ECCOMAS Thematic Conference on Multibody Dynamics
June 19 - 22, 2017, Prague, Czech Republic

Use of analytical derivatives in an optimal control algorithm for the residual elimination problem of gait

Francisco Mouzo[1], Urbano Lugrís[1], Daniel Dopico[1], Benjamin Fregly[2], and Javier Cuadrado[1]

[1]Laboratorio de Ingeniería Mecánica
University of La Coruña
Mendizábal s/n, 15403 Ferrol, Spain
[francisco.mouzo, ulugris, ddopico]@udc.es,
javicuad@cdf.udc.es

[2]Computational Neuromechanics Lab
University of Florida
Gainesville, FL, 32611, USA
fregly@ufl.edu

Abstract

Optimal control is a well-known tool in the biomechanics field for gait analysis and prediction [1]. When solving an optimal control problem, first and second derivatives of the objective function and constraints (which may include the equations of motion in differential form) with respect to the design variables, states and controls are needed [2]. It is common to obtain first derivatives numerically by means of finite difference formulas, and to estimate the Hessian to avoid the calculation of second derivatives, which is very time consuming. Alternatively, it is possible to obtain symbolic expressions for the derivatives or use automatic differentiation [2]. In this work, the problem of analytically obtaining the first derivatives of the objective function and constraints is addressed for an optimal control problem that seeks to track an experimentally measured gait motion while making the simulated motion dynamically consistent. Results and efficiency are compared when using numerical and analytical first derivatives, respectively. For second derivatives, a Hessian approximation method is maintained in both cases.

Experimental walking data were collected from a healthy adult male, 34 years old, mass 85 kg, and height 1.82 m. All test procedures were approved by the local institutional review board, and the subject gave informed consent. The subject walked on a walkway possessing two embedded force plates (AMTI, AccuGait sampling at 100 Hz), and his motion was captured by 12 optical infrared cameras (Natural Point, OptiTrack FLEX:V100 also sampling at 100 Hz) that computed the position of 37 optical markers (red dots in Fig. 1).

A three-dimensional multi-body dynamic walking model of the subject was created by the authors [3], and is illustrated in Fig. 1. The model consisted of 18 anatomical segments: pelvis (base body), torso, neck, head, and two hindfeet, forefeet, shanks, thighs, arms, forearms and hands. The segments were linked by ideal spherical joints, thereby defining a model possessing 57 degrees of freedom (6 for the base body plus 51 for the joints). The computational model was defined using 228 mixed (174 natural + 54 angular) coordinates. Details about the treatment of captured data can be found in [3].

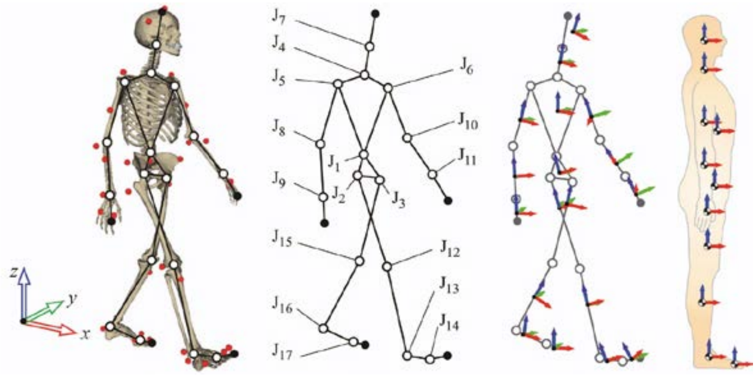

Figure 1. Human multibody model.

Given the experimental marker motion and force plate data, a direct collocation optimal control problem was formulated to find the dynamically consistent model motion that best fit the marker motion data. The optimal control problem possessed the following characteristics. The states were the generalized positions and velocities $\mathbf{z}$ and $\dot{\mathbf{z}}$ for the 57 independent degrees of freedom, and the controls were the associated generalized accelerations $\ddot{\mathbf{z}}$ and the torques $\boldsymbol{\tau}$ acting at the 51 controlled joint degrees of freedom. The objective function minimized the time integral over a complete gait cycle of the squared deviations between the generalized positions, velocities, and accelerations obtained from the model and those obtained from an inverse kinematic analysis, plus the squared deviations between the joint torques obtained from the model and those obtained from an inverse dynamic analysis. The dynamic constraints were trivial, as the generalized accelerations were used as controls. Dynamic consistency was imposed as a path constraint, forcing the residual forces and torques on the base body to be zero. This scheme using inverse dynamics increases the size of the problem, as the accelerations must be included as controls, but proves to be more robust than the forward dynamics alternative.

