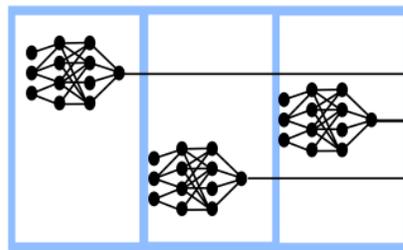
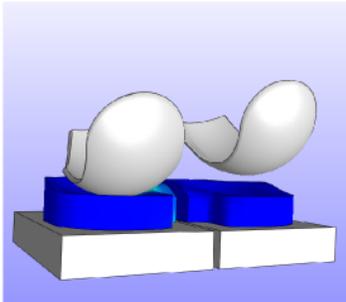
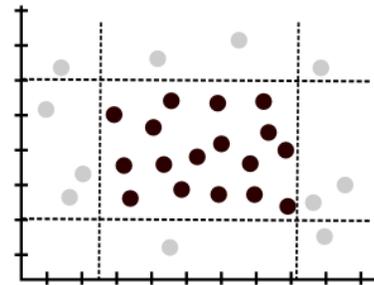
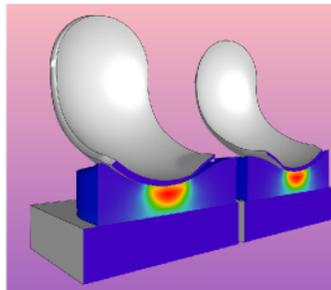
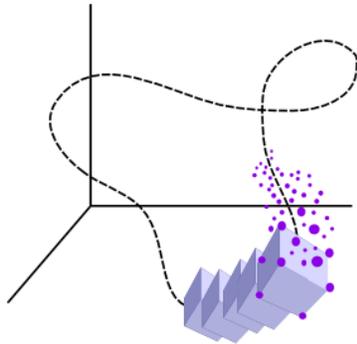


# SCMT Documentation



## Surrogate Contact Modeling Toolbox User Guide

Version 1.0

(updated September 2015)

Ilan Eskinazi

<https://simtk.org/home/scmt/>

# Table of Contents

- 1 Introduction ..... 4
  - 1.1 What is SCMT ..... 4
  - 1.2 What is a surrogate model ..... 4
  - 1.3 Using this manual ..... 4
  - 1.4 How to acknowledge us ..... 4
- 2 Installing SCMT ..... 5
  - 2.1 Pre-requisites ..... 5
  - 2.2 Downloading SCMT ..... 5
  - 2.3 Setup paths ..... 5
- 3 FEBio model ..... 5
  - 3.1 Create meshes ..... 6
  - 3.2 Create FE model ..... 6
    - 3.2.1 Create new file and add materials ..... 6
    - 3.2.2 Create “fake” rigid bodies ..... 6
    - 3.2.3 Sliding contact ..... 6
    - 3.2.4 Constraints and load curves ..... 7
    - 3.2.5 Analysis steps ..... 9
    - 3.2.6 Output request ..... 9
    - 3.2.7 Placement and Center of mass/rotation ..... 9
    - 3.2.8 Export FEB file and test ..... 9
- 4 Sample Point Generator ..... 10
  - 4.1 Create a trajectory file ..... 11
  - 4.2 Select domain type ..... 11
  - 4.3 Select number of Hammersley points ..... 12
  - 4.4 Select expansion vector ..... 12
  - 4.5 Select minima and maxima ..... 12
  - 4.6 Choose Hammersley settings ..... 12
  - 4.7 Add and manage domains ..... 13
- 5 FEBio Model Sampler ..... 14
  - 5.1 Specify FEBio model ..... 15

5.2	Specify FEBio model settings .....	15
5.3	Load sample points and run Sampler .....	15
6	Sample Point Filter .....	16
6.1	Load sample points .....	17
6.2	Add filters .....	17
6.3	Filter and save .....	17
7	Out-of-Contact Sampler .....	18
7.1	Load pre-existing sample points .....	19
7.2	Set general settings .....	19
7.3	Fully out-of-contact .....	19
7.4	One side out-of-contact.....	19
8	Surrogate Model Designer .....	20
8.1	Set inputs and outputs .....	21
8.2	Set neural network training criteria.....	21
8.3	Write formatted data .....	21
9	Surrogate Model Tester .....	22
9.1	Load files.....	23
9.2	Compute errors .....	23
10	Utilities.....	24
10.1	Joining sets of sample points .....	25
10.2	Exporting files as comma separated values (CSV) .....	25
10.3	Conversion between XML and BIN .....	25
11	Plugin for OpenSim .....	26
11.1	What does it do .....	26
11.2	Place the plugin file in the plugin folder.....	26
11.3	Place the DLLs in Opensim installation bin folder .....	27
11.4	Place the surrogate/stageMap XML file in the Opensim installation folder.....	27
11.5	Setup your force component in the osim file .....	27
12	Surrogate Model Use in C++ .....	29
13	Surrogate Model Use in Matlab .....	30

# 1 Introduction

## 1.1 What is SCMT

SCMT stands for surrogate contact modeling toolbox. SCMT is a software package that allows for the semi-automated generation of hyper-fast contact models based on finite element simulations. The surrogate contact models generated with this toolbox can output load quantities as a function of the position and orientation between two contacting bodies, but could also have combinations of loads and pose parameters as inputs and as outputs.

## 1.2 What is a surrogate model

Surrogate models operate similar to a look up table. Plenty of sample points are obtained relating model inputs to model outputs. This is a time consuming operation that can take hours or days to be performed. Once the data is collected, the surrogate model is built which fits or interpolates the data. For this toolbox, the surrogate models are composed of artificial neural networks.

