Iterative refinement implementation on semirecursive algorithm for vehicle dynamics

Yongjun Pan[1] and Yansong He[1]

[1] *School of Automotive Engineering, Chongqing University, {yongjun.pan,hys68}@cqu.edu.cn*

Iterative refinement is a well-established technique to improve the accuracy of the solution to a system of linear differential equations [1]. In recent times, a variety of efficient alternative algorithms have been developed based on the basic iterative refinement algorithm. For instance, the Chebyshev algorithm accelerates the process without loss of numerical stability. These ideas can be applied to the dynamic simulation of multibody systems. Specifically, we consider a semirecursive algorithm whereby two velocity transformations that reduce the number of coordinates are carried out. This formulation, also referred to as the double-step semirecursive formulation, has already been analyzed in a number of papers, covering different aspects of its efficiency such as basic linear algebra subroutines [2], sparse matrix implementation [3], parallel computing, rigid rod approximation [4, 5], sensitivity analysis [6, 7] and flexible multibody application [8]. In this paper, iterative refinement concepts are applied to the above multibody formulation in order to improve its (already high) computational efficiency.

Let us consider a closed-loop, rigid-body system whose topology can be described by a spanning tree after the necessary joints have been temporarily removed according to the cut joint method, and whose configuration is represented by a set of n dependent relative coordinates $\mathbf{z}$. The equations of motion of the multibody system can be written as[4, 9]:

$$\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\mathbf{M}^{\Sigma}\mathbf{R}_{\mathrm{d}}\mathbf{R}_{\mathrm{z}}\ddot{\mathbf{z}}^{\mathrm{i}} = \mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\mathbf{R}_{\mathrm{d}}^{\mathrm{T}}\left(\mathbf{Q}^{\Sigma} - \mathbf{T}^{\mathrm{T}}\bar{\mathbf{M}}\mathbf{D}\right) \tag{1}$$

$$\mathbf{D} \equiv \mathbf{T}\mathbf{R}_{\mathrm{d}}\begin{Bmatrix} -\left(\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{d}}\right)^{-1}\left(\dot{\mathbf{\Phi}}_{\mathbf{z}}\dot{\mathbf{z}}\right) \\ \mathbf{0} \end{Bmatrix} + \mathbf{T}\dot{\mathbf{R}}_{\mathrm{d}}\dot{\mathbf{z}} \tag{2}$$

where $\mathbf{T} \in \mathbb{R}^{6n\times 6n}$ is the so-called path matrix, which represents the topology of the open-loop multibody system; matrices $\mathbf{R}_{\mathrm{d}} \in \mathbb{R}^{6n\times n}$ and $\mathbf{R}_{\mathrm{z}} \in \mathbb{R}^{n\times f}$ are the first and second velocity transformations; $\bar{\mathbf{M}} \in \mathbb{R}^{6n\times 6n}$ is the inertia matrix; $\mathbf{M}^{\Sigma} \in \mathbb{R}^{6n\times 6n}$ and $\mathbf{Q}^{\Sigma} \in \mathbb{R}^{6n}$ are the accumulated inertia matrix and applied force vector; $\mathbf{\Phi}_{\mathbf{z}} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of the loop-closure constraints; $\mathbf{\Phi}_{\mathbf{z}}^{\mathrm{d}} \in \mathbb{R}^{m\times m}$ contains the columns of the Jacobian that correspond to dependent coordinates; $\mathbf{D} \in \mathbb{R}^{6n}$ contains the absolute accelerations corresponding to null independent relative accelerations.

The classic 4th-order Runge-Kutta scheme, which consists of four function evaluations, is used to solve the equations of motion. Experience has shown that this integrator is a good tradeoff between computational efficiency and ease of implementation. Iterative refinement can be applied to the first and third function evaluations after reusing the generalized mass matrix factorization. First, the initial guess of the solution (independent relative accelerations) is produced. Second, the iterative refinement process is carried out on the basis of the initial guess and the generalized mass matrix factorization. This consists of three substeps: residual calculation, solution increment calculation and solution update. Third and last, an appropriate termination criteria is set up to control the refinement process.

By introducing the initial guess $(\ddot{\mathbf{z}}^{\mathrm{i}})^0$, and the termination criteria where the initial weighted error ε is set to be a constant more than 1, the iterative refinement process can be implemented numerically. The inputs of the process are $\bar{\mathbf{M}}^{\Sigma}$, $\mathbf{R}_{\mathrm{z}}$, $\hat{\mathbf{F}}$ and the $\mathbf{L}\mathbf{L}^{\mathrm{T}}$ matrix factorization from the previous evaluation ($\mathbf{R}_{\mathrm{z}}^{\mathrm{T}}\bar{\mathbf{M}}^{\Sigma}\mathbf{R}_{\mathrm{z}} \cong \mathbf{L}\mathbf{L}^{\mathrm{T}}$). Then the following steps are taken:

