = & \text{atan2}(g_{12}, g_{13}) & \text{if } g_{13} \neq 0 \\ & = & \text{acot2}(g_{13}, g_{12}) & \text{else} \end{array}$$

where $\text{atan2}(y,x), \text{acot2}(y,x)$ compute $\tan^{-1}(\frac{y}{x}), \cot^{-1}(\frac{y}{x})$ but use the sign of both x and y to determine the quadrant in which the resulting angle lies.

Step 4: Application Apply for $\frac{T}{3}$ seconds the controls: $(u_1, u_2) = (3\beta_{11}\frac{a_1}{T}, 0)$, $(u_1, u_2) = (3\beta_{12}\frac{a_2}{T}, 3\beta_{22}\frac{a_2}{T})$, $(u_1, u_2) = (3\beta_{11}\frac{a_3}{T}, 0)$

2.3 The One Input Control System

This case corresponds to a satellite with only one rotor. In this case the system is not controllable for there is only one input vector field. The set of all achievable orientations forms a 1-dimensional subgroup of $SO(3)$, S^1. Thus provided the matrix g_d with $g_d = g_i^{-1} g_f \in SO(3)$ lies in this subgroup, we may steer from the initial to final point.

3 Left–Invariant Control Systems on SO(3) with Drift

We consider systems with left invariant drift vector fields; it should be noted that in general satellites with rotors and non-zero angular momentum do **not** have this kind of structure on their drift vector fields. We will try to reduce the problem with drift to one whose structure is similar to those with no drift and apply the results of §2.

3.1 The Case of Three Inputs

When m is equal to 3 we may cancel the drift by means of a combined action of the three inputs. This allows us to apply the steering procedure seen for the corresponding drift–free case.

Proposition 2
Given a control system on SO(3) whose evolution is given by $\dot{g} = g(b_0\times) + g(b_1\times)u_1 + g(b_2\times)u_2 + g(b_3\times)u_3$, *with* b_1, b_2 *and* b_3 *linearly independent,* g_i *and* g_f *both in* $SO(3)$ *and a time* $T > 0$,
Then there exists a $u(\cdot)$ *defined on* $[0, T]$, *piecewise constant, which will steer the system from* g_i *to* g_f *in the interval* $[0, T]$.

Proof: The proof again will be algorithmic. The steering strategy for the system can be outlined as follows:

Step 1: Drift Cancellation Choose the numbers (v_1, v_2, v_3) such that $b_1 v_1 + b_2 v_2 + b_3 v_3 = -b_0$.

Step 2: Computation Find the numbers (u_1, u_2, u_3) that solve the steering problem for the driftless system

$$\dot{g} = g(b_1\times)u_1 + g(b_2\times)u_2 + g(b_3\times)u_3$$

by following the procedure seen in § 2.1.

Step 3: Application Apply for T seconds the inputs $(\frac{u_1}{T} + v_1, \frac{u_2}{T} + v_2, \frac{u_3}{T} + v_3)$ to the system with drift.

3.2 The Two Input Control System with Drift

Let us consider the system

$$\dot{g} = g(b_0\times) + g(b_1\times)u_1 + g(b_2\times)u_2 \tag{5}$$

The problem that we want to solve is to find a pair of inputs (u_1, u_2) that steers the system (5) from a given g_i, with $g_i \in SO(3)$, to a desired $g_f \in SO(3)$ in T seconds. The steering strategy that we present in this section is based on the fact that there exists a time–varying input transformation that allows us to ignore the presence of drift.

Proposition 3
Given a control system on SO(3) whose evolution is described by $\dot{g} = g(b_0\times) + g(b_1\times)u_1 + g(b_2\times)u_2$, *with* b_1 *and* b_2 *linearly independent,* g_i *and* g_f *both in* $SO(3)$ *and a time* $T > 0$,
Then there exists a $u(\cdot)$ *defined on* $[0, T]$, *which will steer the system from* g_i *to* g_f *in the interval* $[0, T]$.

Proof: We will once again give the proof in algorithmic form.

Step 1: Affine Input Transformations Without loss of generality, we will assume b_0 to be orthogonal to both b_1 and b_2. In fact, if b_0 had a non zero component on the space spanned by b_1 and b_2, it would be easy to cancel for it by means of a linear combination of constant inputs. It is not restrictive, in addition, to assume that b_1 and b_2 are orthogonal and have unit norm, as this can also be achieved by means of a linear input transformation of the inputs u_1 and u_2 as seen in the two input drift–free case. Finally, denoting $||b_0||$ by ω, we assume that the orientation of the following vectors b_0, b_1 and b_2 is such that $b_0 \times b_1 = \omega b_2$, $\omega b_1 \times b_2 = b_0$ and $b_2 \times b_0 = \omega b_1$. This may require that we relabel the inputs.

