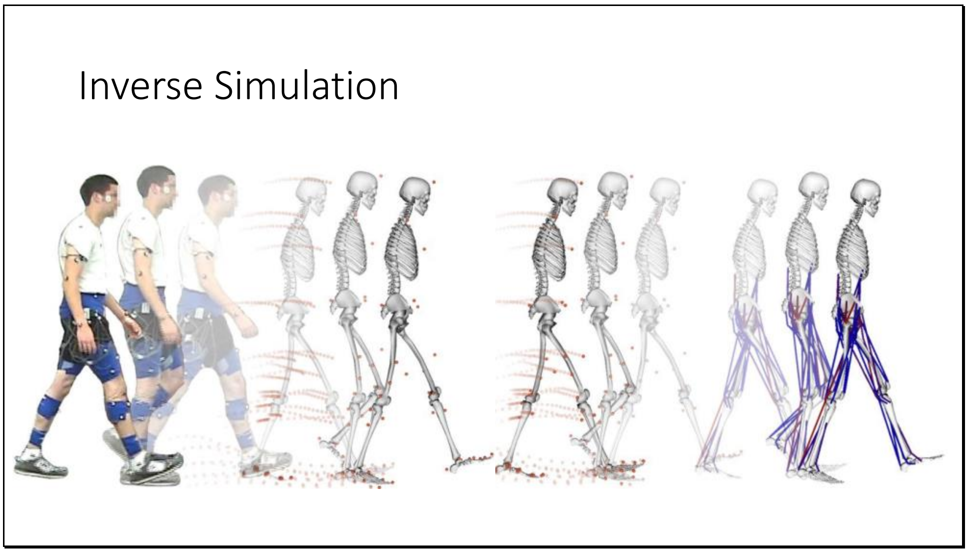
Slide 1

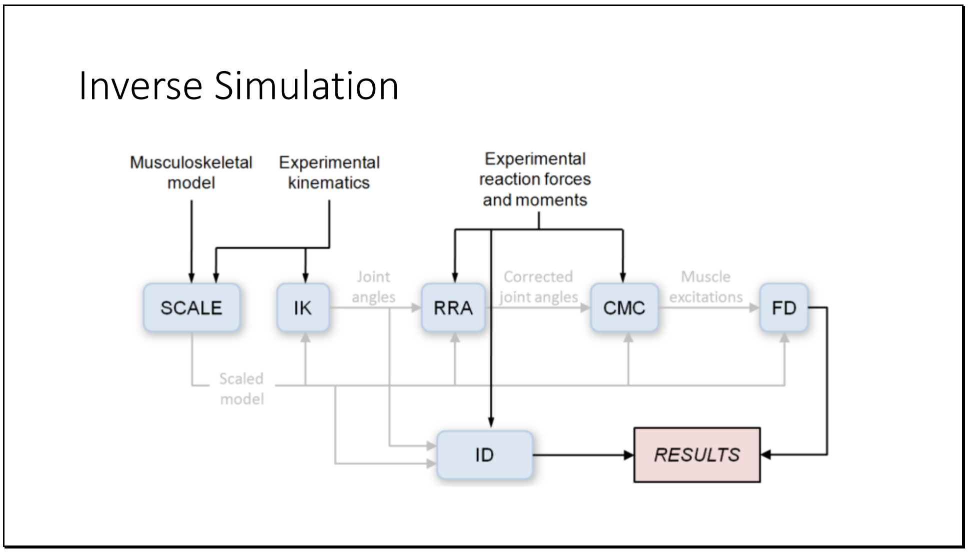
Inverse Simulation



In many cases, you will use OpenSim to analyze experimental data that you have collected in your laboratory. This data typically includes:

- Marker trajectories or joint angles from motion capture
- Force data, typically ground reaction forces and moments and/or centers of pressure
- Electromyography

Slide 3

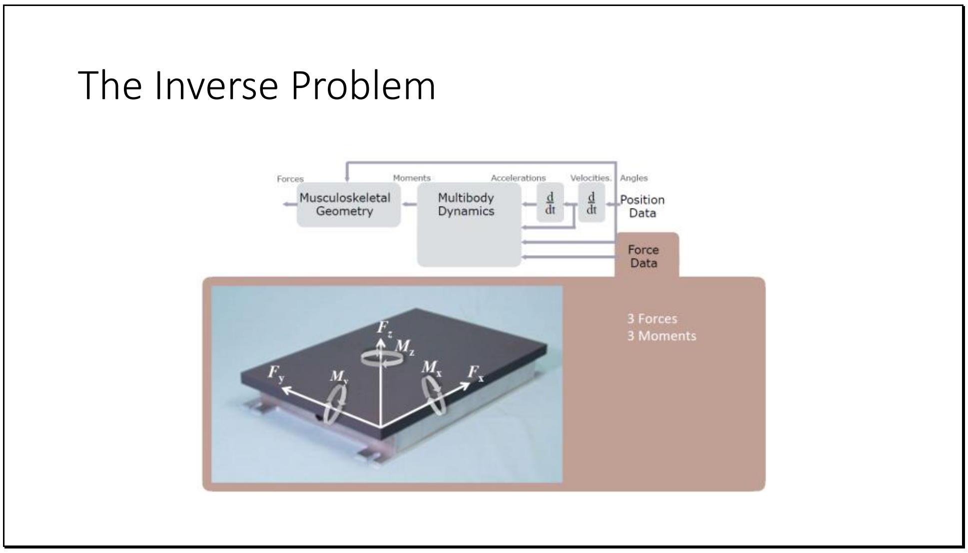


OpenSim enables researchers to solve the Inverse Dynamics problem, using experimental measured subject motion and forces to generate the kinematics and kinetics of a musculoskeletal model (see figure below). In inverse dynamics, experimentally measured marker trajectories and force data are use to estimate a model's kinematics and kinetics.

RRA, CMC and FD will be described in the next lecture.

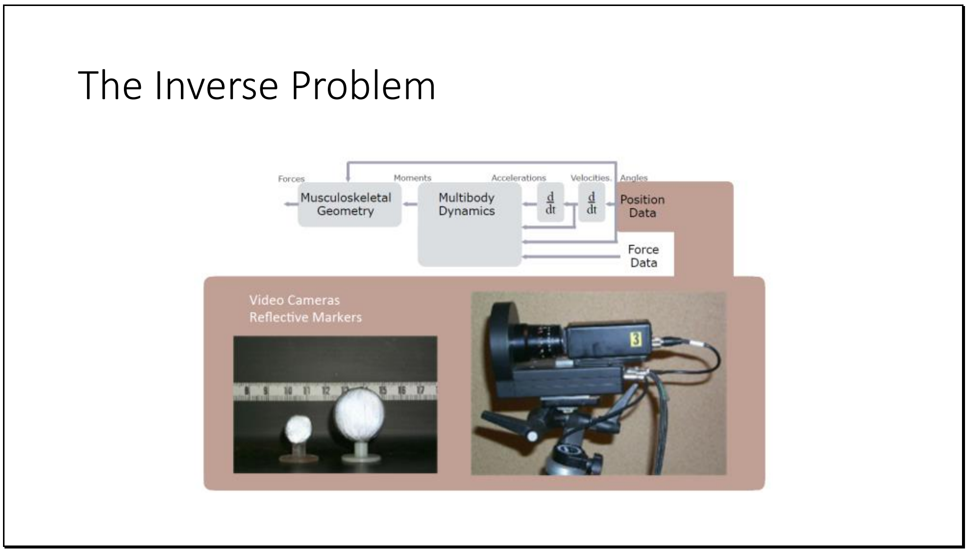
http://simtk-confluence.stanford.edu:8080/display/OpenSim/Overview+of+the+OpenSim+Workflow

