

Interpolated Rigid-Body Motions and Robotics

J.M. Selig

*Faculty of Business, Computing and Info. Management,
London South Bank University,
London SE1 0AA, U.K.
seligjm@lsbu.ac.uk*

Yuanqing Wu

*Dept. Mechanical Engineering,
Shanghai Jiaotong University,
Shanghai, P.R. China.*

Abstract—This work looks at several problems concerned with interpolating rigid-body motions and their application in robotics. Two recently proposed interpolation methods are shown to produce the same results. We also discuss how it might be possible to control a robot in such a way as to follow one of these interpolated motions.

Index Terms—Rigid-body motion, path planning, interpolated motion.

I. INTRODUCTION

This work has several objectives, the first of which is to correct a small mistake in [10]. This concerns the relations which gives the acceleration of points on the end-effector of a serial robot. The correction is given in the next section together with a brief introduction to the geometrical methods used in the paper.

The second objective is to study a pair of special rigid-body motions, the Frenet-Serret motion is given in [3] and the Bishop's motion defined below. These motions could be used for robot path planning. They should be particularly useful when human operators specify the desired path of the robot's end-effector since they are based on curves in 3 dimensions and are hence easy to visualize.

Next we look at some connections with robotics. Specifically, we study how it is possible to guide a serial robot arm along a particular desired motion. Two answers are sketched, a numerical method for the inverse kinematics and a non-linear control method.

Finally, we look at two recently proposed interpolation schemes, [1] and [4]. Although these methods approach the problem differently, the first focuses on the Lie groups while the second concentrates on the paths of points in space, we are able to show that the two methods produce identical curves.

II. SCREW THEORY

Screw theory concerns the group of rigid body motions $SE(3)$ and its Lie algebra $se(3)$. Elements of the groups, that is rigid body motions, can be written as 4×4 matrices,

$$g = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix}$$

with R a 3×3 rotation matrix and \mathbf{t} a translation vector. We denote a point in space by a vector $\mathbf{p} = (x, y, z)^T$, then we extend this vector by adding an extra 1. After the

rigid transformation the new position of the point will be given by,

$$\begin{pmatrix} \mathbf{p}' \\ 1 \end{pmatrix} = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} = \begin{pmatrix} R\mathbf{p} + \mathbf{t} \\ 1 \end{pmatrix}.$$

Elements of the Lie algebra of the group correspond to velocities or small motions, they are called twists or screws. In general, if we think of a path in the group as a continuously parameterised sequence of group elements $g(t)$, then for any parameter value we have a Lie algebra element given by,

$$S = \left(\frac{d}{dt} g(t) \right) g(t)^{-1}.$$

If the group elements $g(t)$ are given as 4×4 matrices then the Lie algebra element will have the form,

$$S = \begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix},$$

where Ω is a 3×3 anti-symmetric matrix corresponding to the angular velocity of the motion. The vector \mathbf{v} is a characteristic linear velocity of the motion.

The action of the group on its Lie algebra elements is given by the conjugation,

$$S' = g S g^{-1}.$$

This corresponds to a left translation in the group. To see this suppose $g(t)$ is a path in the group as above, now left-translate this by a constant group element h to get $hg(t)$. The Lie algebra elements of this path are now,

$$h \left(\frac{d}{dt} g(t) \right) g(t)^{-1} h^{-1} = h S h^{-1}.$$

Clearly a right translation has no effect on the screw.

It is often convenient to write screws as 6-dimensional vectors, corresponding to a screw $S = \begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix}$ we can write a vector,

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

where the vector $\boldsymbol{\omega}$ corresponds to the anti-symmetric matrix Ω in the following way, for any vector \mathbf{u} we have $\Omega \mathbf{u} = \boldsymbol{\omega} \times \mathbf{u}$.

This 6-dimensional representation of screws is useful in many cases. In particular it allows us to define elements

of the dual Lie algebra. Elements of this dual vector space are called wrenches and written,

$$\mathcal{W} = \begin{pmatrix} \mathbf{M} \\ \mathbf{F} \end{pmatrix},$$

where \mathbf{F} is a force and \mathbf{M} is a moment. These could also be linear and angular momenta, if we were considering dynamics.

