OpenSim

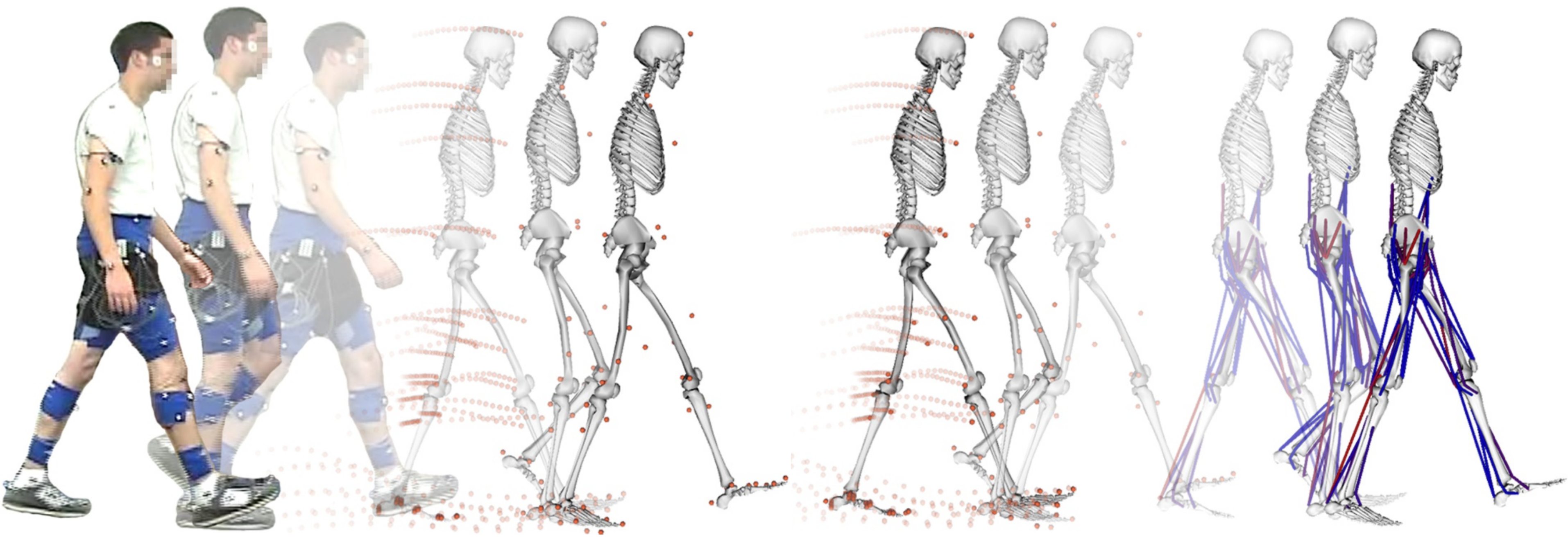
https://simtk.org/home/opensim

Introduction to OpenSim

OpenSim Tutorial SIMPAR 2010

Agenda

8:30 – 8:45	Welcome and goals of workshop – *Sam Hamner & Massimo Sartori*
8:45 – 9:00	**Introduction to the GUI** **– *Sam Hamner***
9:00 – 10:00	Guided GUI Example and Exploration – *Sam & You*
10:00 – 10:30	Break
10:30 – 10:45	Introduction to the API – *Massimo Sartori*
10:45 – 11:55	Guided API Example and Exploration – *Massimo & You*
11:55 – 12:00	Closing remarks – *Sam & Massimo*



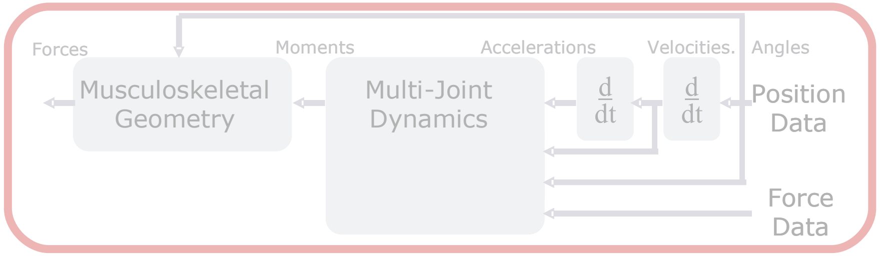
Inverse Kinematics

Inverse Dynamics

Computed Muscle Control

OpenSim Workshop SIMPAR 2010

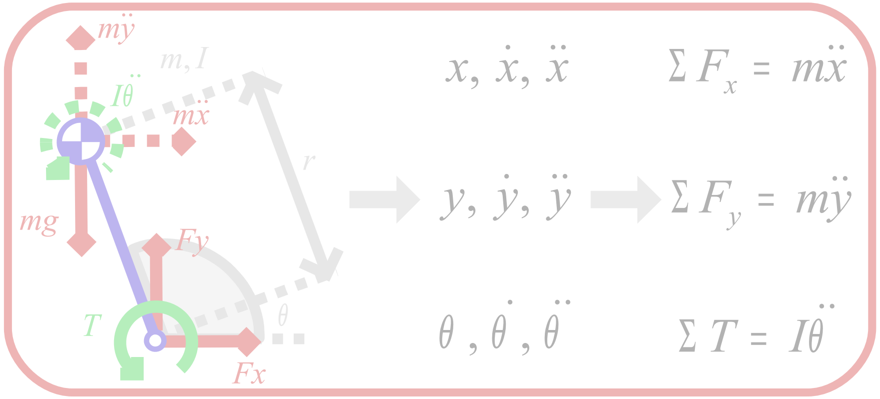
Plan for the Session



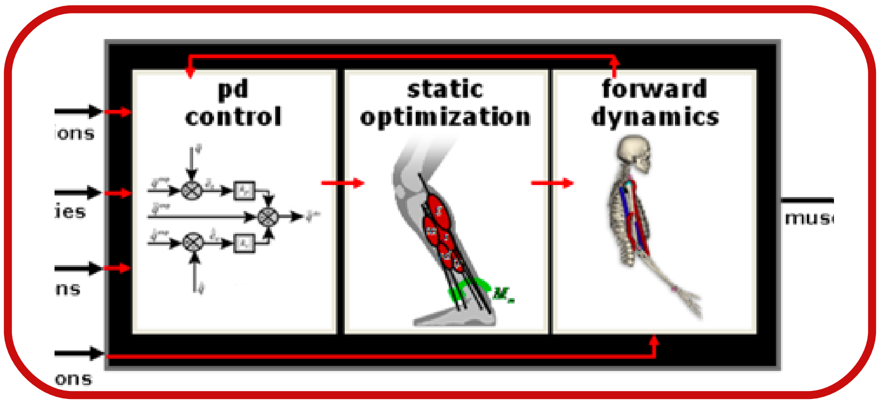
The inverse problem

Going from subject motion to joint kinematics

Plan for the Session (cont.)