## 1.3 Using this manual

This manual will explain how to use the different tools in the SCMT GUI for the purpose of creating a surrogate contact model.

## 1.4 How to acknowledge us

If you are using SCMT for your work, please cite the following paper in your publication:

I. Eskinazi and B. J. Fregly, "An Open-Source Toolbox for Surrogate Modeling of Joint Contact Mechanics," *IEEE Trans Biomed Eng.*, DOI: 10.1109/TBME.2015.2455510.

(Currently e-pub ahead of print)

## 2 Installing SCMT

### 2.1 Pre-requisites

- SCMT requires Microsoft Windows as the operating system.
- The SCMT workflow makes use of FEBio finite element software. This is open-source and free for non-commercial work (<http://febio.org/>). The preprocessing and post-processing software “Preview” and “Postview” that goes along with FEBio are also recommended for finite element model setup and testing.
- You will need Matlab (at least version 2013b) with its Neural network toolbox and its automated code generation toolbox. This allows for neural network training and deployment.
- To make the surrogate models, Matlab, and Opensim work well together you must use 32-bit versions for all of them or 64-bit for all of them. 32-bit is recommended for compatibility because the GUI was compiled as a 32-bit program. If the DLLs are generated with Matlab 64-bit, the SCMT tester from the pre-compiled executable won't work with those DLLs. Of course, since the source code is available, you may compile SCMT as 32-bit or 64-bit on your own.

A computer with many processing cores is recommended to reduce computation time for sampling and training neural networks.

### 2.2 Downloading SCMT

SCMT is available in the downloads page of <https://simtk.org/home/scmt/>

Download the SCMT executable file with supporting DLLs and it should be ready to run.

### 2.3 Setup paths

Make sure Matlab is in your path by opening a command line and typing “matlab.exe.” This should start Matlab.

Make sure FEBio is installed and in your path by typing the executable name in the command line. It could be “febio.exe” or “FEBiox64.exe”, etc.

## 3 FEBio model

SCMT has a few requirements regarding the FEBio model:

- There must be one or two fixed rigid bodies and one moving rigid body

- The output requests for the logfile must be exactly as described in the manual. This is important because SCMT parses the output from the logfile every time a simulation is ran.

### 3.1 Create meshes

SCMT does not care about the meshes, whether or not they are fine or coarse, or what type of elements are used. However it is recommended that hexahedral meshes are used to model the contacting solids in FEBio. One software package that does this meshing is TrueGrid (<http://www.truegrid.com/>).

### 3.2 Create FE model

#### 3.2.1 Create new file and add materials

Create a new PRV file with Preview and import the meshes. Click on “Physics → Add material.” Then set up all properties and the meshes associated with the material.

#### 3.2.2 Create “fake” rigid bodies

Create non-deformable bodies by adding a mesh (could be a simple one element cuboid) and assigning a rigid body material to it. Then attach the rigid bodies to deformable bodies using a rigid interface (Physics → Add Contact → Rigid Interface).

In the figures below, fake rigid bodies (cuboids) are added and rigidly attached to a deformable patellar button in one case and to tibial inserts in the other case. All contact loads will be calculated about the fake rigid bodies’ centers of mass.

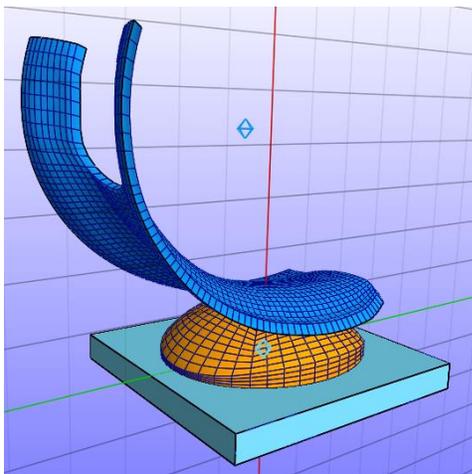


Figure 1. Example of single fake fixed rigid body.

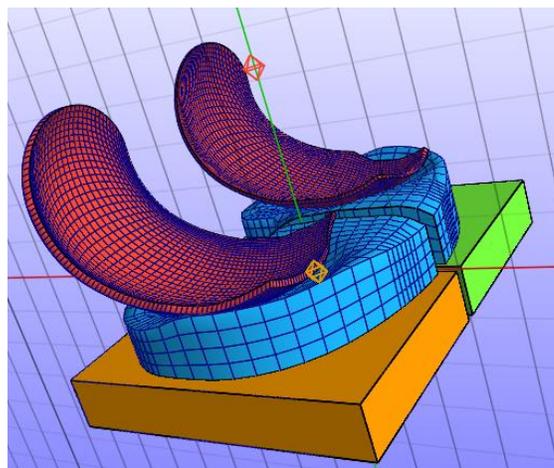


Figure 2. Example of two fake fixed rigid bodies.

#### 3.2.3 Sliding contact

Click on “Physics → Add Contact → Sliding Interface” to add the sliding contact between the solids. Select the elements corresponding master surface and the slave surface. For more information see the Preview/FEBio manuals.

### 3.2.4 Constraints and load curves

The fixed rigid body must have a rigid constraint associated with it. Add one by clicking “Physics → Add Rigid Constraint → Fixed Displacement/Rotation.” For the moving body, apply rigid constraints as well. You’ll probably have both prescribed Displacement/Rotation constraints as well as Prescribed Torque/Force constraints. Make sure all values in the constraint properties are 1.

Now click on the curve editor. It is the symbol with the red line as shown below.



Figure 3. The curve editor on the top menu is the icon with the red line.