If a rigid body is moving with instantaneous velocity given by the screw \mathbf{s} and is acted on by a total force and moment given by the wrench \mathbf{W} then the instantaneous power exerted on the body is given by,

$$\text{power} = \mathcal{W}^T \mathbf{s} = \mathbf{M} \cdot \boldsymbol{\omega} + \mathbf{F} \cdot \mathbf{v}.$$

In robotics we can associate a screw to each joint in of the robot. The possible motions about the joint are given by the exponential,

$$g(\theta) = e^{\theta S^0},$$

where θ is the joint parameter, an angle if the joint is revolute. The superscript 0 in the above refers to the position of the joint screw in the home position. The exponential of a matrix is given by the standard formula,

$$e^S = I + S + \frac{1}{2!}S^2 + \frac{1}{3!}S^3 + \dots$$

In general the position of a point \mathbf{p} attached to the end-effector of the robot will be given by,

$$\begin{pmatrix} \mathbf{p}(t) \\ 1 \end{pmatrix} = e^{\theta_1 S_1^0} e^{\theta_2 S_2^0} e^{\theta_3 S_3^0} e^{\theta_4 S_4^0} e^{\theta_5 S_5^0} e^{\theta_6 S_6^0} \begin{pmatrix} \mathbf{p}(0) \\ 1 \end{pmatrix}.$$

Although the home positions of the joint screws S_i^0 are constants the current positions of these joint screws change, the kinematics of these are given by,

$$\begin{aligned} S_1 &= S_1^0 \\ S_2 &= e^{\theta_1 S_1^0} S_2^0 e^{-\theta_1 S_1^0} \\ S_3 &= e^{\theta_1 S_1^0} e^{\theta_2 S_2^0} S_3^0 e^{-\theta_2 S_2^0} e^{-\theta_1 S_1^0} \\ &\vdots \\ S_6 &= e^{\theta_1 S_1^0} e^{\theta_2 S_2^0} \dots e^{\theta_5 S_5^0} S_6^0 e^{-\theta_5 S_5^0} \dots e^{-\theta_1 S_1^0} \end{aligned}$$

So the time derivatives of these screw are,

$$\frac{d}{dt} S_1 = 0$$

since S_1 is constant.

$$\frac{d}{dt} S_2 = \dot{\theta}_1 e^{\theta_1 S_1^0} (S_1^0 S_2^0 - S_2^0 S_1^0) e^{-\theta_1 S_1^0} = \dot{\theta}_1 [S_1, S_2]$$

the last equality follows because S_1^0 commutes with $e^{\theta_1 S_1^0}$.

$$\begin{aligned} \frac{d}{dt} S_3 &= \dot{\theta}_1 e^{\theta_1 S_1^0} S_1^0 e^{\theta_2 S_2^0} S_3^0 e^{-\theta_2 S_2^0} e^{-\theta_1 S_1^0} \\ &\quad - \dot{\theta}_1 e^{\theta_1 S_1^0} e^{\theta_2 S_2^0} S_3^0 e^{-\theta_2 S_2^0} S_1^0 e^{-\theta_1 S_1^0} \\ &\quad + \dot{\theta}_2 e^{\theta_1 S_1^0} e^{\theta_2 S_2^0} S_2^0 S_3^0 e^{-\theta_2 S_2^0} e^{-\theta_1 S_1^0} \\ &\quad - \dot{\theta}_2 e^{\theta_1 S_1^0} e^{\theta_2 S_2^0} S_3^0 S_2^0 e^{-\theta_2 S_2^0} e^{-\theta_1 S_1^0} \\ &= \dot{\theta}_1 [S_1, S_3] + \dot{\theta}_2 [S_2, S_3] \end{aligned}$$

and so forth until,

$$\begin{aligned} \frac{d}{dt} S_6 &= \dot{\theta}_1 [S_1, S_6] + \dot{\theta}_2 [S_2, S_6] \\ &\quad + \dot{\theta}_3 [S_3, S_6] + \dot{\theta}_4 [S_4, S_6] + \dot{\theta}_5 [S_5, S_6]. \end{aligned}$$

In [10, Chap 3.] a formula is given for the acceleration of a point attached to the end-effector of a serial robot. This formula contains an error, so we present a corrected version here.

Differentiating the position of a point on the end-effector gives,

$$\begin{pmatrix} \dot{\mathbf{p}}(t) \\ 0 \end{pmatrix} = \dot{\theta}_1 S_1^0 e^{\theta_1 S_1^0} \dots e^{\theta_6 S_6^0} \begin{pmatrix} \mathbf{p}(0) \\ 1 \end{pmatrix} + \dots + \dot{\theta}_6 e^{\theta_1 S_1^0} \dots S_6^0 e^{\theta_6 S_6^0} \begin{pmatrix} \mathbf{p}(0) \\ 1 \end{pmatrix}$$

and this simplifies in the usual way to,

$$\begin{pmatrix} \dot{\mathbf{p}}(t) \\ 0 \end{pmatrix} = (\dot{\theta}_1 S_1 + \dots + \dot{\theta}_6 S_6) \begin{pmatrix} \mathbf{p}(t) \\ 1 \end{pmatrix}$$

This gives the velocity of the point.

