Thoughts on Quaternion Interpolation

Noel Hughes

1/10/2007

The type of interpolation to be used depends on several factors:

1) What data are available: The quaternion at each time step; the body rotational rates?
2) What is/is not to be preserved at sampling points.
3) What the expected vehicle behavior is: Are rotational rates high relative to sampling rate?

- note on notation: i's refer to times; j's refer to elements of quaternion, vector, etc.; k's refer to polynomial coefficients.

Quaternion definition:
Euler's theorem:
The rotational relationship between any two coordinate frames can be described by an axis of rotation that is fixed in both coordinate frames and a total rotation angle.

Rotation axis: $\mathbf{e} = e_1 \;\; e_2 \;\; e_3$ (assumed unit magnitude)
Rotation angle: ϕ

$$Q = \begin{array}{l} q_1 = e_1\sin(\phi/2) \\ q_2 = e_2\sin(\phi/2) \\ q_3 = e_3\sin(\phi/2) \\ q_4 = \cos(\phi/2) \end{array}$$

<u>Interpolation of Quaternion elements</u>

Given data: time ordered sets of

1) attitude described by the quaternion from reference frame to body frame, $Q(t_i)$, and
2) rotational rate in the body frame, $\boldsymbol{\omega}(t_i) = \omega_1 \;\; \omega_2 \;\; \omega_3$

<u>Two Point Methods</u>:

using normalized time: $t_N = (t - t_i)/(t_{i+1} - t_i)$, $\quad t_i \le t \le t_{i+1}$

<u>Linear:</u>

$$Q(t) = Q(t_i) + t_N * (Q(t_{i+1})-Q(t_i))$$

or

$$q_1(t) = q_1(t_i) + t_N * (q_1(t_{i+1})-q_1(t_i))$$

$$q_2(t) = q_2(t_i) + t_N * (q_2(t_{i+1}) - q_2(t_i))$$
$$q_3(t) = q_3(t_i) + t_N * (q_3(t_{i+1}) - q_3(t_i))$$
$$q_4(t) = q_4(t_i) + t_N * (q_4(t_{i+1}) - q_4(t_i))$$

Advantages:

Simple to implement

Disadvantages:

Will have instantaneous rate changes at sampling points.

<u>3rd order interpolation:</u>

Assume:

$$q_1(t) = a_{10} + a_{11}t_N + a_{12}\, t_N^2 + a_{13}\, t_N^3$$
$$q_2(t) = a_{20} + a_{21}t_N + a_{22}\, t_N^2 + a_{23}\, t_N^3$$
$$q_3(t) = a_{30} + a_{31}t_N + a_{32}\, t_N^2 + a_{33}\, t_N^3$$
$$q_4(t) = a_{40} + a_{41}t_N + a_{42}\, t_N^2 + a_{43}\, t_N^3$$

to solve for the elements of the a matrix:

setting $t_N = 0.0$

a) $q_j(t_i) = a_{j0}$

setting $t_N = 1.0$

b) $q_j(t_2) = \sum_{k=0,3} a_{jk}$

using

$$\dot{Q} = ½\, Q\omega$$

setting $t_N = 0.0$

c) $\dot{q}_1(t_i) = a_{11} = ½(q_2(t_i)\,\omega_3(t_i) - q_3(t_i)\,\omega_2(t_i) + q_4(t_i)\,\omega_1(t_i))$

(similarly for $\dot{q}_2(t_i)$, $\dot{q}_3(t_i)$, $\dot{q}_4(t_i)$)

setting $t_N = 1.0$

d) $\dot{q}_1(t_{i+1}) = \sum_{j=0,3} j a_{1j} = ½(q_2(t_{i+1})\,\omega_3(t_{i+1}) - q_3(t_{i+1})\,\omega_2(t_{i+1}) + q_4(t_{i+1})\,\omega_1(t_{i+1}))$

(similarly for $\dot{q}_2(t_{i+1})$, $\dot{q}_3(t_{i+1})$, $\dot{q}_4(t_{i+1})$)

Equations a-d provide four equations for each of the quaternion elements from which the four coefficients for each element can be derived.

Advantages:

Preserves both postion and rate at end points – no sudden rate discontinuity.

Disadvantages:

More complex to implement.

Higher Order Interpolation

N+1 things (position, rate, acceleration, etc. at a point) can be preserved, where N is the order of the polynomial.

If order < number of points, position will not be preserved at sampling points.

The higher the order, the greater risk of unacceptable behavior; there may be peaks/valleys corresponding to the order used.

Other types of Interpolation

Interpolation between attitudes can also be performed by calculating the Eigen rotation axis and the total angle between the two attitudes and calculating a delta quaternion from the first using a portion of the angle between at each step. The angle can be a linear or non linear function of time,

$\mathbf{e} = e_1 \;\; e_2 \;\; e_3$	rotation axis
$\phi(t) = \phi_T * f(t)$	rotation angle at time t

Where:	$f(t)$	a function of time, $0.0 \le f(t) \le 1.0$
	ϕ_T	total angle between two attitudes

then

$Q_\Delta = e_1\sin(\phi/2) \;\; e_2\sin(\phi/2) \;\; e_3\sin(\phi/2) \;\; \cos(\phi/2)$

and

$Q(t) = Q\, Q_\Delta$

This is called a “slerp” - spherical linear interpolation – in several literature sources, notably the paper by Ken Shoemake (although, if a non-linear time function is used, this is no longer linear). I used this method to generate attitude maneuvers on another program for the last decade or so.

This formulation ignores rotational rates and their discontinuities at the endpoints; we can devise methods of smoothing to deal with this, if necessary.

For All Interpolation Schemes:

If rotation rate is larger than $\boldsymbol{\pi}/T_s$, where T_s is the sampling interval, interpolation cannot be done unambiguously, given position only data. If rates are given, assumptions about intervening behavior can be made and interpolation performed.

Quaternion Integration:

If the inertia matrix, **I**, is available:

1) calculate rotation rate derivatives in the body frame from:

   $\dot{\omega} = \mathbf{I}^{-1} (\boldsymbol{\tau} - \omega \times \mathbf{I}\, \omega)$ where $\boldsymbol{\tau}$ = external torque

   integrate to generate **ω**

   $\omega = \int \dot{\omega}\, dt$

2) Calculate quaternion derivatives from:

   (after appending "0" (a zero) to **ω** to make it a fourtuple and compatible with quaternion multiplication)

$$\dot{Q} = \tfrac{1}{2} Q\omega$$

$$Q = \int \dot{Q}\, dt$$