In the curve editor, make sure every prescribed pose and load curve has three nodes and the each curve type is linear. The next part is key:

- For prescribed translations/rotations, the first node should be  $(-0.5,0)$ , the second  $(0,0)$ , and the third  $(0.5,y)$  where  $y$  is the prescribed final value for the pose parameter.
- For prescribed forces/torques, the first node should be  $(0,v1)$ , the second  $(0.5,v2)$ , and the third  $(1,v3)$ . The values  $v1$  and  $v2$  are chosen by the user in order by trial and error. The value  $v3$  is the prescribed final value for the

load/torque. For some constraints, v1 and v2 would ideally apply a large load such that the bodies don't come out of contact during the simulation.

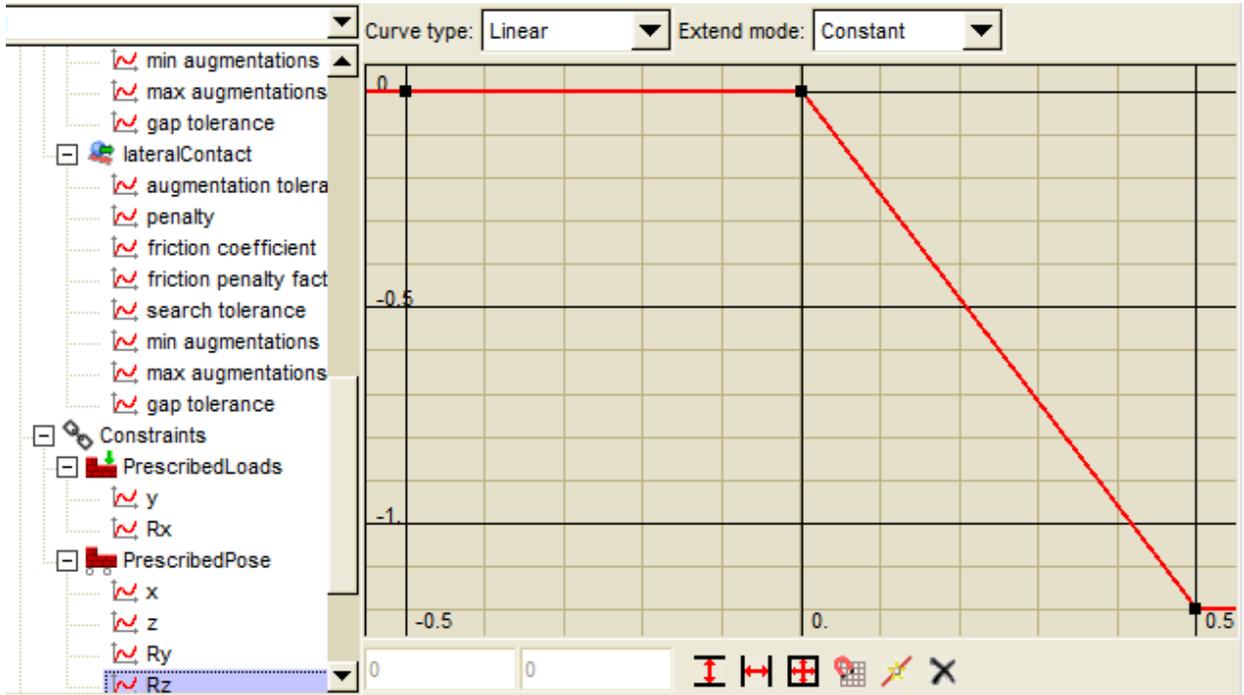


Figure 4. Example of load curve for prescribed pose

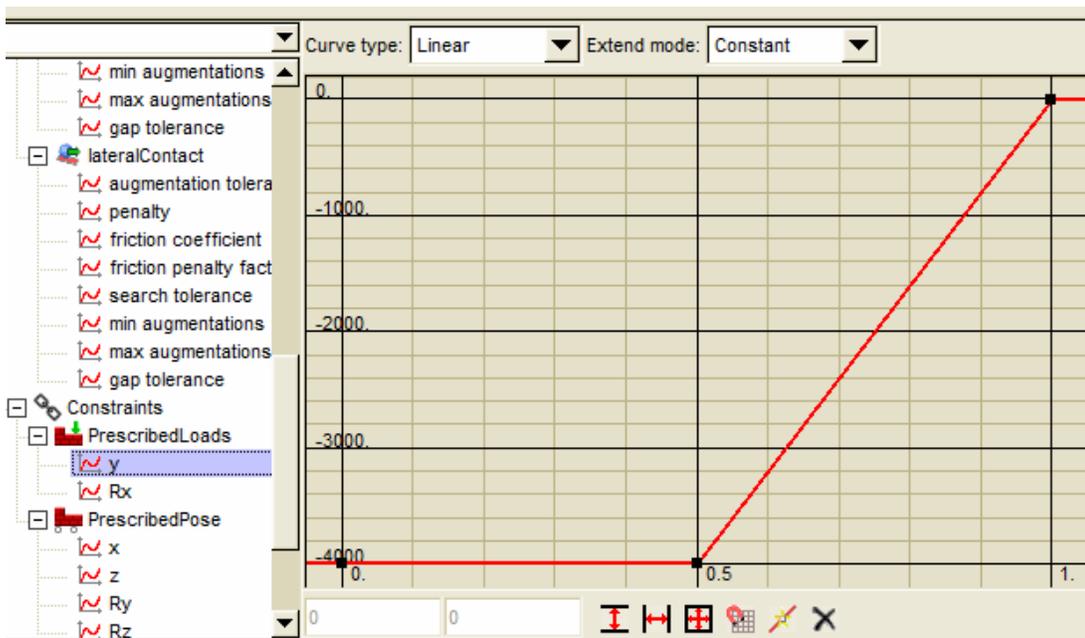


Figure 5. Example of load curve for prescribed load

### 3.2.5 Analysis steps

Add an analysis step by clicking on “Physics → Add Step.” Go to Time Stepping and set to “use auto time stepper” and “use must points.” Set Time steps to 10 with a step size of 0.05. This means that each step is 0.5 s long. The max step to 0.5, and min step to 0.05. Set max retries to 5 and optimal iterations to 10.

