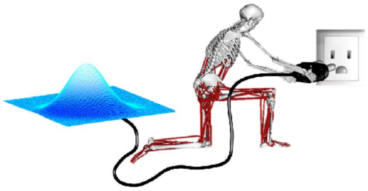
The OpenSim Probabilistic Plugin

An introductory guide to assess uncertainty in musculoskeletal modeling

Casey A. Myers
Kevin B. Shelburne
Peter J. Laz
Bradley S. Davidson



If you perform musculoskeletal simulations using OpenSim at any level, this Probabilistic Plugin is for you. The purpose of the Probabilistic Plugin is to enable OpenSim users to quantitatively assess confidence in outputs from your musculoskeletal simulations. This probabilistic approach provides a systematic framework to quantify uncertainty and report this information. The Probabilistic Plugin is open source, and should be adapted as needed to your specific project.

Where to Start

[Coming Soon] If you are new to probabilistic analyses, visit YouTube to view a presentation on common probabilistic methods in musculoskeletal simulation.

If you have already configured the Matlab Scripting Environment in OpenSim, you are ready to work through the tutorials in order.

(Last mod: 2014-7-8)

Contents

Initialize and Test Interface between OpenSim and Probabilistic Plugin

Set up the Matlab Scripting Environment in OpenSim

To connect Matlab and OpenSim API, follow the instructions on Scripting with Matlab within the OpenSim Confluence documentation.

Download the Probabilistic Plugin

Download zipfile that contains the Plugin distribution and files from https://simtk.org/home/prob_tool

Unzip the file and store folder on your computer.

Save the folder and add this folder name to the Matlab search path.

Test that the interface is correctly

To test that the interface is working correctly, type the following into the Matlab Command Window:

```
    Model('YourFilePath/ProbModel_gait2392.osim')
```

Note: The ProbModel_gait2392.osim is a version of the gait2392 model that has been appropriately scaled for this data set.

Proceed to Tutorial 1 if 1) No errors occur and 2) A model object appears in the Matlab Workspace, proceed to Tutorial 1.

Tutorial 1: Inverse Dynamics and Uncertainty in Body Segment Parameters (Monte Carlo Simulation)

This self-guided tutorial will walk you through a simple analysis performed of the Probabilistic Plugin for OpenSim. A case study is presented that relies on Monte Carlo simulation as applied to lower extremity inverse dynamics in the presence of uncertainties in inertial properties.

Upon completing this tutorial, you will be able to:
- Create valid input distributions for body segment parameters
- Create and interpret outputs of probabilistic analyses: confidence bounds and sensitivity factors
- Develop intuition on convergence of Monte Carlo simulation
- Generate a set up file for future probabilistic analyses

How to consider the effects of uncertainty in inverse dynamics

Inverse dynamics is a fundamental metric in biomechanics

Modeling of inverse dynamics (net moment at a joint) during human movement is a foundational concept in biomechanics. Analyses of joint moments are:
- Taught in every course that covers human movement.
- Frequently applied to assess clinical outcomes.
- A foundational step toward estimating muscle forces (see Tutorial 2).

Where does uncertainty arise in inverse dynamics?

The inverse dynamics solution is mathematically straightforward and depends on three input variables (external reaction forces, segment kinematics, inertial parameters). Each of these inputs is prone to error in the measurement or estimation and is carried through the calculations to the output joint moments.

Effects of input uncertainty

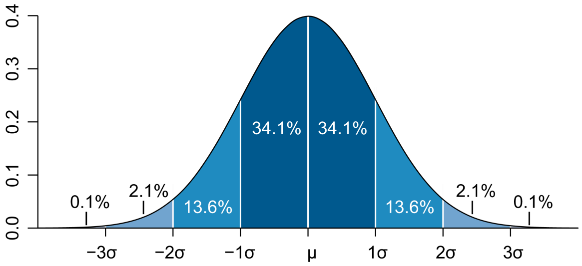
Two important effects of input uncertainty that we should consider when developing a musculoskeletal model:
- The “correct output” at any given time point lies within a range of possible values that are linked to uncertainty in the input.
- The contributions of uncertainty in each input to the model outputs are not equal.

To quantify these effects, we will generate and interpret Confidence Bounds and Sensitivity Factors.

Preparation for Probabilistic Simulation

Create input distributions for body segment parameters

A challenging part of running a probabilistic analysis is correctly modeling the input distribution. The OpenSim Probabilistic Plugin currently accepts the mean and standard deviation to create the Gaussian distribution needed for sampling.



(image taken from Wikimedia commons)

$$f(x,\mu,\sigma) = \frac{1}{\mu\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean value of the parameter and σ is the standard deviation of the parameter.

For your input distributions, we will take each value of μ from the starting model parameters, and define the input σ from previously reported literature.

Coefficient of variation for quantifying the distribution

To obtain a more generalized formulation applicable to all models, we can assume a constant coefficient of variation,

$$CV = \frac{\sigma}{|\mu|}$$

which assumes that the standard deviation is proportional to the magnitude of the mean.

For example, the means and standard deviations reported for the foot segment mass, tibia segment mass, and femur segment mass in Rao et al. (2006) were 0.85(0.11) kg, 2.89(0.19) kg, and 7.59(1.30) kg, respectively.

Therefore, the corresponding coefficients of variation are:

$$CV_{\text{femur}} = 0.171$$
$$CV_{\text{tibia}} = 0.066$$
$$CV_{\text{foot}} = 0.129$$

Make note of these for use when running the probabilistic simulation.

Appendix A. lists papers we have found helpful to quantify distributions for a variety of parameters.

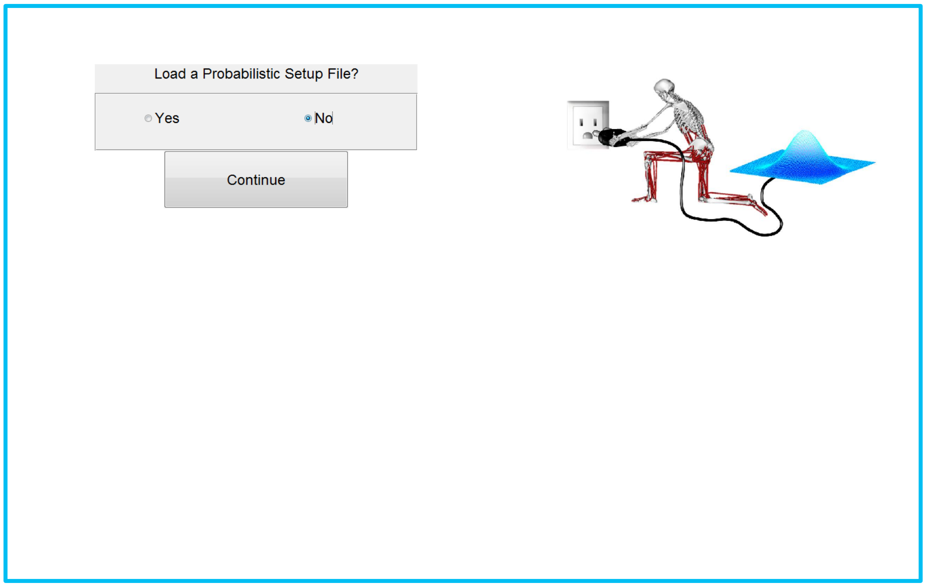
Perform a Monte Carlo Simulation with the OpenSim Probabilistic Plugin

A Monte Carlo simulation is the most familiar probabilistic method. Monte Carlo is a class of data sampling techniques in which the simulation is run for multiple iterations. Each time, the input values are randomly selected from predetermined probability density functions associated with each parameter. The outputs of interest are random and distributed along their own probability density functions.

Run the baseline simulation

The Baseline Simulation is the initial deterministic simulation needed before the probabilistic methods can be performed. In this tutorial, the baseline parameters will be used as the mean values when defining the input distributions.