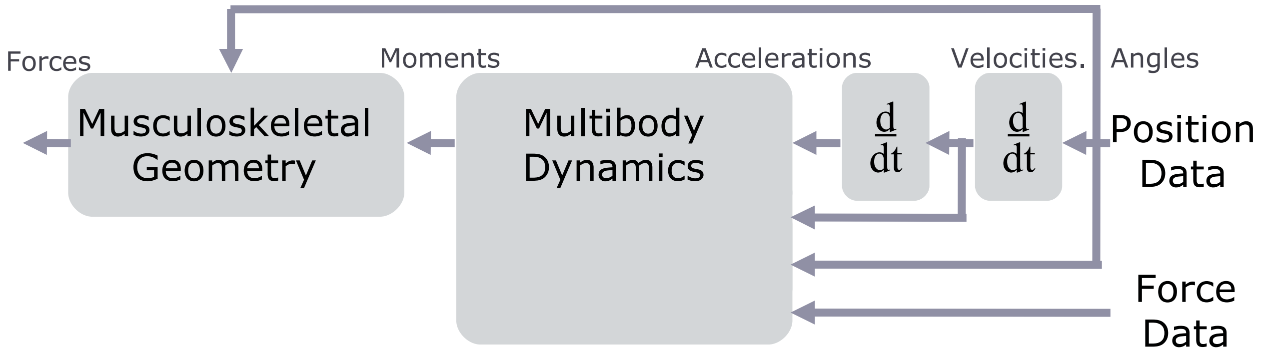


Going from joint kinematics to joint moments

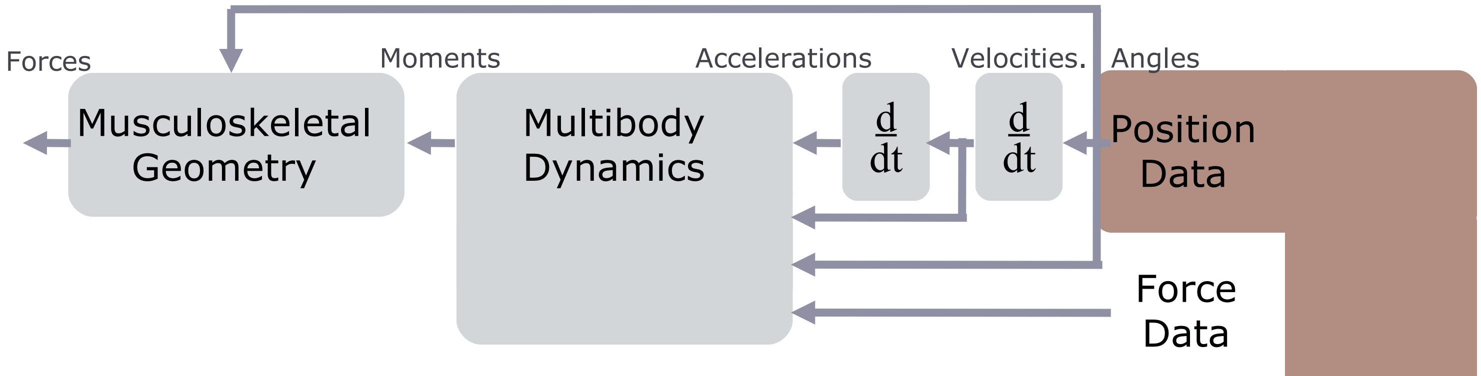


Going from joint moments to muscle forces

The Inverse Problem

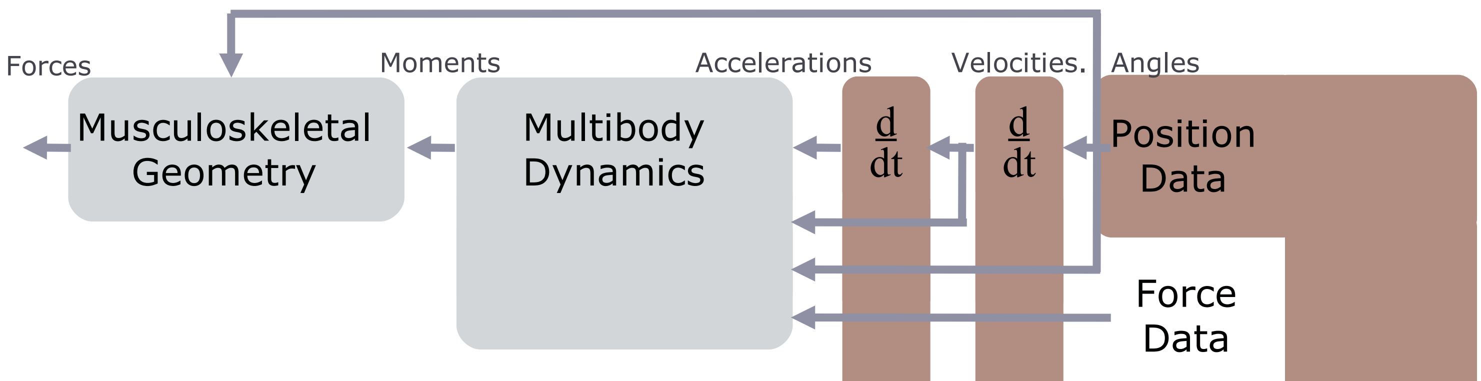


The Inverse Problem



Video Cameras
Reflective Markers

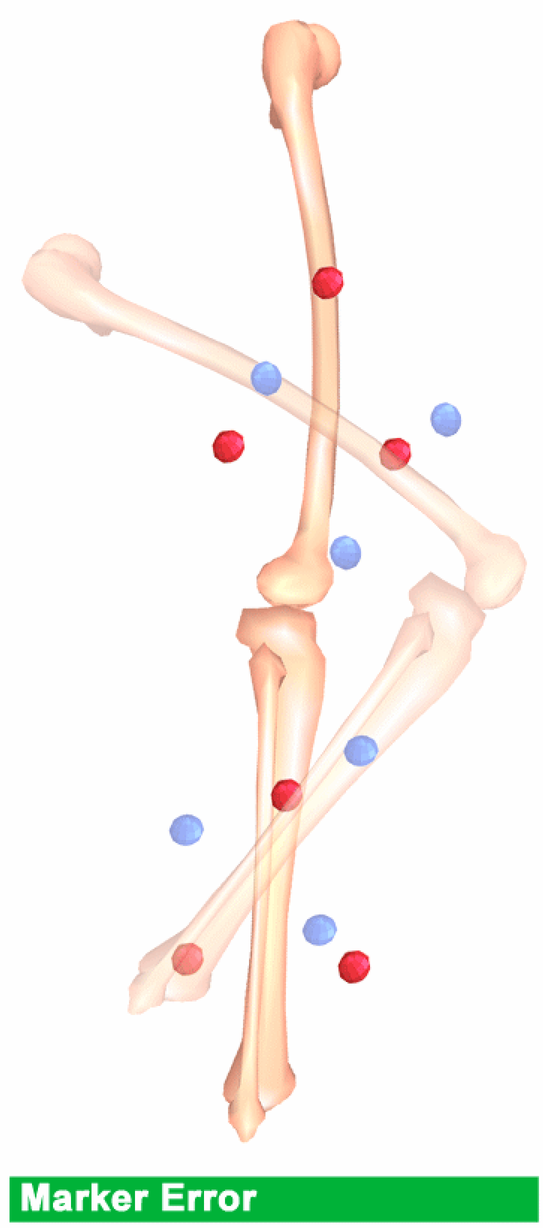
The Inverse Problem



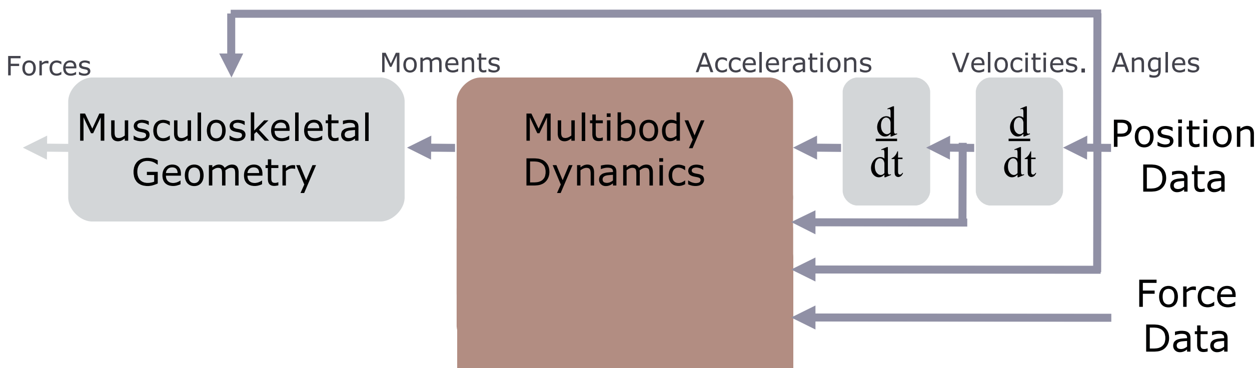
Inverse Kinematics

- Calculate model joint angles that best match the subject's measured motion.
- Uses a least-square optimization.
- Filter and differentiate joint kinematics