- Solve $\mathbf{R}_z^{\mathrm{T}} \bar{\mathbf{M}}^{\Sigma} \mathbf{R}_z \, \ddot{\mathbf{z}}^{\mathrm{i}} = \hat{\mathbf{F}}$
- While $\varepsilon \geq 1$
  - Compute residual $\mathbf{r}^k = \mathbf{R}_z^{\mathrm{T}} \bar{\mathbf{M}}^{\Sigma} \mathbf{R}_z \, (\ddot{\mathbf{z}}^{\mathrm{i}})^k - \hat{\mathbf{F}}$ (initial $k = 0$)
  - Solve $\left(\mathbf{L}\mathbf{L}^{\mathrm{T}}\right) \Delta(\ddot{\mathbf{z}}^{\mathrm{i}})^{k+1} = \mathbf{r}^k$
  - Update $(\ddot{\mathbf{z}}^{\mathrm{i}})^{k+1} = (\ddot{\mathbf{z}}^{\mathrm{i}})^k - \Delta(\ddot{\mathbf{z}}^{\mathrm{i}})^{k+1}$
  - Compute error $\varepsilon = \sqrt{\frac{1}{n} \sum\limits_{k=1}^{n} \left(\frac{\mathbf{e}_k}{\mathbf{w}_k} \right)^2}$
  - Increase $k = k + 1$

where $\mathbf{e}_k = (\ddot{\mathbf{z}}^{\mathrm{i}})^k - (\ddot{\mathbf{z}}^{\mathrm{i}})^{k-1}$, and $\mathbf{w}_k = c_1 \left| (\ddot{\mathbf{z}}^{\mathrm{i}})^k \right| + c_2$. c_1 and c_2 represent the relative and absolute tolerance, respectively. These parameters can be different for each type of variables, such as linear translations and rotations. The weighted error norm reaches convergence when $\varepsilon < 1$.

In order to investigate the accuracy and computational efficiency of the presented iterative refinement algorithm, a medium-size 16DOF sedan vehicle model is simulated here with different time steps. Further, a large-size 40DOF semitrailer truck model is simulated to investigate how the size of the vehicle system affects the efficiency. Results show efficiency gains of 2.7% and 9.5% for the sedan vehicle and semitrailer truck, respectively, along with high accuracy. Further, the algorithm is more efficient for large-size multibody systems. Overall, an iterative refinement algorithm has been presented in the context of efficient time integration schemes for the dynamic simulation of medium-large multibody systems.

References

[1] H. J. Bowdler, R. S. Martin, G. Peters, and J. H. Wilkinson, “Solution of real and complex systems of linear equations,” *Numerische Mathematik*, vol. 8, no. 3, pp. 217–234, 1966.

[2] M. González, F. González, D. Dopico, and A. Luaces, “On the effect of linear algebra implementations in real-time multibody system dynamics,” *Computational Mechanics*, vol. 41, no. 4, pp. 607–615, 2007.

[3] A. F. Hidalgo and J. García de Jalón, “Real-time dynamic simulations of large road vehicles using dense, sparse, and parallelization techniques,” *J. Comput. Nonlinear Dynam*, vol. 10, no. 3, p. 031005, 2015.

[4] J. García de Jalón, E. Álvarez, F. de Ribera, I. Rodríguez, and F. Funes, “A fast and simple semi-recursive formulation for multi-rigid-body systems,” in *Advances in Computational Multibody Systems* (J. Ambrósio, ed.), vol. 2 of *Computational Methods in Applied Sciences*, Chapter 1, pp. 1–23, Springer Netherlands, 2005.

[5] Y. Pan, A. Callejo, J. L. Bueno, R. A. Wehage, and J. García de Jalón, “Efficient and accurate modeling of rigid rods,” *Multibody Syst. Dyn.*, vol. 40, no. 1, pp. 23–42, 2017.

[6] A. Callejo and J. García de Jalón, “Vehicle suspension identification via algorithmic computation of state and design sensitivities,” *J. Mech. Des*, vol. 137, no. 2, p. 021403, 2015.

[7] A. Callejo, J. García de Jalón, P. Luque, and D. A. Mántaras, “Sensitivity-based, multi-objective design of vehicle suspension systems,” *J. Comput. Nonlinear Dynam*, vol. 10, no. 3, p. 031008, 2015.

[8] F. J. Funes and J. García de Jalón, “An efficient dynamic formulation for solving rigid and flexible multibody systems based on semirecursive method and implicit integration,” *J. Comput. Nonlinear Dynam*, vol. 11, no. 5, p. 051001, 2016.

[9] J. García de Jalón, A. Callejo, and A. F. Hidalgo, “Efficient solution of Maggi’s equations,” *J. Comput. Nonlinear Dynam*, vol. 7, no. 2, p. 021003, 2012.