Then go to Nonlinear solver. Select BFGS as solution method with a Displacement tolerance of 0.001, energy tolerance of 0.01, and 0 for Residual tolerance. The Max reformations are limited to 15 and Max BFGS updates to 10. These parameters may all be changed according to user needs. Repeat the process by adding another analysis step.

### 3.2.6 Output request

Right click Output Request in the navigation pane and click “Edit...” Go to the logfile tab. For type select “Rigid body,” then for each fixed rigid body add “Fx;Fy;Fz;Mx;My;Mz” in the field for Data. For the moving body add “x;y;z;qx;qy;qz;qw”. The figure below should illustrate the result in the case of two fixed rigid bodies.

### 3.2.7 Placement and Center of mass/rotation

The contacting bodies must be put in a position where the contact surfaces are penetrating slightly. After doing this transformation, remember to move the center of rotation of the moving body with its mesh.

Type	Data	List
Rigid Body	x;y;z;qx;qy;qz;qw	rigidMovingMat
Rigid Body	Fx;Fy;Fz;Mx;My;Mz	rigidFixedMat_media
Rigid Body	Fx;Fy;Fz;Mx;My;Mz	rigidFixedMat_latera

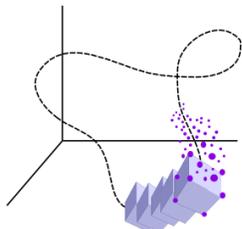
Figure 6. Output requests added each on with proper order.

### 3.2.8 Export FEB file and test

Export the .feb file by clicking “File → Export”

Run file using FEBio’s command line and verify that model works properly.

## 4 Sample Point Generator



The sample point generator takes a 6D trajectory defined by the user to define the domains in 6D space to be sampled.

Surrogate Contact Modeling Toolbox

Help

Sample Point Generator | FEBio Sampler | Data Filtering | Surrogate Designer | Out-of-contact | Testing | Utilities

Load Trajectory D:/Research/SCMT/Surrogate Files/JW\_TF\_ngait\_traj.txt

Domain Name: D1

Bounding Box  Hammersley  Vertices

Number of Hammersley points: 180 Hammersley step: 0

Expansion vector

2 350 0.5 5500 0.05 0.2

offset

Minima

-999999 -999999 -999999 -999999 -999999 -999999

Maxima

999999 -5 999999 999999 999999 999999

Seed

0 0 0 0 0 0

Leap

1 1 1 1 1 1

Base

2 3 5 7 11 13

Domains:

D1  
D2  
D3  
D4

Examine domain

Overwrite domain

Remove selected domain

Clear domains

Save XML

**Generate Sample points**

Save Settings

Load Settings

Add new domain

Sample Point Generator loaded

Figure 7. Screenshot of Sample Point Generator tool.

## 4.1 Create a trajectory file

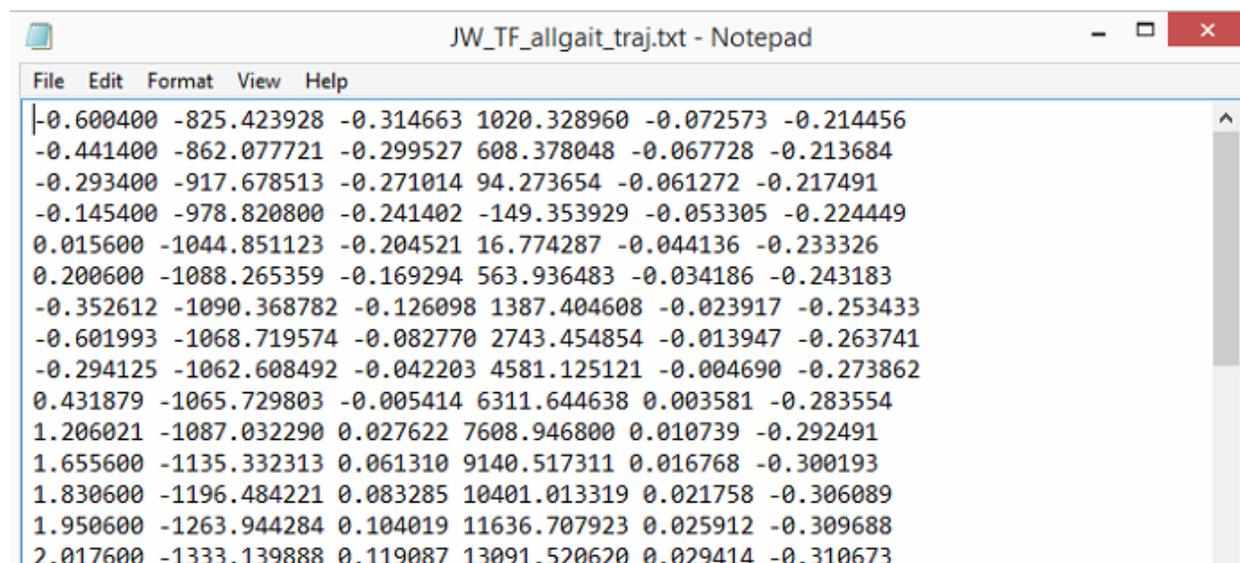
The trajectory should be formatted as a text file. Each row should contain six values, and the order is the following:

x-translation/force    y-translation/force    z-translation/force    x-rotation/torque    y-rotation/torque    z rotation/torque