Computing Joint Kinematics

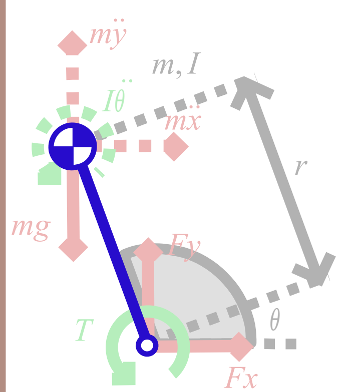


The Inverse Problem



Inverse Dynamics

Inverse Kinematics



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

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

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

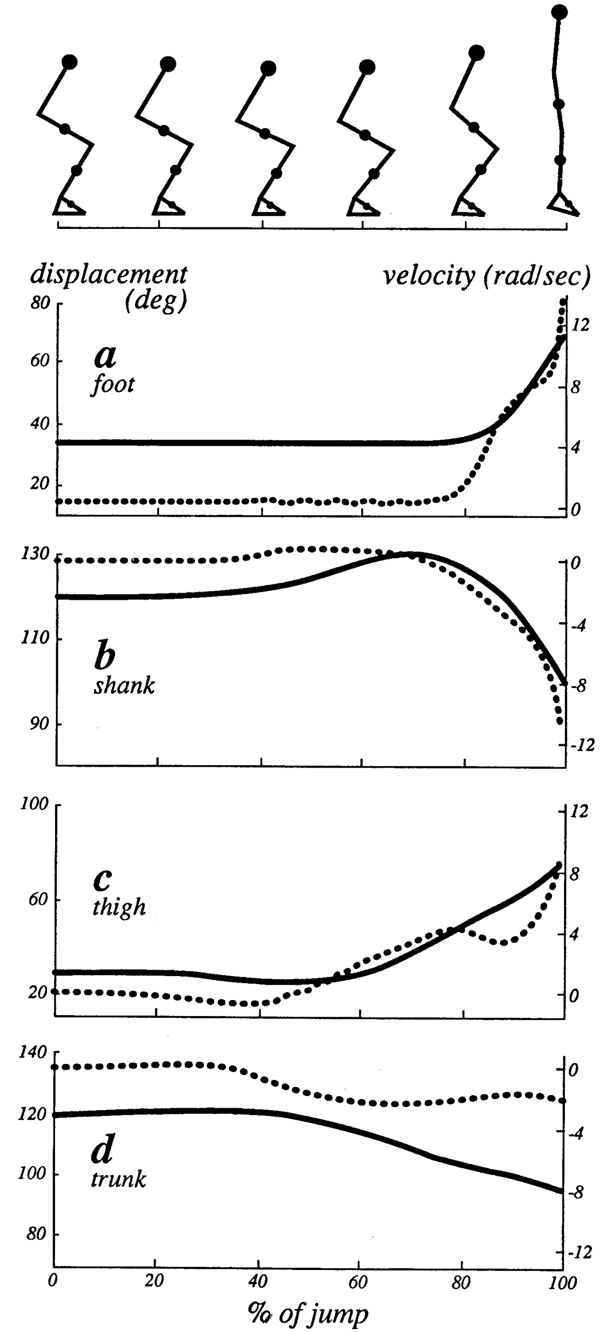
- Derive equations of motion defining the model
- Solve equations of motion for joint moments

Input: Experimental Results

Set-up

Experiment provides

- joint angles
- angular velocities
- ground reaction forces



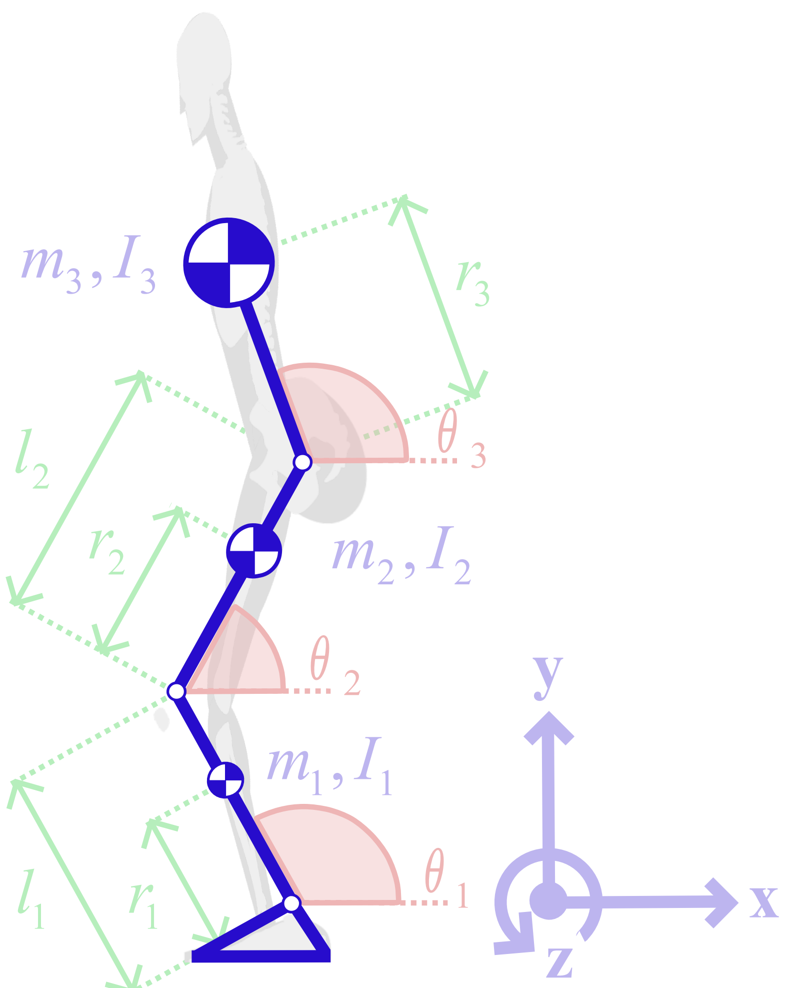
Equations: Multibody Dynamics

- Planar 3 degrees of freedom
- Position (orientation) in global coordinate system
- Segment length = l_i
- Distance to mass center = r_i
- Moments of inertia about mass center
- Foot has no mass and remains on ground

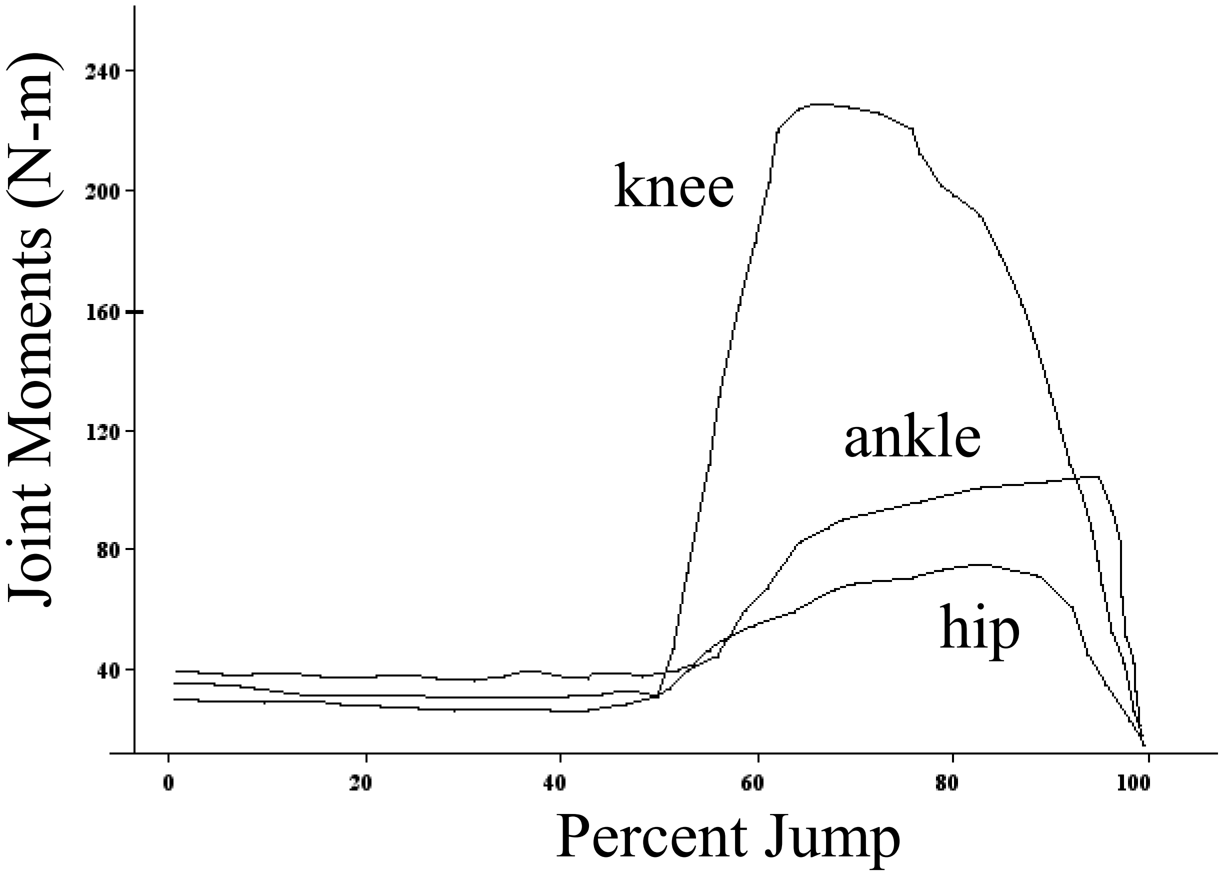
$x, \dot{x}, \ddot{x} \qquad \Sigma F_x = m\ddot{x}$

