Manipulating Robots along Helical Trajectories

J.M. Selig
School of Elec. Electronic & Info. Eng.
South Bank University
London SE1 0AA, U.K.

A.I. Ovseevitch
Institute for Problems in Mechanic
Prospect Vernadsgogo 101
Moscow, Russia

Abstract

Current industrial robots are highly non-linear systems. However,the control strategies in most commercially available robots largely ignore the non-linearity. The resulting linear approximations are only valid at low speeds. Any improvement would allow robots to move faster and hence be more productive. There has been much academic research into robot control, but this has almost always separated the control and the trajectory planning. In this work we seek to combine these tasks and hence simplify the computations required. We investigate how to control a general robot in such a way that it's gripper follows straight line, circular or helical paths. These simple paths are both one parameter subgroups for the group of proper rigid motions and geodesics for certain metrics on the group. This suggests a non-linear feedback control law which turns the closed loop dynamics of the robot into the equations for geodesics. Although these equations are not completely stable we are able to modify the control law so that the resulting closed loop dynamics are stable. Hence, the end-effector of the robot will follow straight line, helical or circular trajectories.

1 Introduction

In many application the end-effector of a robot is required to perform a straight line motion, or a circular motion. Here we study a generalisation of this problem. That is we look at how to drive the end-effector through a helical or screw motion. When the pitch of the screw is zero the motion reduces to a circular path and if the pitch is 'infinite' the motion is a straight line. The study of these screw motions has a long history in mathematics predating the modern description which uses group theory. More recently, some authors have suggested that

screw motions could be used to advantage in robot trajectory planning, see [1] and [2].

Usually in robotics, paths are thought of as the paths of points, however we prefer to think of the path of the rigid body as a whole. This involves using some notions from group theory. The set of positions that a rigid body can take up is the manifold of a six dimensional group. This group is the proper Euclidian group or group of proper rigid motions. If the rigid body undergoes a finite motion we may think of the successive configurations as points on a path in the group manifold. The paths corresponding to helical motions are very special for a number of reasons. Firstly helical motions result from one parameter subgroups of the group. Secondly, they are the geodesics of certain metrics that can be defined on the group. That is, they are the paths of 'minimal length' on the group manifold, and in a sense are optimal paths.

Traditionally, to move a robot through a helical motion we would perform the following steps. First we would generate the successive positions on the path, then perform the inverse kinematics at each point to get the values of the joint angles for each position. The results would then be passed to the robot's control system. If the control system of the robot uses the Computed Torque method, see [3, sect. 7.6.1] then, from the sequence of joint angles we would calculate the torques required to move the robot to the next position in the sequence.

In this work we suggest an alternative approach. The required torques are calculated directly from the measured values of the angles and their velocities. However, the calculation is done in such a way that the closed loop dynamics of this feedback system are the geodesic equations given below. Hence, the path of the robot's end-effector will satisfy the equations and trace out a geodesic, a straight line, circle or helical path depending on the initial conditions. Notice that we avoid the computation of the successive positions on the trajectory and the inverse kinematics. In fact the computations we must perform are roughly comparable in complexity to those involved in the Computed Torque method. However, we stress that the proposed method is a feedback system, see section 4 below.

Before we begin however, we will take the opportunity to review some basic robotics from the point of view of group theory. See also [1] for an introduction to this material.

Robotics is largely concerned with moving solid objects about in space, hence the group of proper rigid motions $SE(3)$, is central to the study of robots. This group is six dimensional, it is the semi-direct product of the rotation group $SO(3)$ with the group of translations $I\!\!R^3$. Historically, the study of this group predates modern group theory, for example; Chasles theorem (1830) states that every proper rigid motion is a screw motion about some axis, see Hunt [4, sect.2.8]. Hence, we will sometimes refer to the group elements as finite screws.

Many current industrial robots consist of six rigid links connected in series via one degree-of-freedom joints, see fig. 1. Six joints are used so that the last link, or end-effector of the robot has six degrees of freedom. Since the group is six dimensional this is the maximum number of degrees of freedom that a

single rigid body can have. The joints are usually simple hinges, known as revolute joints, or sliding joints, called prismatic joints. More rarely, a helical joint may be used, such a joint can move in the same way as a nut does on a bolt. The common thing about these joints is that they can be realised with a pair of mating surfaces, one solid the other hollow. These surfaces can move relative to each other while still remaining in contact. Such joints are known as Reuleaux lower pairs, after Franz Reuleaux (1829–1905). The relative motion allowed by such joints corresponds exactly with the one parameter subgroups of $SE(3)$, indeed each one parameter subgroup is the symmetry group of the mating surface of a joint. Revolute joints can be made using any surface of revolution and prismatic joints from surfaces of translation. The one parameter groups can be parameterised using the exponential map from the Lie algebra to the group.