Slide 4



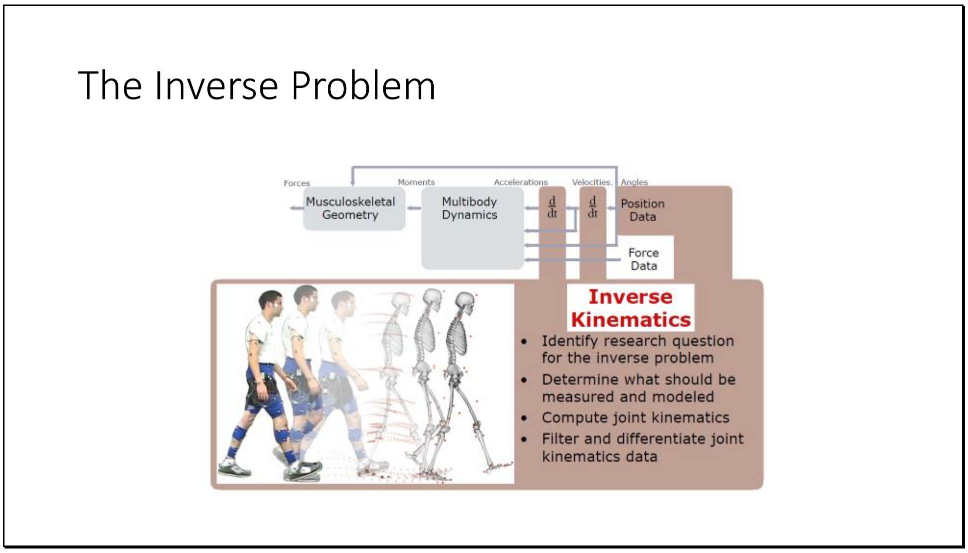
External load data (i.e., ground reaction forces, moments, and center of pressure location). Note that it is necessary to measure and apply or model all external forces acting on a subject during the motion to calculate accurate joint torques and forces.

Slide 5



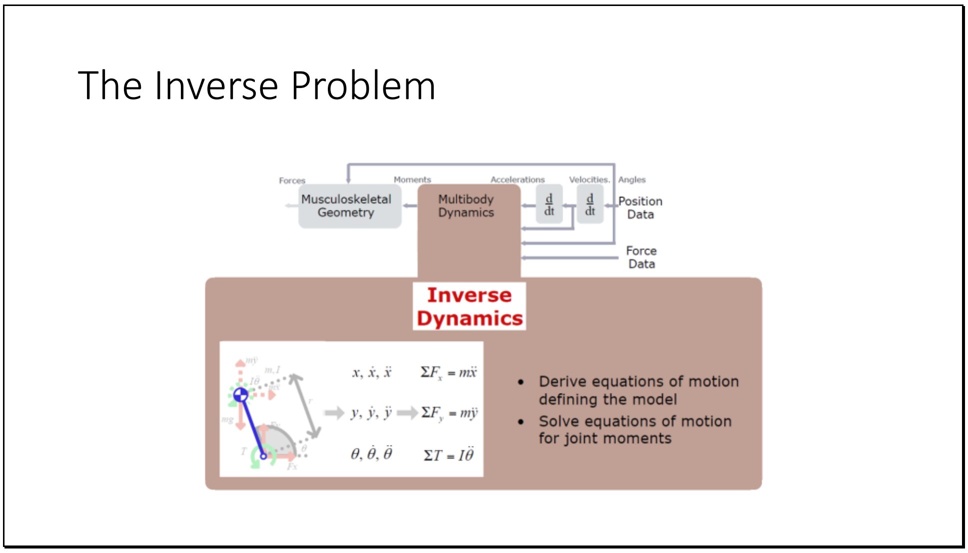
If we know the positions of three points on a rigid body in three-dimensional space, we can completely determine that rigid body's position and orientation. Our marker set is designed to give us the locations of at least three markers on each rigid body segment so that the position and orientation of each body segment of interest can be completely determined. To this end, motion capture systems (Vicon, Motion Analysis) are used to track the position of the markers attached on the body.

Slide 6



The first step of the analysis is to estimate the joint angles of the model and their derivatives with respect to time (Inverse Kinematics performs this task).

Slide 7

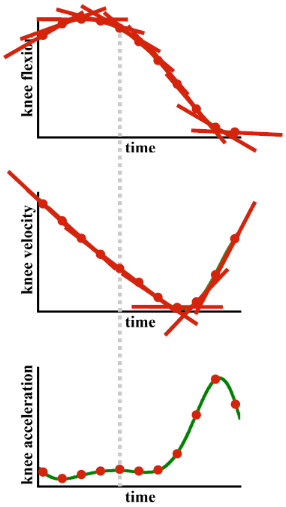


Classical mechanics mathematically expresses the mass-dependent relationship between force and acceleration, **F** = m**a**, with equations of motion. The Inverse Dynamics (ID) Tool solves these equations, in the inverse dynamics sense, to yield the net forces and torques at each joint which produce the movement.

Slide 8

Kinematics

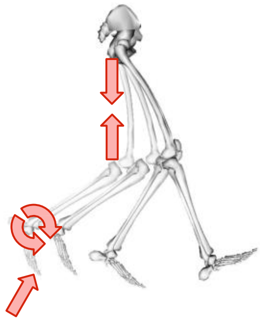
- Coordinate
  - Joint angle or distance specifying relative orientation or location of two body segments
- Coordinate velocity
  - Derivative (rate of change) of a coordinate with respect to time
- Coordinate acceleration
  - Time derivative of a coordinate velocity with respect to time
- Kinematics
  - Set of all coordinates and their velocities and accelerations



