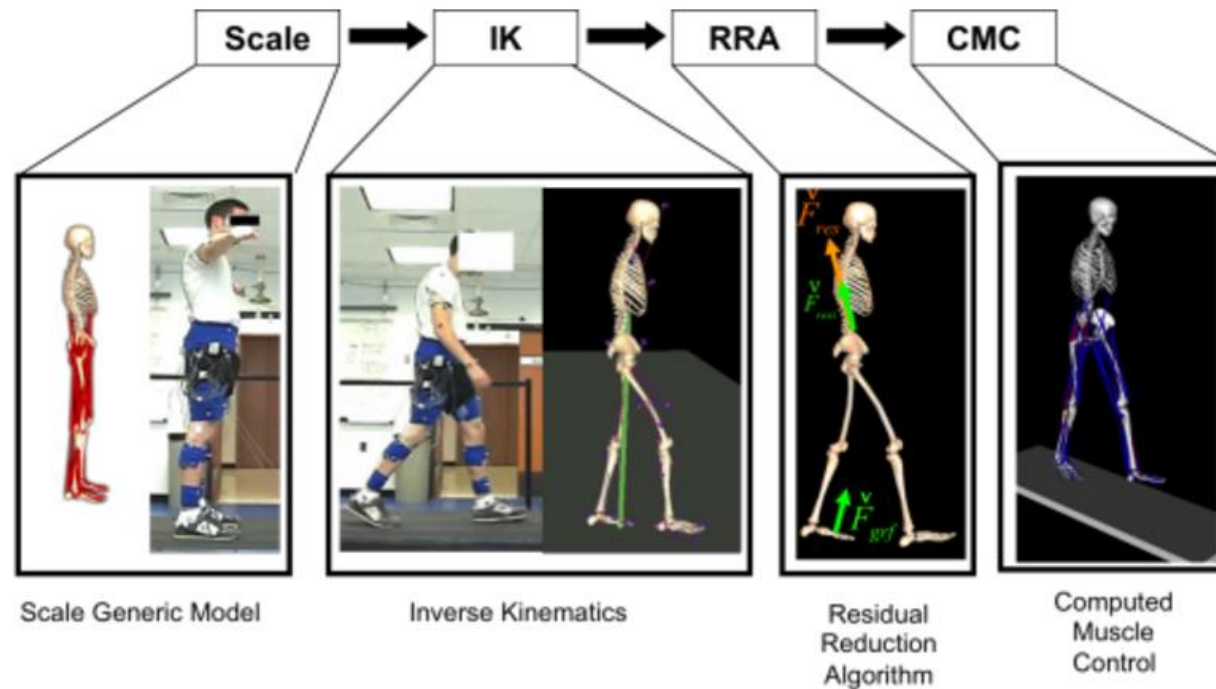




Musculoskeletal modeling

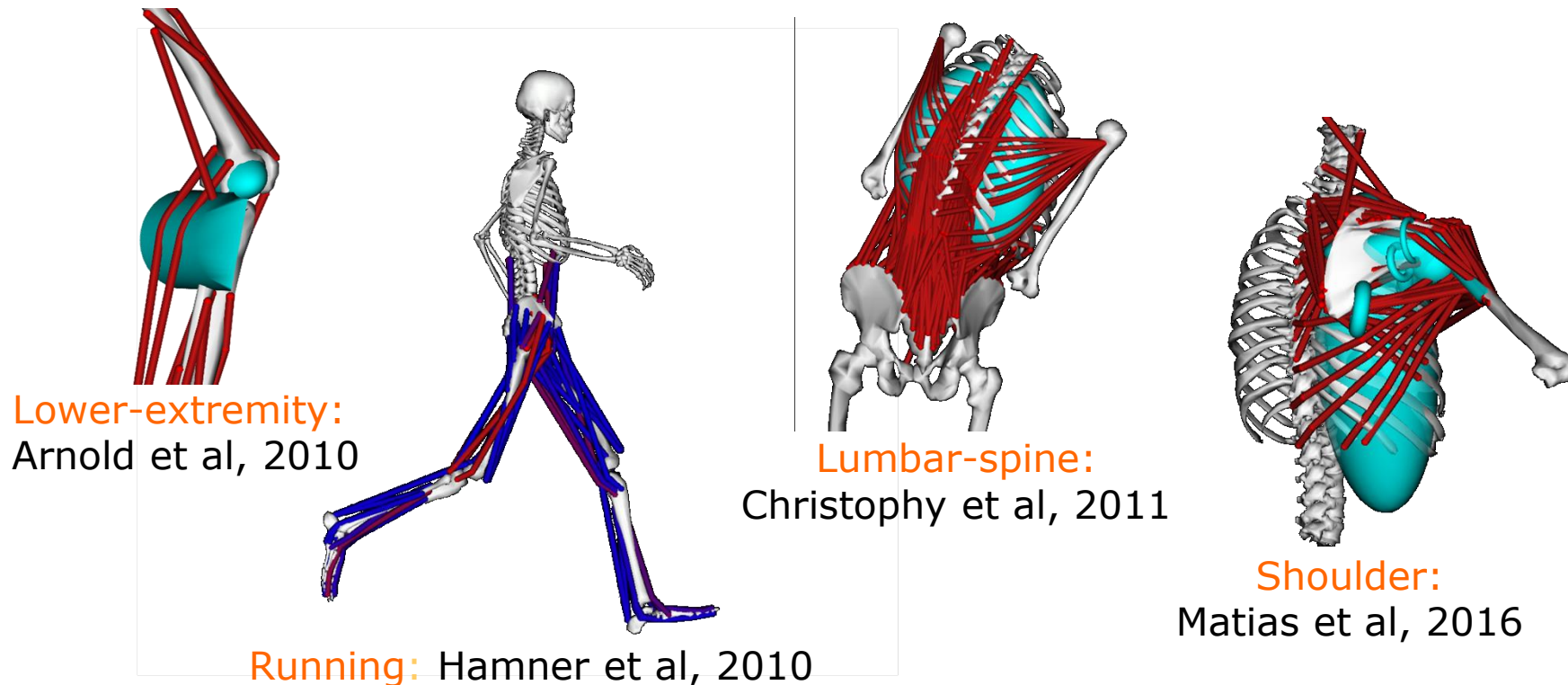
OpenSim workflow

OpenSim enables us to build, exchange, and analyze computer models of the musculoskeletal system and dynamic simulations of movement.