$$g(t) = e^{t\mathbf{s}}$$

The Lie algebra element $\mathbf{s}$ characterises the position and type of the joint, and so will be referred to as the **joint screw**. More properly we should say the **instantaneous** joint screw, to distinguish the Lie algebra elements from the group itself, however the meaning should be clear from the context. Note, that we do not follow [1] where the term screw is used to denote a ray in the Lie algebra, that is an element of the projective space made from the Lie algebra.

For a six joint serial robot the forward kinematics gives a map from joint space, with coordinates:

$$\boldsymbol{\theta} = (\theta^1, \theta^2, \theta^3, \theta^4, \theta^5, \theta^6)$$

to the group $SE(3)$;

$$\rho : \boldsymbol{\theta} \longmapsto e^{\theta^1 \mathbf{s}_1} e^{\theta^2 \mathbf{s}_2} e^{\theta^3 \mathbf{s}_3} e^{\theta^4 \mathbf{s}_4} e^{\theta^5 \mathbf{s}_5} e^{\theta^6 \mathbf{s}_6}$$

see [5].

Although we have indicated the joint variables by theta's it is not intended that only revolute joints can be treated, any one degree-of-freedom joint can be considered. However, if the joints are all revolute, then topology of the joint space will be that of a six-torus. That is, joint space is the Cartesian product of six circles, since a point in this space can be completely specified by the six joint angles. For prismatic or helical joints the joint parameter is a real number, different numbers giving different positions of the joint. Hence, the configuration space for such a joint is topologically a real line. So when the robot has revolute and prismatic or helical joints, the topology of the joint space is the Cartesian product of lines and circles, a circle for each revolute joint and a line for every helical or prismatic joint.

The velocities of the links are also Lie algebra elements. The angular and linear velocities combine to give six component vectors. For a general path in the group:

$$\gamma : t \longmapsto g(t)$$

Figure 1: A Six Joint Serial Robot

the velocity of such a path is given by the Lie algebra element:

$$\dot{\mathbf{q}} = g^{-1}\dot{g}$$

The velocity of the sixth link $\dot{\mathbf{q}}_6$, of a six joint serial robot can be found by differentiating the forward kinematics:

$$\dot{\mathbf{q}}_6 = \dot{\theta}^1 \mathbf{s}_1 + \dot{\theta}^2 \mathbf{s}_2 + \dot{\theta}^3 \mathbf{s}_3 + \dot{\theta}^4 \mathbf{s}_4 + \dot{\theta}^5 \mathbf{s}_5 + \dot{\theta}^6 \mathbf{s}_6$$

This equation is independent of the representation we use for the group and its Lie algebra. However, if we use the six dimensional representation then vectors on the joint space are related to vectors on the group by the Jacobian:

$$\dot{\mathbf{q}}_6 = J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

The columns of the Jacobian matrix given above are simply the joint screws.

$$J(\boldsymbol{\theta}) = \left(\mathbf{s}_1 \,|\, \mathbf{s}_2 \,|\, \cdots \,|\, \mathbf{s}_6 \right)$$

These six component vectors can be partitioned into two 3 component vectors:

$$\mathbf{s} = \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{pmatrix}$$

where $\boldsymbol{\omega}$ is the angular velocity and $\mathbf{v}$ is the linear velocity of any point on the axis of the screw. Note also that these are the joint screws at a given configuration. As the robot moves so do its joints, but in the notation we have

4

suppressed the dependency of the joint screws on the joint angles. We will return to this in a moment but first a little more algebra.

The adjoint action of the group on its Lie algebra can be written using the six dimensional representation of the group:

$$\begin{pmatrix} \boldsymbol{\omega}' \\ \mathbf{v}' \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{XR} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{pmatrix}$$

Where $\mathbf{R}$ is a 3×3 Special Orthogonal matrix and $\mathbf{X}$ represents a translation by $\mathbf{x}$. It is the anti-symmetric matrix which gives $\mathbf{Xu} = \mathbf{x} \times \mathbf{u}$ for any three dimensional vector $\mathbf{u}$, that is:

$$\mathbf{X} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}$$

The adjoint action of the Lie algebra on itself is given by the commutator. In this Lie algebra the commutator of two elements is given by:

$$[\mathbf{s}_1, \mathbf{s}_2] = \begin{pmatrix} \boldsymbol{\omega}_1 \times \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_1 \times \mathbf{v}_2 + \mathbf{v}_1 \times \boldsymbol{\omega}_2 \end{pmatrix}$$