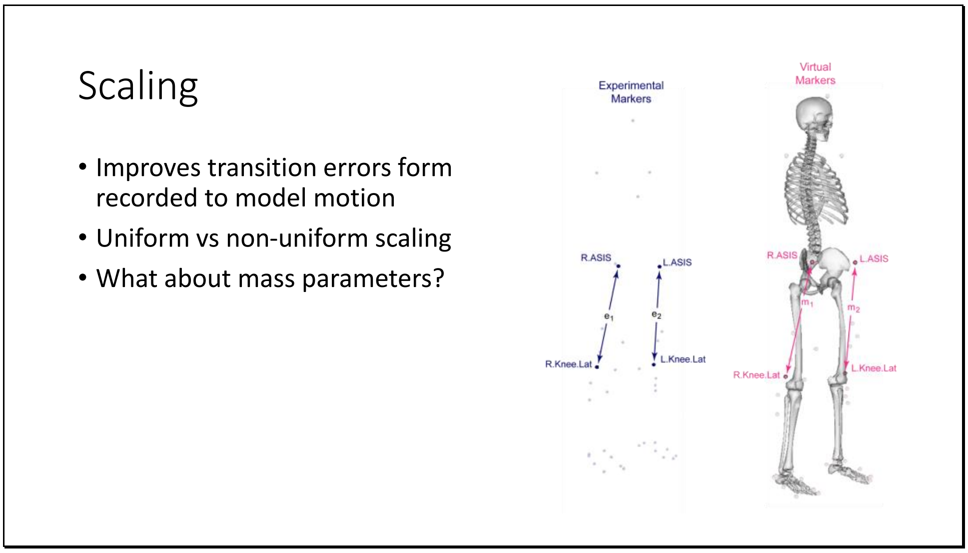
Slide 9

Kinetics

- Kinetics
  - Forces and torques cause the model to accelerate
- Force
  - Applied to points (e.g., ground reactions) or between points (e.g., muscles)
- Torque
  - Applied to a coordinate (e.g., joint torque)

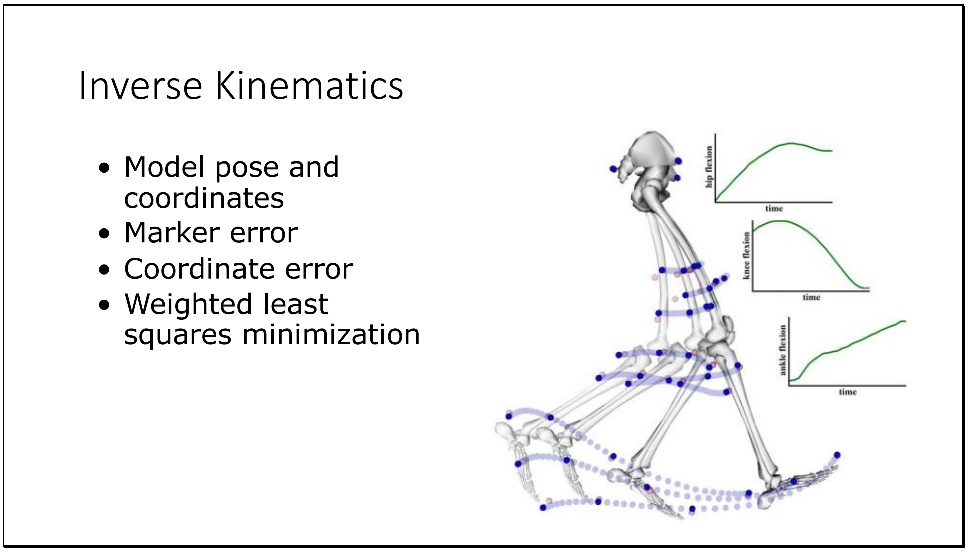


Slide 10



If you are using a generic model from the existing library of models, the next step is to scale the model to match the experimental data collected for your subject—functionality provided by the Scale Tool in OpenSim. The purpose of scaling a generic musculoskeletal model is to modify the anthropometry, or physical dimensions, of the generic model so that it matches the anthropometry of a particular subject. Scaling is one of the most important steps in solving inverse kinematics and inverse dynamics problems because these solutions are sensitive to the accuracy of the scaling step. In OpenSim, the scaling step adjusts both the mass properties (mass and inertia tensor), as well as the dimensions of the body segments

Slide 11



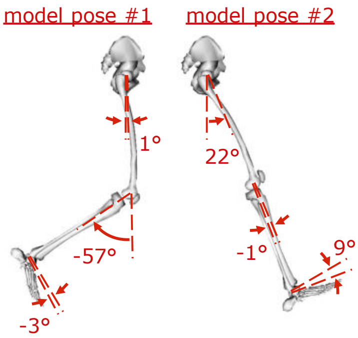
The Inverse Kinematics (IK) Tool in OpenSim finds values for the generalized coordinates (joint angles and positions) in the model that best match the experimental kinematics recorded for a particular subject (see figure below). The experimental kinematics targeted by IK can include experimental marker positions, as well as experimental generalized coordinate values (joint angles). The IK Tool goes through each time step of motion and computes generalized coordinate values which position the model in a pose that "best matches" experimental marker and coordinate values for that time step. Mathematically, the "best match" is expressed as a weighted least-squares problem, whose solution aims to minimize both marker and coordinate errors. Experimental markers are matched by model markers throughout the motion by varying the generalized coordinates (e.g., joint angles) through time.

Obtaining accurate results from the IK Tool is essential for using later tools like Static Optimization, Residual Reduction Algorithm, and Computed Muscle Control.

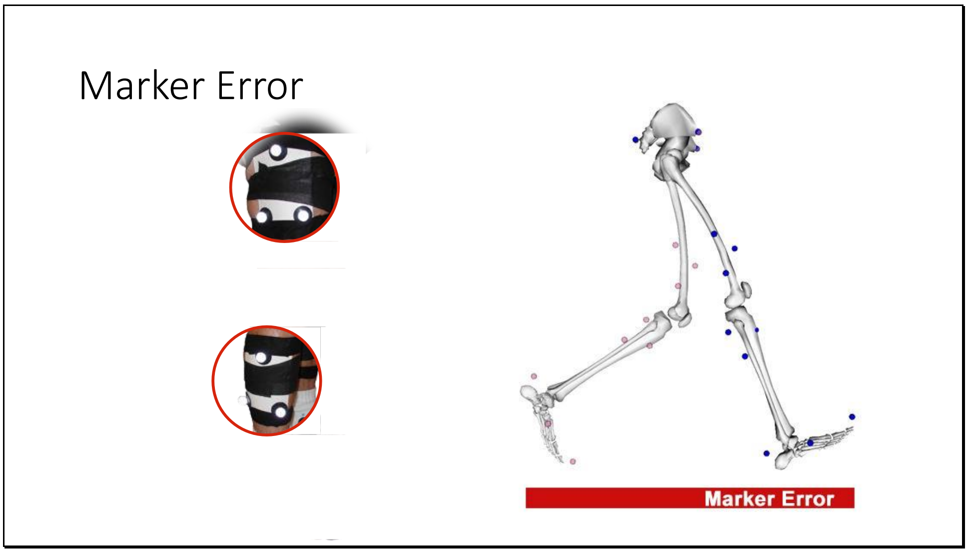
Slide 12

Model Pose and Coordinates

- Model Pose
  - Orientations and locations of body segments in the model
  - Defined by set of model coordinates
- Coordinate
  - Joint angle or distance specifying relative orientation or location of two body segments