The choice of translation vs. force and rotation vs. torque will depend on how the user chooses to simulate the FEBio model. If the FEBio contact model is simulated by applying a y-force and an x-torque as rigid body constraints, and the other spatial directions have prescribed displacement constraints, then the values in the trajectory should be as follows:

x-translation    y-force    z-translation    x-torque    y-rotation    z-rotation

The number of rows in the trajectory file is also important if later on the user does not check the “Bounding Box” option.



```

File Edit Format View Help
|-0.600400 -825.423928 -0.314663 1020.328960 -0.072573 -0.214456
-0.441400 -862.077721 -0.299527 608.378048 -0.067728 -0.213684
-0.293400 -917.678513 -0.271014 94.273654 -0.061272 -0.217491
-0.145400 -978.820800 -0.241402 -149.353929 -0.053305 -0.224449
0.015600 -1044.851123 -0.204521 16.774287 -0.044136 -0.233326
0.200600 -1088.265359 -0.169294 563.936483 -0.034186 -0.243183
-0.352612 -1090.368782 -0.126098 1387.404608 -0.023917 -0.253433
-0.601993 -1068.719574 -0.082770 2743.454854 -0.013947 -0.263741
-0.294125 -1062.608492 -0.042203 4581.125121 -0.004690 -0.273862
0.431879 -1065.729803 -0.005414 6311.644638 0.003581 -0.283554
1.206021 -1087.032290 0.027622 7608.946800 0.010739 -0.292491
1.655600 -1135.332313 0.061310 9140.517311 0.016768 -0.300193
1.830600 -1196.484221 0.083285 10401.013319 0.021758 -0.306089
1.950600 -1263.944284 0.104019 11636.707923 0.025912 -0.309688
2.017600 -1333.139888 0.119087 13091.520620 0.029414 -0.310673

```

Figure 8. Example of a trajectory file.

## 4.2 Select domain type

Once a trajectory has been loaded, the user should give a name to the domain and select from a few options.

If the “Bounding Box” options is checked, then the sample points will cover a domain of large span and low density which will encompass the 6D trajectory. IF the option is not selected, then each row of the trajectory will be expanded into a small subdomain. The sample points generated in this case will cluster about the 6D trajectory.

The “Hammersley” option should normally be checked as it specifies that the space filling method will be used to generate sample points.

The “vertices” option specifies whether or not sample points will be generated at the vertices of the subdomains as well.

### 4.3 Select number of Hammersley points

If the “Bounding Box” option was checked then this is the total number of Hammersley points generated. Otherwise, the total number of Hammersley points would be the one chosen by the user in the field multiplied by the number of rows in the trajectory file. For example, if the user sets 100 in the field listing “number of Hammersley points”, and the trajectory contains 11 rows, and the option of “Bounding Box” is unchecked, then the total number of Hammersley points would be 1100.

### 4.4 Select expansion vector

The expansion vector should be filled with offsets or factors that are to expand the nominal 6D trajectory provided.

Four scenarios are possible and are summarized in the following table

*Table 1. Table illustrating expansion types*

	<b>Expansion factor</b>	<b>Expansion offset</b>
<b>Bounding box</b>	About midpoint of trajectory max and min. bound = midpoint $\pm$ $0.5*(\text{max}-\text{min})*\text{factor}$	About midpoint of trajectory max and min. Upper bound = max + offset Lower bound = min - offset
<b>Not Bounding box</b>	About each trajectory value bound = value $\pm$ $0.5*(\text{max}-\text{min})*\text{factor}$	About each trajectory value bound = value $\pm$ offset

### 4.5 Select minima and maxima

Sometimes expanding the trajectory results in configurations that have unrealistic inputs. These minima and maxima allow users to place limits on the expansions of the subdomains.

### 4.6 Choose Hammersley settings

The seed, leap, base, and step define the Hammersley sequence to be used for sample point generation.

The step is the sequence index. It should be larger or equal to zero.

The seed is the sequence element corresponding to a step of zero. It should be larger or equal to zero.

The leap is the number of successive jumps in the Hammersley sequence.

The base is a series of integers, typically prime, needed for Hammersley sequence generation.

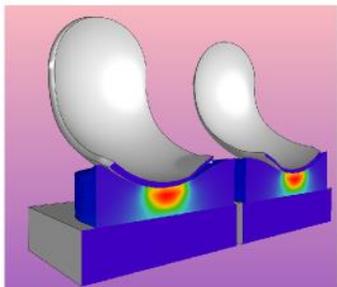
The step<sup>th</sup> sequence element is selected from the original Hammersley sequence as follows:

Seed + step\*leap

#### **4.7 Add and manage domains**

Once the settings are defined, the domain can be added by clicking the “Add New Domain” button. It should now appear on the list to the right. To modify the domain, select it from the list, click on “Examine Domain” and after doing any changes, click on “Overwrite Domain.” Make sure no two domains share the same name. A Sample Point Generator object can be saved and loaded into the GUI by clicking on the save settings load settings buttons. After all domains have been added, click on “Generate Sample Points” and save the data as a .bin file. These sample points generated are only populated with inputs and have to go to the FEBio Sampler to get populated with outputs.

## 5 FEBio Model Sampler



The FEBio Sampler takes sample points populated with inputs only, and populates all the fields via repeated file parsing and calls to FEBio, all done in a multithreaded manner.

