SimTK Documents

OpenSim Tutorial

March 10, 2009

GCMAS Annual Meeting, Denver, CO

Jeff Reinbolt, Ajay Seth, Scott Delp

Website: SimTK.org/home/opensim

OpenSim Tutorial Agenda

10:30am – 10:40am	Welcome and goals of tutorial – *Scott Delp*
10:40am – 11:00am	Inverse kinematics: how it works, exercise, & practice – *Jeff Reinbolt*
11:00am – 11:30am	Inverse dynamics: how it works, exercise, & practice – *Jeff Reinbolt*
11:30am – 11:40am	Questions and answers – *Everyone*
11:40am – 12:25pm	Forward dynamics: how it works, exercises, & practice – *Ajay Seth*
12:25pm – 12:30pm	Closing remarks – *Scott Delp*

Acknowledgments

OpenSim was developed as a part of SimTK and funded by the Simbios National Center for Biomedical Computing through the National Institutes of Health and the NIH Roadmap for Medical Research, Grant U54 GM072970. Information on the National Centers can be found at http://nihroadmap.nih.gov/bioinformatics.

Trademarks and Copyright and Permission Notice

Tutorial Description

Participants who attend this tutorial will learn key features of OpenSim through real-life illustrations and exercises. Participants will create simulations on their computers and analyze those simulations to understand muscle actions.

Audience

This tutorial is recommended for those who desire broader knowledge musculoskeletal simulation with OpenSim and how to begin using this software.

Learning Objectives

- Discuss the need for simulations to understand the complex neuromusculoskeletal system
- Understand why inverse kinematics is needed and how it is used to determine joint motion
- Perform inverse dynamics to determine joint torques from joint motion
- Perform forward dynamics to determine joint motion from muscle excitations
- Edit muscle excitations in a forward dynamics simulation to understand muscle actions

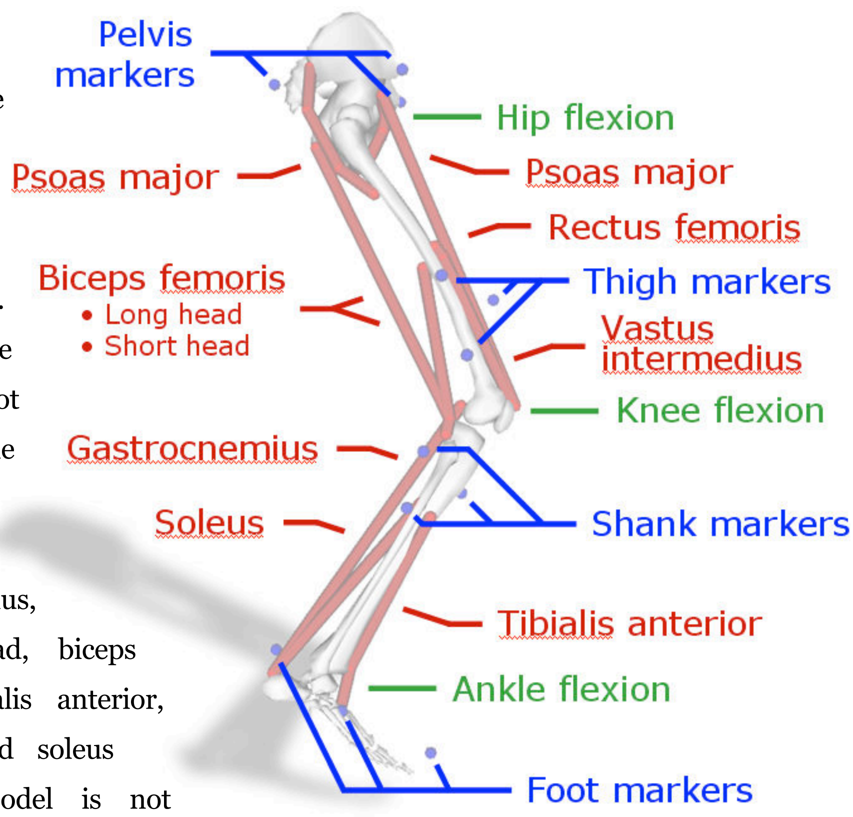
Table of Contents

1 Tutorial Overview

This tutorial provides an introduction to some of the many tools available within OpenSim, a freely available package for musculoskeletal modeling and dynamic simulation of movement. For more information on OpenSim, visit the OpenSim project site at http://simtk.org/home/opensim. The project site provides a forum for users to ask questions and share expertise.