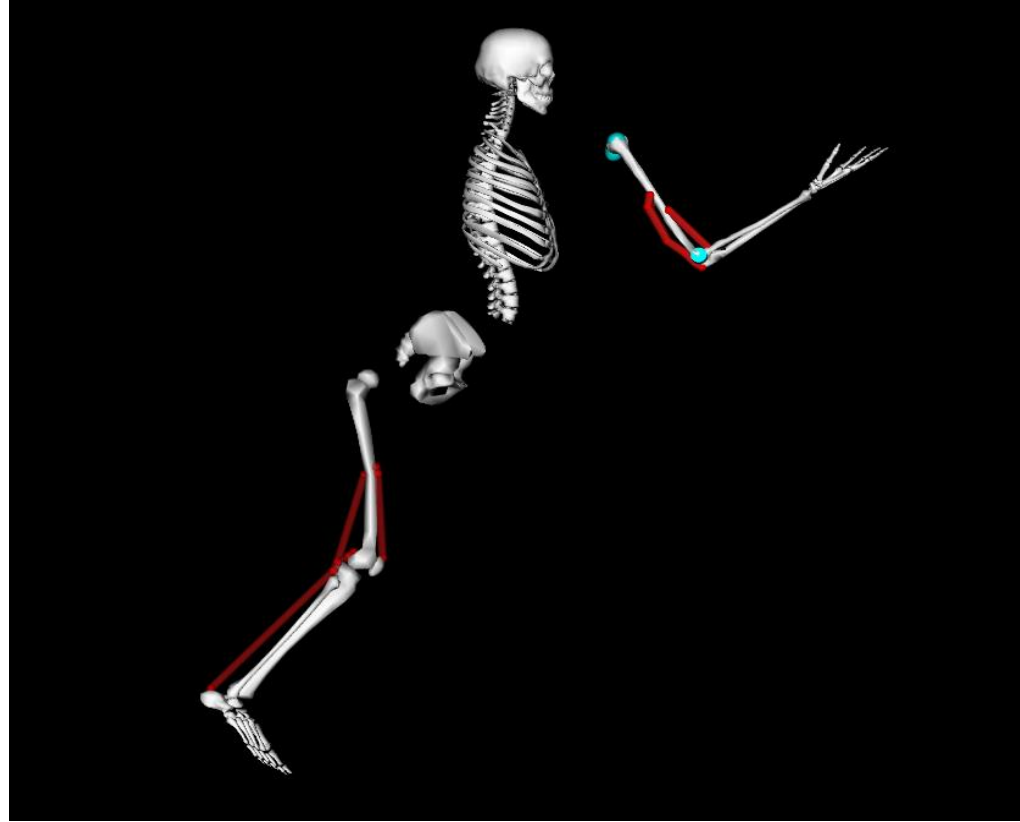


OpenSim model

An OpenSim model represents the dynamics of a system of rigid bodies and joints that are acted upon by forces to produce motion.



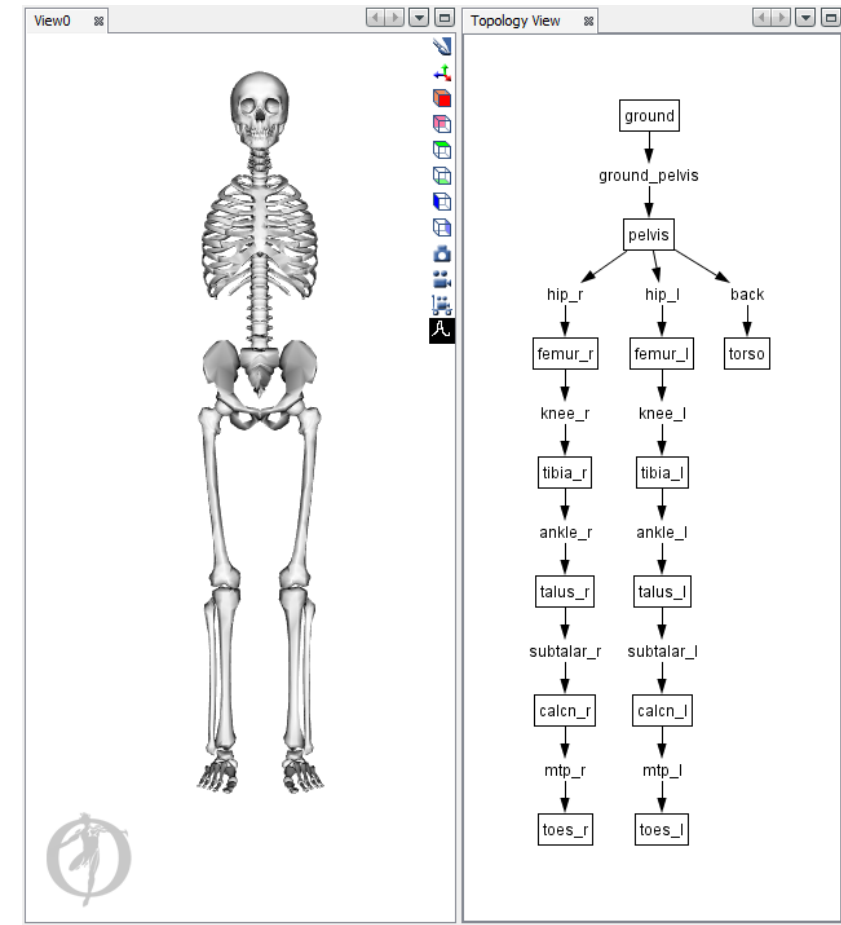
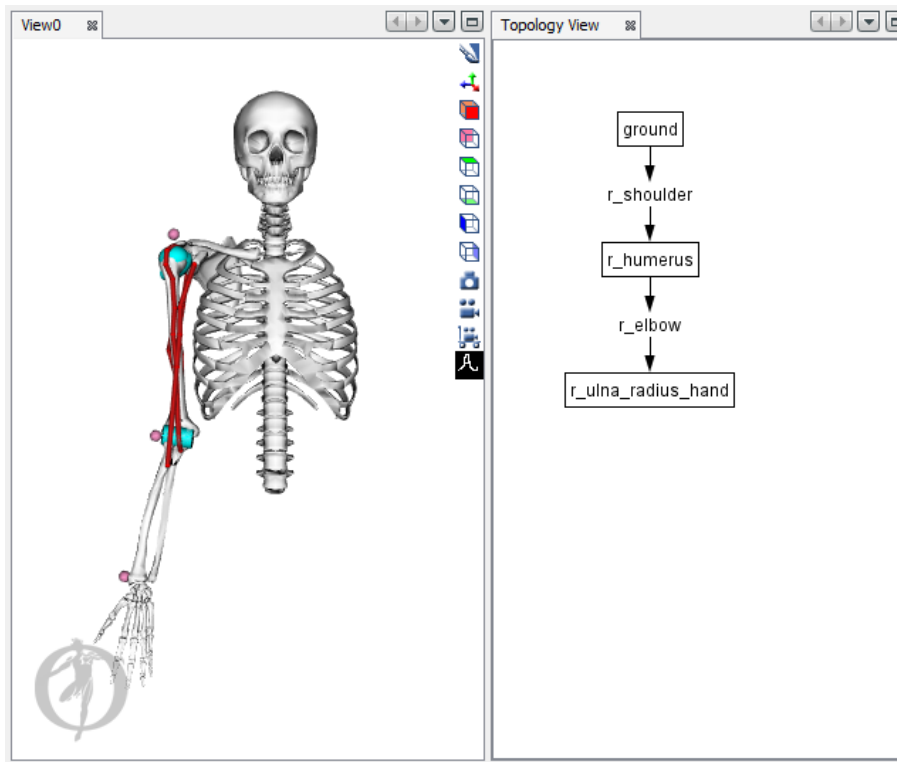
Components of an OpenSim Model