If we differentiate again to get the acceleration we have,

$$\begin{pmatrix} \ddot{\mathbf{p}}(t) \\ 0 \end{pmatrix} = \left\{ \sum_{i=1}^6 \ddot{\theta}_i S_i + \sum_{1 \leq i < j \leq 6} \dot{\theta}_i \dot{\theta}_j [S_i, S_j] + \left(\sum_{i=1}^6 \dot{\theta}_i S_i \right)^2 \right\} \begin{pmatrix} \mathbf{p}(t) \\ 1 \end{pmatrix}$$

We can simplify the above a little by expanding the square term, remembering that S_i and S_j do not generally commute. We get,

$$\begin{pmatrix} \ddot{\mathbf{p}}(t) \\ 0 \end{pmatrix} = \left\{ \sum_{i=1}^6 (\ddot{\theta}_i S_i + \dot{\theta}_i^2 S_i^2) + 2 \sum_{1 \leq i < j \leq 6} \dot{\theta}_i \dot{\theta}_j S_i S_j \right\} \begin{pmatrix} \mathbf{p}(t) \\ 1 \end{pmatrix}$$

The terms $\ddot{\theta}_i S_i + \dot{\theta}_i^2 S_i^2$ are the accelerations would expect from motion about a single joint.

A more comprehensive account of this view of screw theory can be found in [12].

III. FRENET-SERRET MOTION

In [3] Bottema and Roth study a number of ‘special motions’, one of which is the Frenet-Serret motion. Such a motion is determined by a space-curve $\mathbf{p}(t)$. Now in a Frenet-Serret motion a point in the moving body moves along the curve and the coordinate frame in the moving body remains aligned with the tangent \mathbf{t} , normal \mathbf{n} , and binormal \mathbf{b} , of the curve. Using the 4-dimensional representation of $SE(3)$ the motion can be specified as,

$$g(t) = \begin{pmatrix} R(t) & \mathbf{p}(t) \\ 0 & 1 \end{pmatrix}, \quad (1)$$

where $\mathbf{p}(t)$ is the curve and the rotation matrix has the unit vectors \mathbf{t} , \mathbf{n} and \mathbf{b} as columns,

$$R(t) = (\mathbf{t} \mid \mathbf{n} \mid \mathbf{b}). \quad (2)$$

The famous Frenet-Serret relations are,

$$\dot{\mathbf{t}} = v\kappa\mathbf{n}, \quad (3)$$

$$\dot{\mathbf{n}} = -v\kappa\mathbf{t} + v\tau\mathbf{b}, \quad (4)$$

$$\dot{\mathbf{b}} = -v\tau\mathbf{n}, \quad (5)$$

where v , κ and τ are respectively the speed, curvature and torsion of the curve. Our work here will be simplified by introducing the Darboux vector $\boldsymbol{\omega} = v\tau\mathbf{t} + v\kappa\mathbf{b}$ which has the properties that,

$$\dot{\mathbf{t}} = \boldsymbol{\omega} \times \mathbf{t}, \quad \dot{\mathbf{n}} = \boldsymbol{\omega} \times \mathbf{n}, \quad \dot{\mathbf{b}} = \boldsymbol{\omega} \times \mathbf{b}, \quad (6)$$

see [7, §II.4] for example. This means that we can write,

$$\dot{R} = \Omega R, \quad (7)$$

where Ω is the 3×3 anti-symmetric matrix corresponding to $\boldsymbol{\omega}$. Hence we have that,

$$Q = \left(\frac{d}{dt} g \right) g^{-1} = \begin{pmatrix} \Omega & v\mathbf{t} - \boldsymbol{\omega} \times \mathbf{p} \\ 0 & 0 \end{pmatrix}, \quad (8)$$

remember that $\dot{\mathbf{p}} = v\mathbf{t}$.

Using the Frenet-Serret relations (3)–(5) above the derivative of the velocity is,

$$\dot{Q} = \begin{pmatrix} \dot{\Omega} & \dot{v}\mathbf{t} - \dot{\boldsymbol{\omega}} \times \mathbf{p} \\ 0 & 0 \end{pmatrix}. \quad (9)$$

Here, $\dot{\boldsymbol{\omega}} = (\dot{v}\tau + v\dot{\tau})\mathbf{t} + (\dot{v}\kappa + v\dot{\kappa})\mathbf{b}$. In the 6-vector representation this is,

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{v}\mathbf{t} - \dot{\boldsymbol{\omega}} \times \mathbf{p} \end{pmatrix}. \quad (10)$$

In robotics the use of these motions seems to have been first suggested by Wagner and Ravani, see [13]. Their interest was prompted by an arc-welding application where a robot had to follow the intersection of a pair of curved plates keeping the tool aligned with a normal to the intersection curve. They also suggested some applications in computer graphics to blends and sweeping surfaces.