Step 2: Time–Varying Input Transformation For a system of the form (5) there exists an input transformation

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = A(t) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

such that, the system resulting from the state transformation $g_r = g\exp(-(b_0\times)t)$ has no drift

$$\dot{g}_r = g_r(b_1\times)v_1 + g_r(b_2\times)v_2 \tag{6}$$

Notice that $g_r = g\exp(-(b_0\times)t)$ is a time–varying transformation that exactly compensates for the drift by making the reference frame rotate around b_0.

This may be demonstrated in a constructive way. We will derive the input transformation matrix $A(t)$.

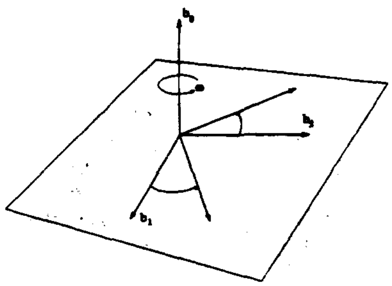

Figure 1: Note the plane Ω which the vectors b_1 and b_2 span remains constant under the action of the drift b_0 even though b_1 and b_2 do not.

From $g_r = g\exp(-(b_0\times)t)$ we can write $g = g_r\exp((b_0\times)t)$, whose derivative is

$$\begin{aligned}\dot{g} &= \dot{g}_r\,\mathrm{Exp}((b_0\times)t) + g_r\,\mathrm{Exp}((b_0\times)t)(b_0\times)\\ &\quad \dot{g}_r\,\mathrm{Exp}((b_0\times)t) + g(b_0\times)\end{aligned} \tag{7}$$

We want to find a new pair of inputs (v_1, v_2) such that

$$\dot{g}_r = g_r(b_1\times)v_1 + g_r(b_2\times)v_2 \tag{8}$$

The two inputs v_1 and v_2 that solve the problem can be determined by using equation (8) in equation (7) to get:

$$\begin{aligned}\dot{g} &= (g_r(b_1\times)v_1 + g_r(b_2\times)v_2)\ \mathrm{Exp}((b_0\times)t) + g(b_0\times)\\ &= g\ \mathrm{Exp}(-(b_0\times)t)(b_1\times)\ \mathrm{Exp}((b_0\times)t)v_1 +\\ &\quad g\ \mathrm{Exp}(-(b_0\times)t)(b_2\times)\ \mathrm{Exp}((b_0\times)t)v_2\\ &\quad +g(b_0\times)\end{aligned} \tag{9}$$

We recall that, given any rotation matrix $R \in SO(3)$ and any skew symmetric matrix $(b\times) \in so(3)$, we have $R(b\times)R^{-1} = (Rb\times)$, therefore equation (9) becomes

$$\dot{g} = g(b_0\times) + g(c_1\times)v_1 + g(c_2\times)v_2 \tag{10}$$

where $c_1 = \exp(-(b_0\times)t)b_1$ and $c_2 = \exp(-(b_0\times)t)b_2$. These two terms can be computed by means of Rodrigues' formula

$$\begin{aligned}c_1 &= \left(I - (b_0\times)\frac{1}{\omega}\sin\omega t + (b_0\times)^2\frac{1}{\omega^2}(1-\cos\omega t)\right)b_1\\ &= b_1 - b_0\times b_1\frac{1}{\omega}\sin\omega t + (b_0\times)(b_0\times b_1)\frac{1}{\omega^2}(1-\cos\omega t)\\ &= b_1 - b_2\sin\omega t + b_0\times b_2\frac{1}{\omega}(1-\cos\omega t)\\ &= b_1 - b_2\sin\omega t - b_1(1-\cos\omega t)\\ &= b_1\cos\omega t - b_2\sin\omega t\end{aligned}$$

Similarly we have

$$\begin{aligned}c_2 &= b_2 - b_0\times b_2\frac{1}{\omega}\sin\omega t + (b_0\times)(b_0\times b_2)\frac{1}{\omega^2}(1-\cos\omega t)\\ &= b_2 - b_1\sin\omega t + b_0\times b_1\frac{1}{\omega}(1-\cos\omega t)\\ &= b_1\sin\omega t + b_2\cos\omega t\end{aligned}$$