This workbook serves as a summary and guide for the tutorial. In the first half hour, we will focus on inverse kinematics to determine the joint angles that best match experimentally collected marker trajectories. In the second half hour, we will focus on inverse dynamics to compute the joint torques necessary to produce joint angles from inverse kinematics. In the final hour, we will focus on forward dynamic simulations that help us understand muscle actions.

To present some of the tools and capabilities of OpenSim, we will use a simplified model (leg39) throughout this workbook. The model consists of the pelvis, thigh, shank, and foot segments along with the psoas major, gluteus maximus, rectus femoris, vastus intermedius, biceps femoris long head, biceps femoris short head, tibialis anterior, medial gastrocnemius, and soleus muscles. This simple model is not intended for research.



leg39 (3 coordinates & 9 muscles) model

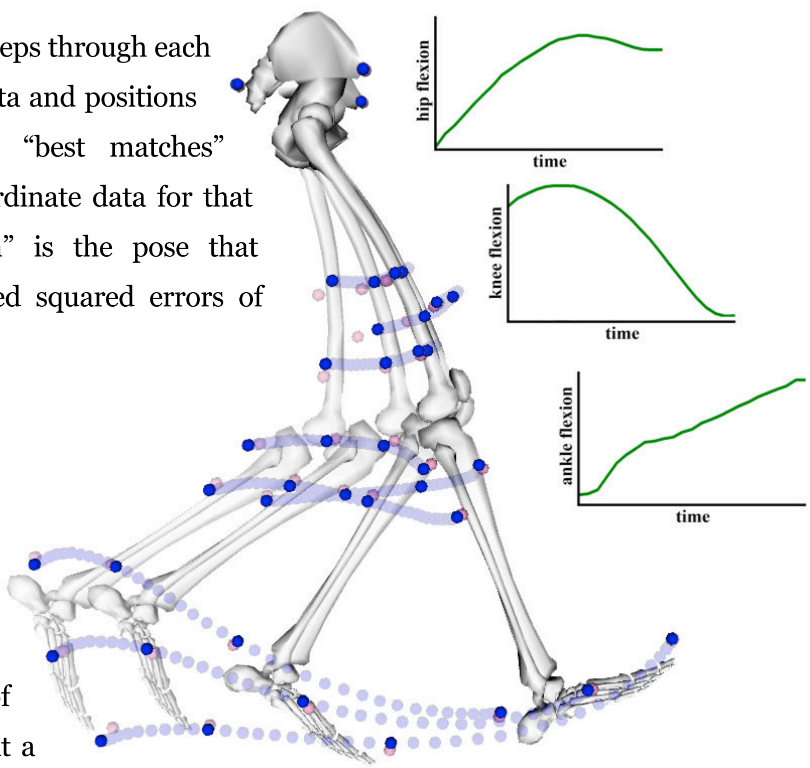
2 Inverse Kinematics

Key Concepts

- Model pose and coordinates
- Marker error
- Coordinate error
- Weighted least squares minimization

2.1 How It Works

The Inverse Kinematics Tool steps through each time frame of experimental data and positions the model in a pose that "best matches" experimental marker and coordinate data for that time step. This "best match" is the pose that minimizes a sum of weighted squared errors of markers and/or coordinates.



2.1.1 Model Pose and Coordinates

The *model pose* is the set of orientations and locations of body segments in the model at a given instant. A *coordinate* is a joint angle or distance that specifies the relative orientation or location of two body segments. Therefore, the set of all model coordinates completely defines the model pose. For example, at the beginning of swing phase, the hip flexion (1°), knee flexion (57°), and ankle flexion (3°) completely define the pose for the leg39 model.

2.1.2 Marker Error

A *marker error* is the distance between an experimental marker (●) and the corresponding model marker (●) at a given instant in time. During inverse kinematics, each marker has an associated weight specifying how strongly that marker's error should be minimized.

2.1.3 Coordinate Error

A *coordinate error* is the difference between an "experimental coordinate" and the corresponding model coordinate (e.g., joint angle or distance) value at a given instant in time. Experimental coordinate values can be joint angles obtained directly from a motion capture system (i.e., built-in mocap inverse kinematics capabilities), or by techniques that involve other measurement devices (e.g., a goniometer). A desired coordinate can also be a specified as constant (e.g., if we know that a specific joint angle should stay at say 0˚). During inverse kinematics, each coordinate has an associated weight specifying how strongly that coordinate's error should be minimized.