The cross here is the standard vector product in three dimensions. Notice that, we can represent this action by a 6×6 matrix:

$$ad(\mathbf{s}) = \begin{pmatrix} \Omega & 0 \\ V & \Omega \end{pmatrix}$$

where $\Omega \mathbf{x} = \boldsymbol{\omega} \times \mathbf{x}$ and $V\mathbf{x} = \mathbf{v} \times \mathbf{x}$ for and three dimensional vector $\mathbf{x}$.

Returning to the movement of joint screw we note that the ith joint moves about the joints lower down in the chain, so that:

$$\mathbf{s}_i(\boldsymbol{\theta}) = e^{\theta^1 \mathbf{s}_1} e^{\theta^2 \mathbf{s}_2} \cdots e^{\theta^{i-1} \mathbf{s}_{i-1}} \mathbf{s}_i(\mathbf{0})$$

where the screws in the exponents are the joint screws in their home positions taken in the adjoint representation. Hence the time derivative of a joint screw is given by:

$$\frac{d}{dt} \mathbf{s}_i = \dot{\theta}^1 [\mathbf{s}_1, \mathbf{s}_i] + \dot{\theta}^2 [\mathbf{s}_2, \mathbf{s}_i] + \cdots$$

$$\cdots + \dot{\theta}^{i-1} [\mathbf{s}_{i-1}, \mathbf{s}_i]$$

Finally in this section we look at the two bilinear forms defined on the Lie algebra which are invariant with respect to the adjoint action. Rather, the one parameter linear system of such forms. These bilinear forms define bi-invariant metrics on the group. A basis for the forms is given by the two, 6×6 matrices, partitioned as:

$$Q_0 = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{pmatrix},$$

and

$$Q_\infty = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Again, these forms have been known for a century or so, the first is usually called the reciprocal product. The other does not seem to have a special name, but it is the Killing form of the Lie algebra. Recall that the Killing form of two elements of a Lie algebra is found by taking the trace of the product of the matrices representing the elements in the adjoint representation of the algebra. Notice that Q_∞ is the only one of these forms which is positive and it is also the only degenerate one.

Almost all screws can be written as:

$$\mathbf{s} = \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{r} \times \boldsymbol{\omega} + p\boldsymbol{\omega} \end{pmatrix}$$

The exception are the screws for which $\boldsymbol{\omega} = \mathbf{0}$. The vector $\boldsymbol{\omega}$ is in the direction of the screw axis and $\mathbf{r}$ is the position vector of any point on the screw axis. The constant p is called the **pitch** of the screw, it is the distance moved along the screw axis for every revolution about the axis. In terms of the bilinear forms, the pitch of a screw $\mathbf{s}$ is given by the ratio:

$$p = \frac{\mathbf{s}^T Q_0 \mathbf{s}}{2\mathbf{s}^T Q_\infty \mathbf{s}}$$

Conventionally, when $\boldsymbol{\omega} = \mathbf{0}$, we say that the screw has infinite pitch, in this case the screw is an infinitesimal pure translation and has the shape:

$$\mathbf{s} = \begin{pmatrix} \mathbf{0} \\ \mathbf{v} \end{pmatrix}$$

We may also associate a pitch to a general bilinear form, suppose $p = -\beta/2\alpha$, then a pitch p form is given by:

$$Q_p = \alpha Q_0 + \beta Q_\infty = \begin{pmatrix} \beta\mathbf{I} & \alpha\mathbf{I} \\ \alpha\mathbf{I} & \mathbf{0} \end{pmatrix}$$

This definition has been adopted so that for a pitch p screw $\mathbf{s}$ we have:

$$\mathbf{s}^T Q_p \mathbf{s} = 0$$

2   Christoffel Symbols

The bilinear forms given above can be thought of as bi-invariant metrics on the group $SE(3)$. In this section we derive the equations for geodesics with respect to these metrics. In particular we find the Christoffel symbols for the metrics as functions of the robot's configuration. That is, we will evaluate the Christoffel symbols for the metric on joint space in terms of the robot's joint screws.

So we begin by pulling-back the metrics to joint space and expressing the geodesic equations in terms of the co-ordinates in joint space. The pull-back to joint space of the metric Q_p is simply given by:

$$Q_p^* = J^T Q_p J$$

where J, is the Jacobian of the forward kinematic mapping.