IV. BISHOP'S MOTION

In [2] Bishop gave an alternative method to associate a moving frame to a curve in 3 dimensions. The Bishop frame can also be used to define a special rigid-body motion. A point in the rigid body follows a curve and an orthonormal frame in the body stays aligned with the Bishop frame. Such a motion will be called a 'Bishop's motion'.

There are some applications of the Bishop frame in Computer graphics to thicken curves. The Bishop frame is used because it doesn't 'twist' about the curve, see Fig. 1 (b). In the Frenet-Serret frame the torsion cannot be defined at points where the curvature vanishes, this is not a problem for Bishop's frame. Another feature of the Bishop frame is that it is not unique, however once the initial frame has been selected the frames for the rest of the curve are uniquely determined. Perhaps the first suggestion that these curves might be useful in robotics appears in [8], they refer to these motions as 'rotation minimizing frames'.

The frame equations for Bishop's frame are:

$$\dot{\mathbf{t}} = vk_1\mathbf{n}_1 + vk_2\mathbf{n}_2, \quad (11)$$

$$\dot{\mathbf{n}}_1 = -vk_1\mathbf{t}, \quad (12)$$

$$\dot{\mathbf{n}}_2 = -vk_2\mathbf{t}. \quad (13)$$

As usual v is the speed of the curve and $\dot{\mathbf{p}} = v\mathbf{t}$, where \mathbf{t} is the unit tangent vector. The vectors \mathbf{n}_1 and \mathbf{n}_2 are unit normal vectors and together with the tangent vector \mathbf{t} , they form an orthonormal frame. So for example $\mathbf{n}_1 \times \mathbf{n}_2 = \mathbf{t}$ and so forth. The parameters k_1 and k_2 are curvature-like functions.

In terms of the Frenet-Serret frame the normal vectors \mathbf{n}_1 and \mathbf{n}_2 are given by,

$$\mathbf{n}_1 = \cos \phi \mathbf{n} - \sin \phi \mathbf{b}, \quad (14)$$

$$\mathbf{n}_2 = \sin \phi \mathbf{n} + \cos \phi \mathbf{b}. \quad (15)$$

where the angle ϕ is given by the indefinite integral,

$$\phi = \int v\tau dt. \quad (16)$$

So, as mentioned above, a curve does not uniquely determine a Bishop frame, there is a single rotational freedom in defining the Bishop frame, given by the constant of integration. But if we choose the unit normal vectors \mathbf{n}_1 and \mathbf{n}_2 at $t = 0$ then the Bishop frame for the rest of the curve is unique.

The path in the group determined by a Bishop's motion will be the same as for the Frenet-Serret motion, see (1) above, but now the rotation matrix will be given by,

$$R(t) = (\mathbf{t} | \mathbf{n}_1 | \mathbf{n}_2). \quad (17)$$

The analogue of the Darboux vector for the Bishop's motion is,

$$\mathbf{a} = -vk_2\mathbf{n}_1 + vk_1\mathbf{n}_2. \quad (18)$$

It is simple to verify that,

$$\dot{\mathbf{t}} = \mathbf{a} \times \mathbf{t}, \quad \dot{\mathbf{n}}_1 = \mathbf{a} \times \mathbf{n}_1, \quad \text{and} \quad \dot{\mathbf{n}}_2 = \mathbf{a} \times \mathbf{n}_2.$$

Note that it is possible to show that $\mathbf{a} = v\kappa\mathbf{b}$. The velocity is,

$$Q = \left(\frac{d}{dt} g \right) g^{-1} = \begin{pmatrix} A & v\mathbf{t} - \mathbf{a} \times \mathbf{p} \\ 0 & 0 \end{pmatrix}, \quad (19)$$

where, as usual, capital A represents the 3×3 anti-symmetric matrix corresponding to the vector \mathbf{a} .

Proceeding as in the previous section we can compute the derivative of the velocity screw, as a 6-dimensional vector this is

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{\mathbf{a}} \\ \dot{v}\mathbf{t} - \dot{\mathbf{a}} \times \mathbf{p} \end{pmatrix}. \quad (20)$$

V. ROBOT KINEMATICS AND CONTROL

In this section we relate the desired motions to a couple of common problems in robotics. In general we want to drive the robot so that its end-effector follows the desired path.