Bodies, joints, constraints, contact geometry,
forces, markers, and controllers

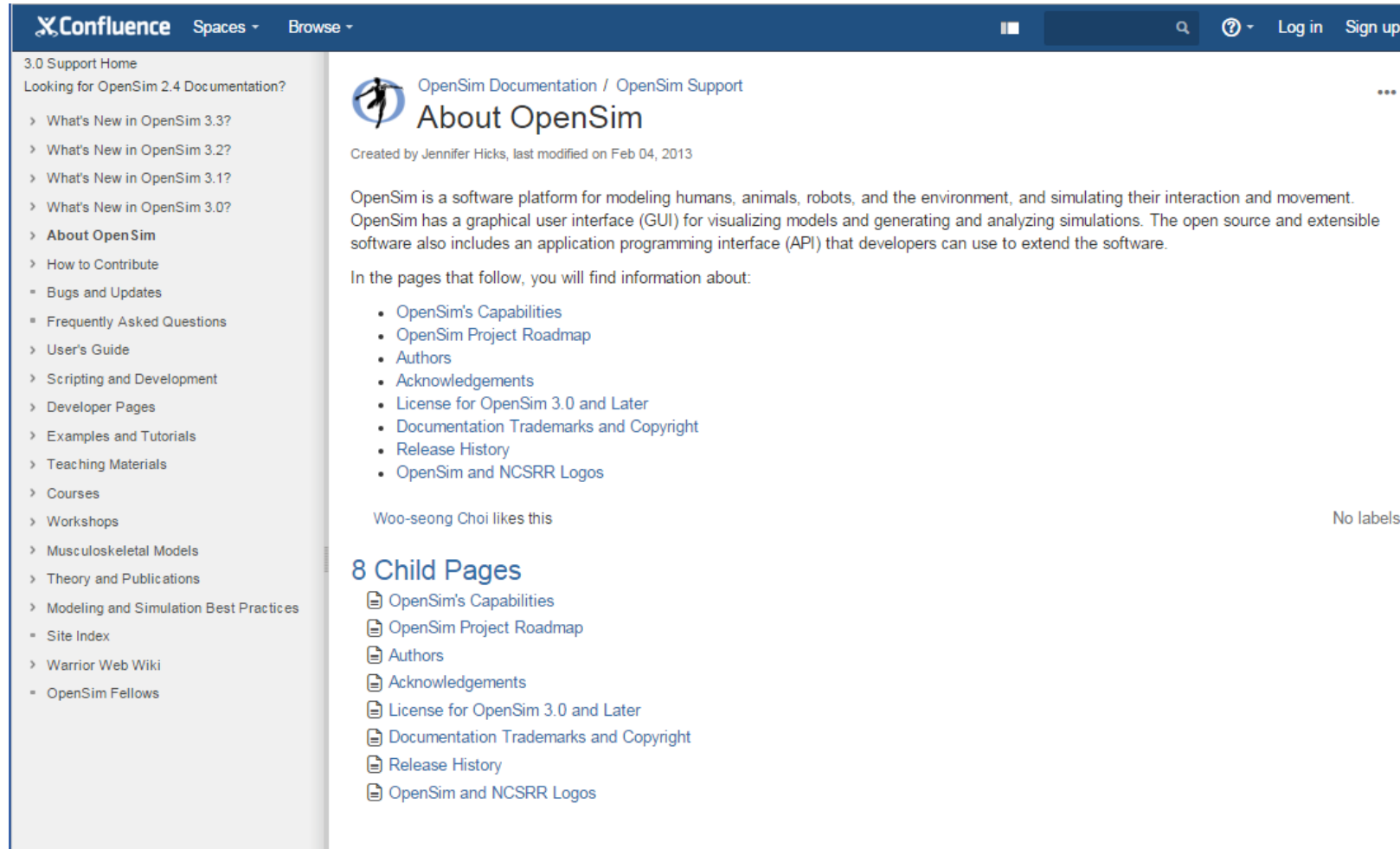
Tree Topology of Multibody Models

- You can view the topology of your model (Window>topology view).