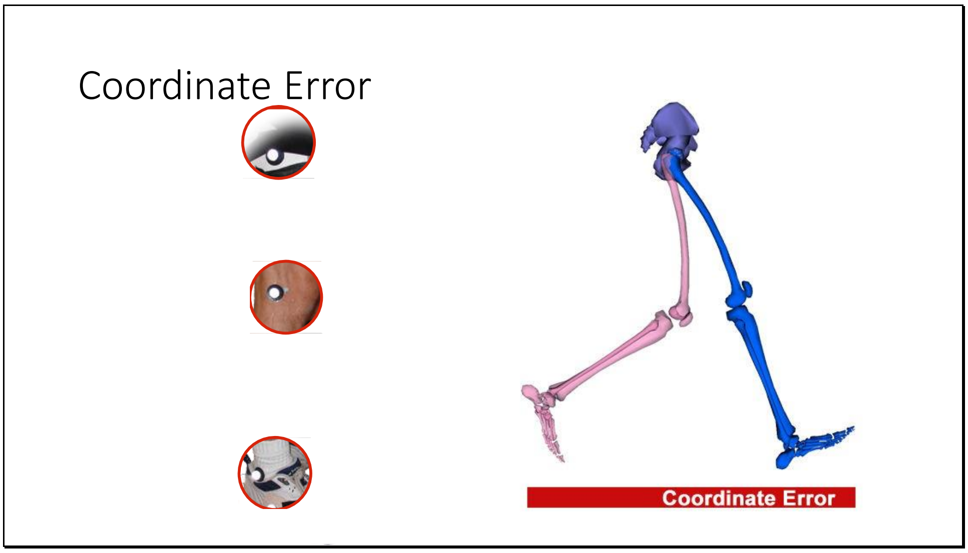


Slide 13



A *marker error* is the distance between an experimental marker and the corresponding marker on the model when it is positioned using the generalized coordinates computed by the IK solver. Each marker has a weight associated with it, specifying how strongly that marker's error term should be minimized.

Slide 14



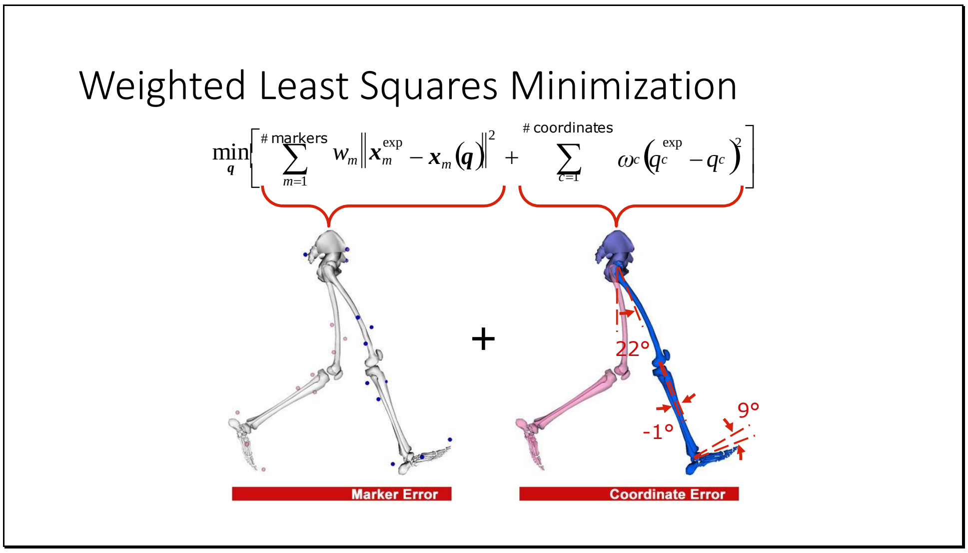
A *coordinate error* is the difference between an experimental coordinate value and the coordinate value computed by IK.

What are "experimental coordinate values?" These can be joint angles obtained directly from a motion capture system (i.e., built-in mocap inverse kinematics capabilities), or may be computed from experimental data by various specialized algorithms (e.g., defining anatomical coordinate frames and using them to specify joint frames that, in turn, describe joint angles) or by other measurement techniques that involve other measurement devices (e.g., a goniometer). A fixed desired value for a coordinate can also be specified (e.g., if you know a specific joint's angle should stay at 0˚). The inclusion of experimental coordinate values is optional; the IK tool can solve for the motion trajectories using marker matching alone.

A distinction should be made between *prescribed* and *unprescribed coordinates*. A prescribed coordinate (also referred to as a *locked coordinate*) is a generalized coordinate whose trajectory is known and which will not be computed using IK. It will get set to its exact trajectory value instead. This can be useful when you have enough confidence in some generalized coordinate value that you don't want the IK solver to change it.

An *unprescribed coordinate* is a coordinate which is not prescribed, and whose value is computed using IK.
Using these definitions, only *unprescribed coordinates* can vary and so only they appear in the least squares equation solved by IK. Each unprescribed coordinate being compared to an experimental coordinate must have a weight associated with it, specifying how strongly that coordinate's error should be minimized.

Slide 15



The weighted least squares problem solved by IK

where **q** is the vector of generalized coordinates being solved for, $\mathbf{x}_i^{exp}$ is the experimental position of marker i, $\mathbf{x}_i(\mathbf{q})$ is the position of the corresponding marker on the model (which depends on the coordinate values), q_j^{exp} is the experimental value for coordinate j. Prescribed coordinates are set to their experimental values. For instance, in the gait2354 and gait2392 examples, the subtalar and metatarsophalangeal (mtp) joints are locked and during IK they are assigned the prescribed value of 0˚.

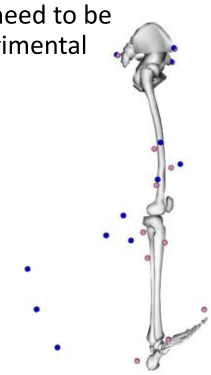
The marker weights (w_i's) and coordinate weights (ω_j's) are specified in the <IKMarkerTask> and <IKCoordinateTask> tags, respectively. These are all specified within a single <IKTaskSet> tag. This least squares problem is solved using a general quadratic programming solver, with a convergence criterion of 0.0001 and a limit of 1000 iterations. These are currently fixed values that cannot be changed in the XML files.

Slide 16

Inverse Kinematics Exercise

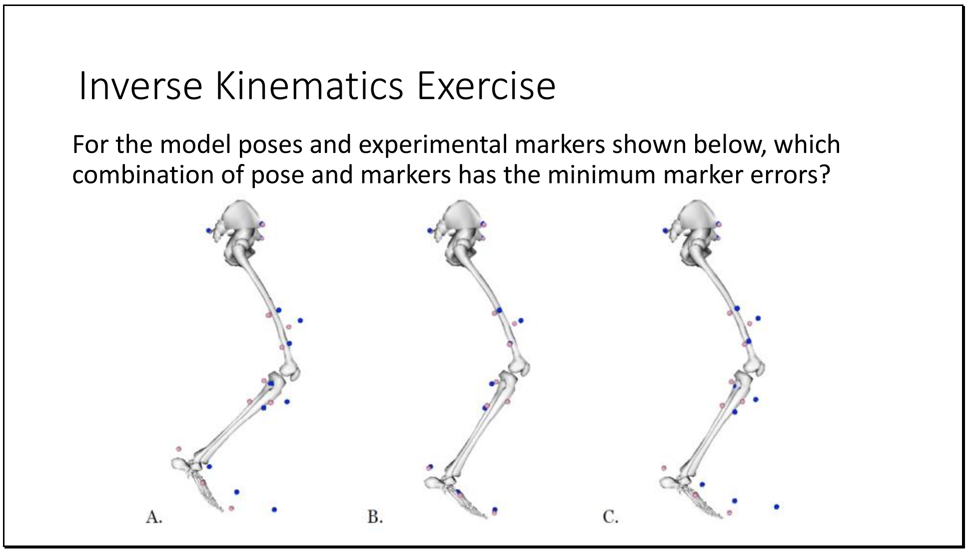
- For the model shown on the right, which coordinate(s) need to be adjusted to create a model pose best matches the experimental markers as shown at the beginning swing phase?