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

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

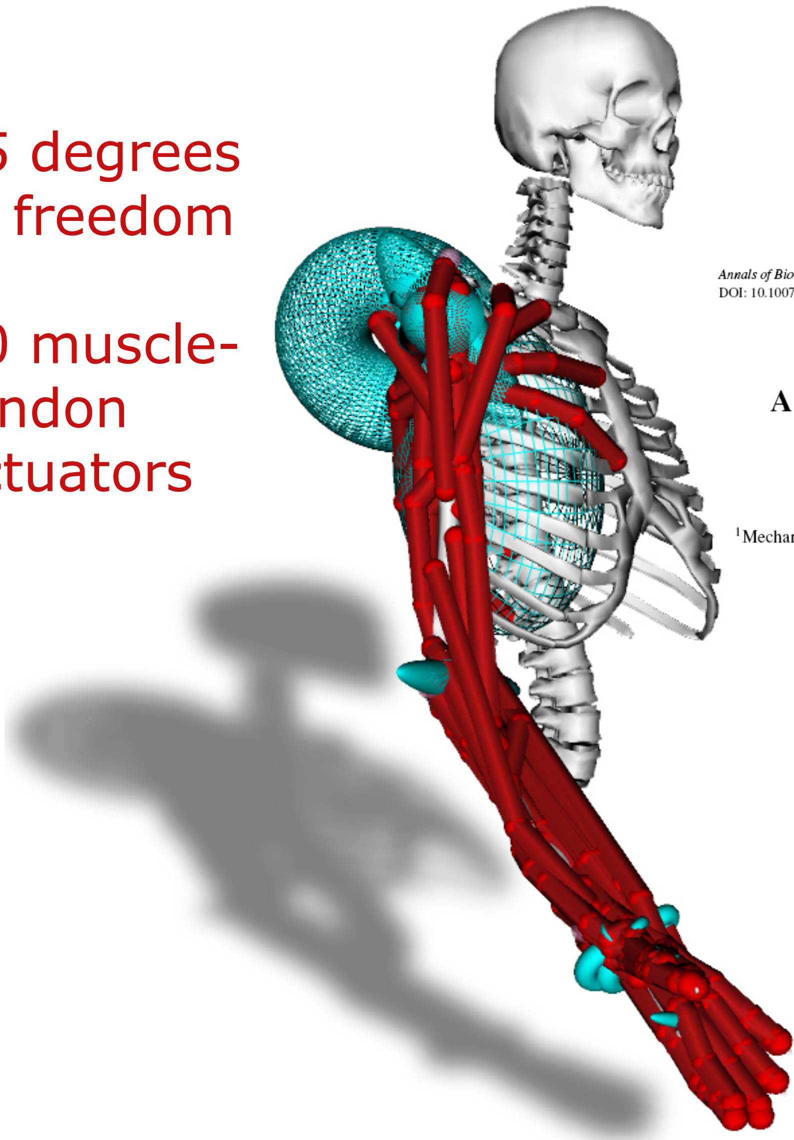


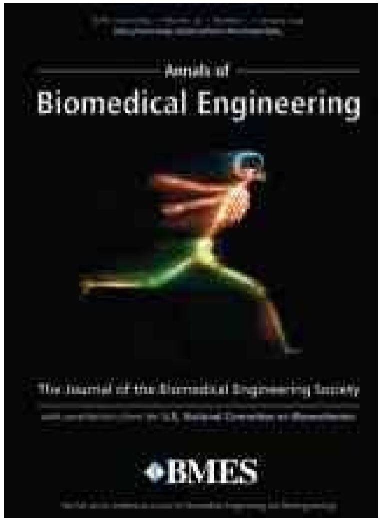
Output: Net Joint Moments



OpenSim Example: Upper Extremity

- 15 degrees of freedom
- 50 muscle-tendon actuators





Annals of Biomedical Engineering, Vol. 33, No. 6, June 2005 (© 2005) pp. 829–840
DOI: 10.1007/s10439-005-3320-7

A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Control

Katherine R. S. Holzbaur,[1] Wendy M. Murray,[2] and Scott L. Delp[2,3]

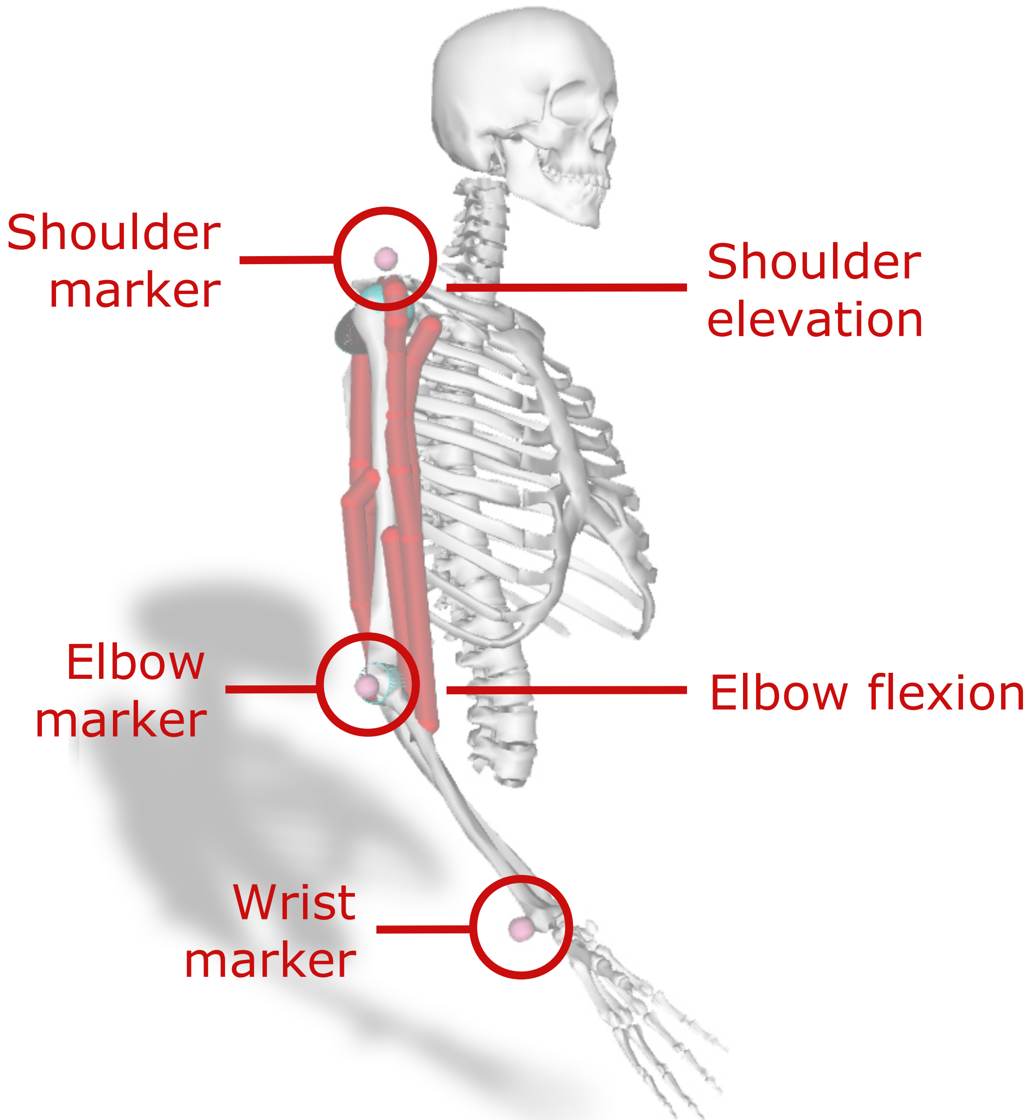
[1]Mechanical Engineering Department, Stanford University, Stanford, California 94305; [2]Bone and Joint Center, VA Palo Alto HCS, Palo Alto, California 94304; and [3]Bioengineering Department, Stanford University, Stanford, California 94305