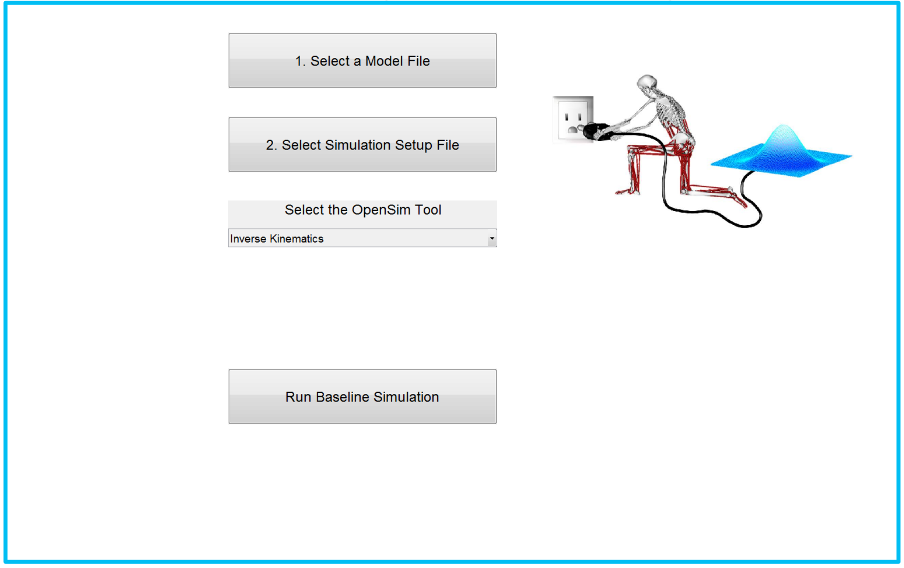
Type `ProbGUI_v8.m` in the Matlab Command Window
This launches the Probabilistic Plugin and you will see the following window.



Load a Probabilistic Setup File?

○ Yes ● No

Select the "No" radio button, then "Continue"
The Probabilistic Setup File is a .xml file that allows the user to bypass the GUI setup. A modifiable setup file will be generated at the end this tutorial, and can be used for future simulations using the Probabilistic Plugin.



Select the OpenSim Tool
Inverse Kinematics

Click "1. Select a Model File"
Select ProbModel_gait2392.osim, which was included in the folder.
This file is the gait2392.osim model that has been appropriately scaled for use with the experimental data. The Probabilistic Plugin will generate a copy of this file and make changes to the copied file. If you restart the plugin, select the original model file.

Click "2. Select Simulation Setup File" and
Select the "OpenSimInverseDynamics_setup.xml" file, which was included in the folder.

!! Important !!
Before proceeding to the next step, open the simulation setup file and the external ground reaction force setup file and ensure that the file paths in these setup files are completely defined.

Select "Inverse Dynamics" from the OpenSim Tool dropdown menu.

Click "Run Baseline Simulation"

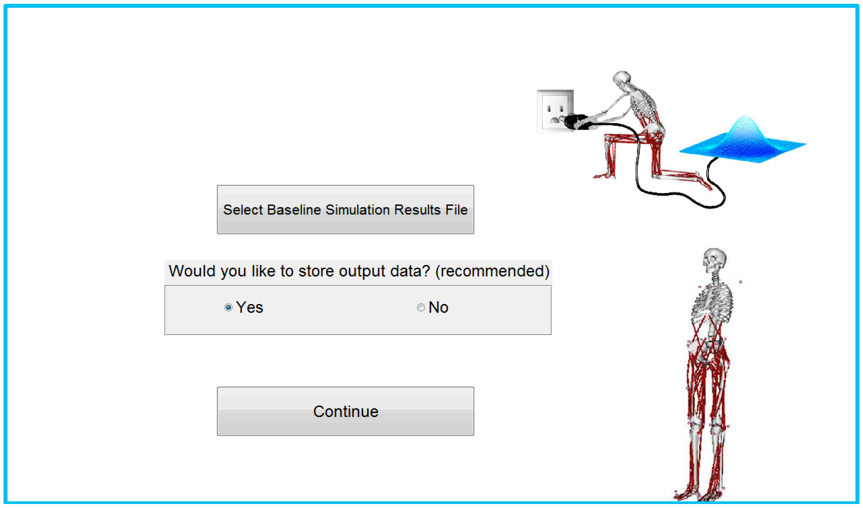
Check that "inverse_dynamics.sto" was written in the "Results" folder located in the current Matlab directory.

If you do not see "inverse_dynamics.sto" with a time stamp equivalent to running the simulation, **examine out.txt for errors that occurred during the baseline simulation.**

Out.txt is written at the conclusion of the baseline simulation and is located in the current Matlab folder.

The most common errors are related to improper path to locate the files needed for the Inverse Dynamics simulation. To correct this, ensure that all paths in the .xml setup files are correctly entered.

Close the Probabilistic Plugin and launch again after correcting the error.



Would you like to store output data? (recommended)

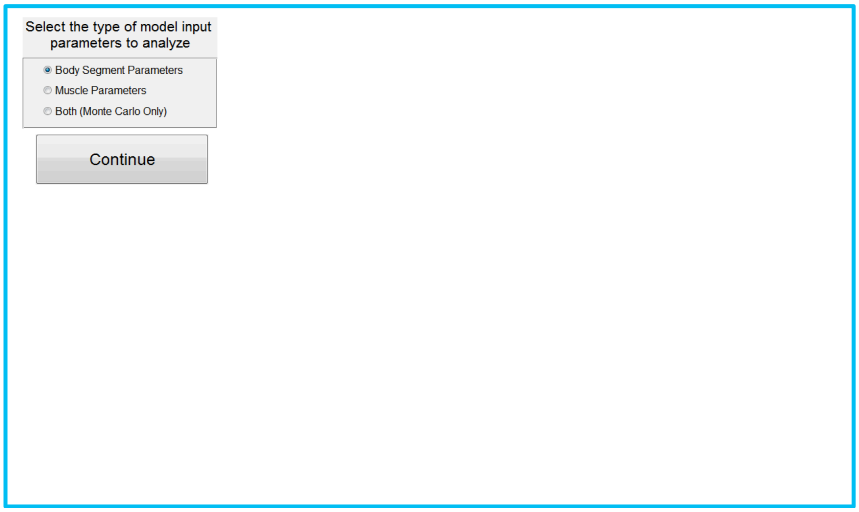
● Yes ○ No

Click "Select Baseline Simulation Results File"
Select the "inverse_dyanmics.sto" file located in the Results folder.

Select the "Yes" radio button located under "Would you like to store the output data from each Monte Carlo simulation?"
The results from each iteration will be stored in the Results folder. The default is "Yes" to ensure future analysis.

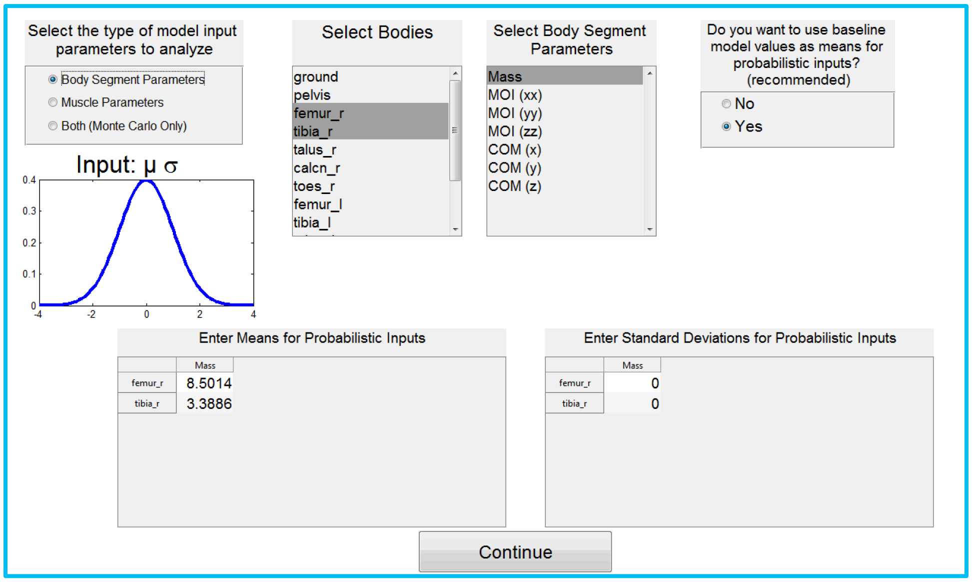
Enter the input distributions

Define the parameters that will be perturbed and define the quantitative distributions.