Firstly we study the computation of the inverse kinematics along such a path. Let us write \mathbf{z}_6 , for the velocity

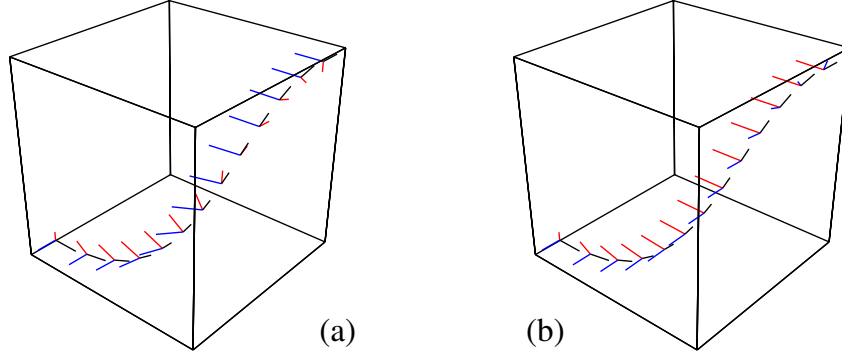


Fig. 1. (a) a Frenet-Serret motion and (b) a Bishop's motion. Both these motions were based on the same cubic spline curve which is approximated by the black tangent lines in either case. The initial frame for the Bishop's motion was chosen to coincide with the initial Frenet-Serret frame. So the frame at the bottom left of each diagram are the same. Notice how the blue binormal line in (a) rotates through 180 degrees as we move along the curve. However, in (b) the blue normal only turns through 90 degrees.

screw of the end-effector of a robot. In terms of the joint variables this velocity screw is,

$$\sum_{i=1}^6 \dot{\theta}_i \mathbf{s}_i = \mathbf{z}_6. \quad (21)$$

This equation shows that the columns of the robot's Jacobian are given by the robot's joint screws \mathbf{s}_i .

Now we can think of the inverse Jacobian of the manipulator as composed of six wrenches $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_6$ such that,

$$\mathcal{W}_i^T \mathbf{s}_j = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}. \quad (22)$$

For many commercially available designs of robot these quantities can be computed symbolically, see [12, §6.7].

Setting the velocity of the robot's end-effector to the desired velocity gives,

$$\dot{\theta}_1 \mathbf{s}_1 + \dot{\theta}_2 \mathbf{s}_2 + \dots + \dot{\theta}_6 \mathbf{s}_6 = \mathbf{q}. \quad (23)$$

Pairing this equation with the wrenches \mathcal{W}_i , the rows of the inverse Jacobian of the robot, yields,

$$\dot{\theta}_i = \mathcal{W}_i^T \mathbf{q}, \quad i = 1, 2, \dots, 6. \quad (24)$$

These differential equations should be straightforward to set up and solve numerically using the Runge-Kutta method for example. Notice that we need to know the starting configuration of the robot but we only need the velocity of the desired motion. Of course, like most methods in robotics, this method will fail near singularities.

As an alternative approach to guiding the robot along a desired path we can design a closed-loop controller for the robot. The scheme here is based on a non-linear feedback control scheme introduced in [9]. The idea of this is to control the end-effector of the robot along a specified path without performing any inverse kinematic computations. The key is to arrange for the feedback to be such that the closed loop dynamics of the system are differential equations in the robot's joint variables that are satisfied by the desired path. This idea is similar to the Passive Velocity Field Control method introduced by Li and Horowitz[5].

However the method presented here is somewhat simpler, since we do not introduce any extra degrees of freedom.

Equation(21) above can be differentiated to give,

$$\sum_{i=1}^6 \ddot{\theta}_i \mathbf{s}_i + \sum_{1 \leq j < k \leq 6} \dot{\theta}_j \dot{\theta}_k [\mathbf{s}_j, \mathbf{s}_k] = \dot{\mathbf{z}}_6. \quad (25)$$

Away from singularities we can pair this equation with the wrenches \mathcal{W}_i , we get,

$$\ddot{\theta}_i + \Gamma_{ijk} \dot{\theta}_j \dot{\theta}_k = \mathcal{W}_i^T \dot{\mathbf{z}}_6, \quad i = 1, 2, \dots, 6 \quad (26)$$

where summing over repeated indices is assumed. The quantities Γ_{ijk} are given by,

$$\Gamma_{ijk} = \begin{cases} \frac{1}{2} \mathcal{W}_i^T [\mathbf{s}_j, \mathbf{s}_k] & j \leq k \\ -\frac{1}{2} \mathcal{W}_i^T [\mathbf{s}_j, \mathbf{s}_k] & j \geq k. \end{cases} \quad (27)$$

If we replace the velocity of the end-effector with the velocity of the desired motion we obtain an equation for the desired motion in the joint space variables of the robot,

$$\ddot{\theta}_i + \Gamma_{ijk} \dot{\theta}_j \dot{\theta}_k = \mathcal{W}_i^T \dot{\mathbf{q}}, \quad i = 1, 2, \dots, 6. \quad (28)$$

Notice, the time derivative of a velocity screw is not exactly an acceleration. If the closed-loop dynamics of the robot take this form then the end-effector will follow the desired motion, (subject to considerations of stability and errors of course).