Kate Wendy Scott

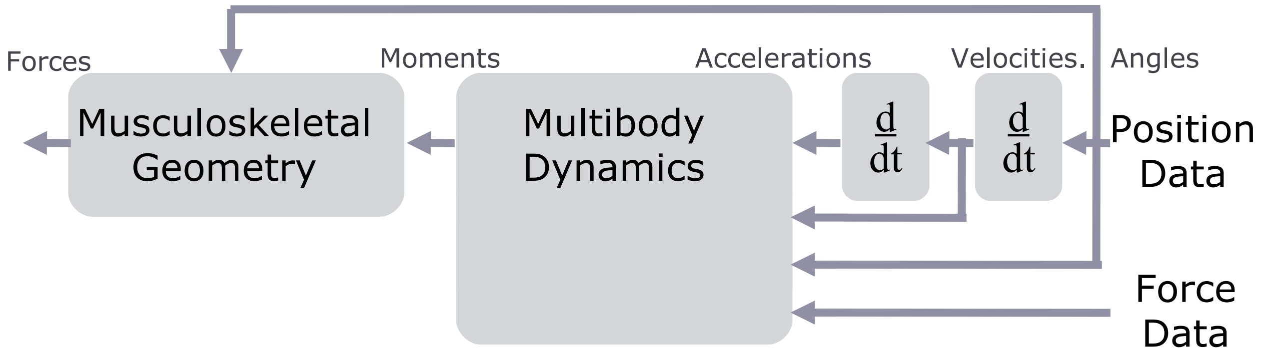
OpenSim Example: Upper Extremity

Inverse Kinematics
- 2 Joints
- 3 Markers

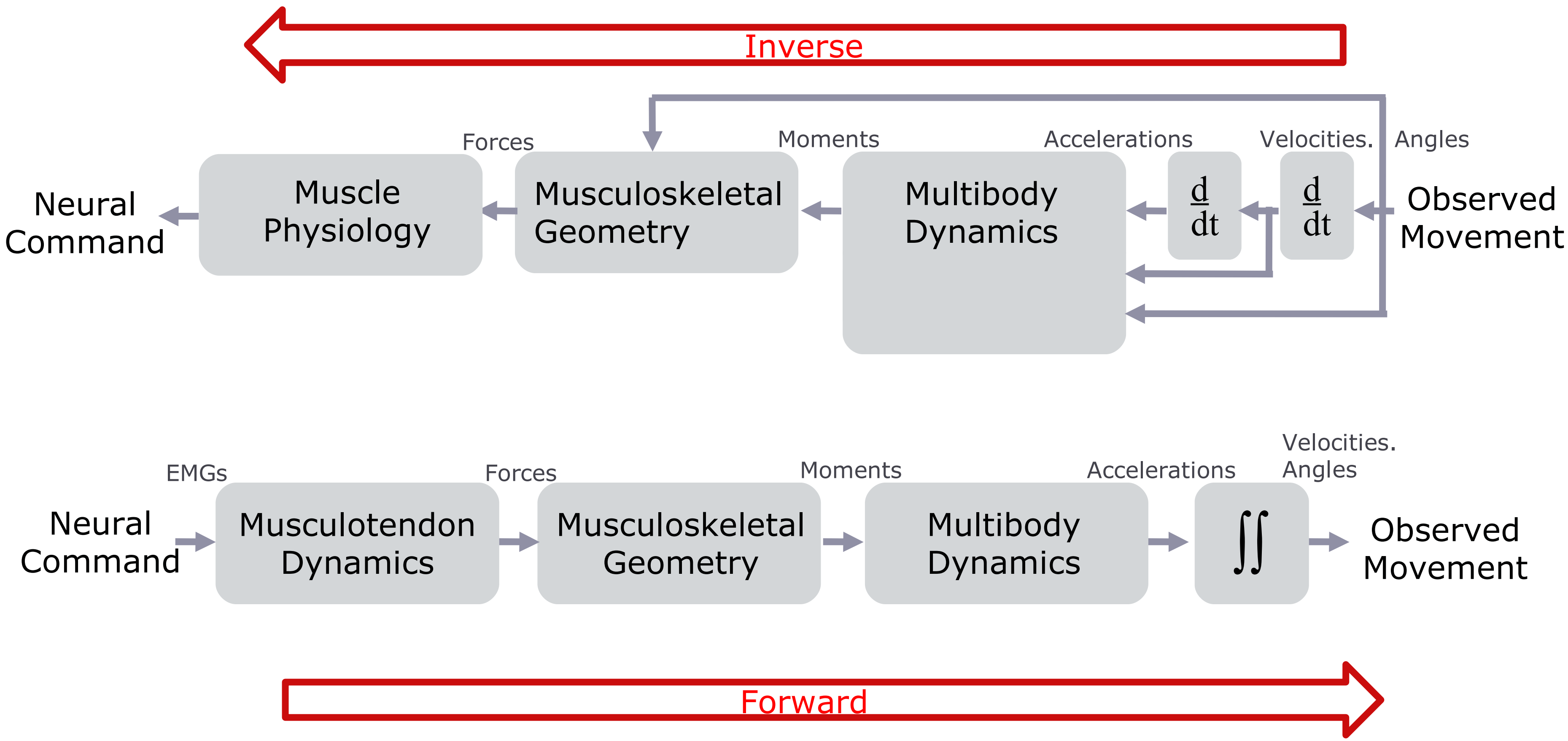
Inverse Dynamics
- 2 Joint Moments



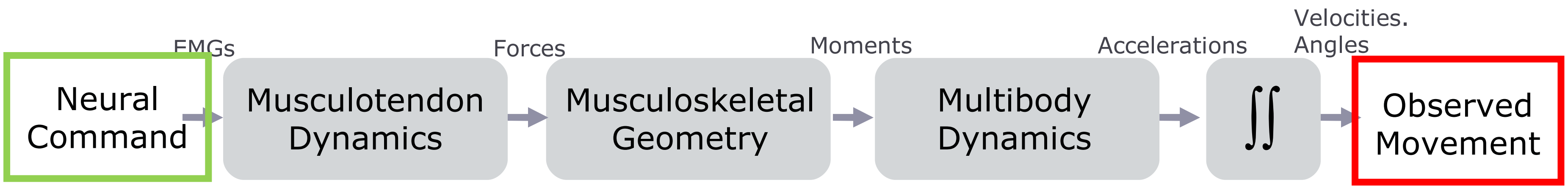
The Inverse Problem



Overview of Forward Dynamics



Elements of a Forward Simulation



CONTROLS

- muscle excitation

SOURCES

- EMG
- Static optimization
- Controller

INITIAL STATES

- joint angles
- joint velocities
- muscle activations
- fiber lengths

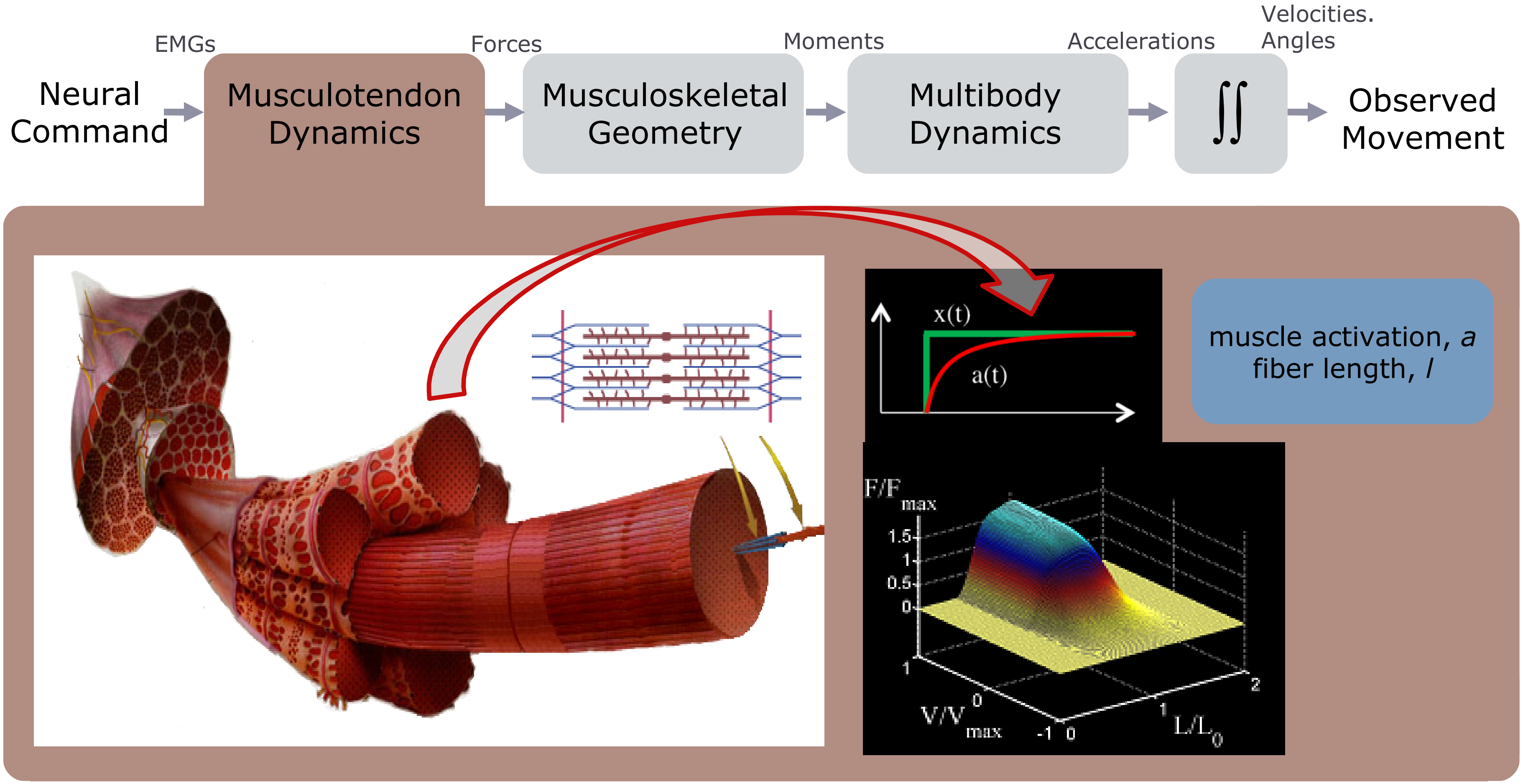
STATES

- joint angles …
- fiber lengths …

ANALYSES