Given a bi-invariant metric on a compact Lie group, the geodesic curves will be exactly the one parameter subgroups and their translates, [6, p.224], [7, p.113] or [8, p.232]. In fact, the proofs in the literature do not depend on the group being compact. So the theorem is true for our group $SE(3)$, which is not compact.

We have many bi-invariant metrics, the difference between them is that although they have the same geodesics the 'length' of such a curve is different under each metric. Recall that the length of tangent vectors along a geodesic is constant. Notice that, under the pitch p metric, the geodesics generated by pitch p screws are null.

According to Milnor [7, p.114], for left invariant vector fields on the group we have:

$$\nabla_{X_i} X_j = \frac{1}{2}[X_i, X_j]$$

where ∇ is the covariant derivative for a bi-invariant metric. Left invariant vector fields on a group are in one to one correspondence with elements of the Lie algebra. So on $SE(3)$ the left invariant vector fields can be thought of as screws. Now given a general vector field on joint space we can use the forward kinematics to 'push-forward' to a vector field on the group, a pair of vector fields will be given by:

$$\begin{aligned} \mathbf{U} &= \dot{\theta}^1 \mathbf{s}_1 + \dot{\theta}^2 \mathbf{s}_2 + \cdots + \dot{\theta}^6 \mathbf{s}_6 \\ \mathbf{V} &= \dot{\phi}^1 \mathbf{s}_1 + \dot{\phi}^2 \mathbf{s}_2 + \cdots + \dot{\phi}^6 \mathbf{s}_6 \end{aligned}$$

Now the covariant derivative can be calculated:

$$\begin{aligned} \nabla_{\mathbf{U}} \mathbf{V} &= \partial_{\mathbf{U}} \mathbf{V} - \frac{1}{2}[\mathbf{U}, \mathbf{V}] \\ &= \sum_{i=1}^{6} \dot{\theta}^i \frac{\partial}{\partial \theta^i} \mathbf{V} - \frac{1}{2}[\mathbf{U}, \mathbf{V}] \end{aligned}$$

Substituting for $\mathbf{U}$ and $\mathbf{V}$ and simplifying we obtain:

$$\nabla_{\mathbf{U}} \mathbf{V} = \sum_{i=1}^{6} \ddot{\phi}^i \mathbf{s}_i +$$

$$\frac{1}{2} \sum_{1 \leq i < k \leq 6} (\dot{\theta}^i \dot{\phi}^k - \dot{\theta}^k \dot{\phi}^i)[\mathbf{s}_i, \mathbf{s}_k]$$

7

This general result tells us how to find the covariant derivative of vector fields and hence how to parallel translate those vectors. For our immediate purposes we want the equations for the geodesics in joint space. A straightforward calculation gives:

$$\nabla_{\mathbf{U}}\mathbf{U} = \sum_{i=1}^{6} \ddot{\theta}^i \mathbf{s}_i + \sum_{1 \le i < j \le 6} \dot{\theta}^i \dot{\theta}^k [\mathbf{s}_i, \mathbf{s}_j]$$

Hence, the geodesic equations in joint space are given by setting the last relation to zero:

$$\nabla_{\mathbf{U}}\mathbf{U} = \sum_{i=1}^{6} \ddot{\theta}^i \mathbf{s}_i + \sum_{1 \le i < j \le 6} \dot{\theta}^i \dot{\theta}^k [\mathbf{s}_i, \mathbf{s}_j] = 0$$

These equations, with very slight modifications, make sense for robots with less than six joints. Our equations describe the geodesics in joint space. However, when the joint space has fewer dimensions than the group, geodesics in joint space do not necessarily correspond to geodesics in the group.

The equations can be tidied up further, away from singularities. When the Jacobian is non-singular we can invert the Jacobian of the robot, let us label the rows of the inverse Jacobian $\mathcal{S}^i$ so that:

$$\mathcal{S}^i(\mathbf{s}_j) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \ne j \end{cases}$$