How to find what you need

- Confluence website



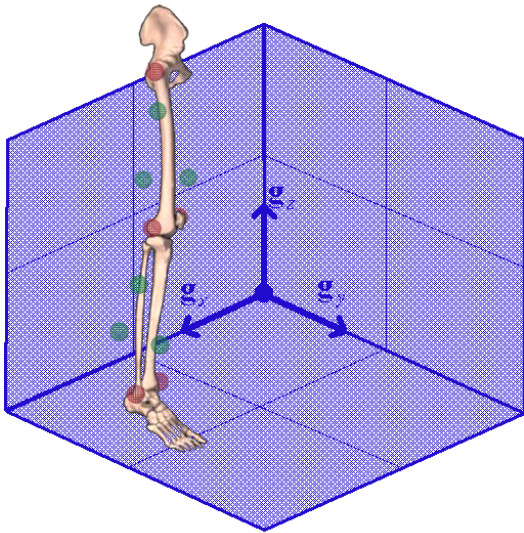
The screenshot displays the Confluence website interface. At the top, there's a navigation bar with the Confluence logo, 'Spaces', 'Browse', a search icon, a help icon, and links for 'Log in' and 'Sign up'. The main content area is titled 'OpenSim Documentation / OpenSim Support' and 'About OpenSim'. It includes a sidebar on the left with a list of links such as '3.0 Support Home', 'Looking for OpenSim 2.4 Documentation?', 'What's New in OpenSim 3.3?', 'What's New in OpenSim 3.2?', 'What's New in OpenSim 3.1?', 'What's New in OpenSim 3.0?', 'About OpenSim', 'How to Contribute', 'Bugs and Updates', 'Frequently Asked Questions', 'User's Guide', 'Scripting and Development', 'Developer Pages', 'Examples and Tutorials', 'Teaching Materials', 'Courses', 'Workshops', 'Musculoskeletal Models', 'Theory and Publications', 'Modeling and Simulation Best Practices', 'Site Index', 'Warrior Web Wiki', and 'OpenSim Fellows'. The main content area describes OpenSim as a software platform for modeling humans, animals, robots, and the environment, and simulating their interaction and movement. It also includes a list of links for 'OpenSim's Capabilities', 'OpenSim Project Roadmap', 'Authors', 'Acknowledgements', 'License for OpenSim 3.0 and Later', 'Documentation Trademarks and Copyright', 'Release History', and 'OpenSim and NCSRR Logos'. At the bottom, it shows '8 Child Pages' with the same list of links.

<http://simtk-confluence.stanford.edu:8080/display/OpenSim/OpenSim+Support>

1 Import data in OpenSim

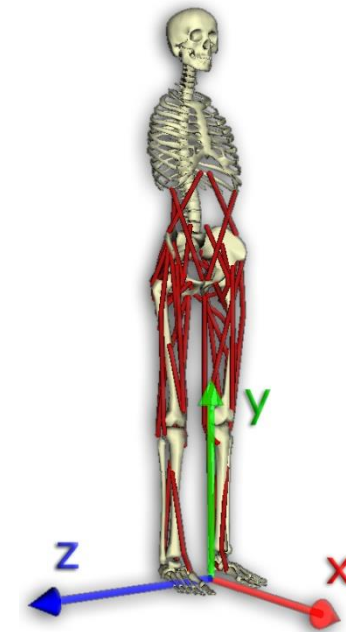
Coordinate Systems

Measure Markers in Lab Coordinate System



coordinate frame g

OpenSim Model Coordinate System



coordinate frame O

Rotation Matrix

$${}^A\mathbf{R}^B = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix}$$

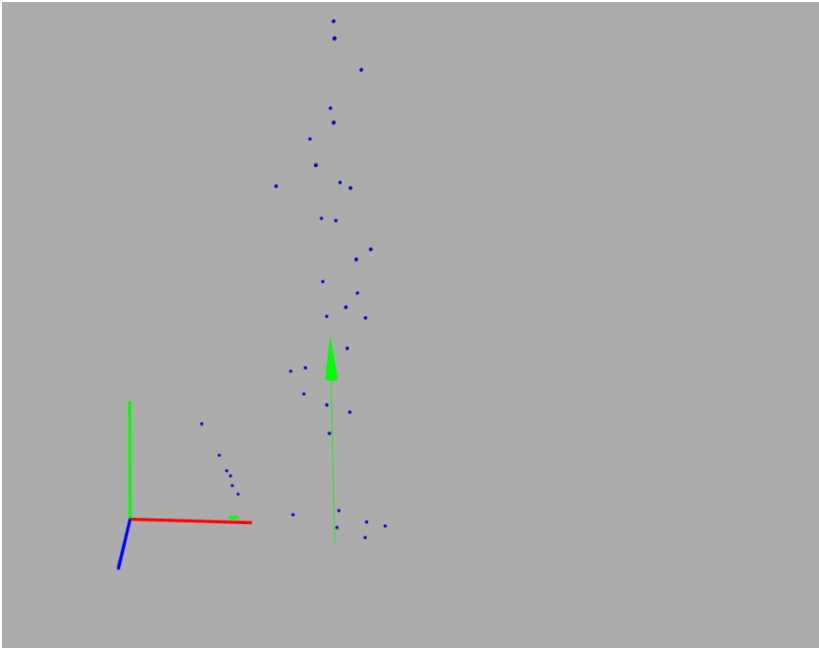
1) Import data in OpenSim – matlab example