- Point Kinematics
- Actuator Power

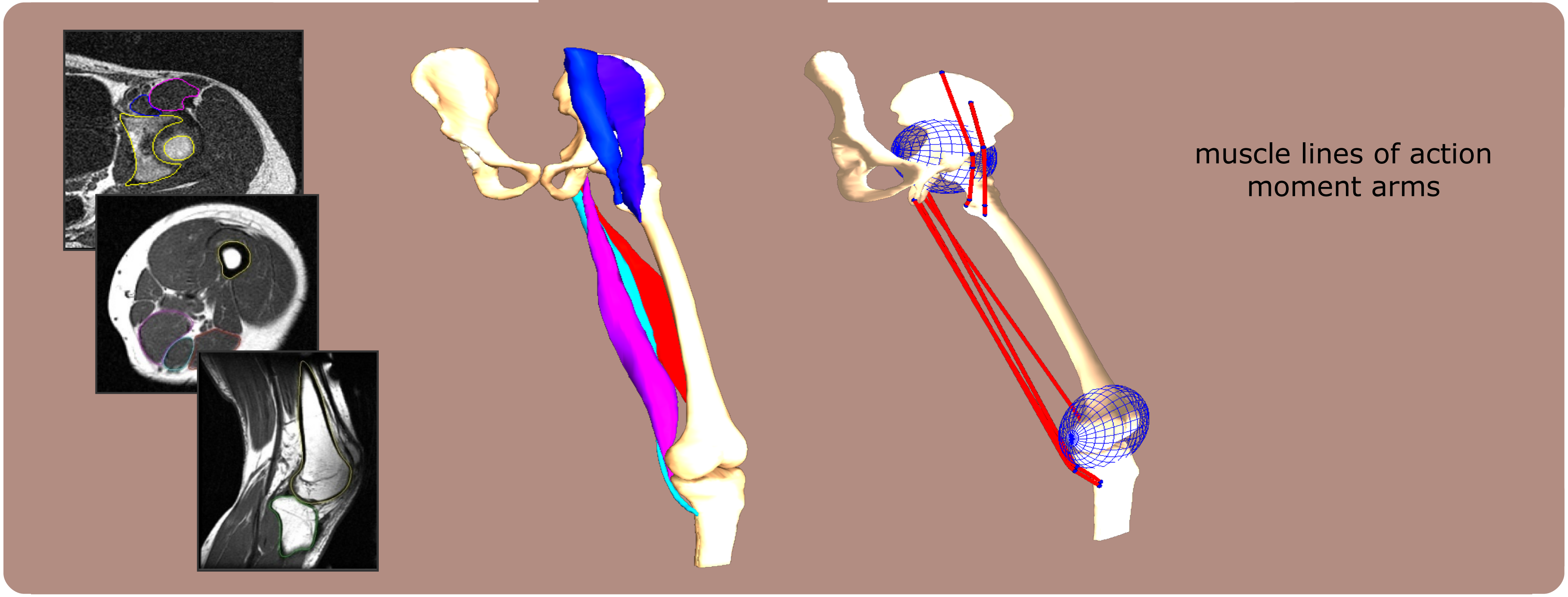
The Neuromusculoskeletal System



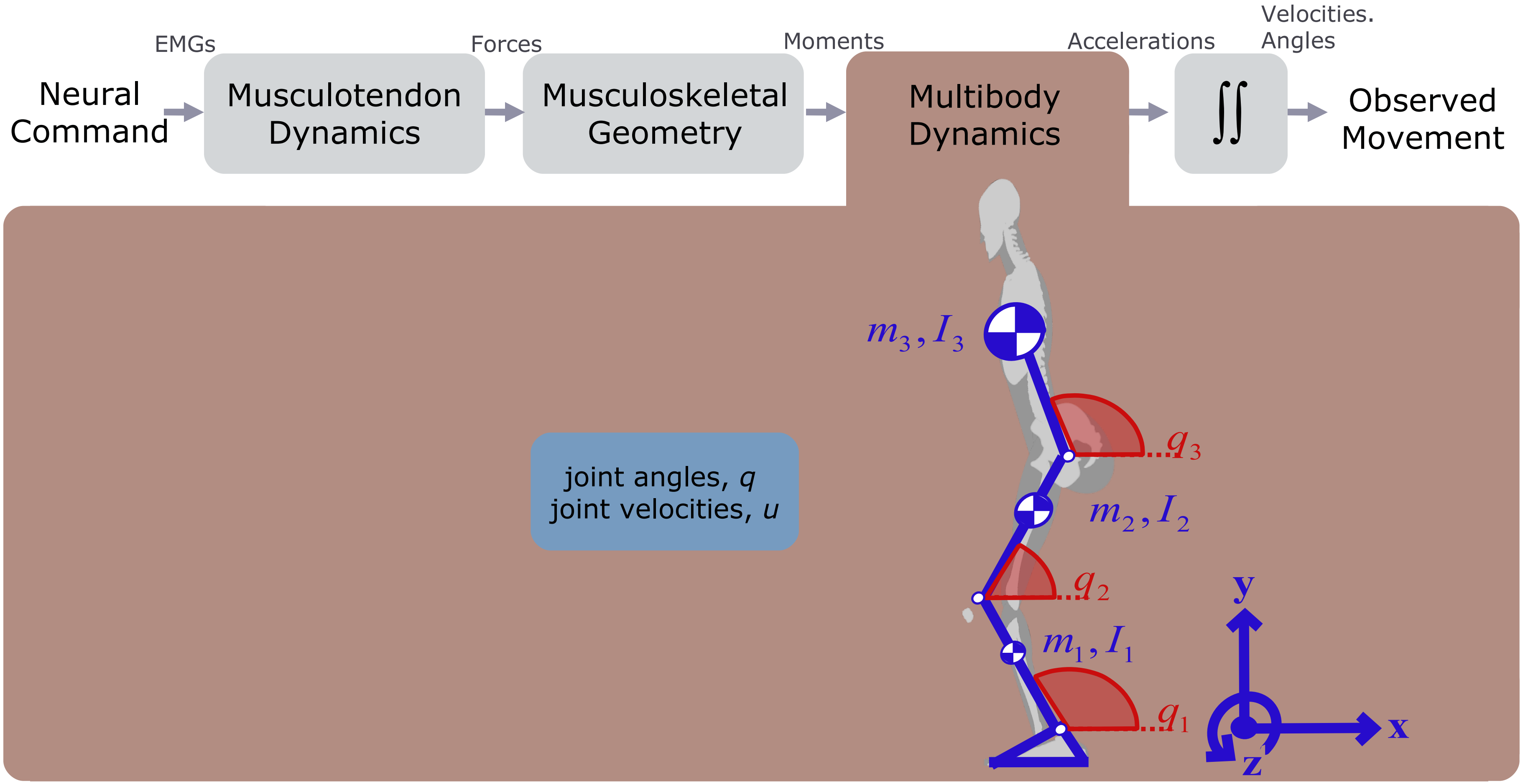
Science

The Neuromusculoskeletal System

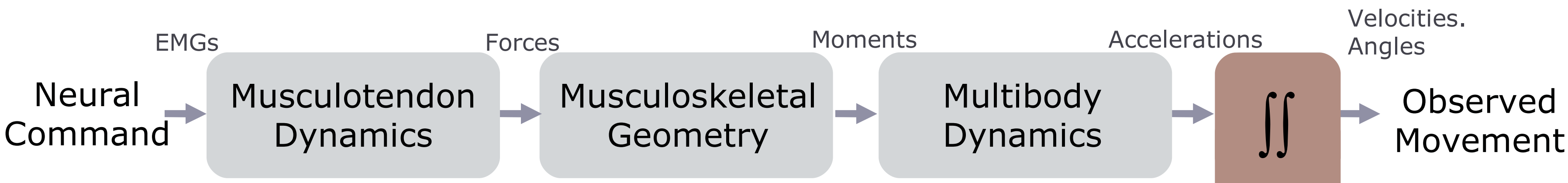
Neural Command → EMGs → Musculotendon Dynamics → Forces → Musculoskeletal Geometry → Moments → Multibody Dynamics → Accelerations → ∫∫ → Velocities. Angles → Observed Movement



The Neuromusculoskeletal System



Forward Integration



Integration of system equations:

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

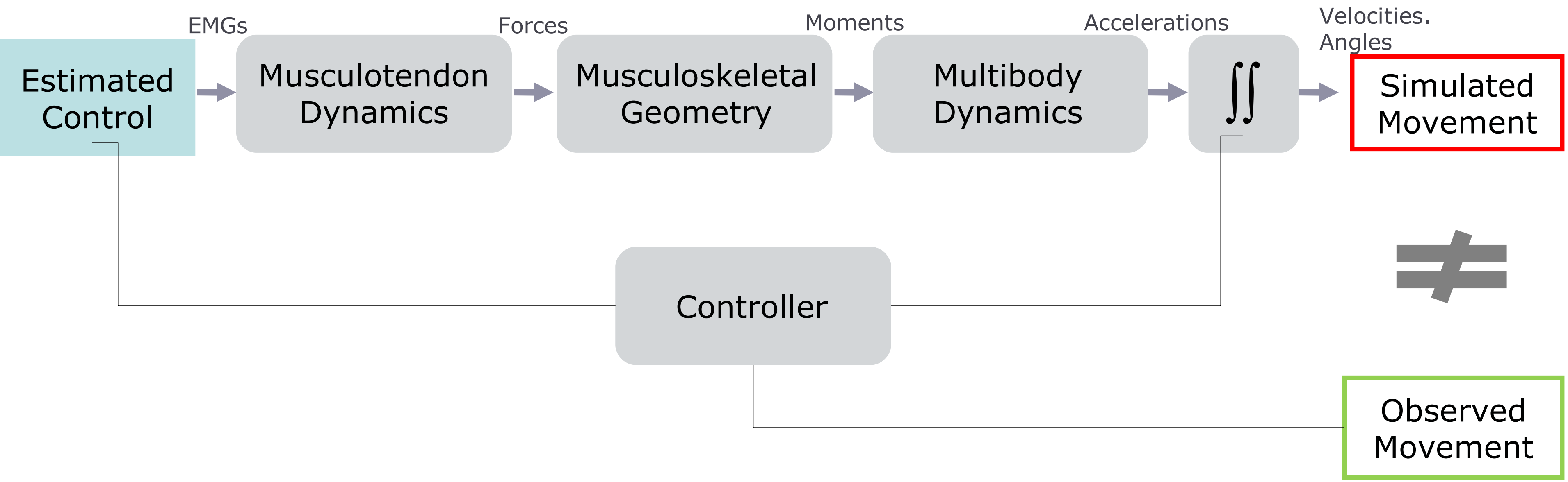
