Derivatives of quaternion spline interpolation function for multibody dynamics

Václav Houdek[1], Olivier Verlinden[2], Michal Hajžman[3]

[1] Faculty of Applied Sciences
University of West Bohemia
Technická 8, 301 00 Pilsen
Czech Republic
vaclavh@kme.zcu.cz

[2] Faculty of Engineering
University of Mons
Place du Parc 20, 7000 Mons
Belgium
olivier.verlinden@umons.ac.be

[3] Faculty of Applied Sciences
University of West Bohemia
Technická 8, 301 00 Pilsen
Czech Republic
mhajzman@kme.zcu.cz

EXTENDED ABSTRACT

Interpolation plays an important role in nowadays world. By interpolating data, we save time and money in general. The main areas where interpolation is applied are robotics, automotive, medicine and biology. One of the possible basis splines for interpolation are B-splines, which are also used in Computer Aided Geometric Design (CAGD) due to their smoothness and locality properties [1]. In this work we consider problems of kinematics, which are, in many cases, characterized by a set of non-liner algebraic equations that have to be assembled and solved in each time step. The computational procedure could be time consuming and therefore it is reasonable to develop suitable methods to overcome such difficulties. Moreover, parametrization of finite rotations is an essential issue in multi-body kinematics and dynamics and therefore the concept of quaternions is employed to describe body rotations in this work. In other words, the main idea is to solve the kinematics prior to the dynamics and to pre-compute the rotation parameters of a car wheel support and then use the interpolation of rotations in the framework of more complex computational tasks. The pre-computation of the rotation parameters leads to a look-up table.

1 Spline, B-spline and quaternion spline interpolation

Generally, *spline interpolation* is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. Spline interpolation fits low-degree polynomials to small subsets of the values. Spline interpolation provides lower interpolation error [3] and also avoids the problem of Runge's phenomenon. *B-spline* is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. Any spline function of given degree can be expressed as a linear combination of B-splines of that degree [4]. The base functions $B_i^k(u)$'s are defined by the recurrence relation [5]

$$B_i^k(u) = \frac{u-u_i}{u_{i+k-1}-u_i}B_i^{k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}}B_{i+1}^{k-1}(u), \quad \text{where} \quad B_i^1(u) = \begin{cases} 1 & u_i \le u \le u_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Spline of order k $\tilde{B}_i^k(u)$ belongs to C^{k-2} class, so that we use k = 4 to reach continuity at the acceleration level. For the simplicity of the text we assume $\tilde{B}_i^k(u) \equiv \tilde{B}_i^4(u) \equiv \tilde{B}_i(u)$). The B-spline quaternion curve with a cumulative basis form is formulated as [5]

$$Q(u) = q_{-1}^{\tilde{B}_0^k(u)} \prod_{i=0}^{n+1} (q_{i-1}^{-1} q_i)^{\tilde{B}_i^k(u)} = q_{-1}^{\tilde{B}_0^k(u)} \prod_{i=0}^{n+1} \exp(\omega_i \tilde{B}_i^k(u)), \quad \text{where} \quad \tilde{B}_i(u) = \sum_{j=i}^{n+1} B_i(u), \quad \text{and} \quad \omega_i = \log(q_{i-1}^{-1} q_i) \tag{2}$$

which interpolates a given sequence of given unit quaternions Q_i (i = 0, 1,...,n) in terms of driving parameter u. The control unit quaternions (control points) q_i have to be precomputed from the sequence of given unit quaternions [5].

2 Derivatives of quaternion B-spline interpolation

With the definition 2 we can define the first derivative as

$$Q'(u) \quad = \quad Q(u) \cdot (\tilde{B}_0'(u) + (\sum_{1}^{n+1} \omega_i \tilde{B}_i'(u))), \quad \text{where} \quad \tilde{B}_i'(u) = \frac{k-1}{u_{i+k-1}-u_i} B_i^{k-1}(u), \tag{3}$$

where $\tilde{B}_i'(t)$ is given by the B-spline differentiation formula [2]. With the first derivative it is now very easy to develop second derivative, by applying the chain rule we obtain

$$Q''(u) \quad = \quad Q'(u) \cdot (\tilde{B}_0'(u) + \sum_{i=1}^{n+1} (\omega_i \tilde{B}_i'(u))) + Q(u) \cdot (\tilde{B}_0''(u) + \sum_{i=1}^{n+1} (\omega_i \tilde{B}_i''(u))), \text{ and } \tilde{B}_i''(u) = \frac{k-1}{u_{i+k-1}-u_i} \cdot \frac{k-2}{u_{i+k-2}-u_i} B_i^{k-2}(u). \tag{4}$$

3 Results

Vertical coordinate of the wheel support (body K) was chosen as the driving parameter (denoted by u resp. $\dot{u}$ and $\ddot{u}$ for first and second time derivative), so that the wheel support travels from u = -0.19 to 0.2 m. The look-up table consisted of 100 rows and was generated so that the relative angle between two successive rotations in the table was constant θ = 0.003 rad. The driving parameter u was imposed to compute the body configuration $\mathbf{q}_K(u) = [x_K\ y_K\ z_K\ \phi_K\ \theta_K\ \psi_K]$, and $\mathbf{q}'_K(u) = \frac{\partial \mathbf{q}_K(u)}{\partial u}$ and $\mathbf{q}''_K(u) = \frac{\partial^2 \mathbf{q}_K(u)}{\partial u^2}$. We analysed and compared (with exact values from kinematic solver) two quantities, vector $\boldsymbol{\delta}_K(u)$ and vector $\frac{\partial \boldsymbol{\delta}_K(u)}{\partial u}$ representing the error of angular velocity and acceleration, respectively. Vector $\boldsymbol{\delta}_K(u)$ is defined with the help of the angular velocity of body K $\boldsymbol{\omega}_K(u)$, so that vectors $\boldsymbol{\delta}_K(u)$ and $\frac{\partial \boldsymbol{\delta}_K(u)}{\partial u}$ are defined as