The equations can be rewritten as:

$$\ddot{\theta}^i + \sum_{1 \le j < k \le 6} \mathcal{S}^i[\mathbf{s}_j, \mathbf{s}_k] \dot{\theta}^j \dot{\theta}^k = 0$$

$$i = 1, 2 \ldots 6$$

Compare this with the usual geodesic equations on a non-degenerate metric space:

$$\ddot{x}^i + \Gamma^i_{jk} \dot{x}_j \dot{x}_k = 0$$

where it is understood that repeated indices are to be summed over. There are many possible ways to choose the Christoffel symbols to give the same connection but if we require the connection to be symmetric (zero torsion) then the Christoffel symbols of the symmetric connection on joint space is given uniquely by:

$$\Gamma^i_{jk} = \begin{cases} \frac{1}{2}\mathcal{S}^i[\mathbf{s}_j, \mathbf{s}_k] & j \le k \\ -\frac{1}{2}\mathcal{S}^i[\mathbf{s}_j, \mathbf{s}_k] & j \ge k \end{cases}$$

We believe this result is new. These are not the same Christoffel symbols that appear in robot dynamics, see for example [3, p.142], there it is the Lagrangian function which is being minimised. There are, however, other ways to find these Christoffel symbols. We could have looked at the 'length' of a curve in joint space. This is given by the integral:

$$\int \dot{\boldsymbol{\theta}}^T J^T Q_P J \dot{\boldsymbol{\theta}} \, dt$$

This is usually considered to be the square of path's length, but our metrics are not positive. It is well known that the geodesic equations can be derived from variations of the path length integral. The Euler-Lagrange equations for this variational problem are quite simple to compute using the results of [9].

The easiest, but least informative, way to find the geodesic equations is to look at the velocity of the robot's last link. From what we know about geodesics in groups, we know that $\dot{\mathbf{q}}_6 = \text{constant}$ is a geodesic in the group. Hence the derivative is zero.

$$\ddot{\mathbf{q}}_6 = \mathbf{0}$$

This can be written in terms of joint angles and their rates, since, from the introduction we know that:

$$\dot{\mathbf{q}}_6 = \sum_{i=1}^{6} \dot{\theta}^i \mathbf{s}_i$$

and

$$\frac{d}{dt}\mathbf{s}_i = \sum_{j=1}^{i} \dot{\theta}^j [\mathbf{s}_j, \mathbf{s}_i]$$

3 Control

Our aim now is to choose a control law so that the closed loop dynamics of the system are just the geodesic equation derived above. If we succeed, the end-effector of the robot will follow a geodesic path; straight line, circle or helix, depending on the initial conditions.

The model we will use for the dynamics of a Robot can be summarised as:

$$A_j^i(\boldsymbol{\theta})\ddot{\theta}^j + B_{jk}^i(\boldsymbol{\theta})\dot{\theta}^j\dot{\theta}^k + C^i(\boldsymbol{\theta}) = \tau^i$$

again, with the summing convention understood. The matrix A contains the information about the inertia of the links while the Bs refer to the interactions between links and also the Coriolis terms. The Cs are the weights of the links and the τs are the generalised forces applied to the joints; for revolute joints these will be torques. See [10, Chap.6] or [3, chap.6].

In order to control the robot we must choose torques at the joints to drive the machine along the desired path. Conventionally, the desired path of the end-effector will be given. From this the corresponding path in joint space can be computed. Typically this involves finding points along the path of the end-effector and then using the robot's inverse kinematics at each point. Once we have the path in joint space, or at least a discrete approximation to it, we can compute the required torques using the equations of motion above.

For geodesic path we can do better than this. We can dispense with the inverse kinematics completely. In the last section we derived the differential equations for paths in joint space corresponding to the geodesics in the group. Now suppose we can choose a control law which makes the closed loop dynamics

Figure 2: Schematic for Control Along Geodesics

of the system into these geodesic equations. That is, we choose the joint torques to be functions of the joint angles and their velocities.

This goal can be achieved we set the joint torques to:

$$\tau^i = (B^i_{jk} - A^i_l \Gamma^l_{jk})\dot{\theta}^j \dot{\theta}^k + C^i$$

$$i = 1, 2 \ldots 6$$

Note that, since this is a feedback law we assume that the values for the joint angles and their velocities are measured values. The closed loop equations of motion are now:

$$A^i_l \ddot{\theta}^l + A^i_l \Gamma^l_{jk} \dot{\theta}^j \dot{\theta}^k = 0$$

The generalised inertia matrix A is always invertible, so this is just the equation for geodesics, as promised. Hence, the solutions will be one parameter subgroups, that is helical motions of the robot's end-effector. The pitch of the screw motion will depend on the initial conditions, essentially the linear and angular velocity of the robot's last link at time $t = 0$. See fig. 2 for a schematic diagram of the proposed controller.