...\Functions\ImportData_API.m

```
2 % c3d import-export
3 import org.opensim.modeling.*
4 c3dPath='C:\Users\u0088756\Documents\PostDoc\Data\DataNexus\ExpFriedl\TrapdoorExp\Friedl_v2\Session
5 c3d = osimC3D(c3dPath,1); % on is with computation COP, 2 is applied as torques
6
7 % Get some stats...
8 nTrajectories = c3d.getNumTrajectories();
9 rMarkers = c3d.getRate_marker();
10 nForces = c3d.getNumForces();
11 rForces = c3d.getRate_force();
12
13 % Get Start and end time
14 t0 = c3d.getStartTime();
15 tn = c3d.getEndTime();
16
17 % Rotate the data
18 c3d.rotateData('x',-90);
19 c3d.rotateData('y',180);
20
21 % Write the marker and force data to file
22 c3d.writeTRC('C:\Users\u0088756\Desktop\test_data_markers.trc');
23 c3d.writeMOT('C:\Users\u0088756\Desktop\test_data_forces.mot');
24
```

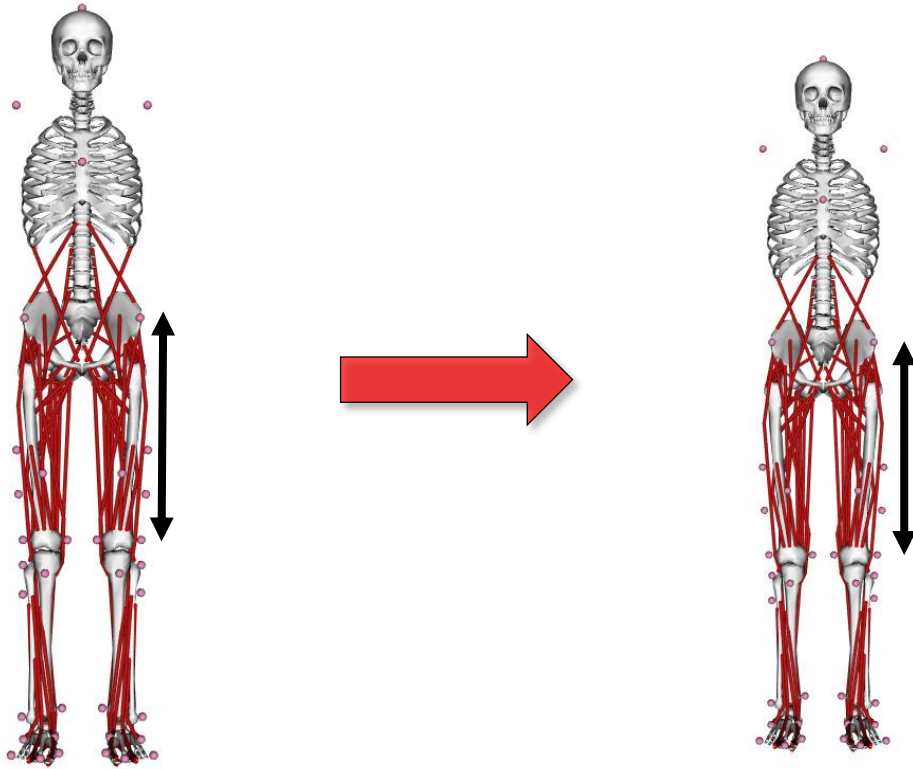

1) Import data in OpenSim – matlab example

Preview experimental data



2) Scaling Tool: Scale Model

Scale tool in OpenSim



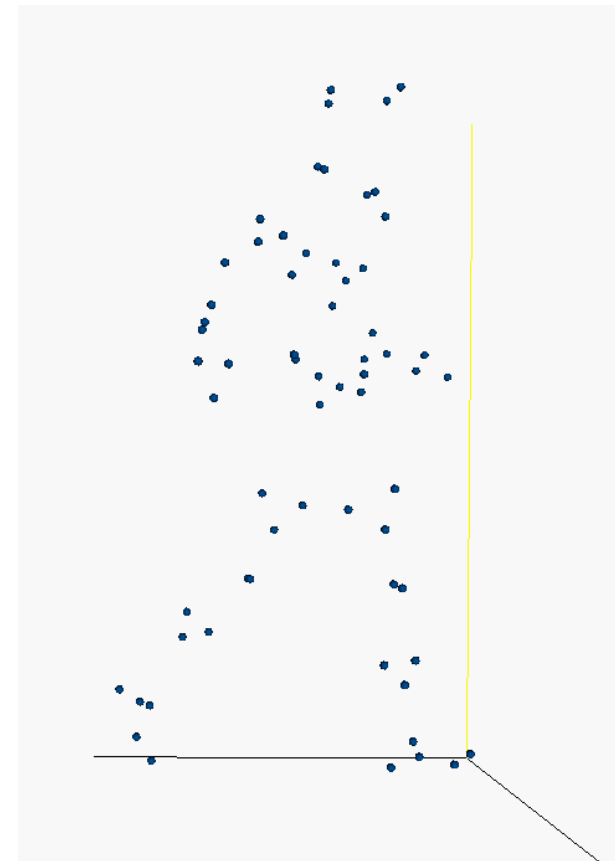
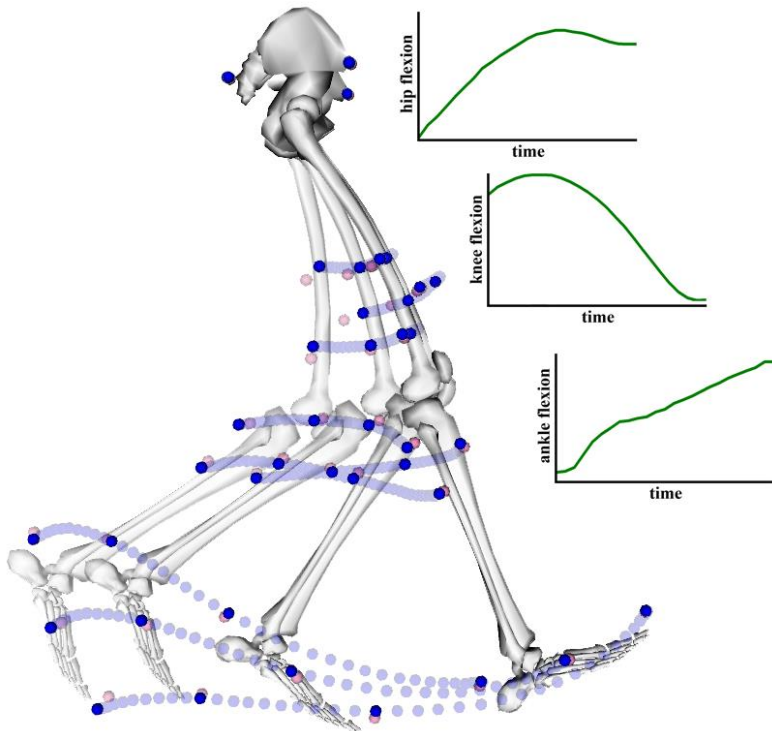
This step was already done