Depending on whether we want to follow a Frenet-Serret motion of a Bishop's motion we must use $\dot{\mathbf{q}}$ as given by either (10) or (20).

Now the dynamics of the robot can be represented by the equations,

$$A_{ij} \ddot{\theta}_j + B_{ijk} \dot{\theta}_j \dot{\theta}_k = \tau_i, \quad i = 1, 2, \dots, 6 \quad (29)$$

again assuming summation over repeated indices, see [12, Chap. 13] for example. Here A_{ij} is the generalised mass matrix and B_{ijk} represents the coupling and Coriolis terms. The joint torques, which we assume that the control system can apply to robot's motors, are represented by τ_i . For simplicity gravity will be ignored.

Now suppose that our control system measures the joint variables and their rates, the motor torques can be set to,

$$\tau_i = (B_{ijk} - A_{il}\Gamma_{ljk})\dot{\theta}_j\dot{\theta}_k + A_{il}\mathcal{W}_l^T\dot{\mathbf{q}}, \quad i = 1, \dots, 6 \quad (30)$$

This is a non-linear feedback control law. The closed-loop equations of the system are found by substituting for τ_i in the robot dynamics equation (29) above. The result will be the equation for the desired motion (28) above (multiplied by the positive-definite mass matrix A_{ij}).

The development of the control law above is based on the task space equation,

$$\dot{\mathbf{z}}_6 = \dot{\mathbf{q}}. \quad (31)$$

For a stable controller it is probably better to base the method on the equation,

$$\dot{\mathbf{z}}_6 = \dot{\mathbf{q}} + \lambda(\mathbf{z}_6 - \mathbf{q}), \quad (32)$$

where λ is a positive constant. Clearly, $\mathbf{z}_6 = \mathbf{q}$ is a solution to this equation. Moreover, if we assume that $\mathbf{z}_6 = \mathbf{q} + \mathbf{e}$, where \mathbf{e} is an error vector, then the dynamics of the error vector obey,

$$\dot{\mathbf{e}} = -\lambda\mathbf{e}. \quad (33)$$

So the errors in the velocity decrease exponentially.

The joint space version of equation (32) is,

$$\ddot{\theta}_i + \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k = \mathcal{W}_i^T\dot{\mathbf{q}} + \lambda(\mathcal{W}_i^T\mathbf{q} - \dot{\theta}_i), \quad i = 1, \dots, 6 \quad (34)$$

The corresponding closed loop control law will be,

$$\tau_i = (B_{ijk} - A_{il}\Gamma_{ljk})\dot{\theta}_j\dot{\theta}_k + A_{il}\mathcal{W}_l^T\dot{\mathbf{q}} + \lambda A_{il}(\mathcal{W}_l^T\mathbf{q} - \dot{\theta}_l), \quad i = 1, \dots, 6 \quad (35)$$

VI. PROJECTION BASED INTERPOLATION

Recently, two new methods of producing rigid body motions have been proposed, see [1] and [4]. Both these methods rely on projecting affine motions to rigid-body motions. Belta and Kumar's method interpolates the motions directly to produce a curve in the group of affine motions $GA^+(3)$ and then projects these motions to the 'nearest' rigid-body motion. The method described by, Hofer, Pottmann and Ravani is based on paths of points in space. In this method several points on the rigid body are selected and interpolation curves are defined for each of the points. At some particular instant the positions of the interpolated points may not be a rigid transformation of the original points. To produce a rigid body motion a least squares problem is solved to find the positions of the points subject to the constraint that they are rigidly related to the original points. We show here that these methods are essentially the same.