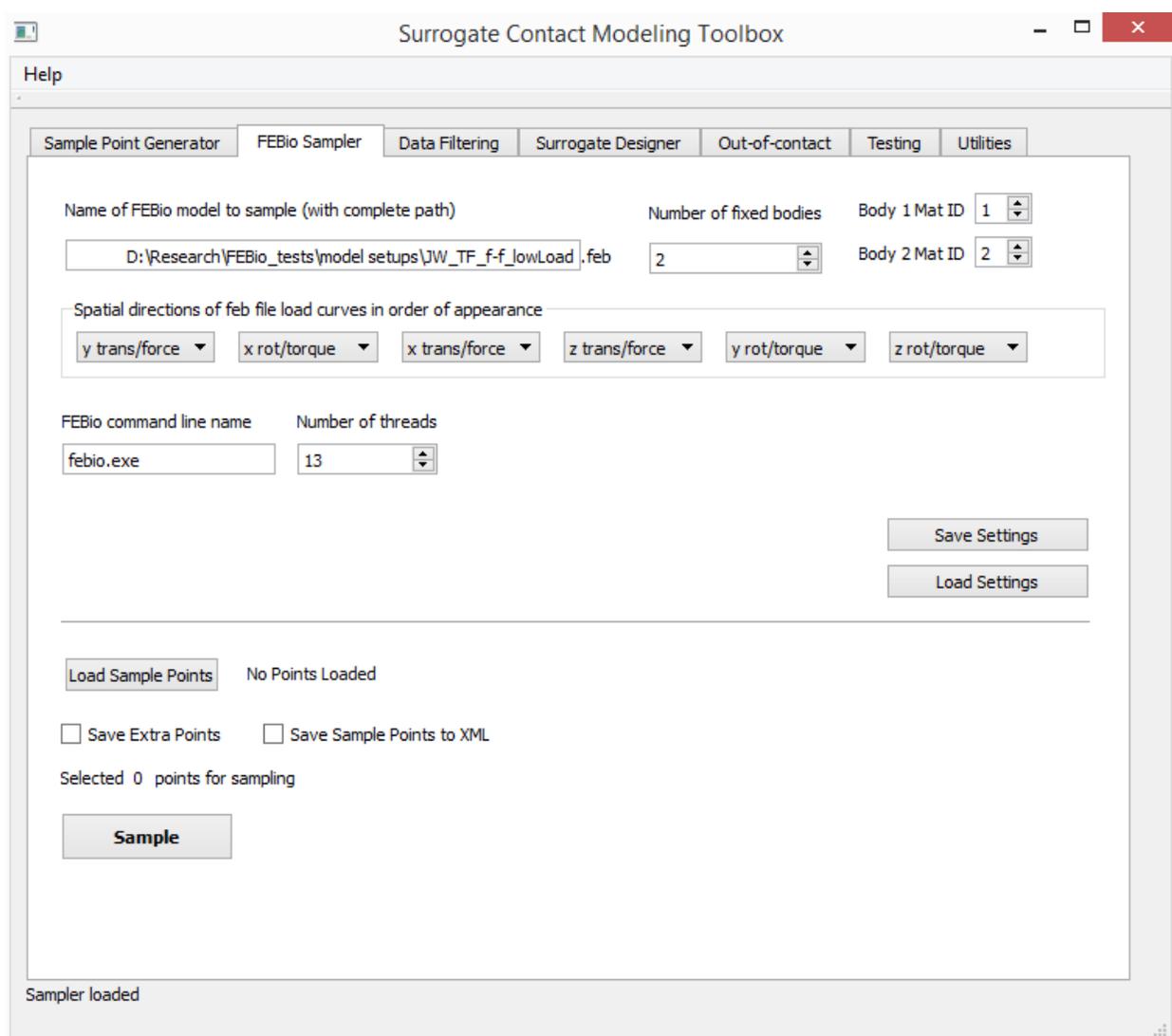


Figure 9. Screenshot of FEBio Sampler tool.

### **5.1 Specify FEBio model**

Copy the name of the FEBio model into the appropriate field. Make sure it includes the path unless it is in the same folder as the SCMT executable.

### **5.2 Specify FEBio model settings**

Specify the number of fixed rigid bodies in the FEBio model. The only options are 1 or 2. Now, specify the material IDs for the fixed rigid bodies. This can be obtained directly from the FEB file.

Now scroll through the FEB file to the load curve section. There should be two load curves of type “Step” associated with the analysis steps. After those, there should be six load curves associated with the rigid body constraints. The spatial direction of each of those load curves should be listed in order in the dropdown lists (or comboboxes).

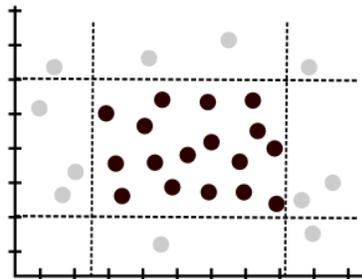
List the command line name for FEBio in the appropriate field. Lastly, specify the number of threads to be used for sampling. This should be equal or less to the number of processing cores in the computer.

### **5.3 Load sample points and run Sampler**

Click on the “Load Sample Points” button and select the .bin file containing the sample points populated with inputs.

Before running using the “Sample” button, notice two optional check boxes. The “Save Extra Points” options allows to save intermediate sample points from the simulations. These extra sample points are not uniformly distributed and should not be used directly as part of the surrogate model creation process. Advanced users may use these points for testing, etc. The option to “Save Sample Points to XML” saves the sample points populated with inputs and outputs to and XML file. This is for testing only with a small number of sample points as the memory requirements of XML files are large and can crash the program.

## 6 Sample Point Filter



The data filter is used to remove sample points that are beyond what's deemed necessary or physically realistic.

