Interpolated Rigid-Body Motions and Robotics

J.M. Selig

London South Bank University

and

Yuanqing Wu

Shanghai Jiaotong University.

Introduction

Interpolation of rigid motions important in robotics and graphics.

Many suggestions so far: Finite screw motions, spline rotations and translations independently, minimum jerk motions and several others.

Here introduce a couple more possibilities and show that two existing suggestions give the same results.

The Velocity Screw

Motion of a rigid body given by a sequence of rigid transformations $g(t)$, curve in the group $SE(3)$. Screws are velocities (Lie algebra elements),

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

The Velocity Screw (cont.)

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

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

Ω is angular velocity as 3×3 anti-symmetric matrix, $\mathbf{v}$ is linear velocity of the motion.

Product of Exponentials

Kinematics of serial robot can be written as product of exponentials. For a single joint,

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

θ, the joint parameter, an angle for a revolute joint. Superscript 0 refers to home position of the joint screw. Exponential of a matrix is given by the standard formula,

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

Product of Exponentials (cont.)

Position of point $\mathbf{p}$ attached to end-effector given by,

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

Velocity of Point on End-effector

Velocity of point $\mathbf{p}$ given by differentiating,

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

Note: S_i is current position of joint screw.
Standard result on Robot Jacobians.

Acceleration of Point on End-effecto

Differentiating again gives,

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

Error in Selig(2000), corrected here.

Inverse Kinematics

Knowing the velocity screw and acceleration are important for robot control. Here give application to inverse kinematics. Suppose want end-effector to follow given motion $g(t)$. Can turn inverse kinematics problem along the path into set of differential equation and use standard numerical methods to find joint angles.

Using 6×6 representation of the group the velocity screw of the end-effector is,

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

Inverse Kinematics(cont.)

where $\dot{\boldsymbol{\theta}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_6 \end{pmatrix}$ and the Jacobian J has rows,

$$J = (\mathbf{s}_1|\mathbf{s}_1|\cdots|\mathbf{s}_6).$$

Here s_i is the 6-vector corresponding to the 4×4 matrix S_i above.

Inverse Kinematics(cont.)

Setting the end-effector velocity to the velocity of the curve and inverting the Jacobian get a system of 1st order ODEs,

$$\dot{\boldsymbol{\theta}} = J^{-1}\mathrm{s},$$

where s is the 6-vector corresponding to $\left(\frac{d}{dt}g(t)\right)g^{-1}(t)$.

For most common robots J^{-1} can be computed symbolically.

Frenet-Serret Motion

Frenet-Serret motion has been suggested before in robotics (Wagner and Ravani 1997). Attractive because based on curves in 3D so easy to visualise.

Point on end-effector follows specified 3D curve, orientation of the end-effector fixed with respect to Frenet frame of the curve. That is, the tangent, principal normal and binormal vectors are fixed in the end-effector.

Frenet-Serret Motion (cont.)

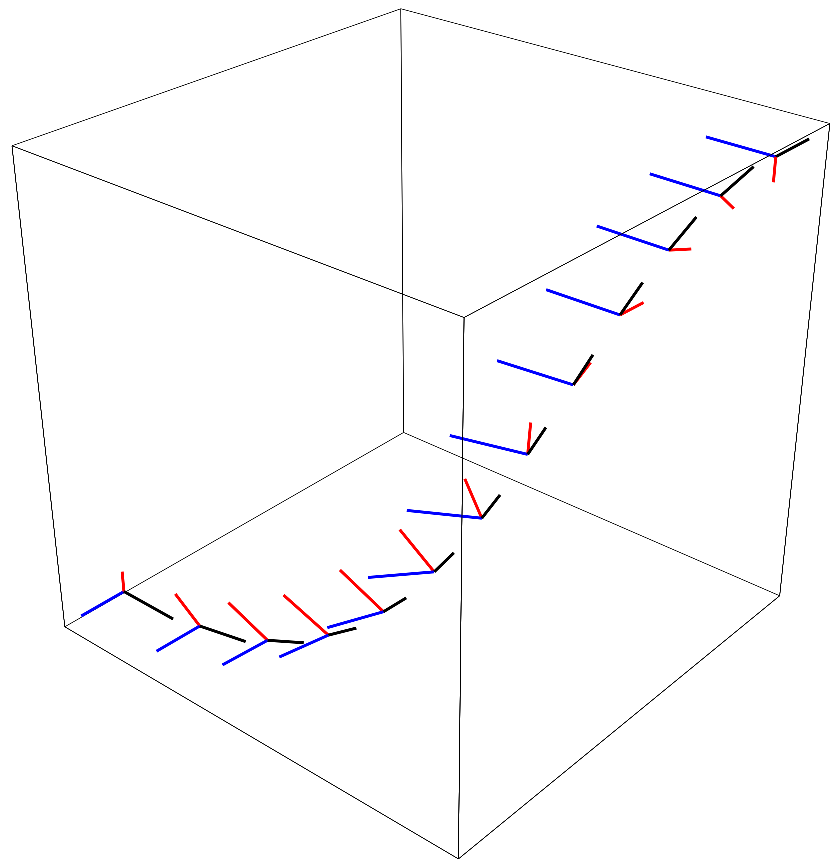
Suppose the curve in 3D is given by $\mathbf{p}(t)$ then the velocity screw of the Frenet-Serret motion is,