The equations of motion were provided using the so-called matrix-R formulation [4], which leads to a system in the minimum number of 57 coordinates. Using this formulation, the inverse dynamic equations that provided the generalized forces were:

$$\mathbf{Q}_{ID} = \mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} - \mathbf{R}^{\mathrm{T}}(\mathbf{Q} - \mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{z}}) = \mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\ddot{\mathbf{z}} - \overline{\mathbf{Q}} \tag{1}$$

where **R** is the matrix that projects the independent set of coordinates onto the dependent set, **M** is the mass matrix in dependent coordinates, and **Q** is the generalized force vector in dependent coordinates.

For the optimal control problem to be solved, the partial derivatives of the path constraints with respect to the states, $\mathbf{z}$ and $\dot{\mathbf{z}}$ and controls $\ddot{\mathbf{z}}$ and $\boldsymbol{\tau}$ were necessary. The derivatives with respect to the torques were trivial. The complexity arises in the derivatives of equation (1) with respect to $\mathbf{z}$, $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$, which are outlined using the following equations:

$$\frac{\partial \mathbf{Q}_{ID}}{\partial \mathbf{z}} = \left(\frac{\partial \mathbf{R}^{\mathrm{T}}}{\partial \mathbf{z}}\mathbf{M}\mathbf{R} + \mathbf{R}^{\mathrm{T}}\mathbf{M}\frac{\partial \mathbf{R}}{\partial \mathbf{z}}\right)\ddot{\mathbf{z}} - \frac{\partial \overline{\mathbf{Q}}}{\partial \mathbf{z}} \tag{2}$$

$$\frac{\partial \mathbf{Q}_{ID}}{\partial \dot{\mathbf{z}}} = -\frac{\partial \overline{\mathbf{Q}}}{\partial \dot{\mathbf{z}}} \tag{3}$$

$$\frac{\partial \mathbf{Q}_{ID}}{\partial \ddot{\mathbf{z}}} = \mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R} \tag{4}$$

The development of each of these partial derivative terms leads to extremely complex expressions involving tensor products that can be found in detail in [5]. Equations (1) through (4) were implemented in the software package MBSLIM [6], written in FORTRAN. For solving the optimal control problem, the commercial software GPOPS-II [7] was used.

Table 1. Results for the initial mesh.

Derivatives method	*Iterations until convergence*	*Time (s)*	*Time per iteration (s)*	*Objective*
GPOPS-II Sparse Central Diff.	16	399	24.94	1.084e-2
GPOPS-II Sparse Forward Diff.	18	225	12.5	1.14e-2
MBSLIM	17	123	7.24	1.033e-2

Results, summarized in Table 1, show that the three methods lead to almost the same solution in a similar number of iterations, but the use of analytical derivatives provides slightly better accuracy and much better efficiency.

Acknowledgments

The support of the Spanish Ministry of Economy and Competitiveness (MINECO) under projects DPI2015-65959-C3-1-R (cofinanced by the European Union through the EFRD program) and DPI2016-81005-P is greatly acknowledged.

References

[1] A.J. van den Bogert, D. Blana, D. Heinrich. Implicit methods for efficient musculoskeletal simulation and optimal control. *Procedia IUTAM*, 2:297-316, 2011.

[2] J.T. Betts. *Practical methods for optimal control using nonlinear programming*. SIAM, Philadelphia, 2001.

[3] U. Lugris, J. Carlin, A. Luaces, J. Cuadrado. Gait analysis system for spinal cord injured subjects assisted by active orthoses and crutches. *Journal of Multi-body Dynamics*, 227(4):363-374, 2013.

[4] J. Garcia de Jalon, E. Bayo. *Kinematic and dynamic simulation of multibody systems*. Springer-Verlag, New York, 1994.

[5] D. Dopico, Y. Zhu, A. Sandu, C. Sandu. Direct and adjoint sensitivity analysis of ordinary differential equation multibody formulations. *Journal of Computational and Nonlinear Dynamics*, 10(1):011012, 2015.

[6] MBSLIM multibody simulation library: `http://lim.ii.udc.es/MBSLIM/`.

[7] M.A Patterson, A.V. Rao. GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software*, 41(1):1-37, 2014.