Select the type of model input parameters to analyze

● Body Segment Parameters
○ Muscle Parameters
○ Both (Monte Carlo Only)

Select “Body Segment Parameters” radio button and click “Continue”


Select the type of model input parameters to analyze

● Body Segment Parameters
○ Muscle Parameters
○ Both (Monte Carlo Only)

Do you want to use baseline model values as means for probabilistic inputs? (recommended)

○ No
● Yes

Enter Means for Probabilistic Inputs

	Mass
femur_r	8.5014
tibia_r	3.3886

Enter Standard Deviations for Probabilistic Inputs

	Mass
femur_r	0
tibia_r	0

Select the “femur_r” and “tibia_r” segments in the list of bodies available to perform analyses.

note: To select multiple items in the list hold the Ctrl key.

Select “Mass” as the parameter to perturb on each segment.

note: Although Mass is already highlighted, you must click on it to avoid an error.

Select “Yes” radio button to indicate use the baseline model values.

Click “Continue”

Because we chose to use the segment parameters from the baseline model as the mean value for each distribution, the means table will be populated. If you chose "No", the means must be manually input into the table.

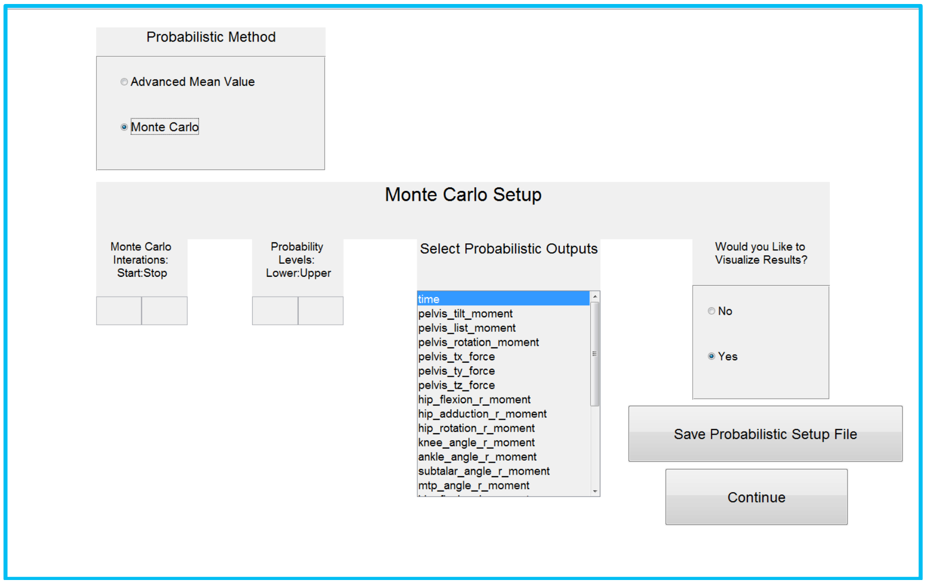
Calculate the standard deviations using the coefficients of variation defined in the earlier section and enter standard deviations in the GUI.

$$\sigma_{\text{femur}} = CV_{\text{femur}} \times |\mu_{\text{femur}}| = 0.171 \times 8.5014 = 1.454$$
$$\sigma_{\text{tibia}} = CV_{\text{tibia}} \times |\mu_{femur}| = 0.066 \times 3.3886 = 0.223$$

Click "Continue".

Initialize the Monte Carlo Simulation

After the distributions are defined, the probabilistic simulation must be initialized to perform the probabilistic analysis.



Select "Monte Carlo" radio button under Probabilistic Method.
Click "Continue".

Enter 1 and 30 as the Monte Carlo iterations Start:Stop
This will run the Monte Carlo simulation 30 times.

Enter 5 and 95 as the lower and upper Probability Levels.
This specifies the program to create lower and upper limits of a 90% confidence bound (between the 5th and 95th percentiles of the distribution).

Select "hip_flexion_r_moment", "knee_angle_r_moment", and "ankle_angle_r_moment" as the Probabilistic Outputs.
This list is constructed from the possible outputs located in your Results File.

Select "Yes" under "Would you like to visualize the results?".

Click "Save Probabilistic Setup File".

Name the file "Tutorial1_MonteCarlo30_Setup" and save
This selection will generate an .xml file that can be loaded in place of the Probabilistic Plugin GUI.

Click "Continue"
The Monte Carlo Simulation will run and produce output information in the Matlab Command Window.

On a PC with 16.0 GB of RAM and a 3.60 GHz processor, 30 iterations in the Monte Carlo Simulation will take approximately 60 seconds.

Visualization from the Monte Carlo Simulation

After the simulation has completed, several plots will be displayed that include interpretable results and information about the simulation.

Confidence Bounds

Confidence bounds represent the range in which the output of the simulation can lie. In this tutorial, we chose a two-sided confidence bound with limits at 5th and 95th percentile of the output distribution.

> "There is a 90% probability that that true result of this simulation lies between the lower and upper confidence bounds."

Currently, standards do not exist on selection of confidence bound sizes.

Confidence Bounds versus a Confidence Interval

Confidence Bounds approximate the value of a model output and is calculated from repeated numerical simulations whereas a Confidence Interval approximates the mean of an entire population mean based on a sample data set that includes multiple participants (Curran-Everett, 2009). The

However, when the output distribution of your probabilistic simulation is Gaussian, the two-sided confidence bounds can be interpreted in a similar manner a confidence interval. For example, when the output distribution is

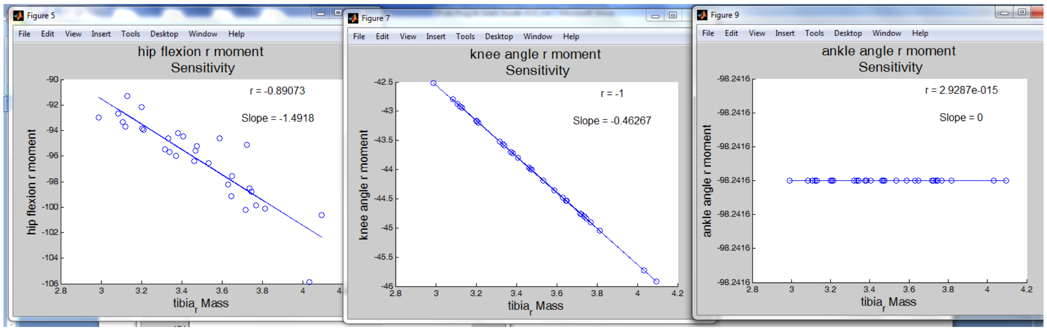
Gaussian you can test if the outputs from two different models, given the same input data, are different by stating the null hypothesis (h_0) and alternative hypothesis (h_1) as

h_0: Model A Output = Model B Output.
h_1: Model A Output $\neq$ Model B Output

If the acceptable Type I Error is limited to 5%, then we reject h_0 when the two-sided 95% confidence bounds (2.5th percentile and 97.5th percentile) from each Monte Carlo Simulation do not overlap.

Interpret sensitivity factors

A Sensitivity Factor is generated for every combination of input varied and the output of interest. The value of the sensitivity factor is quantified by Pearson Product-Moment Correlation between the input parameter and the output.



The value of Sensitivity Factor indicates the degree of sensitivity. For example: weakly sensitive (*r*=0.2-0.4), moderately sensitive (*r*=0.4-0.6), and highly sensitive (*r*=0.6-1.0).

We recommend categorizing the degree of sensitivity on Sensitivity Factors that are statistically different from zero (when the 95% confidence interval of the correlation coefficient does not contain zero).

In addition, the slope of the regression provides information about how the average change in the input will affect the output. Note that this interpretation assumes a linear relationship between the input and output.

Output Distribution

A plot is generated that shows the histogram of each output in the simulation and the normal probability plot. This information can be used to examine the qualitative features of your distribution.

If you intend to calculate a confidence interval (see panel above), the normal probability plot will help you decide if the data already satisfy the Gaussian criterion. If not, the value and histogram will assist deciding on an appropriate transform.