$$\boldsymbol{\omega}_K = \underbrace{\begin{pmatrix} \cos(\psi_K)\cos(\theta_K) & -\sin(\psi_K) & 0 \\ \sin(\psi_K)\cos(\theta_K) & \cos(\psi_K) & 0 \\ -\sin(\psi_K) & 0 & 1 \end{pmatrix}}_{\mathbf{A}_K} \cdot \begin{Bmatrix} \phi'_K \\ \theta'_K \\ \psi'_K \end{Bmatrix} \cdot \dot{u} = \boldsymbol{\delta}_K \cdot \dot{u}, \quad \text{and} \quad \frac{\partial \boldsymbol{\delta}_K}{\partial u} = \mathbf{A}'_K \cdot \begin{Bmatrix} \phi'_K \\ \theta'_K \\ \psi'_K \end{Bmatrix} + \mathbf{A}_K \cdot \begin{Bmatrix} \phi''_K \\ \theta''_K \\ \psi''_K \end{Bmatrix}. \tag{5}$$

The exact orientation (of the body K) will be compared to the interpolated one (K_{int}). We compare the evolution of the norm of the vector difference between the $\boldsymbol{\delta}$ vectors, and the norm of the vector difference between the $\frac{\partial \boldsymbol{\delta}_K}{\partial u}$ vectors, whereas

$$\delta_E(u) = \frac{||\boldsymbol{\delta}_K(u) - \boldsymbol{\delta}_{Kint}(u)||}{||\boldsymbol{\delta}^u_K(u)||}, \quad \text{and} \quad d\delta_E(u) = \frac{||\frac{\partial \boldsymbol{\delta}_K}{\partial u}(u) - \frac{\partial \boldsymbol{\delta}_{Kint}}{\partial u}(u)||}{||\frac{\partial \boldsymbol{\delta}_K}{\partial u}(u)||}. \tag{6}$$

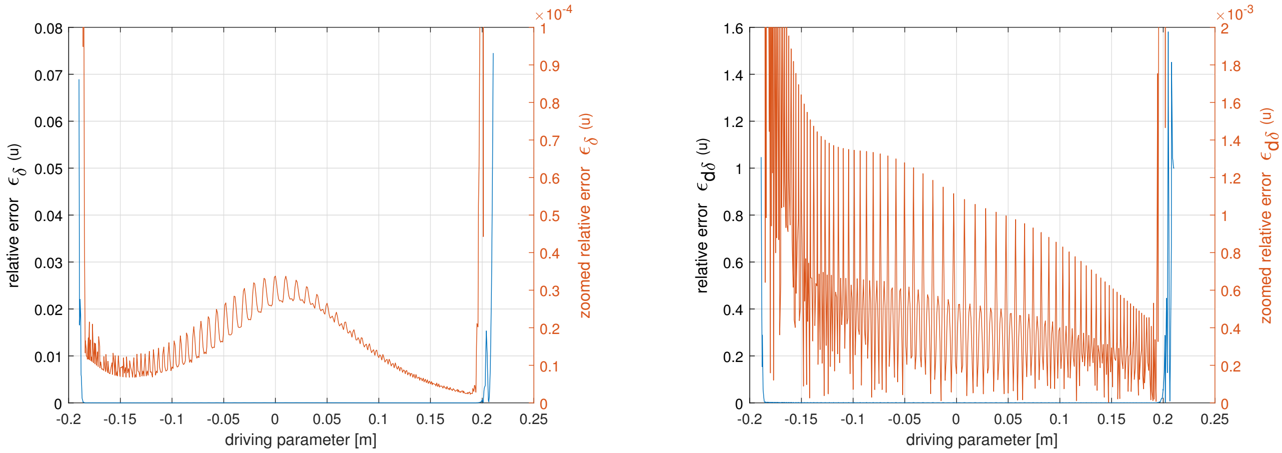

Figure 1: $\delta_E(u)$ error and $d\delta_E(u)$ error

4 Conclusion

The performed interpolation shows that $d\delta_E(u)$ shows higher error than $\delta_E(u)$ which is logical result, because computation of $\boldsymbol{\delta}_K(u)$ is based on more points. The errors are lower than 0.3%, which is sufficient for the dynamical simulations. The highest errors are achieved in the beginning and the end of the interpolated interval, this is caused by bad boundary conditions and by the inner properties of the method. One of the biggest advantage of this interpolation is that it is based on the orientations, not on angular velocities or accelerations as it is in case of Hermite spline interpolation.

Acknowledgments

This research work was supported by the project SGS-2023-008.

References

[1] Lim, Choong-Gyoo. A universal parametrization in B-spline curve and surface interpolation. Computer Aided Geometric Design. 1999, 16(5), 407-422.

[2] Kim, M.-J., Kim M.-S., Shin S. Y. A C2-continuous B-spline quaternion curve interpolating a given sequence of solid orientations. In: Proceedings Computer Animation'95 [online]. IEEE Comput. Soc. Press, 1995, pp. 72-81

[3] J. H. Ahlberg, E. N. Nilson, J. L. Walsh. The theory of Splines and Their Applications. Academic Press. New York and London, 1967.

[4] G. D. Knott. Interpolating Cubic Splines Birkhäuser Boston, 2000.

[5] M. J. Kim, M. S. Kim, and S. Y. Shin. A c2-continous b-spline quaternion curve interpolating a given sequence of solid orientations. Computer Animation '95, IEEE CS Press, 1995.