SpineBushing Algorithm Notes/Application
Nur Adila Faruk Senan

This folder contains

- The C++ code for the SpineBushingForce function - I did my best to comment it as thoroughly as possible, so hopefully, you'll be able to understand everything. It is set up to compute forces and moments assuming that the experimental stiffness matrix, K^E (which you are planning on using as input), was determined using

  1. relative motion between the two vertebras measured in the frame of reference fixed to the lower body.
  2. changes in forces and moments acting at the Centers of Mass (CoMs) of the 2 bodies with force components given in the frame of reference attached to the lower body, moment components given in the Dual Euler basis, expressed in the frame of reference attached to the lower body.

- The SpineBushing plugin which you can just insert into the OpenSim 'plugins' folder, and use like a normal user-created plugin if you don't want to go through the hassle of building the plugin in C++ (... and trust that I know what I'm doing) :) I compiled it in Visual Studio Express 2010 using CMake2.8 for OpenSim version 2.4.0. To use it,

  Start OpenSim → tools → plugins → SpineBushing.

  Only then will it show up in the rest of your loaded models. If you are running OpenSim from the command window, then loading a plugin involves adding in something along the lines of

  −L ... `/plugins/SpineBushing`

  to the end of your command line.

Inputs

The SpineBushing takes in the following inputs:

- `body_1` and `body_2`.
  These specify the lower and upper vertebras respectively and correspond to the `parent` and `child` bodies.

- `stiffness_row1`, `stiffness_row2`, ... , `stiffness_row5`, `stiffness_row6`
  Together, these constitute the input stiffness matrix,

$$\mathsf{K} = \begin{bmatrix} \texttt{stiffness\_row1} \\ \dots \\ \texttt{stiffness\_row6} \end{bmatrix} = \begin{bmatrix} \mathsf{K}_1 & \mathsf{K}_3 \\ \mathsf{K}_3^T & \mathsf{K}_2 \end{bmatrix}.$$

  Here,

  - K_1 relates the change in (1-2-3 sequence) Euler angles to moments and has units of Nm/rad,
  - K_2 relates the change in relative position between the vertebral centers of mass/geometry to moments and has units of N/m.
  - K_3 relates the change in relative position between the vertebral centers of mass/geometry to forces and has units of N/rad.

- `reportForcesAtCoM` and `reportForcesInBodyFrames`.
  These two *do not* affect the computations. They're just for force reporting and give you the options of reporting the forces/moment acting at the body origins or the body centers of mass, and to report these in either the ground frame or the body frames affixed to each body.

  The default setting is true (reports forces/moments at centers of mass, and in the body frames) so you don't have to specify them in your code if that's what you're using. Otherwise, set whichever option you want to `false`.

Notice that, unlike the existing bushing function, I did away with the bushing frame altogether since I realized that they were unnecessary in most cases. Specifically, for most spine models,

1. The joint frame `orientation` and `orientation_in_parent` are both (0 0 0).

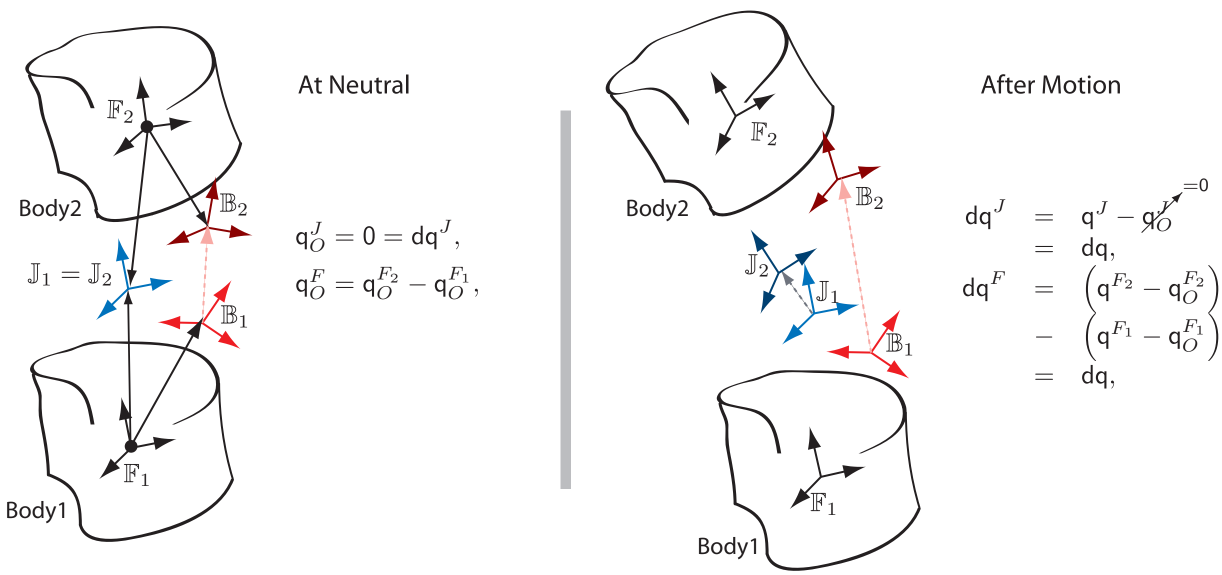
   This lets us the relative position/orientation between the joint frames to compute the change in motion between the centers of mass. That is, since joint frames are initially coincident,

$$\mathsf{q}^J(t=0) = 0,$$

   the relative motion between the joint frames is also the *change in relative motion* between the joint frames,

$$\mathsf{dq}^J = \mathsf{q}^J,$$

   where q^J and dq^J are both generalized displacement vectors (three rotations first, then three translations). Now, we know that the motion of the bodies are specified by rigid body dynamics so all points rotate by the same amount about the joints. Hence, the *change in relative motion* between the joint frames dq^j is also the same as the relative motion between the joint frames, dq (see the figure below).