We begin with a number of rigid body motions we wish to interpolate, assume these are given by,

$$\begin{pmatrix} R_i & \mathbf{t}_i \\ 0 & 1 \end{pmatrix}, \quad i = 1, \dots, n.$$

Now choose a number, say k , of points $\mathbf{a}^{(j)}$, at least 4 non-coplanar points according to [4]. The knot-points for the interpolation are then,

$$\mathbf{b}_i^{(j)} = R_i\mathbf{a}^{(j)} + \mathbf{t}_i, \quad i = 1, \dots, n. \quad (36)$$

So we get a set of interpolated curve,

$$\mathbf{p}^{(j)}(t) = \sum_{i=1}^n f_i(t)\mathbf{b}_i^{(j)}, \quad j = 1, \dots, k \quad (37)$$

where $f_i(t)$ are the interpolating functions. In Belta and Kumar's method we would interpolate in the space of matrices to get,

$$\begin{aligned} X(t) &= \begin{pmatrix} M(t) & \mathbf{d}(t) \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^n f_i(t)R_i & \sum_{i=1}^n f_i(t)\mathbf{t}_i \\ 0 & 1 \end{pmatrix}. \end{aligned} \quad (38)$$

Clearly we have,

$$\begin{pmatrix} \mathbf{p}^{(j)}(t) \\ 1 \end{pmatrix} = X(t) \begin{pmatrix} \mathbf{a}^{(j)} \\ 1 \end{pmatrix}, \quad j = 1, \dots, k. \quad (39)$$

In general the matrix $X(t)$ will lie in the group $GA^+(3)$, this is because the 3×3 block $M(t) = \sum_{i=1}^n f_i(t)R_i$ will generally be an element of $GL^+(3)$. However, it may happen that for some particular values of t , $\det(M(t)) = 0$, some conditions for this are studied in [1]. The corresponding problem for points, that is values of t where the set of points $\mathbf{p}^{(j)}(t)$ may be coplanar, does not seem to have been anticipated in [4].

In Hofer *et al*'s method we find the rigid motion at some time t by minimising the quantity $\sum_{j=1}^k |\mathbf{p}^{(j)}(t) - R\mathbf{a}^{(j)} - \mathbf{t}|^2$. That is we seek a rotation matrix R and a translation vector \mathbf{t} such that this expression is minimal. A solution to this problem can be found in [11] for example. The first step is to choose the origin of coordinates to be at the centroid of the $\mathbf{a}^{(j)}$ points, this is so that $\sum_{j=1}^k \mathbf{a}^{(j)} = \mathbf{0}$. In these coordinates the minimal translation is given by,

$$\mathbf{t} = \frac{1}{k} \sum_{j=1}^k \mathbf{p}^{(j)}. \quad (40)$$

To simplify notation we have dropped the explicit dependence on t , but we are assuming some definite value of t . Now using the fact that $\mathbf{p}^{(j)} = M\mathbf{a}^{(j)} + \mathbf{d}$ and the fact that the sum of the $\mathbf{a}^{(j)}$ s is zero we have,

$$\mathbf{t} = \mathbf{d}. \quad (41)$$

The minimal rotation can be found from the polar decomposition of the matrix,

$$P = \sum_{j=1}^k \mathbf{p}^{(j)} \left(\mathbf{a}^{(j)} \right)^T. \quad (42)$$

This simplifies to,

$$P = M \sum_{j=1}^k \mathbf{a}^{(j)} \left(\mathbf{a}^{(j)} \right)^T. \quad (43)$$

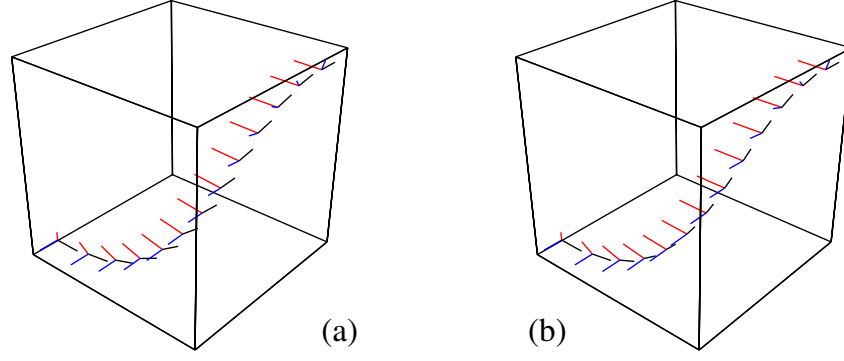


Fig. 2. (a) a cubic projected interpolated motion based on the end-points of the Bishop's motion given in Fig. 1(b) and (b) the Bishop's motion from Fig. 1(b) for comparison.

The solution is the rotation R such that $P = RK$ where K is symmetric. Almost any numerical methods text will give details on computing this and its relation to the singular value decomposition of P .