Equation (10) thus becomes

$$\begin{aligned}\dot{g} &= g((b_1\times)\cos\omega t - (b_2\times)\sin\omega t)v_1 +\\ &\quad g((b_1\times)\sin\omega t + (b_2\times)\cos\omega t)v_2 +\\ &\quad g(b_0\times)\end{aligned} \tag{11}$$

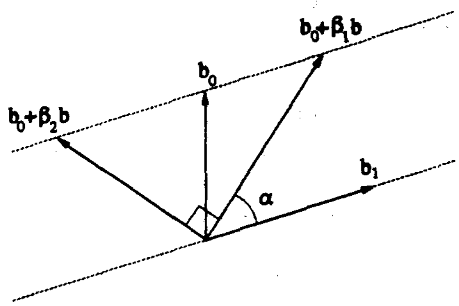

Figure 2: This figure shows geometrically how the constants β_1 and β_2 are chosen in order to insure that the resulting inputs are orthogonal. Notice that there is one degree of freedom, α.

It is now clear that by setting

$$\begin{bmatrix}u_1\\u_2\end{bmatrix} = A(t)\begin{bmatrix}v_1\\v_2\end{bmatrix} = \begin{bmatrix}\cos\omega t & \sin\omega t\\ -\sin\omega t & \cos\omega t\end{bmatrix}\begin{bmatrix}v_1\\v_2\end{bmatrix} \tag{12}$$

we obtain the system (5).

Step 3: Computation Find the input sequence that drives the drift–free system

$$\dot{g}_r = g_r(b_1\times)v_1 + g_r(b_2\times)v_2$$

from g_i to $g_f\ \mathrm{Exp}(-(b_0\times)T)$ in the interval $(0, T)$ by using the method of theorem (1).

Step 4: Application Apply the resulting controls to the system

$$\begin{aligned}\dot{g} = g(b_0\times) + g(b_1\times)(\cos(\omega t)v_1 + \sin(\omega t)v_2)\\ g(b_2\times)(-\sin(\omega t)v_1 + \cos(\omega t)v_2)\end{aligned}$$

3.3 The One Input Control System with Drift

Now we will exploit rather than ignore the drift provided that we have fewer constraints on the arrival time.

Proposition 4

Given a control system on $SO(3)$ whose evolution is described by $\dot{g} = g(b_0\times) + g(b_1\times)u_1$, with b_0 and b_1 linearly independent, g_i and g_f both in $SO(3)$

Then there exists a $T \in \mathbb{R}_+$ and a $u(\cdot)$ defined on $[0, T]$, piecewise constant, which will steer the system from g_i to g_f in the interval $[0, T]$.

Proof: The proof will be algorithmic, as before.

Step 1: Orthogonalization There is one degree of freedom assuming there are no input constraints. Find the numbers β_1, β_2 so that $b_0 + \beta_1 b_1$ is orthogonal to $b_0 + \beta_2 b_1$. While the plane in which these vectors lie in is fixed, the two vectors may rotate in a limited way in the plane. Call the normalized versions of the resulting two vectors h_1, h_2.

Step 2: Computation Apply the procedure of Section 2.2 to the system:

$$\dot{g} = g(h_1u_1 + h_2u_2)$$

with g_i and g_f as before, with $T = 3$.

Step 3: Time Scaling There will be two problems with the solutions that may arise from the computation step. First, they might have negative values. The drift can not be reversed, but luckily we may just add 2π to any negative result and convert the input to a positive one.

Also, the inputs may not be varied. However, the amount of time they are applied can be adjusted. Instead of applying the first input for one second for example, we can apply β_1 for $\frac{a_1}{\|b_0+\beta_1 b_1\|}$ seconds. Set the scaling constants c_1 and c_2 to $\|b_0 + \beta_1 b_1\|$ and $\|b_0 + \beta_2 b_1\|$ respectively.

Step 4: Application Apply the control $u_1 = \beta_1$ for $\frac{a_1}{c_1}$ seconds, $u_1 = \beta_2$ for $\frac{a_2}{c_2}$ seconds and $u_1 = \beta_1$ for $\frac{a_1}{c_1}$ seconds.

Again, the choice of β_1 and β_2 must be done by taking into account possible constraint on the time necessary to steer the system and (or) on the magnitude of the input.