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



The correct is E. Knee and ankle

Slide 17


Inverse Kinematics Exercise
For the model poses and experimental markers shown below, which combination of pose and markers has the minimum marker errors?
A.
B.
C.

B.

Slide 18

Inverse Kinematics Exercise

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

D. All of the above

Slide 19

Tips and Tricks

- Marker weights are relative
- Check max and RMS marker errors in messages window
- Weight “motion” marker triads on body segments higher than anatomical markers
- Max marker error should be < 2 cm with RMS error <1 cm

Data Collection and Other Preparation:

- When collecting experimental data, place three non-collinear markers per body segment that you want to track. You need at least three markers to track the 6-degree-of-freedom motion (position and orientation) of a body segment.
- Place markers on anatomical locations with minimum skin/muscle motion.

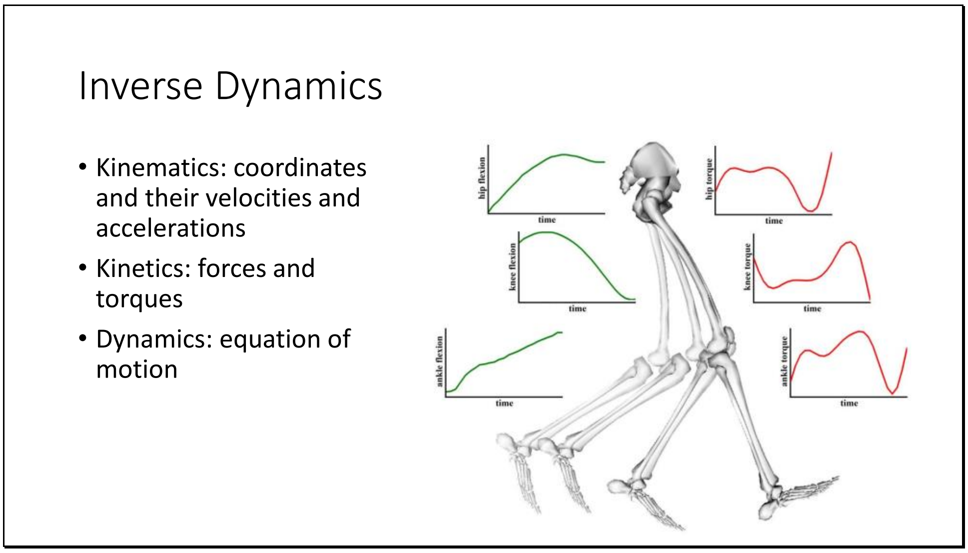
Inverse Kinematics Settings:

- Weight "motion" segment markers (from a triad placed on the thigh segment, for example) more heavily than anatomical markers affixed to landmarks like the greater trochanter and the acromion, which can be helpful for scaling, but are influenced by muscle and other soft tissue movements during motion.
- Relative marker weightings are more important than their absolute values. Therefore, a weighting of 10 vs. 1 is 10 times more important, whereas 20 vs. 10 is only twice as important. Markers are not necessarily tracked better because they both have higher weightings.

Evaluating your Results:

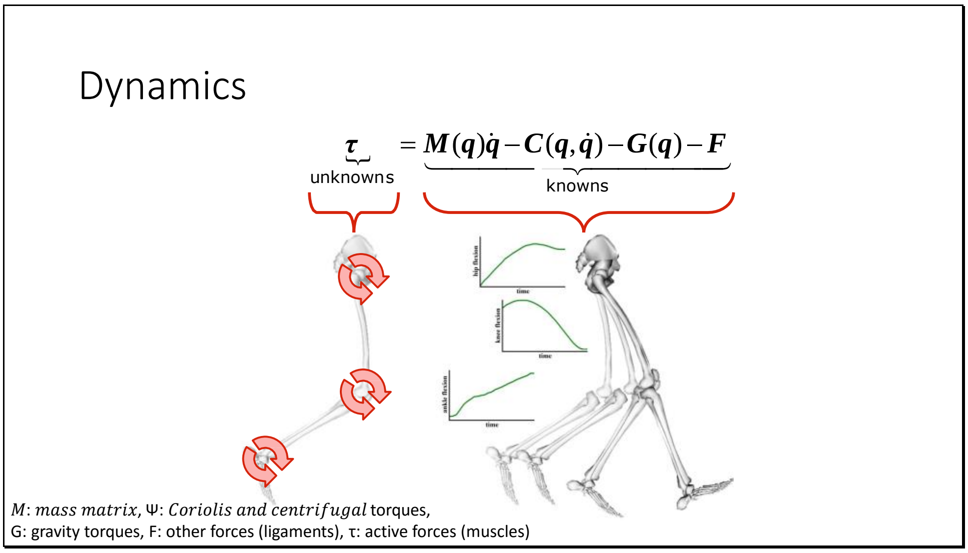
- Total RMS and maximum marker errors are reported in the "Messages" window. Use these values to guide changes in weightings or, if necessary, to redo marker placement and possibly scaling. Maximum marker error should generally be less than 2-4 cm, and RMS under 2 cm is achievable. These guidelines will vary depending on the nature of the model and the motion being examined.
- If using coordinates from a motion capture system, make sure that the joint/coordinate definitions match—otherwise, you may cause more harm than good.
- Compare your results to similar data reported in the literature. Your results from an unimpaired average adult should generally be within one standard deviation.
- If you are unsatisfied with the results, recheck the results of scaling.

Slide 20



Dynamics is the study of motion *and* the forces and moments that produce that motion. The Inverse Dynamics (ID) Tool determines the generalized forces (e.g., net forces and torques) that cause a particular motion, and its results can be used to infer how muscles are actuated to generate that motion. To determine these internal forces and moments, the equations of motion for the system are solved with external forces (e.g., ground reaction forces) and accelerations given (estimated by differentiating angles and positions twice). The equations of motion are automatically formulated using the kinematic description and mass properties of a musculoskeletal model in Simbody™.

Slide 21



The motion of the model is completely defined by the generalized positions, velocities, and accelerations. Consequently, all of the terms on the right-hand side of the equations of motion are known. The remaining term on the left-hand side of the equations of motion is unknown. The inverse dynamics tool uses the known motion of the model to solve the equations of motion for the unknown generalized forces.