Belta and Kumar's method is really a family of methods depending on a positive definite symmetric matrix W . The method requires us to find the polar decomposition of the matrix product MW . So if we identify W with the matrix,

$$W = \sum_{j=1}^k \mathbf{a}^{(j)} \left(\mathbf{a}^{(j)} \right)^T, \quad (44)$$

the curves in $SE(3)$ given by the two approaches will be identical.

On the other hand, Belta and Kumar give particular attention to the case where $W = I$ the identity matrix. Now if we choose the points $\mathbf{a}^{(j)}$ carefully we can make the matrix $\sum_{j=1}^k \mathbf{a}^{(j)} \left(\mathbf{a}^{(j)} \right)^T = I$. A suitable choice here would be the vertices of a regular tetrahedron,

$$\mathbf{a}^{(1)} = \begin{pmatrix} 2/\sqrt{6} \\ 0 \\ -1/2\sqrt{3} \end{pmatrix}, \mathbf{a}^{(2)} = \begin{pmatrix} -1/\sqrt{6} \\ 1/\sqrt{2} \\ -1/2\sqrt{3} \end{pmatrix}, \\ \mathbf{a}^{(3)} = \begin{pmatrix} -1/\sqrt{6} \\ -1/\sqrt{2} \\ -1/2\sqrt{3} \end{pmatrix}, \text{ and } \mathbf{a}^{(4)} = \begin{pmatrix} 0 \\ 0 \\ \sqrt{3}/2 \end{pmatrix}. \quad (45)$$

In fact, by Schur's lemma, any collection of points $\mathbf{a}^{(j)}$, symmetrical with respect to some finite subgroup of rotation (crystallographic point groups), will give a matrix W that is a scalar multiple of the identity. An example of such a motion is shown in Fig. 2(a).

VII. CONCLUSIONS

The Frenet-Serret motion has been known in the mechanisms community for many years but does not seem to have found much acceptance in robotics, except [13]. The Bishop's motion does not seem to be familiar to the mechanisms community. Either of these motions could be the basis of an interpolation scheme. This would involve finding a spline curve with specified Frenet frames at its end-points. Hence, these simple motions may be useful in practical circumstances.

The purpose of section V was really to show that for applications to robotics we need to know the velocity, and its derivative, of the rigid body motion. The control method presented was rather simple and not to be taken too seriously. However, the idea of designing the control system of a robot in such a way that it follows a desired trajectory using a knowledge of the geometry of the desired path rather than just employing inverse kinematics is a worthwhile goal.

This implies that interpolation techniques which result in motions whose velocities are difficult to calculate are less desirable for robotic applications. At present it is difficult to see how to compute the velocity of a motion given by the schemes studied in section VI.

REFERENCES

- [1] C. Belta and V. Kumar, "An SVD-projection method for interpolation on $SE(3)$ ", *IEEE Trans. Robotics and Automation*, vol. 18, pp. 334-345, 2002.
- [2] R.L. Bishop, "There is more than one way to frame a curve", *Am. Math. Monthly*, vol. 82 pp. 246-251, 1975.
- [3] O. Bottema and B. Roth, *Theoretical Kinematics*, Dover Publications, New York, 1990.
- [4] M. Hofer, H. Pottmann and B. Ravani, "From curve design algorithms to the design of rigid body motions". *The Visual Computer* vol. 20, pp. 279-297, 2004.
- [5] P.Y. Li and R. Horowitz, "Passive velocity field control of mechanical manipulators", *IEEE Trans. Robotics and Automation*, vol. 15, pp. 751-763, 1999.
- [6] D. Marsh, *Applied Geometry for Computer Graphics and CAD* 2nd ed., Springer Verlag, London, 2005.
- [7] B. O'Neill, *Elementary Differential Geometry*, Academic Press, New York, 1966.
- [8] R. Ravani and A. Meghadari, "Spatial rational motions based on rational Frenet-Serret Curves", *Proc. IEEE Intl. Conf. Systems, Man and Cybernetics* pp. 4456-4461, 2004.
- [9] J.M. Selig and A.I. Ovseevitch, "Manipulating robots along helical trajectories", *Robotica* vol. 14 pp. 261-267, 1996.
- [10] J.M. Selig, (ed.) *Geometric foundations of robotics* World Scientific, Singapore, 2000.
- [11] J.M. Selig, "Three Problems in Robotics", *Proc. Inst. Mech. Eng. part C: J. Mechanical Engineering Science*, vol 216, pp.73-80, 2002.
- [12] J.M. Selig, *Geometric Fundamentals of Robotics*. Springer Verlag, New York, 2005.
- [13] M.G. Wagner and B. Ravani, "Curves with rational Frenet-Serret motion", *Computer Aided Geometric Design* vol. 15 pp. 79-101, 1997.