2.1.4 Weighted Least Squares Minimization

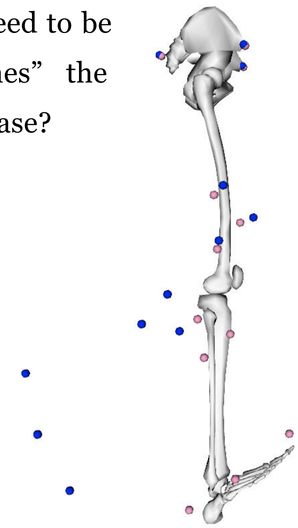
The weighted least squares minimization solved by the Inverse Kinematics Tool is

$$\min_{\boldsymbol{q}}\left[\sum_{m=1}^{\#\text{ markers}} w_m \left\|\boldsymbol{x}_m^{\exp} - \boldsymbol{x}_m(\boldsymbol{q})\right\|^2 + \sum_{c=1}^{\#\text{ coordinates}} \omega_c \left(q_c^{\exp} - q_c\right)^2\right]$$

where $\boldsymbol{q}$ is the set of coordinates to be determined, $\boldsymbol{x}_m^{\exp}$ is the experimental position of marker m, $\boldsymbol{x}_m(\boldsymbol{q})$ is the position of the corresponding model marker (as a function of the coordinates, $\boldsymbol{q}$), w_m is the weight associated with the m^{th} marker, $q_c^{\exp}$ is the experimental value for coordinate c, q_c is the corresponding model coordinate, and ω_c is the weight associated with the c^{th} coordinate.

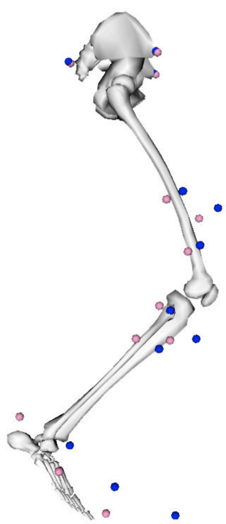
Exercise

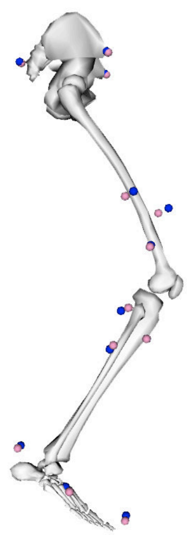
1. For the model shown on the right, which coordinate(s) need to be adjusted to create a model pose that “best matches” the experimental markers shown at the beginning of swing phase?

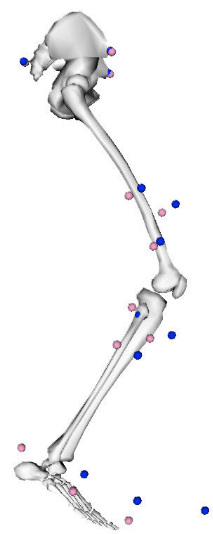


   A. Hip
   B. Knee
   C. Ankle
   D. Hip and ankle
   E. Knee and ankle

2. For the model poses and experimental markers shown below, which combination of pose and markers has the minimum marker errors?


A.


B.


C.

3. In theory, experimental markers on the thigh and shank could have more skin movement artifacts compared with the foot markers; which of the following scenarios would be most appropriate for the weighted least squares minimization solved by the Inverse Kinematics Tool?

   A. Decrease tracking weights on thigh markers
   B. Decrease tracking weights on shank markers
   C. Increase tracking weights on foot markers
   D. All of the above

2.2 Inverse Kinematics Tool

To launch the Inverse Kinematics Tool, select **Inverse Kinematics…** from the **Tools** menu. The **Inverse Kinematics Tool** dialog, like all other OpenSim tools, operates on the *Current Model* open and selected in OpenSim.

2.2.1 Inputs

Three items are required as input for the Inverse Kinematics Tool:

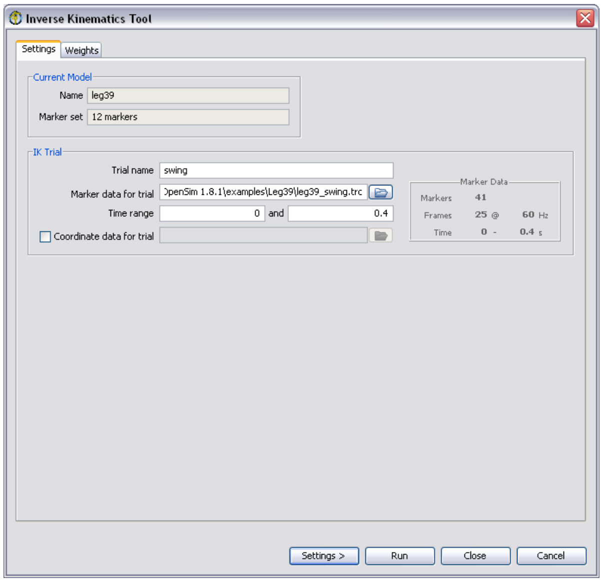
- The current model loaded in OpenSim.
- Experimental marker positions and/or coordinate values for a motion trial.
- Inverse kinematics tasks which specify the experimental markers positions and/or coordinate values that should be matched and their respective weightings.