Use the Probabilistic Setup File to generate results with different parameter distributions

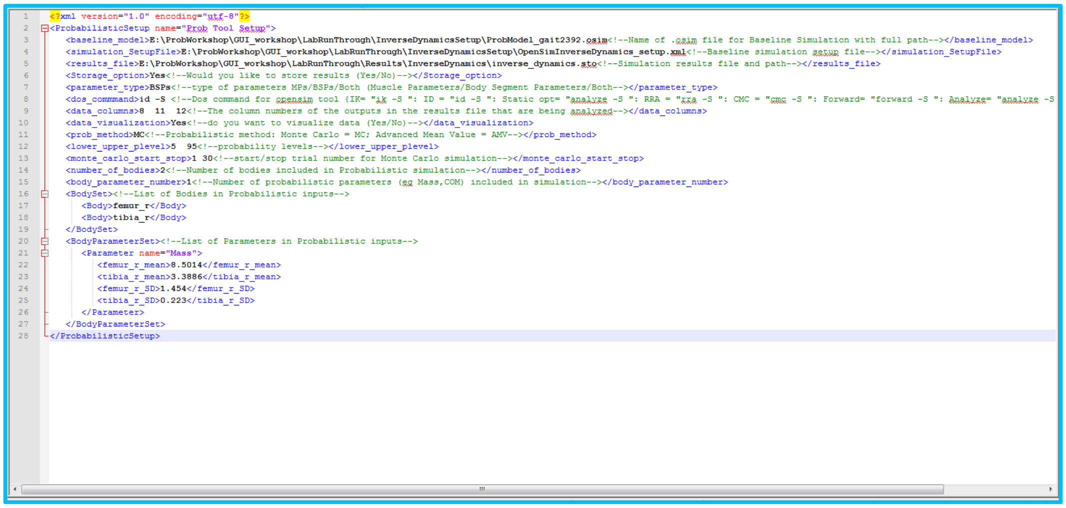
After completing the first simulation, the Plugin generated a new XML file that allows running the same or modified version of the probabilistic simulation without navigating the Plugin GUI each time.

Modify the probabilistic setup file

Navigate to the file named “filename.xml” which is located in the local directory with the Plugin files.

Open the file in an XML viewer of your choice

Explore the set up file created.
You will recognize many of the decisions you made when using the PlugIn GUI



```
<?xml version="1.0" encoding="utf-8"?>
<ProbabilisticSetup name="Prob Tool Setup">
    <baseline_model>E:\ProbWorkshop\GUI_workshop\LabRunThrough\InverseDynamicsSetup\ProbModel_gait2392.osim<!--Name of .osim file for Baseline Simulation with full path--></baseline_model>
    <simulation_SetupFile>E:\ProbWorkshop\GUI_workshop\LabRunThrough\InverseDynamicsSetup\OpenSimInverseDynamics_setup.xml<!--Baseline simulation setup file--></simulation_SetupFile>
    <results_file>E:\ProbWorkshop\GUI_workshop\LabRunThrough\Results\InverseDynamics\inverse_dynamics.sto<!--Simulation results file and path--></results_file>
    <Storage_option>Yes<!--Would you like to store results (Yes/No)--></Storage_option>
    <parameter_type>BSPs<!--type of parameters MPs/BSPs/Both (Muscle Parameters/Body Segment Parameters/Both--></parameter_type>
    <dos_commmand>id -S <!--Dos command for opensim tool {IK= "ik -S ": ID = "id -S ": Static opt= "analyze -S ": RRA = "rra -S ": CMC = "cmc -S ": Forward= "forward -S ": Analyze= "analyze -S
    <data_columns>8  11  12<!--The column numbers of the outputs in the results file that are being analyzed--></data_columns>
    <data_visualization>Yes<!--do you want to visualize data (Yes/No)--></data_visualization>
    <prob_method>MC<!--Probabilistic method: Monte Carlo = MC; Advanced Mean Value = AMV--></prob_method>
    <lower_upper_plevel>5  95<!--probability levels--></lower_upper_plevel>
    <monte_carlo_start_stop>1 30<!--start/stop trial number for Monte Carlo simulation--></monte_carlo_start_stop>
    <number_of_bodies>2<!--Number of bodies included in Probabilistic simulation--></number_of_bodies>
    <body_parameter_number>1<!--Number of probabilistic parameters (eg Mass,COM) included in simulation--></body_parameter_number>
    <BodySet><!--List of Bodies in Probabilistic inputs-->
        <Body>femur_r</Body>
        <Body>tibia_r</Body>
    </BodySet>
    <BodyParameterSet><!--List of Parameters in Probabilistic inputs-->
        <Parameter name="Mass">
            <femur_r_mean>8.5014</femur_r_mean>
            <tibia_r_mean>3.3886</tibia_r_mean>
            <femur_r_SD>1.454</femur_r_SD>
            <tibia_r_SD>0.223</tibia_r_SD>
        </Parameter>
    </BodyParameterSet>
</ProbabilisticSetup>
```

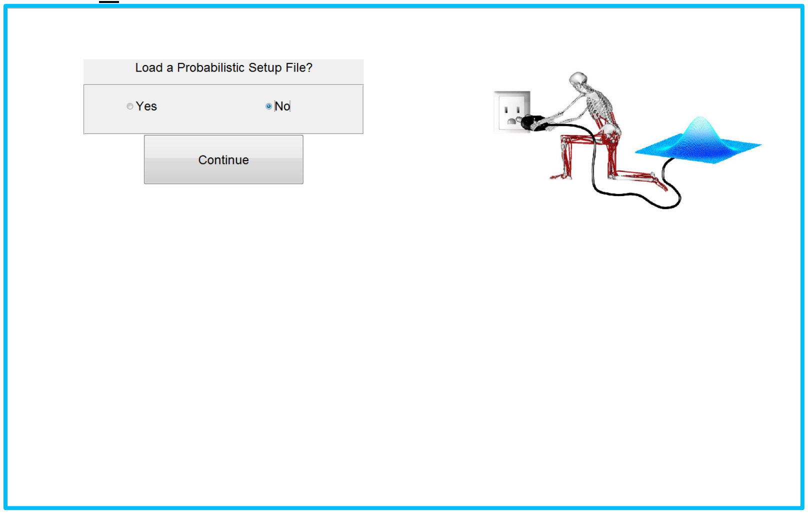
Change the standard deviations for the mass of the femur and tibia to 2x the original value.

```
<femur_r_SD>2.908<femur_r_SD>
<tibia_r_SD>0.466<tibia_r_SD>
```

Leave the number of iterations the same

Run the Monte Carlo Simulation with altered parameters

Type `ProbGUI_v8.m` in the Matlab Command Window

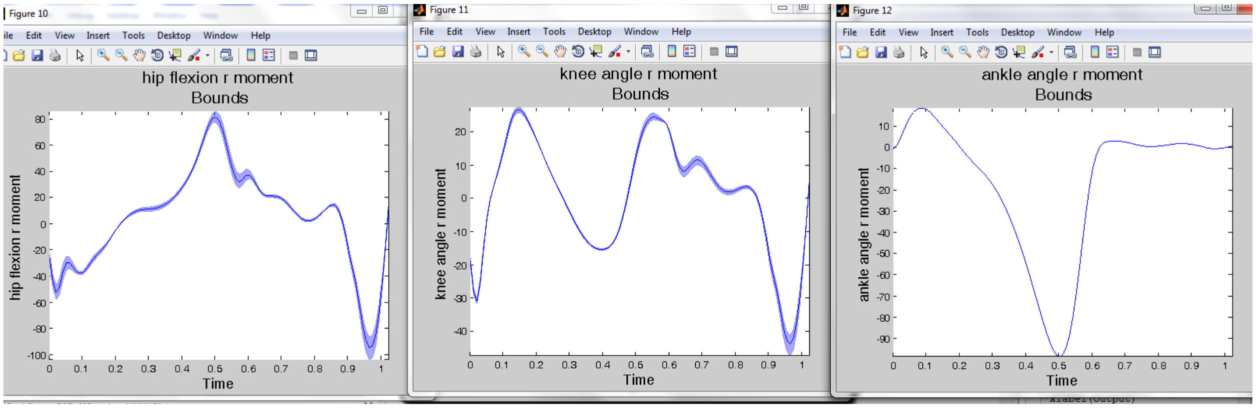


Select the “Yes” radio button, then “Continue”.

Select the Probabilistic Setup file that you saved. The simulation will begin with the baseline simulation and then proceed to the Monte Carlo iterations.

Examine New Results

The updated plots of the 90% confidence bounds are now larger than in the initial simulation for the hip and knee.