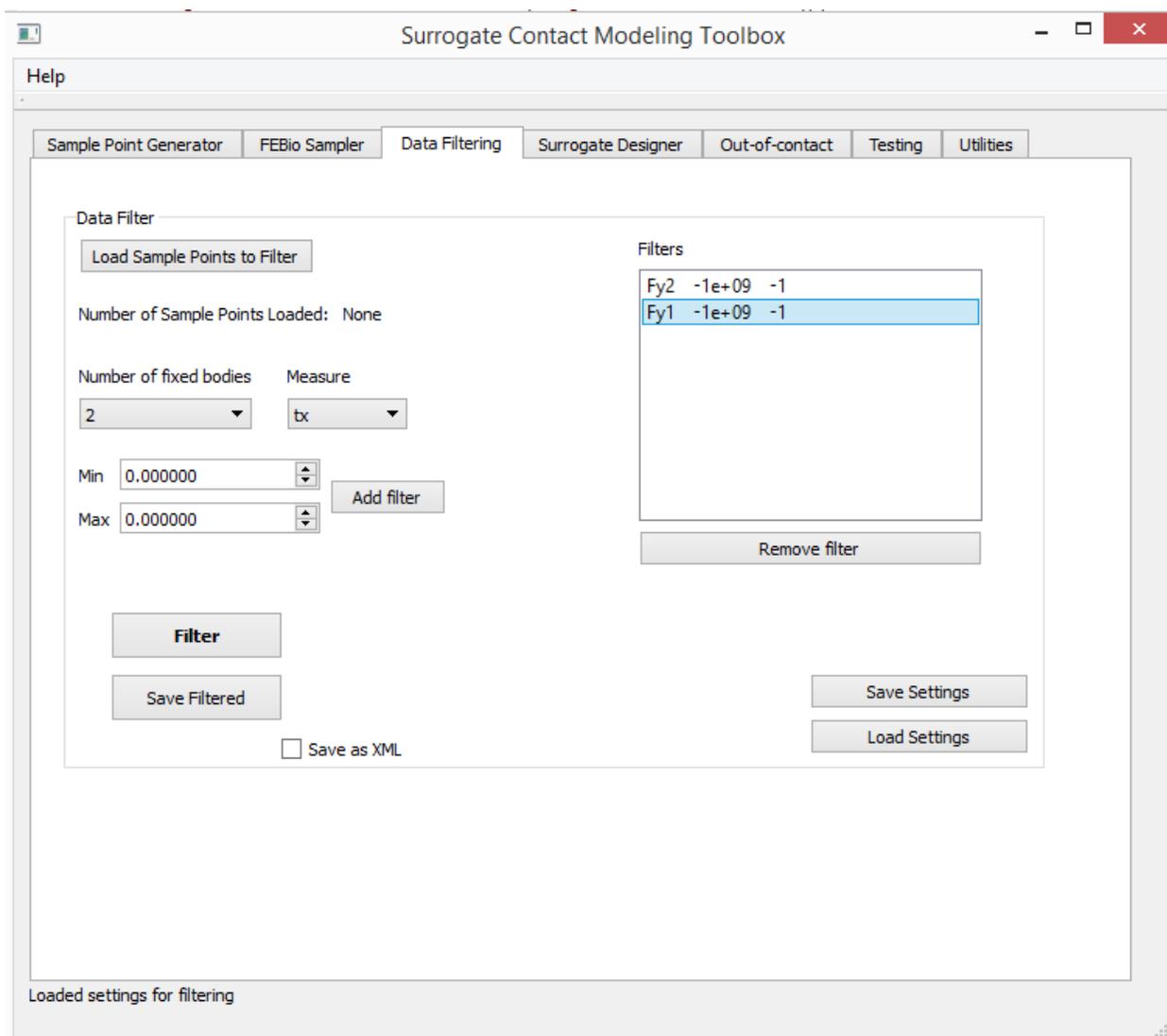


Figure 10. Screenshot of Data Filtering tool.

### **6.1 Load sample points**

Click on the “Load Sample Points to Filter” button and select the .bin file containing your data. The number of sample points in the file loaded is then shown in the label below the button.

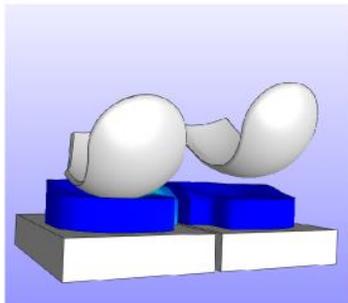
### **6.2 Add filters**

Select the number of fixed rigid bodies in your FEBio model and select a measure from the list/comboBox by which to filter. Now choose values for the minimum and maximum allowed values for the selected measure. When done, click the “Add filter” button. The filter information should show up on the list to the right. Continue to add filters as required.

### **6.3 Filter and save**

When all filters have been added, click the “Filter” button. The number of remaining sample points after filtering should show up in the status bar below. You may now continue to adjust your filters if necessary. When satisfied, press “Filter” again, and save the filtered sample points by clicking “Save Filtered.”

## 7 Out-of-Contact Sampler



The “Out-of-Contact” tool allows users to generate sample points in out-of-contact configurations.