$$\mathbf{s} = \begin{pmatrix} \boldsymbol{\omega} \\ v\mathbf{t} - \boldsymbol{\omega} \times \mathbf{p} \end{pmatrix},$$

where $\boldsymbol{\omega} = v\tau\mathbf{t} + v\kappa\mathbf{b}$ is the Darboux vector, $v,\ \kappa,\ \tau$ are the speed, curvature and torsion and $\mathbf{t},\ \mathbf{b}$ are the tangent and binormal to the curve $\mathbf{p}$.

Can also find second derivative, see paper.

Frenet-Serret Motion Example

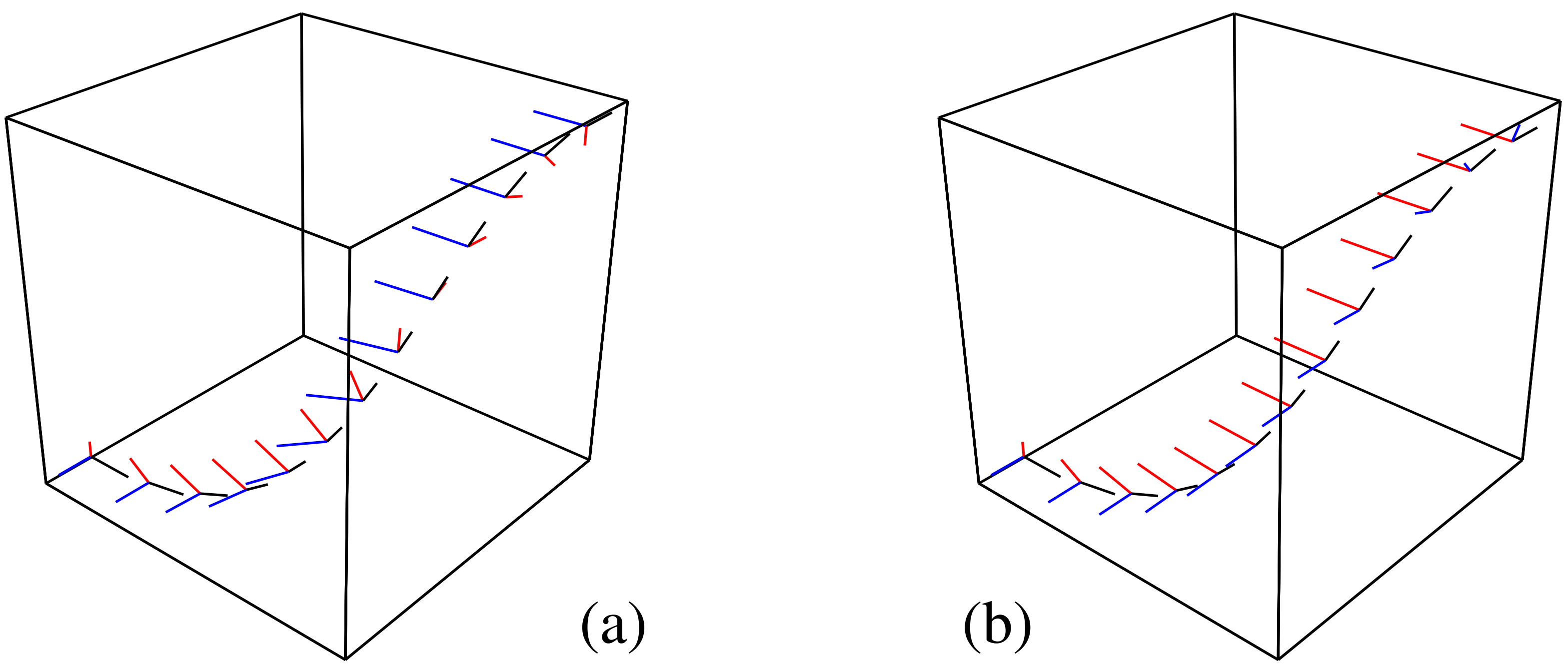


A Frenet-Serret motion based on a cubic spline. Black lines—tangent vectors, blue—principal normals and red—binormals.

Bishop's Motion

"More than one way to frame a curve"—Bishop (1975)

Same as above but keep orientation fixed w.r.t Bishop's frame.



(a) (b)

Projection Based Interpolation

Two recent approaches.

- Belta and Kumar (2002). Interpolate matrices, project to $SE(3)$, polar decomposition.
- Hofer, Pottmann and Ravani (2004). Interpolate points on a rigid body, project to rigid body motion using least squares.