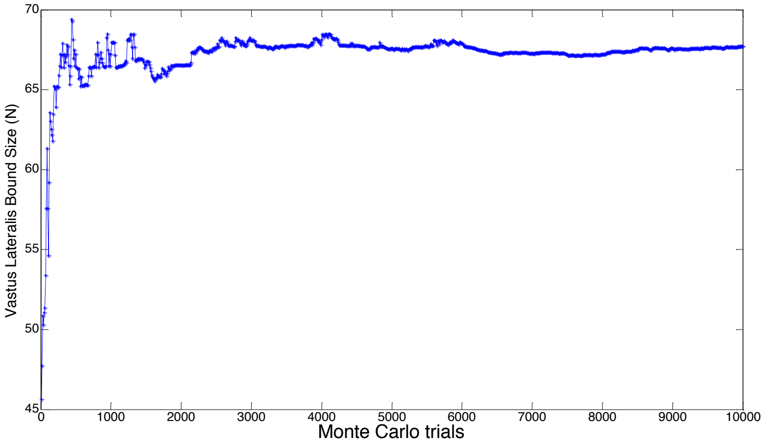
How many iterations are necessary in a Monte Carlo Simulation?

Accuracy of the Monte Carlo simulation improves with the number of iterations

It is important to perform enough iterations in the Monte Carlo simulation to obtain the results for interpretation. The confidence bounds and sensitivity will change with additional iterations.

There are multiple ways to examine convergence. The most common is to set a convergence criterion on the change on confidence bounds between iterations.

The plot below demonstrates how the bound size changed with each successive iteration of a Monte Carlo simulation that used bound size of the Vastus Lateralis muscle force. In the Monte Carlo simulation shown, the results converged around 3000 iterations.



Without prior knowledge of how a system will behave in the Monte Carlos simulation, selecting the convergence criterion may be difficult. As a result, convergence may be assessed after the simulation.

To generate your own convergence for the inverse dynamics example, use the data in the results files created during the Monte Carlo Simulation, which are located in the results folder defined earlier and specified in the Probabilistic Setup File. The output you choose to converge upon must be plotted against the iteration using a custom Matlab script.

Refer to Valente et al. (2013) for an excellent example of reporting convergence of a Monte Carlo simulation.

Monte Carlo Simulation Exercises

Perform the following "homework assignments" to develop better understanding the Monte Carlo Simulation results and the file handling within the Probabilistic Plugin.

Exercise 1: Run full Monte Carlo Simulation

Modify the probabilistic input file to add 500 iterations to the last simulation. Did the 5th and 95th percentiles change compared to the simulation with 30 iteration?

Exercise 2: Create convergence plot

Write a Matlab script to plot the value of the 95th percentile for peak hip extension moment for iterations 1 through 500. Steps:

1) Load the results file for an iteration from the output folder.
2) Find value for peak hip extension moment.
3) Using all previous iterations, calculate the 95th percentile for that iteration
4) Plot results versus each iteration.

Tutorial 2: Muscle Force Prediction and Uncertainty in Muscle Properties (Advanced Mean Value)

This self-guided tutorial will walk you through using the Advanced Mean Value (AMV) method of the OpenSim Probabilistic Plugin.

Upon completing this tutorial, you will be able to:
- Run the AMV method within the Probabilistic Plugin GUI
- Create and interpret outputs of probabilistic analyses: confidence bounds and sensitivity factors
- Characterize the tradeoff of computational efficiency and amount of information available between Monte Carlo and AMV (within the number of most probable points selected)
- Generate and interpret importance factors

Muscle force prediction and uncertainty in muscle parameters

Static optimization and muscle force prediction

Static optimization is currently the most common tool used to resolve the over-determined system of muscles forces within a musculoskeletal model. In OpenSim, the Static Optimization Tool is standard in the GUI.

Uncertainty in muscle parameters

It is important to consider the effects of selecting muscle properties on force prediction processes. Muscles and parameters do not share equal importance in a given simulation. However, it is clear that muscles play an important role in accelerating segments they do not span (Zajac, 1993).

Large number of parameters included in the simulations

In a Hill-Type muscle model, multiple parameters must be quantified for each muscle. These include physiological cross-sectional area (PCSA), pennation angle, maximum velocity, tendon slack length. These values are specific to each muscle, and are quantified for each subject.

Most current lower-extremity models include large numbers of muscles to actuate the system. For example, the gait2392 model we are using for these tutorials includes 92 muscles. If we are required to quantify four parameters per muscle, then 92x4=368 parameters, each with a level of uncertainty.

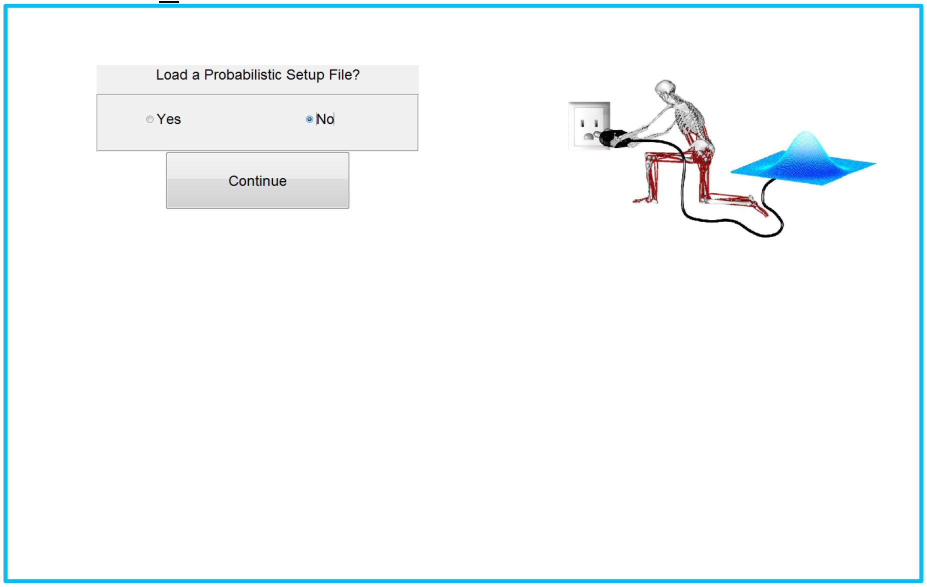
Perform Most Probable Point Analysis (Advanced Mean Value method) on Muscle Forces

When the number of input parameters gets large, the computational expense can drastically increase. When this occurs, we can estimate the reliability metrics through an optimization procedure called the Most Probable Point (Wu et al., 1990). Like the Monte Carlo Simulation, the results provide confidence bounds; however, sensitivity factors are not possible because the entire input probability density function is not considered. A metric of sensitivity called an importance factor is available in the MPP methods.

Run the baseline simulation

The Baseline Simulation is the initial deterministic simulation needed before the probabilistic methods can be performed. In this tutorial, the baseline parameters will be used as the mean values when defining the input distributions.

Type `ProbGUI_v8.m` in the Matlab Command Window.



Select the "No" radio button, then "Continue".

Click "1. Select a Model File" and
Select ProbModel_gait2392.osim.

Click "2. Select Simulation Setup File" and
Select the "ProbGait_StaticOp_Setup.xml" file.

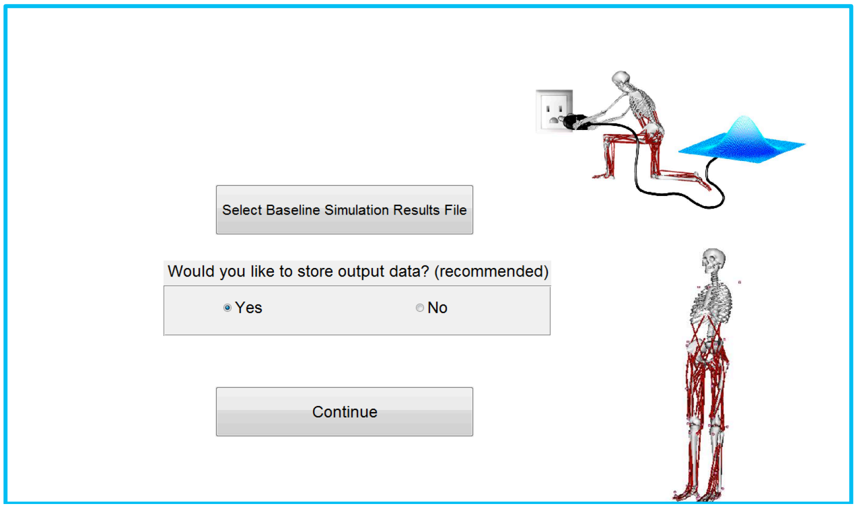
!! Important !!

Before proceeding to the next step, open the simulation setup file and the external ground reaction force setup file and ensure that the file paths in these setup files are completely defined.

Select "Static Optimization" from the OpenSim Tool dropdown menu.

Click "Run Baseline Simulation".

Check that "_force.sto" was written in the "Results" folder located in the current Matlab directory.