2.2.2 Outputs

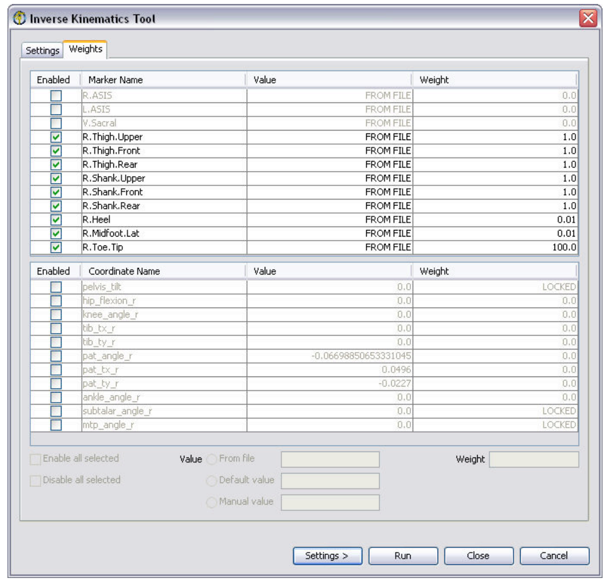
The Inverse Kinematics Tool generates a single motion containing the time histories of the coordinates that describe the movement of the model.

Practice: Peak knee flexion in swing and marker weights

Settings



Weights



1. What is the subject's peak knee flexion in swing?
2. When did peak knee flexion occur during swing?
3. What is the RMS marker error at peak knee flexion in swing (*Note: check output to the messages window; to open this window, select Messages from the Window menu*)?
4. By changing the marker weights, can you reduce the RMS marker error at peak knee flexion in swing (*Note: peak knee flexion in swing changes with different marker weights*)? If yes, what were the weights and how much did the error decrease?

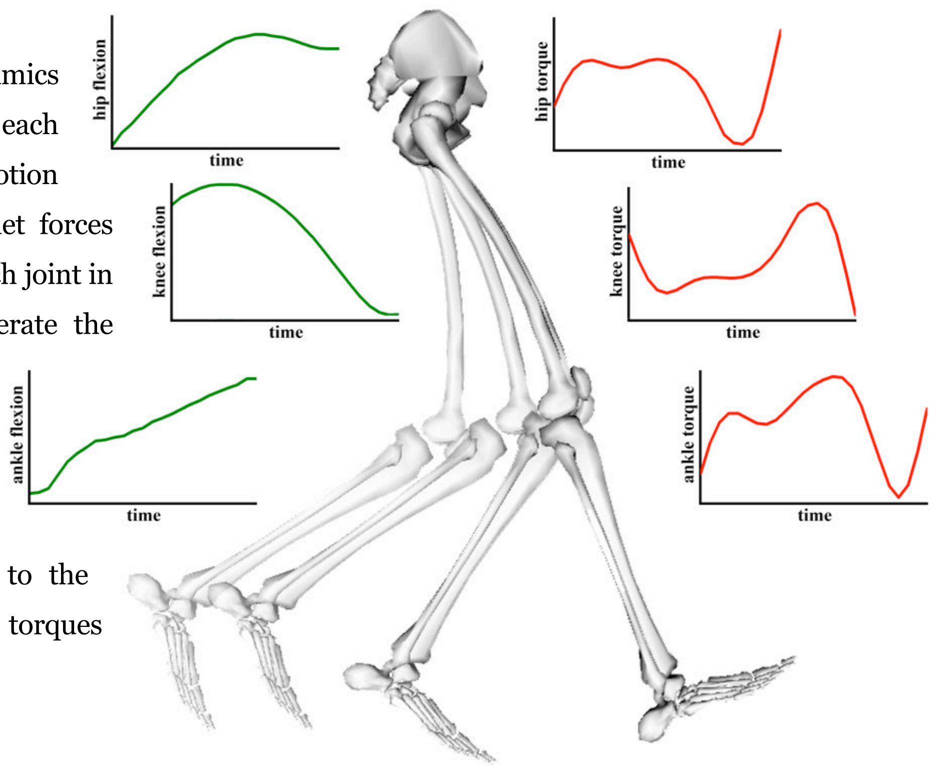
3 Inverse Dynamics

Key Concepts

- Kinematics: coordinates and their velocities and accelerations
- Kinetics: forces and torques
- Dynamics: equations of motion

3.1 How it Works

The Inverse Dynamics Tool steps through each time frame of a motion and computes the net forces and/or torques at each joint in the model that generate the experimental kinematics. The equations of motion relate the model accelerations to the forces and/or joint torques applied to the model.