In the figure, the body centers of geometry are situated at the body origins frames ($\mathbb{F}_1$ and $\mathbb{F}_2$), and the joint orientation and orientation_in_parent both equal zero. This allows us to use the relative motion between the joint frames ($\mathbb{J}_1$ and $\mathbb{J}_2$) to specify the change in relative motion between the centers of geometry.

If $\mathbb{J}_1$ and $\mathbb{J}_2$ need to be oriented differently, then you would add in bushing frames $\mathbb{B}_1$ and $\mathbb{B}_2$, oriented, and positioned appropriately. Be sure to use dq^B, not q^B though.

2. K^E is computed using forces/moments exerted at the centers of mass of the vertebral bodies [1] Hence, given `dq`, and OpenSim's `getBodyMassCenterStation` function (to get the center of mass locations of the two bodies). we compute the generalized force vector f acting on the *center of mass* of the upper body as

$$\mathsf{f} = -\mathsf{K}^E \mathsf{dq}.$$

Implicit in the equation above is the assumption that the bodies experience zero SpineBushing forces and moments in the neutral posture.

Items 1 and 2 eliminate the necessity for bushing frames altogether since we can

- use the relative motion between the joint frames to define the *change* in relative motion (from neutral) between the two bodies.
- use OpenSim's `getBodyMassCenterStation` to get the points about which to apply the SpineBushing forces/moments.

If you are using a model with `orientation` $\neq (0, 0, 0)$, `orientation_in_parent` $\neq (0, 0, 0)$ and/or K^E measured using some other landmark point on the vertebras, then you'll need to incorporate SpineBushing frames and change some of the computation.

I toyed with the idea of including more options, such as the ability to specify the frames of reference of the coordinates, and forces/moments used to measure the stiffness matrix but decided that instead of having all those `if`/`else` statements in the code, it'd be easier to edit the algorithm itself depending on the data you're using. Specifically, go through the `computeForce` function in the `SpineBushing.cpp` file, and edit the variables appropriately.

Miscellaneous

Extensive details for just the compilation/building plugin (i.e. logistics part) of this are given on the

- OpenSim website: https://simtk.org/home/opensim
- OpenSim Developer's guide: https://simtk.org/project/xml/downloads.xml?group_id=91
- OpenSim forum: https://simtk.org/forum/forum.php?forum_id=282
- Musculoskeletal Modeler's Kitchen: https://simtk.org/home/modelerskitchen (see, especially, the wiki page)

There are a lot of other good tips and tricks posted on these sites/documents so they're worth visiting. If you actually take the time to go through and really understand all the included APIExamples included, you'll have enough under your belt to code like a pro. (Trust me. It's worth the investment. It's what I would do if I could go back in time and do it all over.)

Michael Sherman, (who wrote the original "LinearBushing" code for SimTK) is a really, really helpful and knowledgeable person when it comes to the underlying SimTK framework. His documenation is also very comprehensive so if you're interested in learning more about the SimTK aspect of things, I strongly encourage going through the examples posted (https://simtk.org/home/simtk, and accompanying sub-components).

[1] See, e.g. Crisco et al. [2007], Gardner-Morse and Stokes [2004], Panjabi et al. [1976], Stokes et al. [2002], Stokes and Iatridis [2005]).These studies computed K^E using the forces/moments measured at the vertebral centers of geometry, but as there is no data with regards to the vertebral centers of mass, we assumed that the vertebral centers of mass $\approx$ vertebral centers of geometry.

References

J. J. Crisco, L. Fujita, and D. Spenciner. The dynamic flexion/extension properties of the lumbar spine in vitro using a novel pendulum system. *Journal of Biomechanics*, 40(12):2767–2773, Jan 2007. doi: 10.1016/j.jbiomech.2006.12.013.

M. G. Gardner-Morse and I. A. F. Stokes. Structural behavior of the human lumbar spinal motion segments. *Journal of Biomechanics*, 37(2):205–212, 2004. doi: 10.1016/j.jbiomech.2003.10.003.

M. M. Panjabi, R. A. Brand Jr., and A. A. White III. Three-dimensional flexibility and stiffness properties of the human thoracic spine. *Journal of Biomechanics*, 9(4):185–192, 1976. doi: 10.1016/0021-9290(76)90003-8.

I. A. F. Stokes and J. C. Iatridis. *Basic Orthopaedic Biomechanics and Mechano-Biology*, chapter Biomechanics of the Spine, pages 529–561. Lippincott Williams & Wilkins, Philadelphia, third edition, 2005. Edited by Mow, V. C. and Huiskes, R.

I. A. F. Stokes, M. G. Gardner-Morse, D. Churchill, and J. P. Laible. Measurement of a spinal motion segment stiffness matrix. *Journal of Biomechanics*, 35(4):517–521, 2002.