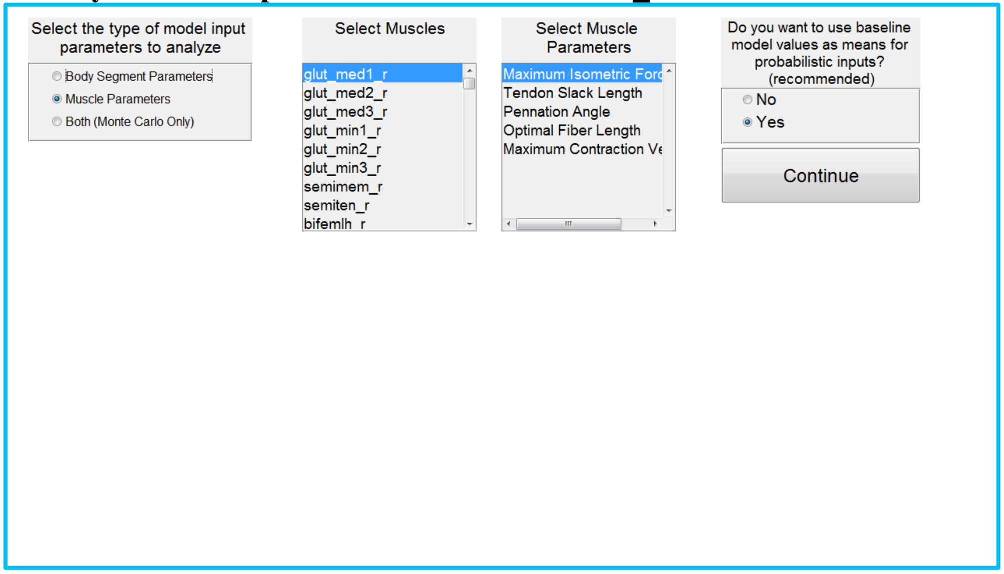


Click "Select Baseline Simulation Results File"
Select the "_force.sto" file located in the Results folder.

Select the "Yes" radio button located under "Would you like to store output data?"
The results from each iteration will be stored in the Results folder. The default is "Yes" to ensure future analysis.

Click "Continue".

Select your static optimization results file for '_force.sto'.

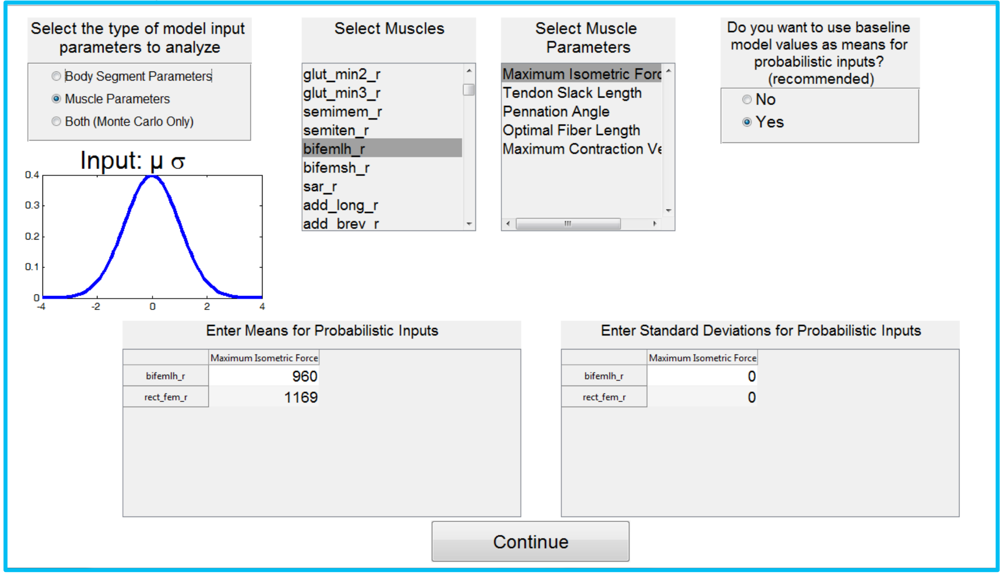


Select the Muscle Parameters radio button to analyze.

Select the biceps femoris long head (bifemlh_r) and rectus femoris (rect_fem_r) on the right side from the list of muscles in the model.

Select Maximum Isometric Force from the list of parameters.

Select the 'yes' radio button to use initial model values and continue.



Enter Means for Probabilistic Inputs

	Maximum Isometric Force
bifemlh_r	960
rect_fem_r	1169

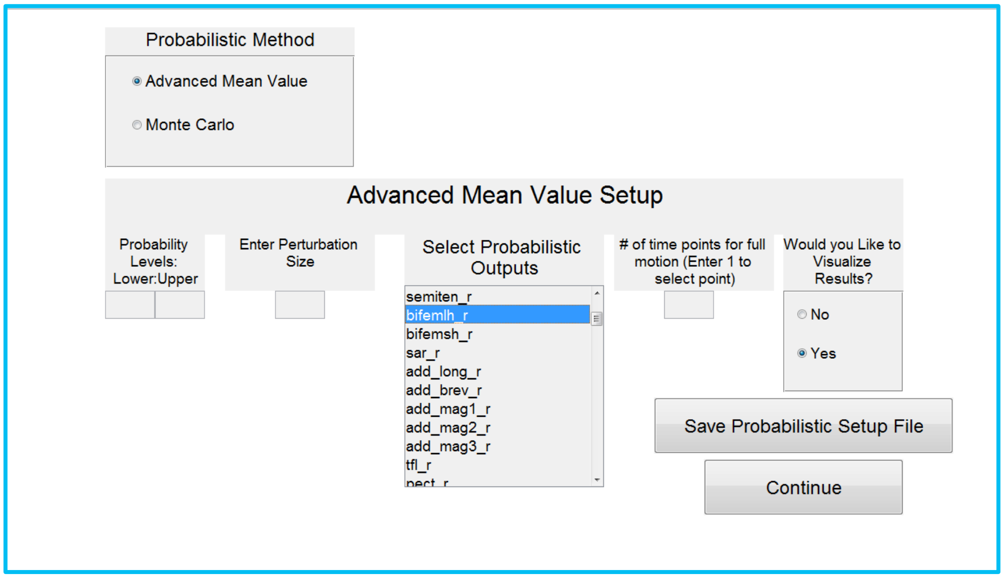
Enter Standard Deviations for Probabilistic Inputs

	Maximum Isometric Force
bifemlh_r	0
rect_fem_r	0

Enter values for standard deviations:

$$\sigma_{\mathrm{BF}} = CV_{\mathrm{BF}} \times \left|\mu_{\mathrm{BF}}\right| = 0.0682 \times 960 = 65.45$$

$$\sigma_{\mathrm{RF}} = CV_{\mathrm{RF}} \times \left|\mu_{\mathrm{RF}}\right| = 0.0456 \times 1169 = 76.71$$



Select the Advanced Mean Value radio button.

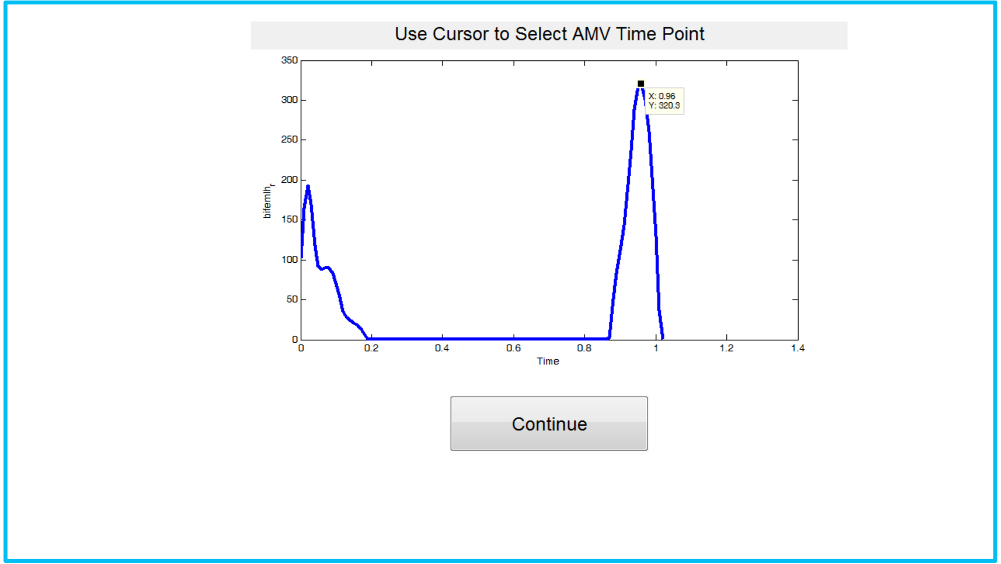
Enter 5 and 95 for the upper and lower probability levels.