3.1.1 Kinematics: Coordinates and their Velocities and Accelerations

A *coordinate* is a joint angle or distance that specifies the relative orientation or location of two body segments in the model. The derivative (rate of change) of a coordinate with respect to time is the coordinate's *velocity*. In turn, the time derivative of its velocity is the coordinate's *acceleration*. The collection of coordinates and their velocities and accelerations describe the *kinematics* of the model.

3.1.2 Kinetics: Forces and Torques

Forces and *torques* cause the model to accelerate according to Newton's second law. A force can be applied to points (e.g., ground reactions) or between points (e.g., muscles) on the model. A torque can be applied to a coordinate (e.g., joint torque) to accelerate that joint angle directly.

3.1.3 Dynamics: Equations of Motion

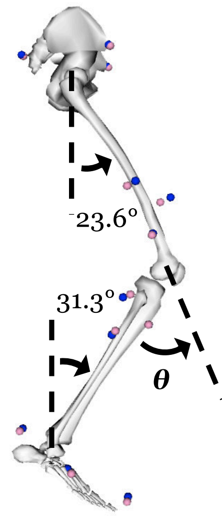
From Newton's second law, we can determine the kinetics necessary to accelerate the model by treating the skeleton as a set of interconnected rigid-bodies with inertial properties such that:

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

where $\boldsymbol{\tau}$ is the set of joint torques, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ are the coordinates and their velocities and accelerations, respectively; $\boldsymbol{M}(\boldsymbol{q})$ is the mass matrix, which depends on the coordinates and inertial properties of the model; $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the combination of Coriolis and centrifugal forces, which depend on the coordinates and their velocities; $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational force, which depends on the coordinates; and $\boldsymbol{F}$ is other forces applied to the model.

The motion of the model is completely defined by the coordinates and their velocities and accelerations. Consequently, all of the terms on the right-hand side of the equations of motion are known. The remaining set of joint torques on the left-hand side is unknown. The Inverse Dynamics Tool uses the known motion of the model to solve the equations of motion for the unknown joint torques.

Exercise



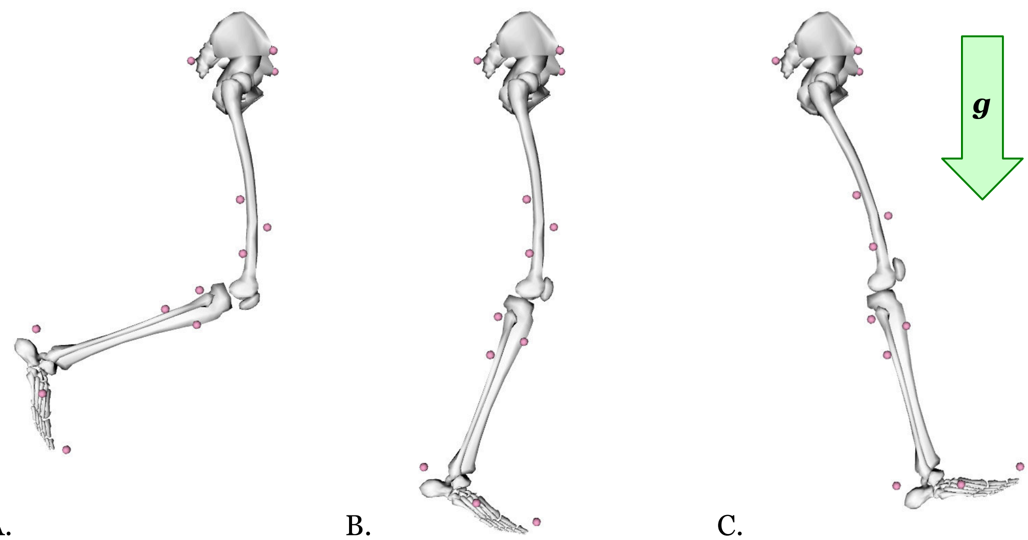
1. For the model shown on the right, what is the value (θ) of the knee coordinate (*Note: extension is* +)?

   A. 23.6° B. -54.9° C. 31.3° D. -125.1°

2. Given that the model shown on the right is at rest, what is the velocity of the knee?

   A. 23.6°/s B. -54.9°/s C. 3.89°/s D. 0°/s

3. For the model poses shown below at rest and with gravity ($\boldsymbol{g}$) as the only force acting on the model, which pose requires the largest torque at the knee joint?