Scaled_model.osim

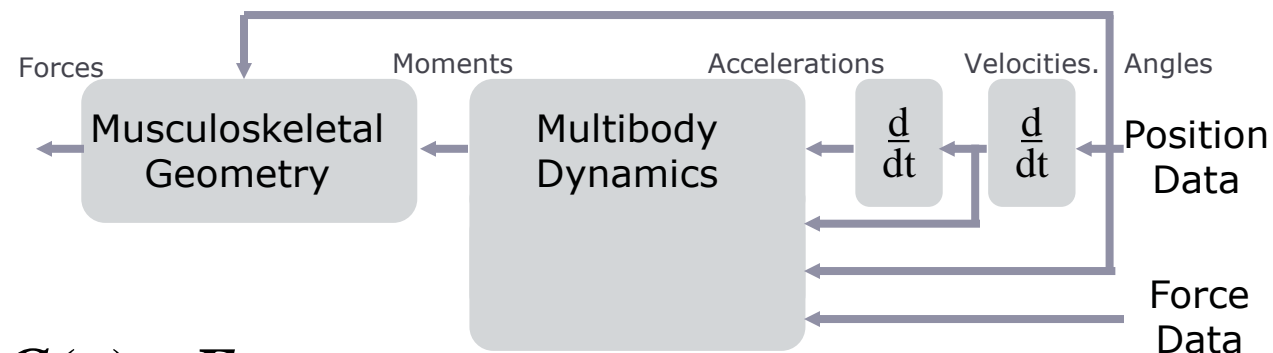
3) Inverse kinematics

Weighted Least Squares Minimization

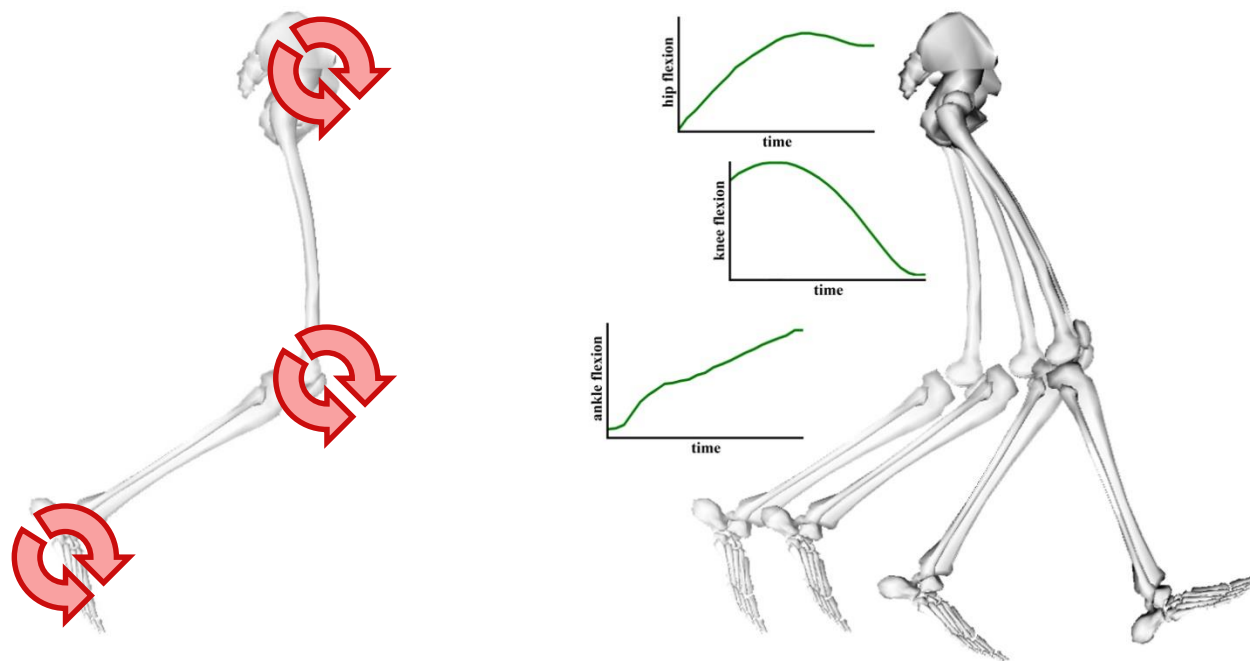
$$\min_q \left[\sum_{m=1}^{\# \text{ markers}} w_m \left\| \mathbf{x}_m^{\text{exp}} - \mathbf{x}_m(\mathbf{q}) \right\|^2 \right]$$



4) Run Inverse dynamics



$$\underbrace{\tau}_{\text{unknowns}} = \underbrace{M(q)\ddot{q} - C(q, \dot{q}) - G(q) - F}_{\text{knowns}}$$



q = generalized coordinates
 F = external forces
 $M(q)$ = mass matrix
 $C(q, \dot{q})$ = coriolis terms
 $G(q)$ = gravitational terms