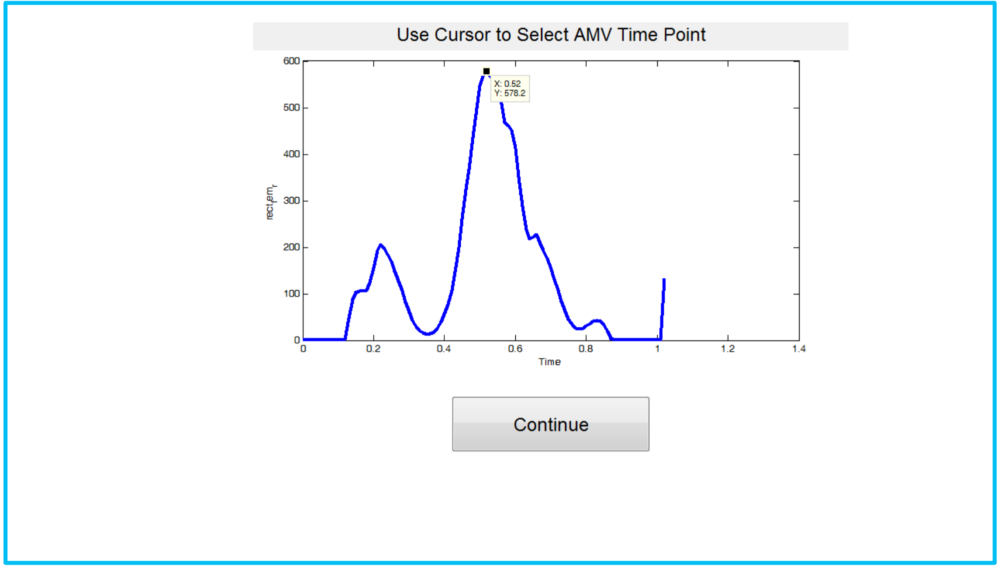
Enter 0.5 for the perturbation size. 0.5 is recommended but the user can use any.

Select the muscles that you chose in your analysis from the list of outputs to assess their results muscle force outputs. (bifemlh_r; rect_fem_r).

Enter 1 for '# the time points for full motion'.
When you continue you will be prompted to select where in the motion you would like the time point to be.

Save the probabilistic setup file, continue.

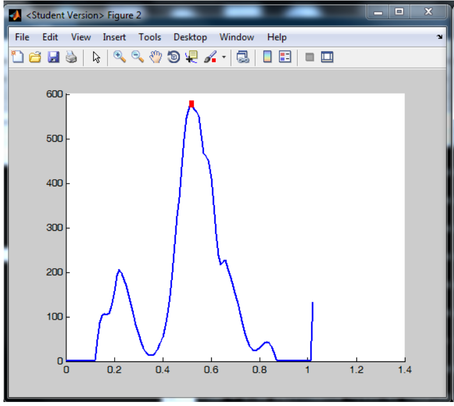


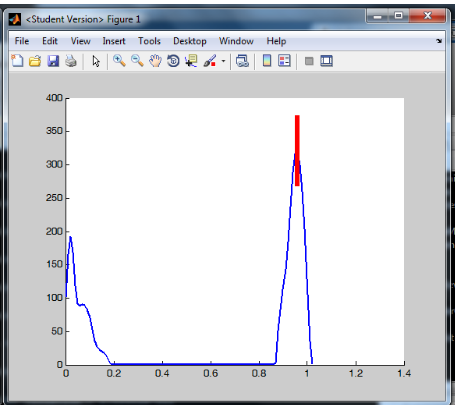


Use the cursor to select the point of maximum force outputs for each muscle, and click Continue.

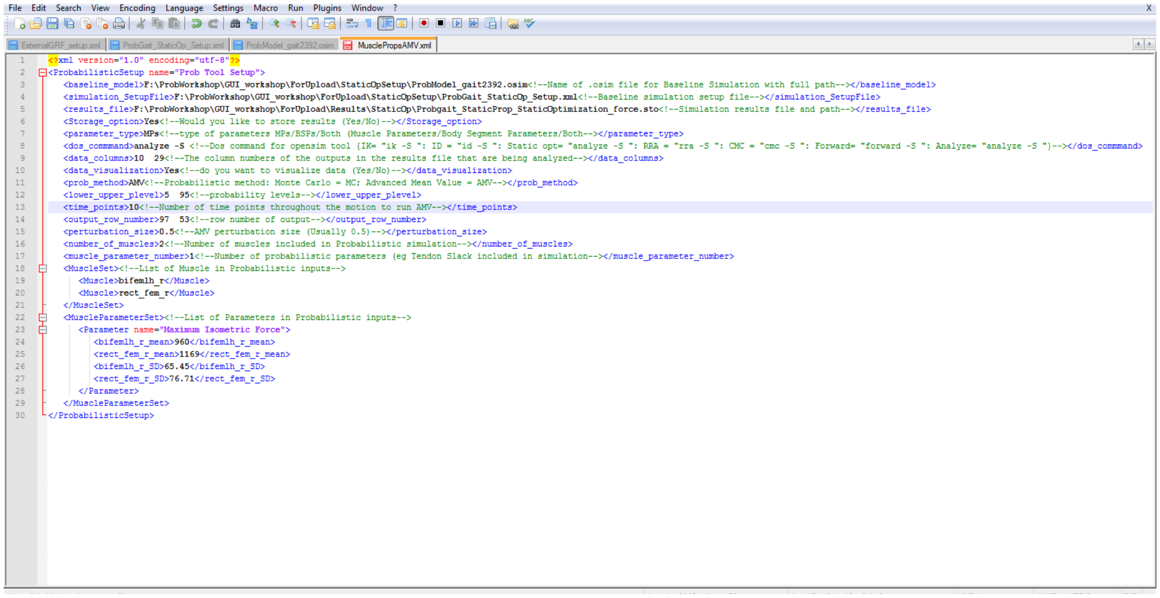
Evaluate Results

Size of 5-95% bounds for the one time point are denoted by the height of the red line. Does it make sense for the rectus femoris bounds to be so small? (Likely due to the peak force occuring during peak hip extension with the knee in a flexed position, putting the rectus femoris in a strentched position where changes in maximum isometric force would have a small effect).





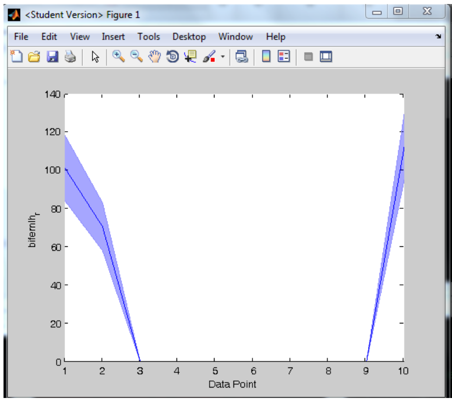
Open the Probabilistic setup file and change the number of time points from 1 to 10.