A. B. C.

3.2 Inverse Dynamics Tool

To launch the Inverse Dynamics Tool select **Inverse Dynamics…** from the **Tools** menu. The **Inverse Dynamics Tool** dialog, like all other OpenSim tools, operates on the *Current Model* open and selected in OpenSim.

3.2.1 Inputs

Two items are required as input for the Inverse Dynamics Tool:

- The current model loaded in OpenSim.
- Motion file containing the time histories of coordinates that describe the movement of the model. This file may be generated by the Inverse Kinematics Tool.

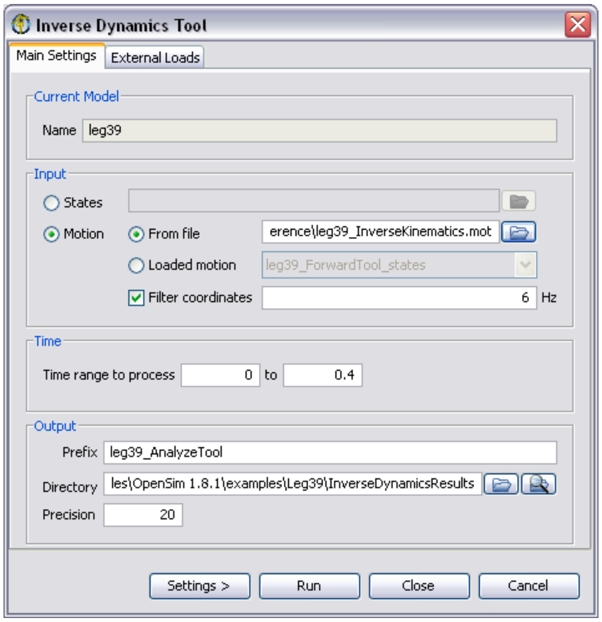
One *additional* item may be provided to the Inverse Dynamics Tool: motion file containing external loads (e.g., ground reaction forces).

3.2.2 Outputs

The Inverse Dynamics Tool generates a single storage file containing the time histories of the the net forces and/or torques at each joint responsible for the movement of the model.

Practice: Knee torque and filtering

Main Settings



1. What is the subject's knee joint torque at peak knee flexion during swing?
2. By changing the filter cutoff frequency from 6 Hz to 12Hz, how does this change the subject's knee joint torque at peak knee flexion in swing?
3. Without filtering the motion, what is the subject's knee joint torque at peak knee flexion in swing?
4. Is it practical to perform inverse dynamics without filtering the motion?

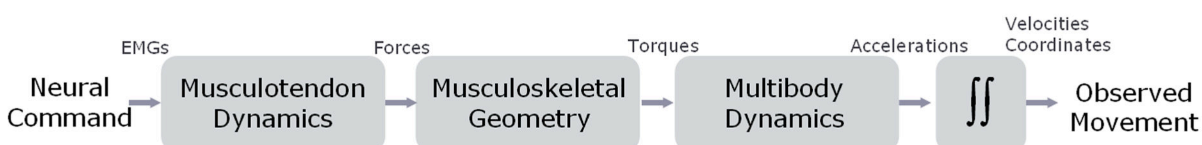
4 Forward Dynamics

Key Concepts

- Musculoskeletal model dynamics
- States of a musculoskeletal model
- Controlling a musculoskeletal simulation
- Numerical integration of dynamical equations

4.1 How it Works

The Forward Dynamics Tool uses the model together with the initial states and controls to run a muscle-driven forward dynamics simulation. A forward dynamics simulation is the solution (integration) of the differential equations that define the dynamics of a musculoskeletal model.



4.1.1 Musculoskeletal Model Dynamics

In contrast to inverse dynamics where the motion of the model was known and we wanted to determine the forces and torques that generated the motion, in forward dynamics, a mathematical model describes how coordinates and their velocities change due to applied forces and torques (moments).