Best Practices and Troubleshooting

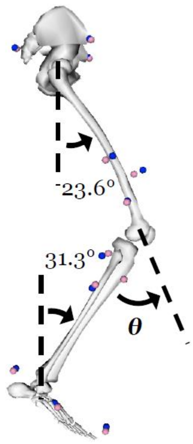
- Filter your raw coordinate data, since noise is amplified by differentiation. Without filtering, the calculated forces and torques will be very noisy.
- Compare your results to data reported in the literature. Your results should be within one standard deviation of reported values.
- Inspect results from Inverse Dynamics to check whether ground reaction forces were applied correctly or not. Are there large and unexpected forces at the pelvis? For gait, applying ground reaction forces should help reduce the forces computed by Inverse Dynamics at the pelvis.

Slide 22

Inverse Dynamics Exercise

- 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°



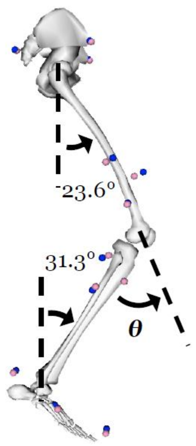
B

Slide 23

Inverse Dynamics Exercise

- 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

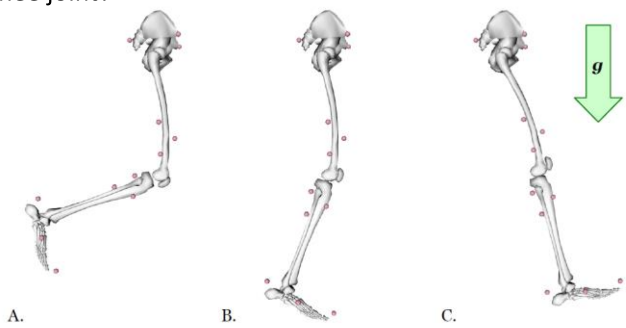


D.

Slide 24

Inverse Dynamics Exercise

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



A.

Slide 25

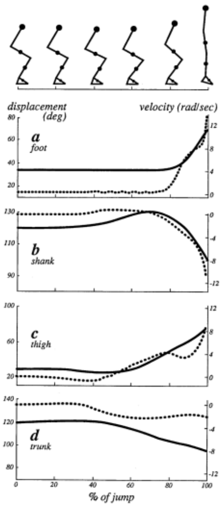
A Possible Inverse Dynamics Question

- What are the sagittal plane moments about the ankle, knee, and hip during a maximum height jump?

Slide 26

Inverse Dynamics Input

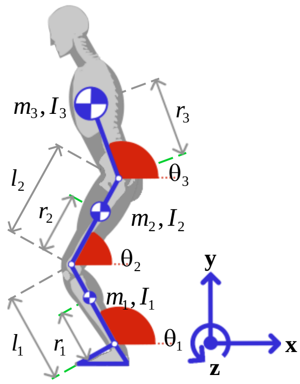
- Joint angles
- Angular velocities
- Angular accelerations
- Ground reaction forces



Slide 27

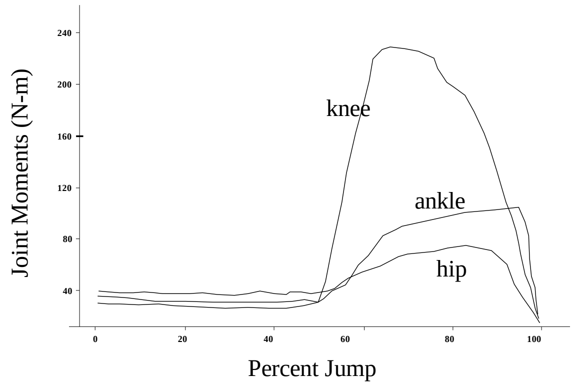
Multibody Dynamics

- 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



Slide 28

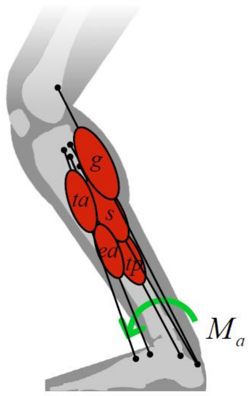
Net Joint Moment



Slide 29

Static Optimization

- Kinematics
  - coordinates and their velocities and accelerations
- Kinetics
  - muscle forces
- Muscle physiology
  - activation-contraction dynamics and force-length-velocity relations
- Dynamics
  - equations of motion
- Musculoskeletal geometry
  - muscle moment arm
- Optimization
  - the “distribution” problem

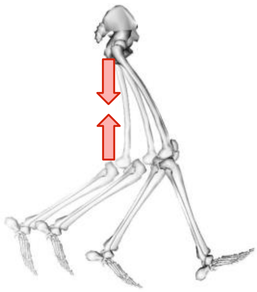


Static optimization is an extension to inverse dynamics that further resolves the net joint moments into individual muscle forces at each instant in time. The muscle forces are resolved by minimizing the sum of squared (or other power) muscle activations.

Slide 30

Kinetics: Muscle Forces

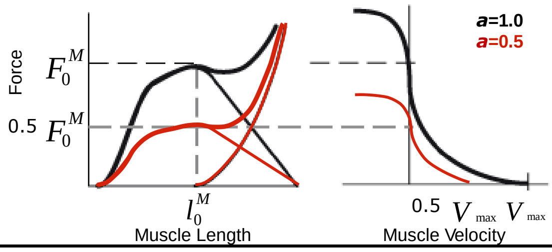
- Kinetics
  - Muscle forces cause the model to accelerate
- Muscle force
  - Applied between origin and insertion points

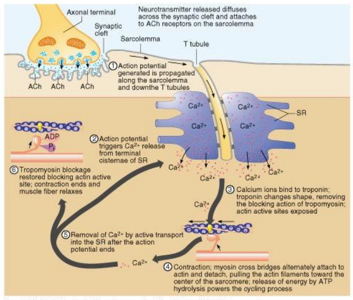


Slide 31

Muscle Physiology

- Muscle activation-contraction
  - Biochemical reaction that causes a muscle's fibers to contract which produces force
- Muscle force-length-velocity
  - Force production diminishes for short, long, and fast fibers



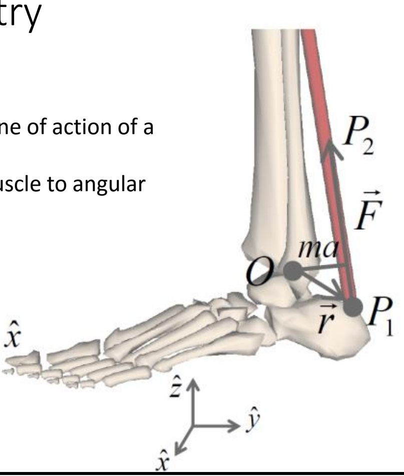


Slide 32

Musculoskeletal Geometry

- Muscle moment arm
  - The perpendicular distance from the line of action of a muscle to the joint center of rotation
  - Transformation from linear force of muscle to angular moment about a joint center

$$ma_x = \frac{\vec{r} \times \vec{F}}{|\vec{F}|} \bullet \hat{x}$$



Slide 33

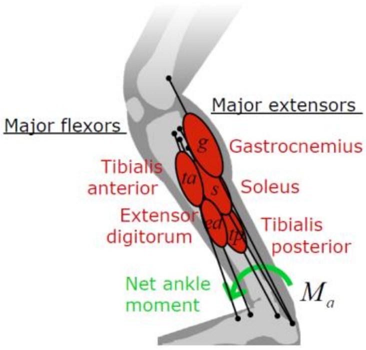
Tips and Tricks

- Filter your raw coordinate data
- Check residuals for RRA and to make sure GRFs were applied correctly
- Compare to previous literature data (if available)