4) Run Inverse dynamics

Solved algebraically
from the ground up

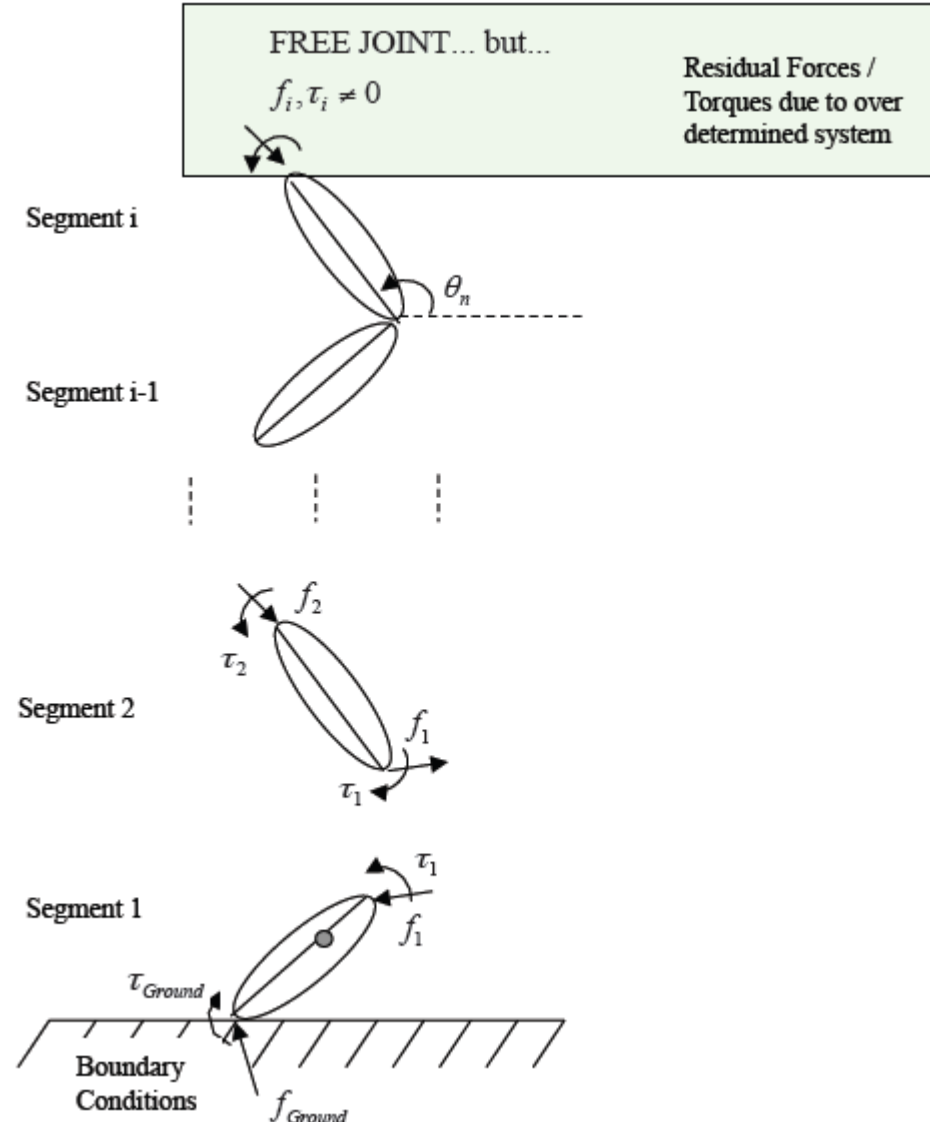
$$x, \dot{x}, \ddot{x} \quad \Sigma F_x = m\ddot{x}$$