From Newton's second law, we can describe the accelerations (rate of change of velocities) of the coordinates in terms of the inertia and forces applied on the skeleton as a set of rigid-bodies:

$$\ddot{\boldsymbol{q}} = \left[\boldsymbol{M}(\boldsymbol{q})\right]^{-1}\left\{\boldsymbol{\tau} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{F}\right\}$$ Multibody dynamics

where □ is the coordinate accelerations due to joint torques, $\boldsymbol{\tau}$, Coriolis and centrifugal forces, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$, as a function of coordinates, $\boldsymbol{q}$, and their velocities, $\dot{\boldsymbol{q}}$, gravity, $\boldsymbol{G}(\boldsymbol{q})$, and other forces applied to the model, $\boldsymbol{F}$, and $\left[\boldsymbol{M}(\boldsymbol{q})\right]^{-1}$ is the inverse of the mass matrix.

$$\boldsymbol{\tau}_m = \left[\boldsymbol{R}(\boldsymbol{q})\right]\boldsymbol{f}(\boldsymbol{a},\boldsymbol{l},\dot{\boldsymbol{l}})$$ Moments due to muscle forces

$$\dot{\boldsymbol{l}} = \Lambda(\boldsymbol{a},\boldsymbol{l},\boldsymbol{q},\dot{\boldsymbol{q}})$$ Muscle contraction dynamics

$$\dot{\boldsymbol{a}} = \mathrm{A}(\boldsymbol{a},\boldsymbol{x})$$ Muscle activation dynamics

The net muscle moments, $\boldsymbol{\tau}_m$, in turn, are a result of the moment arms, $\boldsymbol{R}(\boldsymbol{q})$, multiplied by muscle forces, $\boldsymbol{f}$, which are a function of muscle activations, $\boldsymbol{a}$, and muscle fiber lengths, $\boldsymbol{l}$, and velocities, $\dot{\boldsymbol{l}}$. Muscle fiber velocities are governed by muscle contraction dynamics, Λ, which is dependent on the current muscle activations and fiber lengths as well as the coordinates and their velocities. Activation dynamics, A, describes how the activation rates, $\dot{\boldsymbol{a}}$, of the muscles respond to input neural excitations, $\boldsymbol{x}$, generally termed the model's controls. These form a set of differential equations that model *musculoskeletal dynamics*.

4.1.2 States of a Musculoskeletal Model

The *state* of a model is the collection of all model variables defined at a given instant in time that are governed by dynamics. The model dynamics describe how the model will advance from a given state to another. In a *musculoskeletal model* the states are the coordinates and their velocities and muscle activations and muscle fiber lengths. The dynamics of a model require the state to be known in order to calculate the rate of change of the model states (joint accelerations, activation rates, and fiber velocities) in response to forces and controls.

4.1.3 Controlling a Musculoskeletal Model

The forces (e.g., muscles) in a *musculoskeletal model* are governed by dynamics and have inputs that affect their behavior. In OpenSim, these inputs are called the *controls* of a model, which can be excitations for muscles or torque generators. Ultimately, controls determine the forces and/or torques applied to the model and therefore determine the resultant motion.

4.1.4 Numerical Integration of Dynamical Equations

A simulation is the integration of the musculoskeletal model's dynamical equations starting from a user-specified initial state. After applying the controls, the activation rates, muscle fiber velocities, and coordinate accelerations are computed. Then, new states at small time interval in the future are determined by numerical integration. A 5^{th}-order Runge-Kutta-Feldberg integrator is used to solve (numerically integrate) the dynamical equations for the trajectories of the musculoskeletal model states over a definite interval in time. The Forward Dynamics Tool is an open-loop system that applies muscle/actuator controls with no feedback, or correction mechanism, therefore the states are not required to follow a desired trajectory.

Exercise

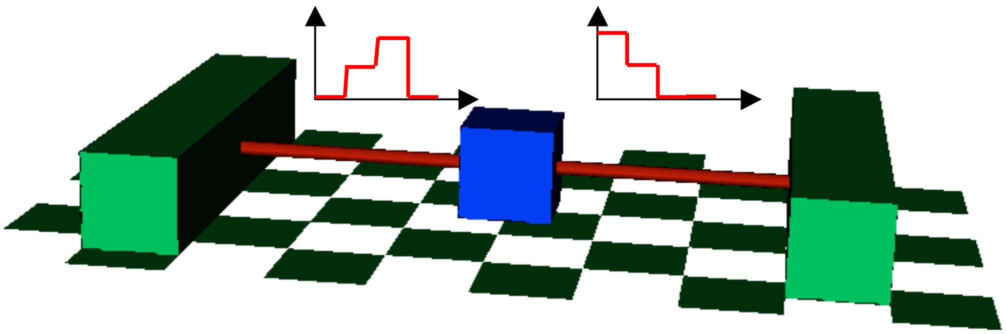
1. A forward dynamics simulation is

   A. a musculoskeletal model
   B. muscle-driven
   C. a simulation that uses feed-back
   D. the integration of dynamical equations

2. The musculoskeletal model for the tutorial (leg39) has how many states?

   A. 3 B. 9 C. 12 D. 24

3. Given the model below with two identical muscles and their levels of control plotted versus time, which way will the block initially move if starting from rest?