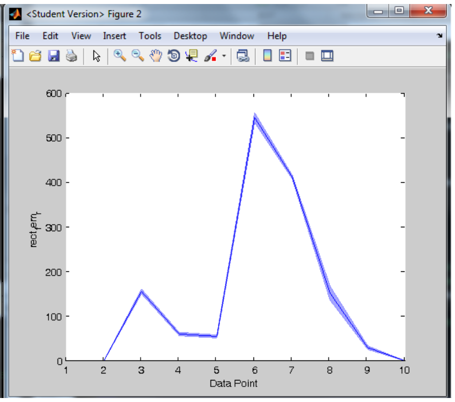


```
<?xml version="1.0" encoding="utf-8"?>
<ProbabilisticSetup name="Prob Tool Setup">
   <baseline_model>F:\ProbWorkshop\GUI_workshop\ForUpload\StaticOpSetup\ProbModel_gait2392.osim<!--Name of .osim file for Baseline Simulation with full path--></baseline_model>
   <simulation_SetupFile>F:\ProbWorkshop\GUI_workshop\ForUpload\StaticOpSetup\ProbGait_StaticOp_Setup.xml<!--Baseline simulation setup file--></simulation_SetupFile>
   <results_file>F:\ProbWorkshop\GUI_workshop\ForUpload\Results\StaticOp\Probgait_StaticProp_StaticOptimization_force.sto<!--Simulation results file and path--></results_file>
   <Storage_option>Yes<!--Would you like to store results (Yes/No)--></Storage_option>
   <parameter_type>MPs<!--type of parameters MPs/BSPs/Both (Muscle Parameters/Body Segment Parameters/Both--></parameter_type>
   <dos_command>analyze -S <!--Dos command for opensim tool {IK= "ik -S ": ID = "id -S ": Static opt= "analyze -S ": RRA = "rra -S ": CMC = "cmc -S ": Forward= "forward -S ": Analyze= "analyze -S "}--></dos_commmand>
   <data_columns>10  29<!--The column numbers of the outputs in the results file that are being analyzed--></data_columns>
   <data_visualization>Yes<!--do you want to visualize data (Yes/No)--></data_visualization>
   <prob_method>AMV<!--Probabilistic method: Monte Carlo = MC; Advanced Mean Value = AMV--></prob_method>
   <lower_upper_plevel>5  95<!--probability levels--></lower_upper_plevel>
   <time_points>10<!--Number of time points throughout the motion to run AMV--></time_points>
   <output_row_number>97  53<!--row number of output--></output_row_number>
   <perturbation_size>0.5<!--AMV perturbation size (Usually 0.5)--></perturbation_size>
   <number_of_muscles>2<!--Number of muscles included in Probabilistic simulation--></number_of_muscles>
   <muscle_parameter_number>1<!--Number of probabilistic parameters (eg Tendon Slack included in simulation--></muscle_parameter_number>
   <MuscleSet><!--List of Muscle in Probabilistic inputs-->
      <Muscle>bifemlh_r</Muscle>
      <Muscle>rect_fem_r</Muscle>
   </MuscleSet>
   <MuscleParameterSet><!--List of Parameters in Probabilistic inputs-->
      <Parameter name="Maximum Isometric Force">
         <bifemlh_r_mean>960</bifemlh_r_mean>
         <rect_fem_r_mean>1169</rect_fem_r_mean>
         <bifemlh_r_SD>65.45</bifemlh_r_SD>
         <rect_fem_r_SD>76.71</rect_fem_r_SD>
      </Parameter>
   </MuscleParameterSet>
</ProbabilisticSetup>
```

Re-run the simulation: evaluate results.

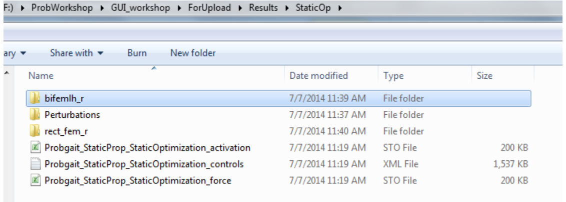
In the graphs you will see the bounds for a more complete gait cycle. Ten time points for this simulation should take approximately 25 minutes.



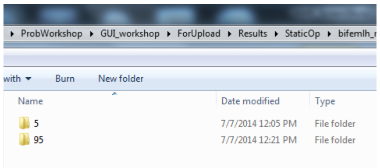


Try with even more points to increase the detail over the gait cycle. Next, take some time and run a Monte Carlo of 250 trials replicating the same same inputs as the AMV (should complete in a little over 2 hours on a computer as decribed above). Compare the results for the size of the bounds. How many time points in AMV were needed to adequately follow the Monte Carlo result?

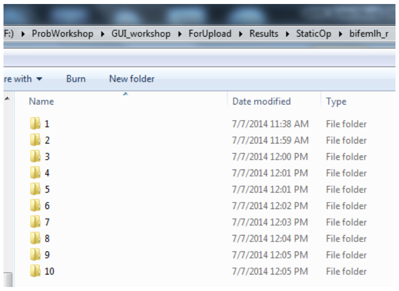
For Future analysis, results appear in three folders. First, the mean and each of the pertubations are run and stored in separate folders in the 'pertubations' folder. Second each muscle is run for the 5 and 95% probability level and results are stored for each time point.



Name	Date modified	Type	Size
bifemlh_r	7/7/2014 11:39 AM	File folder	
Perturbations	7/7/2014 11:37 AM	File folder	
rect_fem_r	7/7/2014 11:40 AM	File folder	
Probgait_StaticProp_StaticOptimization_activation	7/7/2014 11:19 AM	STO File	200 KB
Probgait_StaticProp_StaticOptimization_controls	7/7/2014 11:19 AM	XML File	1,537 KB
Probgait_StaticProp_StaticOptimization_force	7/7/2014 11:19 AM	STO File	200 KB



Name	Date modified	Type
5	7/7/2014 12:05 PM	File folder
95	7/7/2014 12:21 PM	File folder



Name	Date modified	Type
1	7/7/2014 11:38 AM	File folder
2	7/7/2014 11:59 AM	File folder
3	7/7/2014 12:00 PM	File folder
4	7/7/2014 12:01 PM	File folder
5	7/7/2014 12:01 PM	File folder
6	7/7/2014 12:02 PM	File folder
7	7/7/2014 12:03 PM	File folder
8	7/7/2014 12:04 PM	File folder
9	7/7/2014 12:05 PM	File folder
10	7/7/2014 12:05 PM	File folder

First Run:
One time point (max)
Two muscles (ham + quad)
Max isometric force

Second Time Point:
10 time points Multiple time points
Max isometric force

Assignmnent:
Run Monte Carlo Simualtion with Max isometric force

References

Curran-Everett, D., 2009. Explorations in statistics: confidence intervals. Adv. Physiol. Educ. 33, 87–90.

Valente, G., Taddei, F., Jonkers, I., 2013. Influence of weak hip abductor muscles on joint contact forces during normal walking: probabilistic modeling analysis. J. Biomech. 1–8.

Wu, Y.-T., Millwater, H., Cruse, T., 1990. Advanced probabilistic structural analysis method for implicit performance functions. AIAA J. 28, 1663–1669.

Zajac, F.E., 1993. Muscle coordination of movement: a perspective. J. Biomech. 26, 109–124.

Appendix A: Literature that contain quantitative information for parameter uncertainty

This list of peer-reviewed literature has been helpful for estimating the coefficients of variation for model parameters.

Marker Placement/Movement Artifact

Della Croce, U., A. Cappozzo, and D. Kerrigan. Pelvis and lower limb anatomical landmark calibration precision and its propagation to bone geometry and joint angles. *Med Biol Eng Comput* 37:155–161, 1999.

Benoit, D. L., D. K. Ramsey, M. Lamontagne, L. Xu, P. Wretenberg, and P. Renström. Effect of skin movement artifact on knee kinematics during gait and cutting motions measured in vivo. *Gait Posture* 24:152–64, 2006.

Gao, B., and N. N. Zheng. Investigation of soft tissue movement during level walking: translations and rotations of skin markers. *J Biomech* 41:3189–95, 2008.

Body Segment Parameters

De Leva, P. (1996). Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of Biomechanics*, *29*(9), 1223–1230.

Langenderfer, J. E., Laz, P. J., Petrella, A. J., & Rullkoetter, P. J. (2008). An efficient probabilistic methodology for incorporating uncertainty in body segment parameters and anatomical landmarks in joint loadings estimated from inverse dynamics. *Journal of Biomechanical Engineering*, *130*(1), 014502.

Rao, G., Amarantini, D., Berton, E., and Favier, D., 2006, "Influence of Body Segments' Parameters Estimation Models on Inverse Dynamics Solutions During Gait," J. Biomech, **39**, pp. 1531–1536.

Reinbolt, J. a, Haftka, R. T., Chmielewski, T. L., & Fregly, B. J. (2007). Are patient-specific joint and inertial parameters necessary for accurate inverse dynamics analyses of gait? *IEEE Transactions on Biomedical Engineering*, *54*(5), 782–93.

Muscle Properties

Friederich, J. A., and R. A. Brand. Muscle fiber architecture in the human lower limb. *J Biomech* 23:91–95, 1990.

Ward, S. R., C. M. Eng, L. H. Smallwood, and R. L. Lieber. Are current measurements of lower extremity muscle architecture accurate? *Clin Orthop Relat Res* 467:1074–82, 2009.