$$y, \dot{y}, \ddot{y} \quad \rightarrow \quad \Sigma F_y = m\ddot{y}$$

$$\theta, \dot{\theta}, \ddot{\theta} \quad \Sigma T = I\ddot{\theta}$$

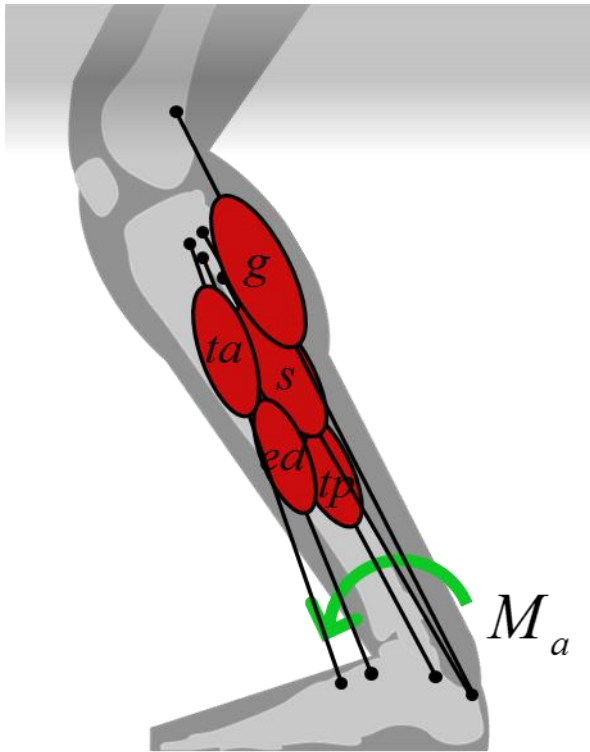


Joint Moments that
generate the motion



5) Solve for muscle forces

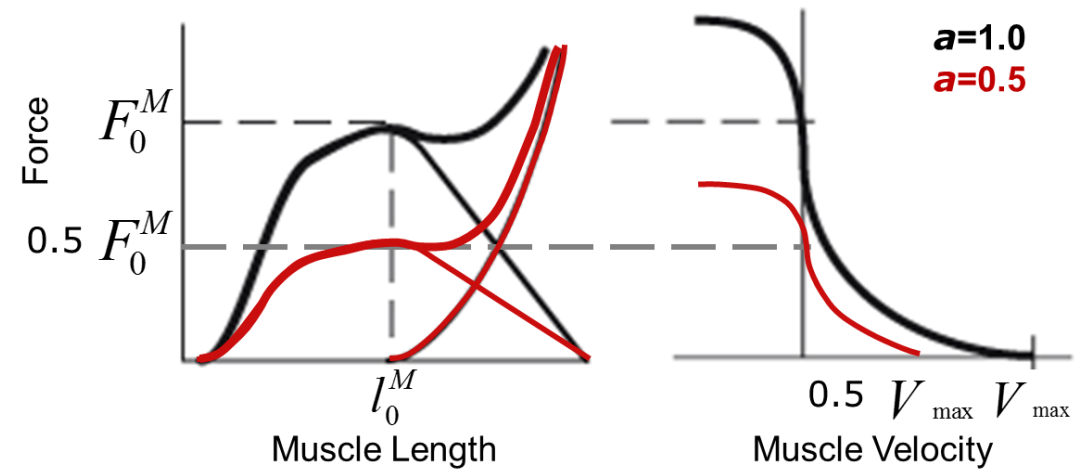
Static optimization



$$\tau = \sum_i^{nM} d_i F_i + \tau_{res}$$

With force as a function of:

Muscle fiber length, velocity and activation



6) Limitations of static optimization

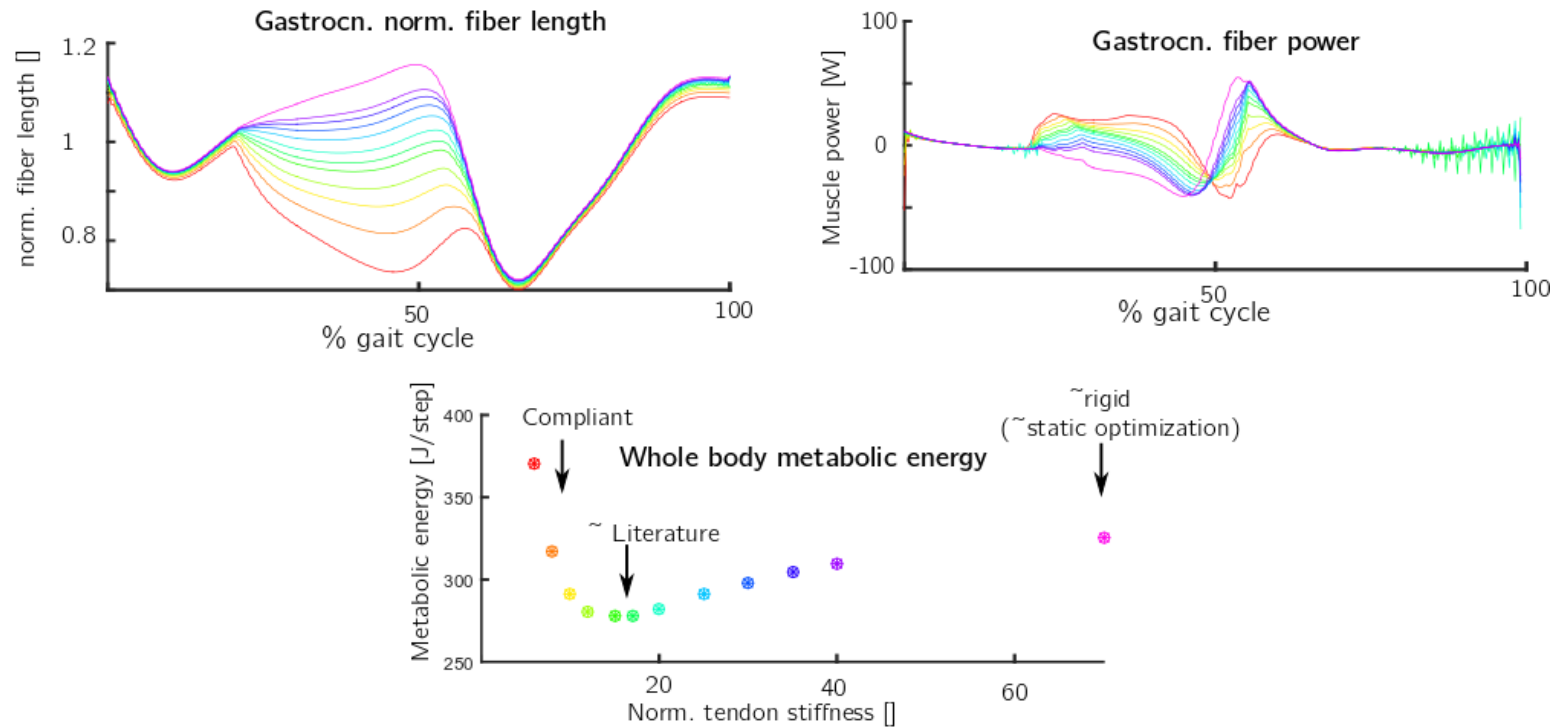
Accounting for contraction dynamics

- Activation dynamics ?
- Compliant tendons ?

Minizing activations squared ?

...

6) Simulation muscle-tendon dynamics



<https://simtk.org/projects/optcntrlmuscle>

Direct Collocation Optimal Control for Solving the Muscle Redundancy Problem

Share Follow Project

About Downloads Documents Forums Issues News

This project provides Matlab code for solving the muscle redundancy problem using direct collocation.

Download Latest Releases

License: MuscleRedundancySolver_v0.1, MuscleRedundancySolver_v1.1, MuscleRedundancySolver_v2.1

215 downloads
1 forum posts
7 followers
Last updated Nov 08, 2018
Project Statistics

Suggest Idea

Estimation of muscle forces during motion involves solving an indeterminate problem (more unknown muscle forces than joint moment constraints), frequently via optimization methods. When the dynamics of muscle activation and contraction are modeled for consistency with muscle physiology, the resulting optimization problem is dynamic and challenging to solve. This project proposes two computationally efficient formulations for solving these dynamic optimization problems using direct collocation optimal control methods. Both formulations rely on implicit representation of the contraction dynamics and use either muscle length or tendon force as state variable.