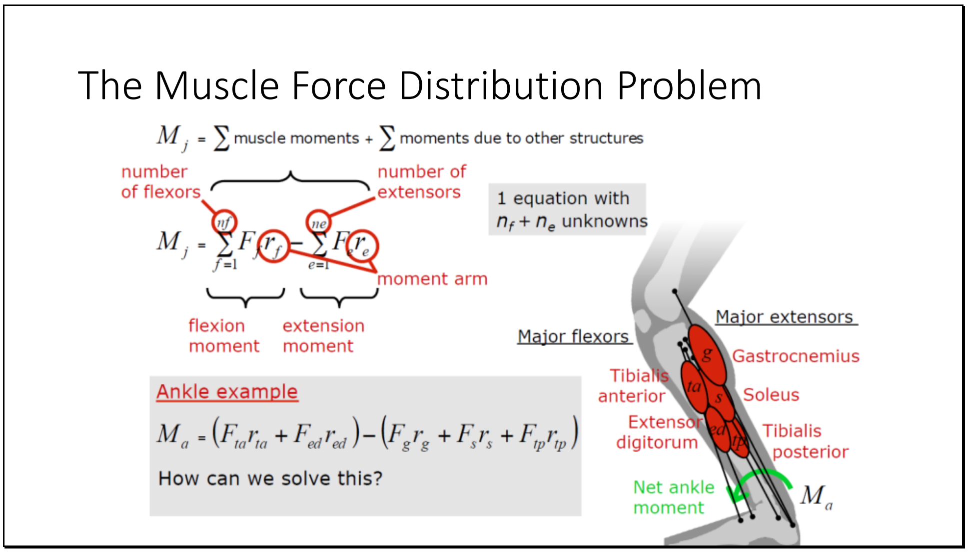
Slide 34

Static Optimization

- Determines the "best" set of muscle forces that
  - Produce net joint moments at a discrete time
  - Do not violate muscle force limits
  - Optimize a performance criterion
- Performance criterion attempts to capture the goal of the neural control system
  - Minimize muscle force?
  - Minimize muscle stress?

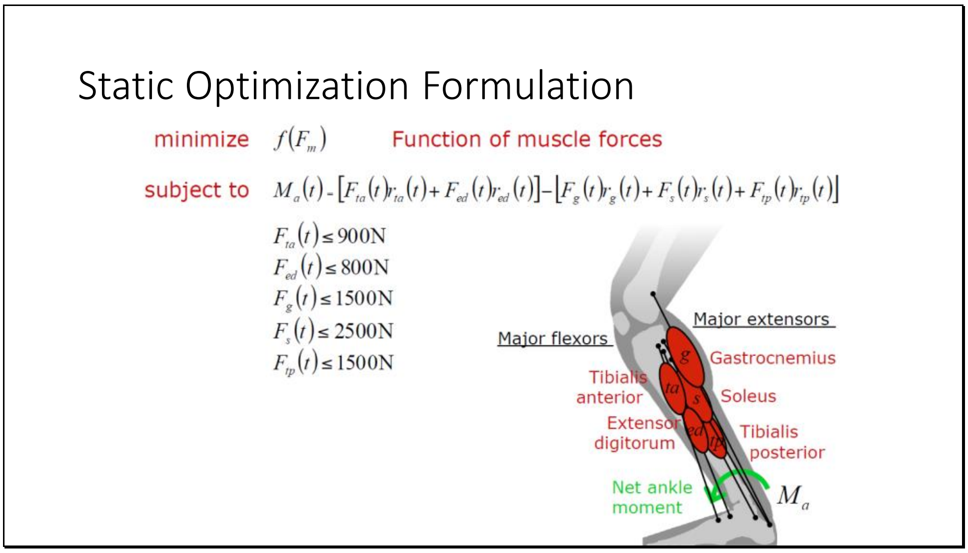


Slide 35



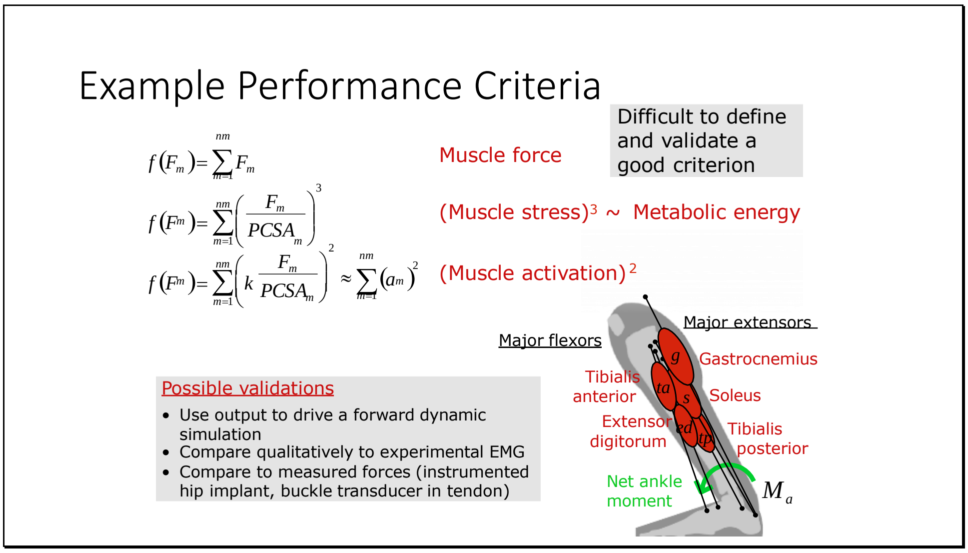
Many unknowns in one equation. Solution to this unconstraint problem can be achieved, by introducing additional equations, constraints and an objective function that we want to minimize.

Slide 36



Minimize some objective function, so that the muscle forces will produce the desired torque around a joint, while accounting for the force capability of each muscle.

Slide 37



The choice of the objective function depends on the task under investigation. The most common objective function is the minimum effort of muscle activation, which gives good correlation for tasks such as walking, running.

Slide 38

Formulation of the Optimization Problem

- Ideal force generators
- Constrained by force-length-velocity properties
- Minimizing the objective function

$$\sum_{m=1}^{n}(a_m F_m^0) r_{m,j} = \tau_j$$

$$\sum_{m=1}^{n}[a_m f(F_m^0, l_m, v_m)] r_{m,j} = \tau_j$$

$$J = \sum_{m=1}^{n}(a_m)^p$$

As described in ID, the motion of the model is completely defined by the generalized positions, velocities, and accelerations. The Static Optimization Tool uses the known motion of the model to solve the equations of motion for the unknown generalized forces (e.g., joint torques).