There is a standard way of studying the stability of geodesic equations, see [11, sect.6.9] or [7, sect.19]. The geodesic equations can be linearised about a particular geodesic. This gives the equations of geodesic deviation:

$$\nabla_{\mathbf{X}} \nabla_{\mathbf{X}} \mathbf{Z} + \mathbf{R}(\mathbf{X}, \mathbf{Z})\mathbf{X} = 0$$

The vector field $\mathbf{X}$ is tangent to the particular geodesic. While $\mathbf{Z}$ is a field on the geodesic which expresses the difference between the geodesic and nearby geodesics, such a field is called a Jacobi field. The term $\mathbf{R}$ is the Riemann curvature tensor of the metric. It is defined by:

$$\mathbf{R}(\mathbf{U}, \mathbf{V})\mathbf{W} =$$

$$-\nabla_{\mathbf{U}} \nabla_{\mathbf{V}} \mathbf{W} + \nabla_{\mathbf{V}} \nabla_{\mathbf{U}} \mathbf{W} + \nabla_{[\mathbf{U}, \mathbf{V}]} \mathbf{W}$$

10

(Notice, it is also common to give $\mathbf{R}$ the opposite sign, the conventions in [7] will be followed here).

According to Milnor [7, sect.21], on a Lie group with bi-invariant metric, we have that:

$$\mathbf{R}(\mathbf{X}, \mathbf{Y})\mathbf{Z} = \frac{1}{4}[[\mathbf{X}, \mathbf{Y}], \mathbf{Z}]$$

Hence, we may write the equations of geodesic deviation as:

$$\nabla_{\mathbf{X}}\nabla_{\mathbf{X}}\mathbf{Z} - \frac{1}{4}ad^2(\mathbf{X})\mathbf{Z} = 0$$

The stability of this equation depends on the eigenvalues of the matrix $ad^2(\mathbf{X})$ where ad is the adjoint representation of the Lie algebra on itself, see section 1 above. A simple computation shows that for the group $SE(3)$, the matrix $ad^2(\mathbf{s})$, generally has eigenvalues; -1 with multiplicity four and 0 with multiplicity two. Exceptionally, when $\mathbf{s}$ is an infinitesimal pure translation, all six eigenvalues are 0. Hence we see that most geodesics near a finite pitch geodesic stay near. However, some will drift away, those corresponding to the zero eigenvectors. If the original geodesic is generated by the Lie algebra element $\mathbf{s}$ then the eigenvectors corresponding to the zero eigenvalue of $ad^2(\mathbf{s})$ are just those elements which commute with $\mathbf{s}$. These are simply the screws with the same axis as $\mathbf{s}$.

It is possible to modify the proposed control law so that the closed loop dynamics are absolutely stable. Consider the following equations in the group:

$$\ddot{\mathbf{q}}_6 = \lambda(\mathbf{s}_d - \dot{\mathbf{q}}_6)$$

Certainly, $\dot{\mathbf{q}}_6 = \mathbf{s}_d$ is a solution to this equation for any value of the constant λ. Moreover, nearby solutions converge to this solution. To see this, assume that $\dot{\mathbf{q}}_6$ is given by a linear combination of constant screws which form a basis of the Lie algebra:

$$\dot{\mathbf{q}}_6 = a^d \mathbf{s}_d + a^1 \mathbf{z}_1 + a^2 \mathbf{z}_2 + a^3 \mathbf{z}_3 + a^4 \mathbf{z}_4 + a^5 \mathbf{z}_5$$

where the a^is are non-constant coefficients. This equation now reduces to the six equations for the coefficients:

$$\dot{a}^d = \lambda(1 - a^d), \quad \text{and} \quad \dot{a}^i = -\lambda a^i$$

$$i = 1, 2 \ldots 5$$

The solutions of which are:

$$a^d(t) = (1 - e^{-\lambda t}) + a^d(0)e^{-\lambda t}$$

and

$$a^i(t) = e^{-\lambda t} a^i(0) \quad i = 1, 2 \ldots 5$$

If we choose λ to be positive, then it is clear that extraneous solutions will decay away and a^d will tend to 1.