$$\dot{a} = \mathrm{A}\,(a,x)$$

$$\dot{l} = \Lambda\,(a,l,q)$$

Numerical Integration:

- 5^{th} order Runge-Kutta-Feldberg Variable Step Integrator

How did your Simulations Compare to Observed?



Solution: Close the loop!

Computed Muscle Control

Estimated Control → Musculotendon Dynamics → (Forces) → Musculoskeletal Geometry → (Moments) → Multibody Dynamics → (Accelerations) → ∬ → (Velocities. Angles) → Simulated Movement

Accelerations(t+T)

Controls=x(t)

Static Optimization+

PD Control

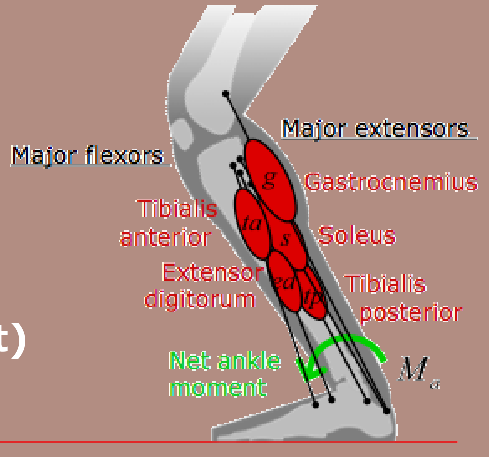
Observed Movement

Min: J(f(t+T))

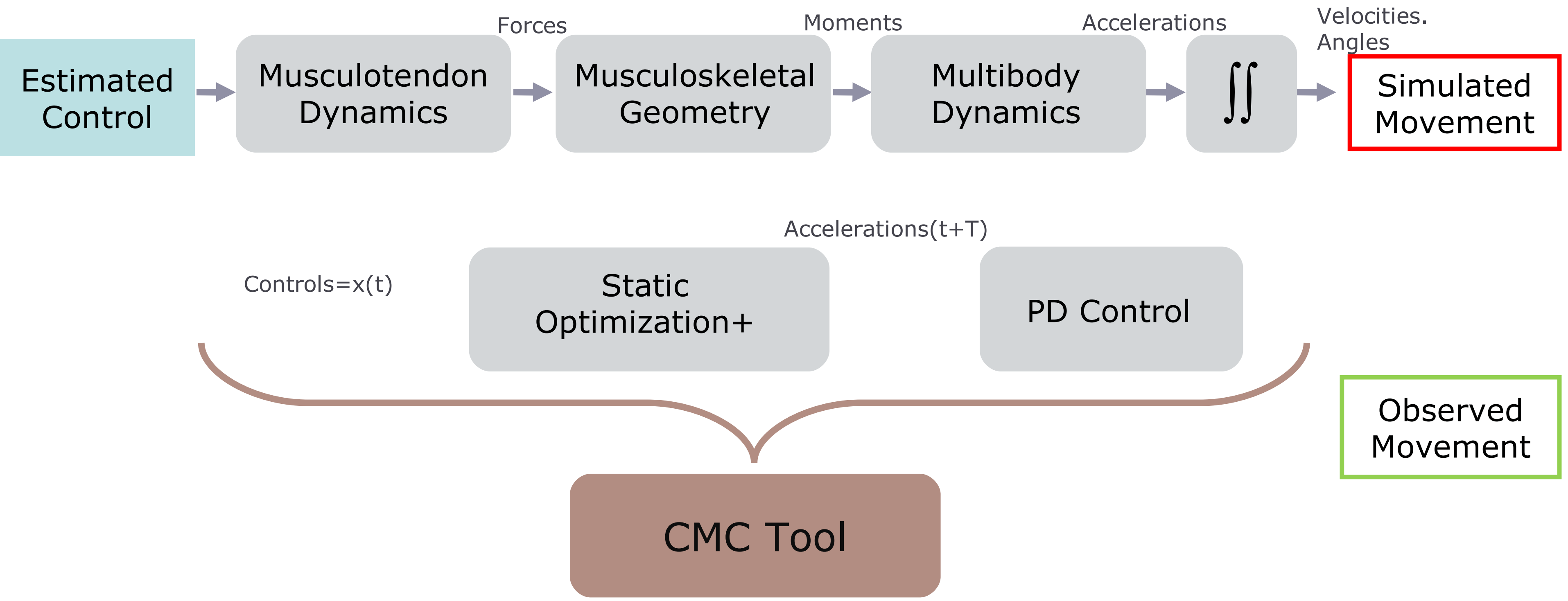
Subject to:

$f(x_{min}(t)) < f(t+T) < f(x_{max}(t))$

f(t+T) — Musculotendon^{-1} Dynamics — **x(t)**



Computed Muscle Control Tool:

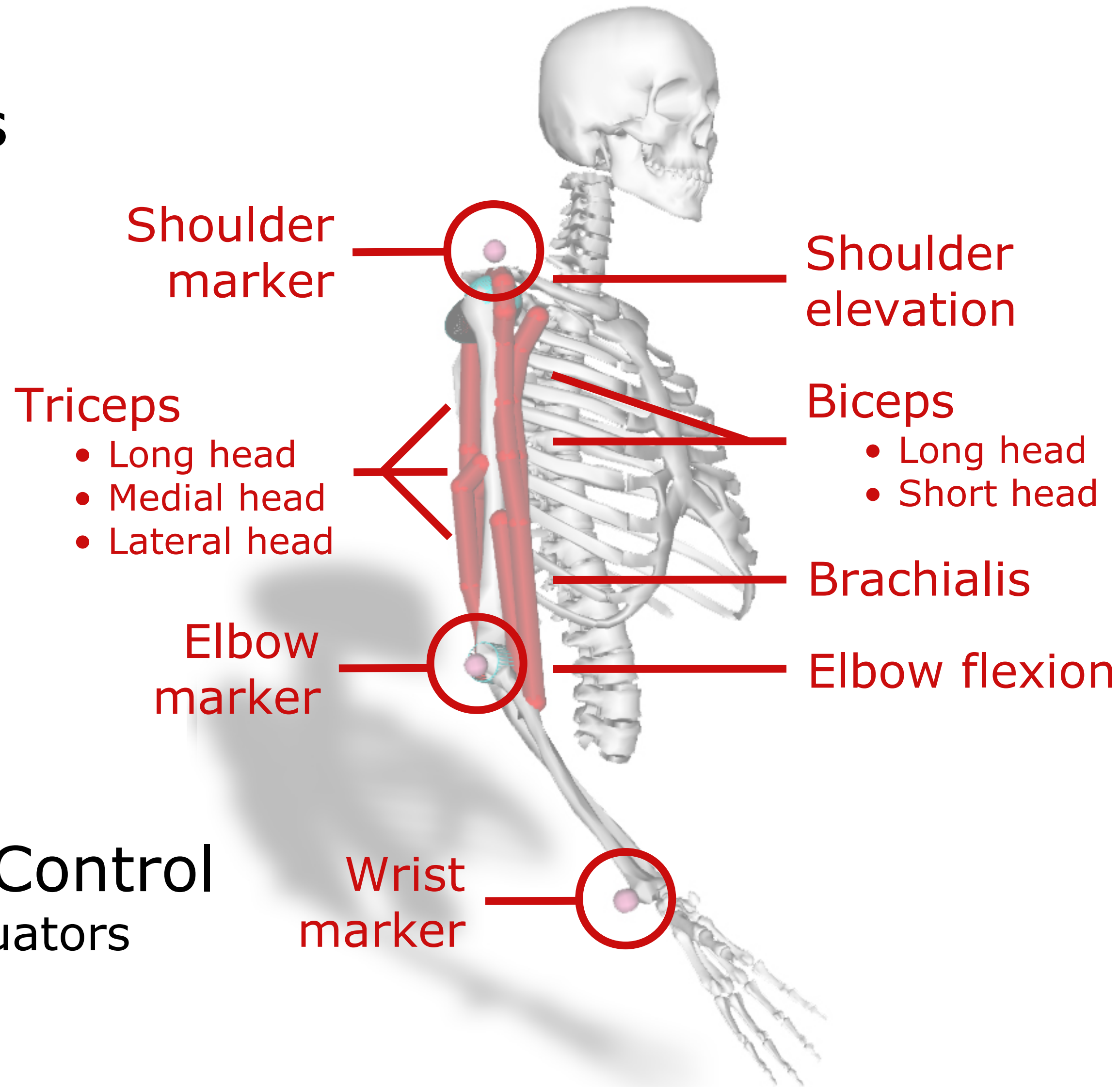


OpenSim Example: Upper Extremity

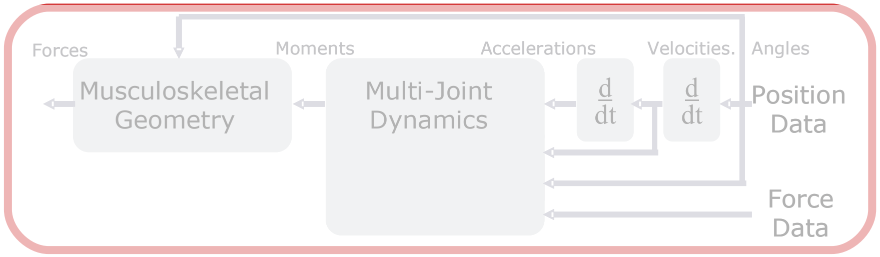
Inverse Kinematics
- 2 Joints
- 3 Markers

Inverse Dynamics
- 2 Joint Moments

Computed Muscle Control
- 6 Muscle-tendon actuators



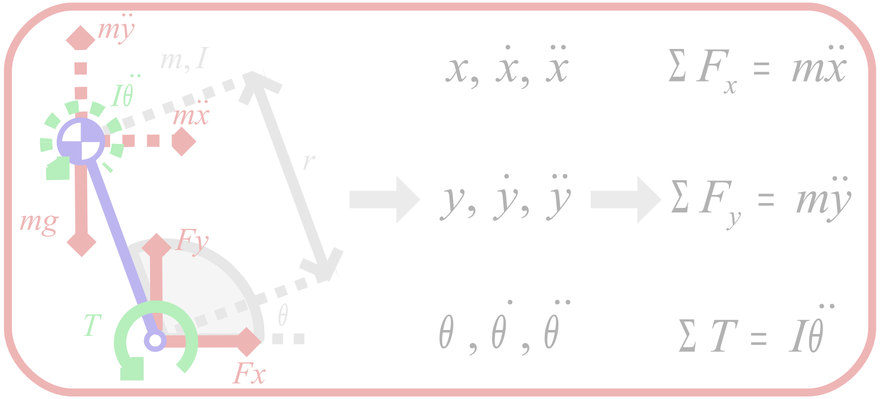
Main Points of the Session



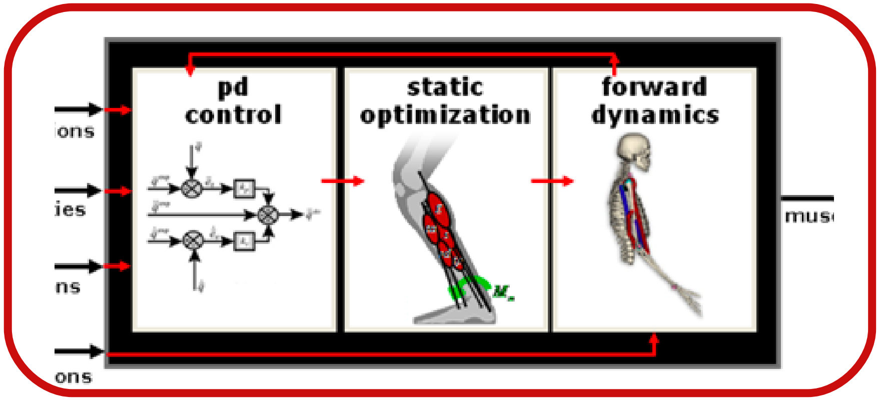
The inverse problem:
from position data to joint moments

Inverse kinematics determines joint kinematics from subject motion

Main Points of the Session (cont.)



Inverse dynamics determines joint moments from joint kinematics



CMC estimates muscle forces from joint moments & tracking motion

Continue Using OpenSim

- **User's Forum on SimTK.org**
- **OpenSim Development Team**
- **OpenSim Workshops at Stanford University**
  - **March 16-18, 2011**
  - **August 15-17, 2011**
- **OpenSim Fellows Program – Summer 2011**