Projection Based Interpolation (cont

Begin with a number of rigid body motions to interpolate, assume these are,

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

Choose k points $\mathbf{a}^{(j)}$. Knot-points for interpolation then, $\mathbf{b}_i^{(j)} = R_i\mathbf{a}^{(j)} + \mathbf{t}_i, \quad i = 1, \ldots, n$. Get interpolated curves, ($f_i(t)$ interpolating functions)

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

Belta and Kumar's method

Interpolate the matrices,

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

Clearly,

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

Hofer *et al*'s method

Find the rigid motion at time t by minimising the quantity $\sum_{j=1}^{k} |\mathbf{p}^{(j)}(t) - R\mathbf{a}^{(j)} - \mathbf{t}|^2$. Can show that minimal translation is given by, $\mathbf{t} = \mathbf{d}$, centroid of the points. Minimal rotation can be found from the polar decomposition of the matrix,

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

The solution is the rotation R such that $P = RK$ where K is symmetric.

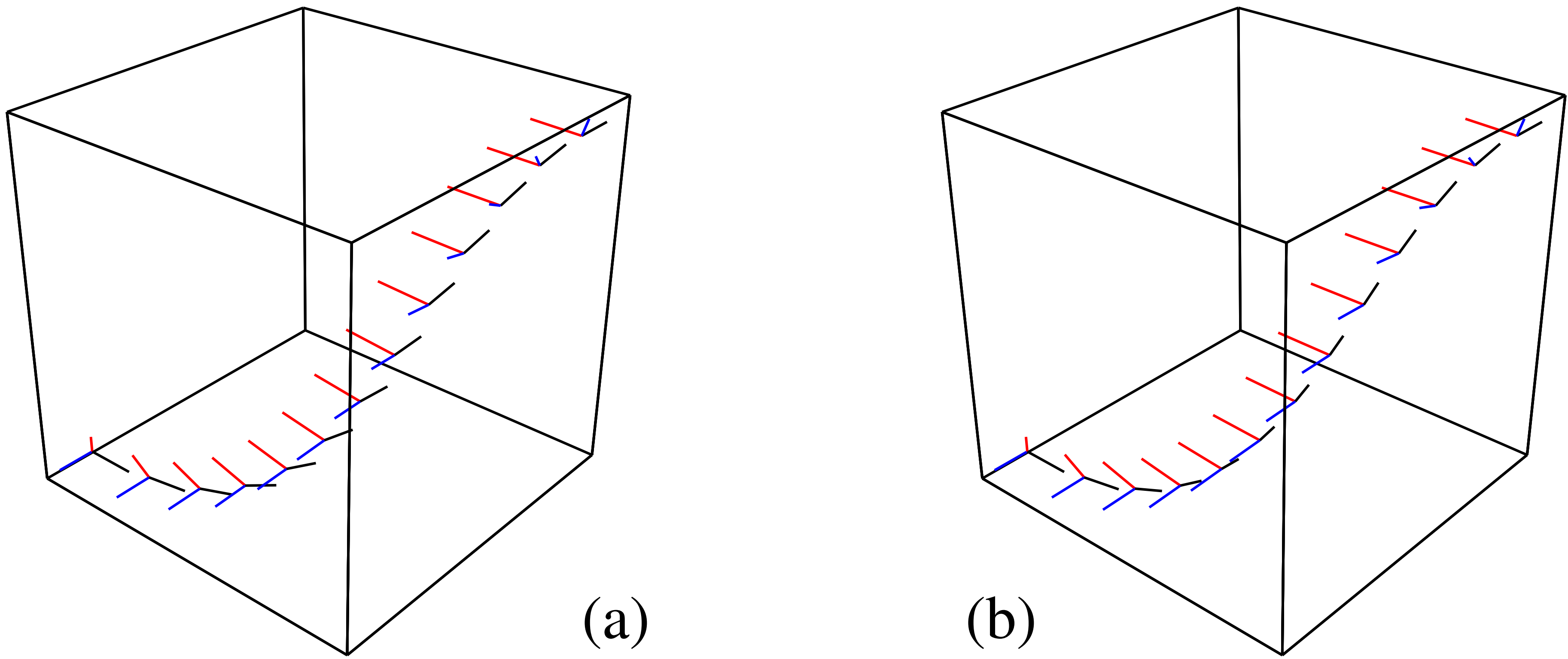
Belta and Kumar's method (cont.)

Project $X(t)$ to $SE(3)$, translation is $\mathbf{t} = \mathbf{d}$ as above and rotation comes from polar decomposition of MW where W is any positive definite symmetric 3×3 matrix. If we choose,

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

then methods are identical. In particular can have $W = I_3$, identity matrix if $\mathbf{a}^{(j)}$ are chosen to be symmetrical.

Projection *vs* Bishop's motion



(a) a cubic projected interpolated motion based on the end-points of the Bishop's motion (b).

Conclusions

- Frenet-Serret and Bishop's motion simple and probably useful.
- Good idea to design Robot's control system to follow paths rather than point-to-point control.
- Cannot find the velocity screw for a projection based motion (yet!).