Just as with the geodesic equations, this equation can be pulled-back to joint space to give equations in the joint variables:

$$\ddot{\theta}^i + \sum_{j<k} \mathcal{S}^i([\mathbf{s}_j, \mathbf{s}_k])\dot{\theta}^j\dot{\theta}^k =$$

$$\lambda \mathcal{S}^i(\mathbf{s}_d - \sum_{j=1}^{6} \dot{\theta}^j \mathbf{s}_j)$$

$$i = 1, 2 \ldots 6$$

Since:

$$\ddot{\mathbf{q}}_6 = \sum_{j=1}^{6} \ddot{\theta}^j \mathbf{s}_j + \sum_{1 \leq k < l \leq 6} \dot{\theta}^k \dot{\theta}^l [\mathbf{s}_k, \mathbf{s}_l]$$

Again it is possible to choose a control law which makes the closed loop dynamics of the robot look like the above equations:

$$\tau^i = (B^i_{jk} - A^i_l \Gamma^l_{jk})\dot{\theta}^j\dot{\theta}^k +$$

$$C^i + \lambda A^i_l \mathcal{S}^l(\mathbf{s}_d - \dot{\theta}^j \mathbf{s}_j)$$

$$i = 1, 2 \ldots 6$$

Notice that our simple model of the robot's dynamics takes no account of friction at the joints, the dynamics of the motors nor any flexibility that may be present in the links or joints. It may be argued that the dynamics of the model is too simple especially since the motor dynamics are not included. This is however, a common first step in modelling robot dynamics. It can also be justified by the common practice of controlling each motor separately and then having a supervisory level of control for coordinating the motions of the joints.

We can however, introduce a simple model of friction into the dynamics as follows. Assume that the frictional forces on the joints are proportional to the joint rates. The equations of motion for the robot including this type of friction are just:

$$A^i_j \ddot{\theta}^j + B^i_{jk}\dot{\theta}^j\dot{\theta}^k + C^i + \mu_i\dot{\theta}^i = \tau^i$$

where the μs are the friction coefficients, so no sum over i is intended here. The same technique as above can be used in this case also, we set the joint torques to:

$$\tau^i = (B^i_{jk} - A_{il}\Gamma^l_{jk})\dot{\theta}^j\dot{\theta}^k + C^i +$$

$$\mu_i\dot{\theta}^i + \lambda A^i_l \mathcal{S}^l(\mathbf{s}_d - \dot{\theta}^j \mathbf{s}_j)$$

$$i = 1, 2 \ldots 6$$

In many situations we would like the robot's end-effector to accelerate from rest or decelerate along a geodesic. In other words we would like the end-effector to follow a path given by:

$$\ddot{\mathbf{q}}_6 = e^{\gamma t^2 \mathbf{s}_d / 2}$$

12

where the constant γ depends on the rate of acceleration along the geodesic. For this situation we consider the equation:

$$\ddot{\mathbf{q}}_6 = \gamma \mathbf{s}_d + \lambda(\gamma t \mathbf{s}_d - \dot{\mathbf{q}}_6)$$

Clearly this has a solution:

$$\dot{\mathbf{q}}_6 = \gamma t \mathbf{s}_d$$

as required. As above we can see that this is stable when λ is positive. Hence, to make the closed loop dynamics of the robot match this equation we set the joint torques to:

$$\tau^i = (B_{jk}^i - A_l^i \Gamma_{jk}^l)\dot{\theta}^j \dot{\theta}^k + C^i +$$

$$A_l^i \mathcal{S}^l(\gamma(1 - \lambda t)\mathbf{s}_d - \lambda \dot{\theta}^j \mathbf{s}_j)$$

$$i = 1, 2 \ldots 6$$

Notice that all of our results above depend on the exact cancellation between the modelled and actual dynamics. That is, the quantities A, B, C and so on, must be known with a high degree of precision. This will clearly be a problem when the robot picks up an unknown payload.

4   Comparison with other Methods

In this section we describe how the control method outlined above compares with some of the standard Robot control techniques. The main reference used here is Spong and Vidyasagar [3]. They distinguish between the open loop Computed Torque method and the closed loop version which they call Inverse Dynamics, see [3, sect. 8.3]. In other work both these methods often go by the name Computed Torque. A schematic diagram of both methods is given in fig. 3. Notice that both these methods require an inverse kinematics stage to compute the desired joint angles and their derivatives. The non-linear compensator referred to in the diagram is essentially the dynamic equations of the robot. Given the desired trajectories for the joints we can compute the required torques, the overall effect is to turn the compensator plus robot into a linear system. In the Inverse Dynamics method, this linear system is controlled and stabilised using feedback.