A. To the left B. To the right C. Does not move D. Upward

4. Given initial coordinates and their velocities and muscle activations and muscle fiber lengths, how are these states determined at a small instant ahead in time?

   A. Specify controls and compute muscle activation rates, fiber velocities, and coordinate accelerations from model dynamics
   B. Numerically integrate forces and controls from model differential equations
   C. Numerically integrate muscle activation rates, fiber velocities, and coordinate accelerations
   D. Numerically differentiate forces and controls from the dynamical equations
   E. A & C

4.2 Forward Dynamics Tool

To launch the Forward Dynamics Tool select **Forward Dynamics…** from the **Tools** menu. The **Forward Dynamics Tool** dialog like all other OpenSim tools operates on the *Current Model* open and selected in OpenSim.

4.2.1 Inputs

Three items are required by the Forward Dynamics Tool:

- The current model loaded in OpenSim
- XML file containing the time histories of the model controls (e.g., muscle excitations) to the muscles and/or joint torques. This file may be generated by the user, Static Optimization Tool, or Computed Muscle Control Tool.
- Storage file containing the initial states of the musculoskeletal model that includes coordinates and their velocities and muscle activations and muscle fiber lengths

Two *additional* items may be provided to the Forward Dynamics Tool:

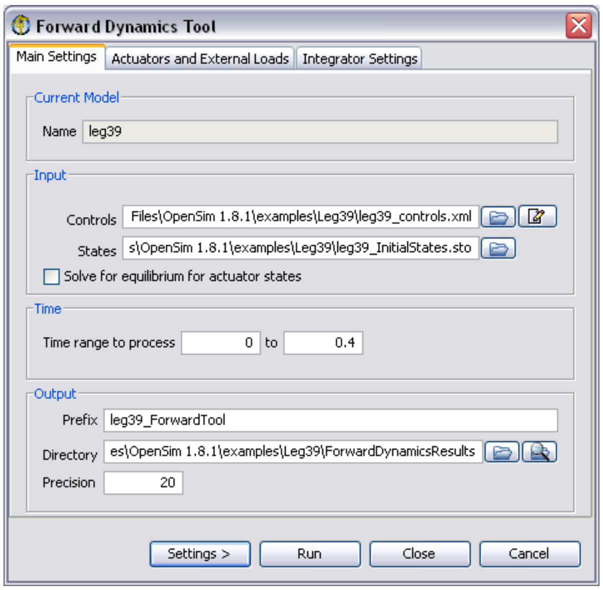
- Motion file containing external loads (e.g., ground reaction forces)
- Settings for numerical integration

4.2.2 Outputs

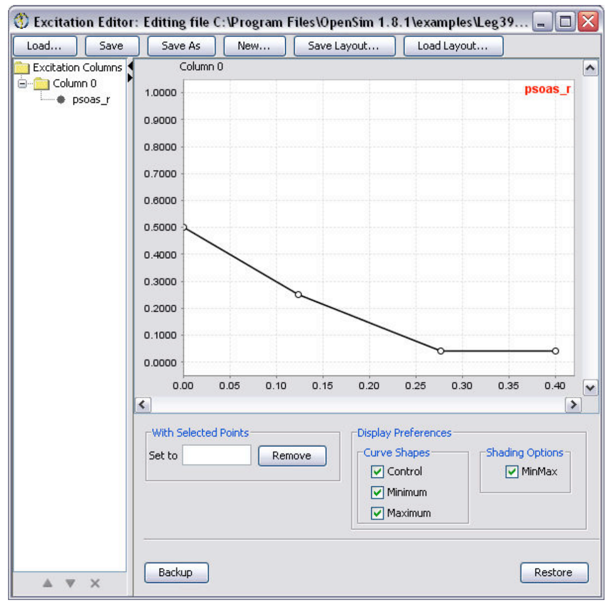
The Forward Dynamics Tool generates a storage file containing the time histories of the model's controls and states that result from integration of the model's dynamical equations.

Practice: Muscle-driven swing leg simulation

Main Settings



Excitation Editor



1. Without exciting any muscles, what are the hip, knee, and ankle angles at end of swing? How do they compare to angles you determined from inverse kinematics?
2. By exciting a single muscle in early swing, can you exceed the hip flexion that you determined from inverse kinematics? If yes, what muscle?
3. By exciting a second muscle in late swing, can you avoid hyper-extending the knee and achieve the hip flexion from inverse kinematics? If so, which muscle?
4. By exciting a third muscle, can you avoid excessive ankle plantarflexion? If yes, what muscle?
5. What methods could you use to automate the process of creating a forward dynamics simulation from the inverse kinematics results?