4 Simulation Strategies

While convenient for algebraic manipulation, matrix-form differential equations for $SO(3)$ are not suitable for numerical simulation. Recall that the Lie group $SO(3)$ is a three-dimensional submanifold of $GL(3)$. If matrix differenitial equations for $SO(3)$ are used, numerical error may slowly drive the nine states off the sub-manifold. One may resolve this by using a smooth map from $\mathbb{R}^3$ to to $SO(3)$, for example the *roll-pitch-yaw* coordinate chart, and simulating the system in $\mathbb{R}^3$ instead. These maps, however, are prone to singularities and if used they require frequent change of coordinates.

We chose to use the quaternions representing $SO(3)$, avoiding the singularities. Given any $g \in SO(3)$, there exists an $\omega \in \mathbb{R}^3$ of unit length and a θ such that g is a rotation about ω through θ degrees. The quaternion parameters are then given by:

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} \cos(\frac{\theta}{2}) \\ \omega_1 \sin(\frac{\theta}{2}) \\ \omega_2 \sin(\frac{\theta}{2}) \\ \omega_3 \sin(\frac{\theta}{2}) \end{pmatrix}$$

Given the quaternians $q \in \mathbb{R}^4$, the matrix g may be computed directly.

$$g = 2\begin{bmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 2 \\ q_1 q_2 + q_0 q_3 & q_0^2 + q_2^2 - \frac{1}{2} & q_0 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & q_0^2 + q_3^2 - \frac{1}{2} \end{bmatrix}$$

While Cayley's formula may be applied to compute ω and θ and hence the quaternion, there are more direct methods. Designating the i, j^{th} element of g by g_{ij}, we obtain

$$\begin{aligned} q_0 &= \sqrt{\frac{\operatorname{trace}(g) + 1}{4}} \\ q_1 &= \frac{g_{32} - g_{23}}{4q_0} \\ q_2 &= \frac{g_{13} - g_{31}}{4q_0} \\ q_3 &= \frac{g_{21} - g_{12}}{4q_0} \end{aligned}$$

The above holds unless $q_0 = 0$. If it is, $g_{ij} + g_{ji} \neq 0$ implies that q_i and q_j are not equal to zero. Then the diagonal terms may be used to compute that $q_i^2 = g_{ii} + \frac{1}{2}$.

Finally, given a left-invariant vector field given by $g(\omega\times)$, the evolution of the quaternion parameters is given by:

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} q$$

There are many nice properties to this parameterization, for example matrix multiplication maps simply to quaternion multiplication. For details, see [6].

In the conference presentation, we will present animated simulations for steering the attitude of satellites and space robots using the algorithms developed here.

5 Conclusion

This paper was an attempt to make a complete solution for a class of steering problems with algorithms. The four theorems embody the the different approaches applied. The first employs an input and coordinate transformation to put a general system into a canonical form for which we have precomputed formulas. The others which follow employ various input and coordinate transformations to once again put a more general system into this canonical form. In future work, we will generalize these approaches to more general matrix Lie groups.

References

[1] V. I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, New York, New York, second edition, 1989.

[2] J. Baillieul. Geometric methods for nonlinear optimal control problems. *Journal of Optimization Theory and Application*, 25(4):519 – 548, 1978.

[3] W-L. Chow. Über systeme von linearen partiellen differentialgleichungen erster ordnung. *Math. Annalen*, 117:998–105, 1940-41.

[4] J. Marsden. *London Mathematical Society Lecture Note Series 174: Lectures on Mechanics.* Cambridge University Press, Cambridge, England, 1992.

[5] J. M. McCarthy. *Introduction to Theoretical Kinematics.* MIT Press, Cambridge, MA, 1990.

[6] Parviz E. Nikravesh. *Computer-Aided Analysis and Design of Mechanical Systems.* Prentice-Hall, 1989.

[7] P. J. Olver. *Applications of Lie Groups to Differential Equations.* Springer-Verlag, New York, New York, 1986.

[8] M. Reyhanoglu and N. H. McClamroch. Conrollability and stabilizability of planar multibody systems with angular momentum preserving control torques. In *American Controls Conference*, 1991. Boston, Mass.

[9] V. S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations.* Springer-Verlag, New York, New York, 1984.

[10] G. Walsh and S. Sastry. On reorienting rigid linked bodies using internal motions. In *IEEE Conference on Decision and Control*, pages 1190 – 1195, 1991. Also as Electronic Research Laboratory memo M91/35.