The method outlined in this paper is most similar to the technique called; Inverse Dynamics in Task Space, see [3, sect. 9.4]. We may identify task space with the group manifold of $SE(3)$, that is the space of positions and orientations for the robot's end-effector. The Inverse Dynamics in Task Space method begins by specifying a co-ordinate system for task space. This is only possibly locally since there is no global co-ordinate system on the group $SE(3)$. However, this is not a serious problem since it is only the derivative of the forward kinematic map; from joint space to task space, that is used subsequently. The derivative, that is the Jacobian matrix, is not the standard Jacobian of the robot as used above. Recall, the Jacobian of a robot relates the velocity screw of the end-effector to the joint rates. Simply integrating the components of such a velocity

Figure 3: Schematics for Computed Torque (top) and Inverse Dynamics Control (bottom)

screw will not give a co-ordinate system on the group. However, we can follow the style of the derivation given in [3, sect. 9.4] to get the simplest of the control laws given above.

For a constant motion along a screw we have:

$$J_j^i\ddot{\theta}^j + \dot{J}_j^i\dot{\theta}^j = 0$$

Substituting this in the equations of motion gives the required torques as:

$$\tau^i = B_{jk}^i\dot{\theta}^j\dot{\theta}^k - A_l^i(J^{-1}\dot{J})_k^l\dot{\theta}^k + C^i$$

From the section 2 we can see that:

$$(J^{-1}\dot{J})_k^i = \Gamma_{jk}^i\dot{\theta}^j$$

5 Conclusions

The idea of using non-linear feedback in robotics is not in itself novel, see for example [12] and references therein. However, our introduction of some of the differential geometry of $SE(3)$ into robot control is new. Our approach eliminates the need for much of the computations which would otherwise have to be done. The scale of the computational task seems to be comparable with that for the Computed Torque method, which it would also replace. Like Computed Torque, the technique depends on the exact cancellation of terms in the real and modelled dynamics of the system. The effect of the mismatch between the model and the robot, that is the robustness of the method, has yet to be studied.

The price that is paid for simplifying the control task is that only particular trajectories can be followed, namely the helical ones. More general trajectories could be realised in the same manner if they were geodesics for known metrics or using a scheme similar to Inverse Dynamics in Task Space. Note that there are many other approaches to simplifying the control of robots, for example see [13] where the simplification is achieved by careful design of the arm's dynamics.

Finally, we come to the problem of singularities, that is points where the robot's Jacobian has less than maximal rank. This has been avoided throughout the above. The limitations imposed on the performance of the robot by the singularities are the same as with any control strategy. Near singularities very large forces or torques are required to move the robot, and in general singular positions must be avoided.

It does not seem unreasonable to hope that practical implementation of these ideas will lead to interesting and useful robot controllers.

References

[1] A.E. Samuel, P. R. McAree, and K. H. Hunt. Unifying screw geometry and matrix transformations. *International Journal of Robotics Research*, 10(5):454–472, 1991.

[2] K. H. Hunt. Manipulating a body through a finite screw displacement. In *Proceedings of the 7th World Congress, Seville*, volume 1, pages 187–191, 1987.

[3] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Controll.* John Wiley and Sons, New York, 1989.

[4] K. H. Hunt. *Kinematic Geometry of Mechanisms*. Clarendon Press, Oxford, 1978.

[5] R. W. Brockett. *Robotic Manipulators and Products of Exponential Formula*, volume 58 of *Lecture Notes in Computer Science*, pages 120–129. Springer Verlag, New York, 1984. Proceedings of conference held in Beer Sheva 1983.

[6] S. Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*, volume 80 of *Pure and Applied Mathematics*. Academic Press, New York, 1978.

[7] J. Milnor. *Morse Theory*, volume 51 of *Annals of Mathematics Studies*. Princeton University Press, New Jersey, 1969.

[8] S. Sternberg. *Lectures on Differential Geometry*. Prentis Hall, New Jersey, 1964.

[9] J. M. Selig. Robot dynamics using screw theory. In K. Warwick, editor, *Robotics: Mathematical and Computational Aspects*, pages 293–307. Institute of Mathematics and its Applications, 1993.

[10] R.P. Paul. *Robot Manipulators:Mathematics, Programming and Control.* M.I.T. Press, Cambridge, Mass., 1981.

[11] B. F.Schutz. *Geometrical Methods of Mathematical Physics.* Cambridge University Press, Cambridge, 1980.

[12] M. Cotsaftis and C. Vibet. A direct application of the decoupling method to euler-lagrange formalism. *International Journal of Robotics Research*, 11(1):64–74, 1992.

[13] K. Youcef-Toumi and A.T.Y. Kuo. High-speed trajectory control of a direct-drive manipulator. *IEEE Trans. Robotics and Automation*, 9(1):102–108, 1993.

List of Figures