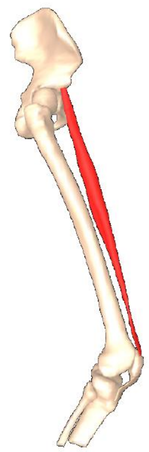
n is the number of muscles in the model; a_m is the activation level of muscle m at a discrete time step; is its maximum isometric force; l_m is its length; v_m is its shortening velocity; is its force-length-velocity surface; $r_{m,j}$ is its moment arm about the j^{th}joint axis; is the generalized force acting about the j^{th} joint axis; and p is a user defined constant. Note that for static optimization computes the active fiber force along the tendon assuming an inextensible tendon and does not include contribution from muscles' parallel elastic element.

Slide 39

Static Optimization Exercise

- Given that the rectus femoris muscle has a peak isometric force of 1169 N and it is at its optimal fiber length and zero velocity, what is the force generated for an activation of 0.86?

A. 164 N
B. 952 N
C. 1005 N
D. 1058 N



C.

Slide 40

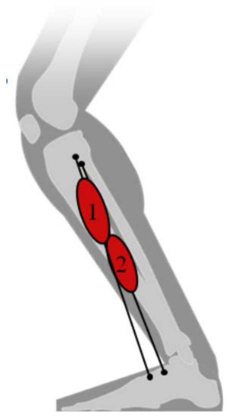
Static Optimization Exercise

- For the model shown on the right, which muscle has the largest moment arm about the **ankle** joint?

A. 1

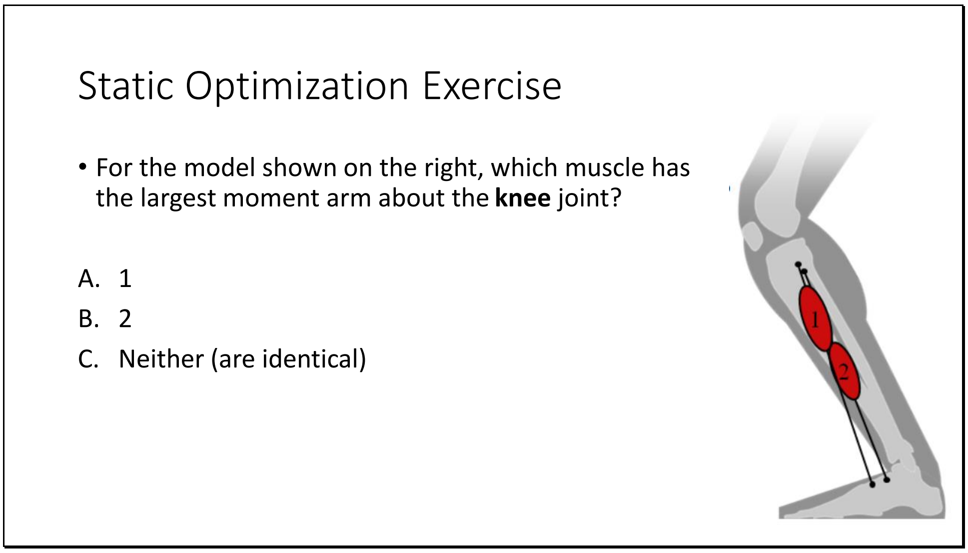
B. 2

C. Neither (are identical)



A.

Slide 41



Static Optimization Exercise
• For the model shown on the right, which muscle has the largest moment arm about the **knee** joint?
A. 1
B. 2
C. Neither (are identical)
1
2

C.

Slide 42

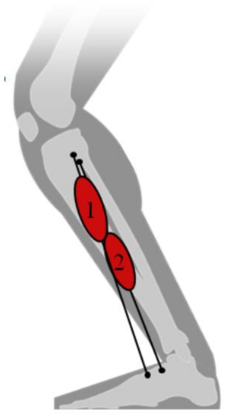
Static Optimization Exercise

For the model shown on the right, muscle 1 and 2 have the following properties

Muscle	Peak Isometric Force (N)	Moment Arm (cm)
1	905	3.6
2	512	3.0

To solve the “distribution” problem minimizing the sum of squared activations, which muscle would be activated more for a given dorsiflexion moment?

A. 1
B. 2
C. Neither (are identical)



A.

Slide 43

Tips and Tricks

- **Inputs:** Can use kinematics from IK or RRA. If using IK, need to filter kinematics
- **Residuals:** Add residual actuators to pelvis
- **Reserves:** Add reserve torque actuators to trouble-shoot a weak model
- **Minimizing residuals & reserves:** Increase maximum control value (default = 1) and lower the maximum force -> penalizes activity
- **Command Line:** analyze –S setup_!le.xml

Best Practices and Troubleshooting
Static Optimization Settings:

- You can use IK or RRA results as input kinematics. If using IK results, you usually need to filter them, either externally or using the OpenSim analyze/static optimization field; if using RRA results, you usually do not have to filter.
- For gait and many other motions, you need to add (append) residual actuators to the first free joint in the model (typically the ground-pelvis joint).

    - There should be one actuator for each degree-of-freedom (i.e., F_X, F_Y, F_Z, M_X, M_Y, M_Z).
    - These residual actuators are required because there is dynamic inconsistency between the estimated model accelerations and the measured ground reaction forces. This inconsistency can result from marker measurement error, differences between the geometry of the model and the subject, and inertial parameters.
    - Running RRA will reduce—but not eliminate—these residuals. Thus, appending actuators is still necessary.

Troubleshooting:

- If the residual actuators or the model's muscles are weak, the optimization will take a long time to converge or will never converge at all.

    - If the residual actuators are weak, increase the maximum control value of a residual, while lowering its maximum force. This allows the optimizer to generate a large force (if necessary) to match accelerations, but large control values are penalized more heavily. In static optimization, ideal actuator excitations are treated as activations in the cost function.

    - If the muscles are weak, append Coordinate Actuators to the model at the joints in the model. This will allow you to see how much "reserve" actuation is required at a given joint and then strengthen the muscles in your model accordingly.
    - If troubleshooting a weak model and optimization is slow each time, try reducing the parameter that defines the maximum number of iterations.
- Static optimization works internally by solving the inverse dynamics problem, then trying to solve the redundancy problem for actuators/muscles using the accelerations from the inverse dynamics solution as a constraint. If a constraint violation is reported, this could be a sign that the optimizer couldn't solve for muscle forces while enforcing the inverse dynamics solution.

    - This likely means that there is noise in the data or there is a sudden jump in accelerations in one frame.
    - In this case, you should examine the inverse dynamics solution to determine the problematic frame, and fix/interpolate the data during this portion of the motion.
- If your model has passive elements (e.g., ligaments or springs), you should use OpenSim version 3.3 or later. Prior to version 3.3, the forces generated by passive elements were not properly accounted for.

Evaluating your Results:

- Are there any large or unexpected residual actuator forces?
- Find EMG or muscle activation data for comparison with your simulated activations. Does the timing of muscle activation/deactivation match? Are the magnitudes and patterns in good agreement?