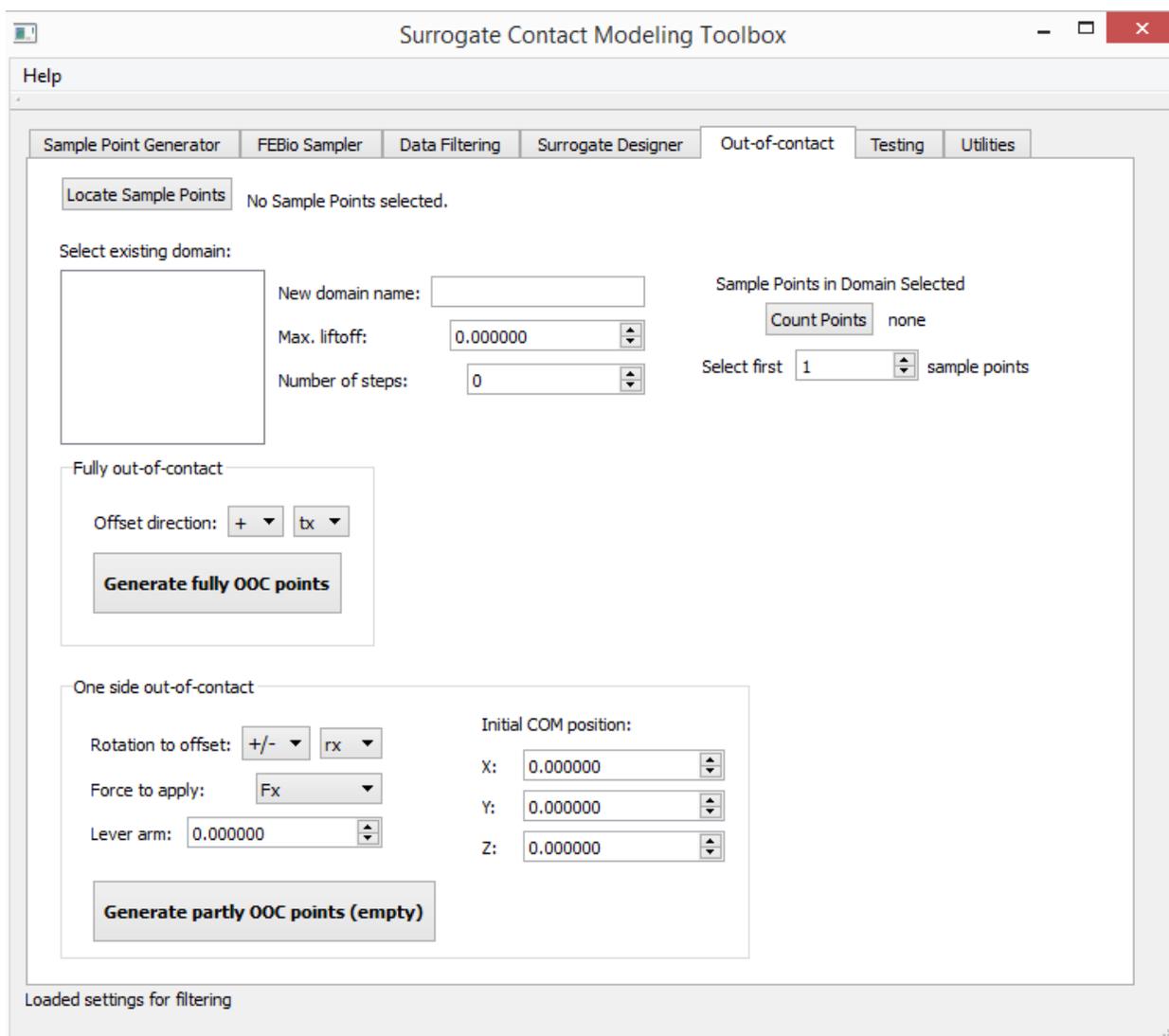


Figure 11. Screenshot of Out-of-Contact tool.

## 7.1 Load pre-existing sample points

Sample points already populated with pose parameters and loads in the Sampler are loaded using the “Locate Sample Point” button. After loading sample points, the list to the left will indicate the existing domains in the data set. If generating fully out-of-contact sample points, select a domain containing configurations at the contact boundary. If generating partly out-of-contact sample points, select a domain with a large range of contact load amplitudes.

## 7.2 Set general settings

The following steps should be performed in any order:

- Give a name to the new domain with out-of-contact points
- Select the maximum liftoff for the contacting surfaces
- Select number of sampling steps in between contacting and maximum liftoff configurations
- Press the “Count Points” button and select how many sample points to draw from the selected domain.

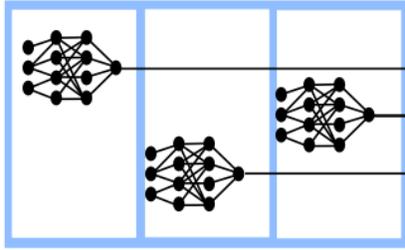
## 7.3 Fully out-of-contact

Select the offset direction  $\pm tx$ ,  $\pm ty$ , or  $\pm tz$  in which to translate the moving body in order to obtain all loads of zero. Click the “Generate fully OOC points” to save the new sample points populated with loads of zero.

## 7.4 One side out-of-contact

This option is for FEBio models with two fixed rigid bodies. This tool takes each pre-existing sample point, and changes the rotation angle by  $\pm rx$ ,  $\pm ry$ , or  $\pm rz$ . Then it keeps one compressive load  $F_x$ ,  $F_y$ , or  $F_z$  from the pre-existing sample point to be applied. The lever arm, or estimated distance between the two contact patches should be put in as well as the initial position of the center of mass/center of rotation of the moving body. Click on the “Generate partly OOC points (empty)” button to save the generated sample points. In contrast with the fully out-of-contact sample points, these sample points are empty. That is, they must be loaded into the FEBio Sampler and simulations should be ran to populate the sample points’ fields.

## 8 Surrogate Model Designer



The surrogate designer has various functions. It allows the user to specify the architecture of the surrogate model in terms of inputs and outputs to each one of its stages. It also specifies the neural network training criteria. Additionally, it converts populated sample points into training and testing data files that are read by Matlab.

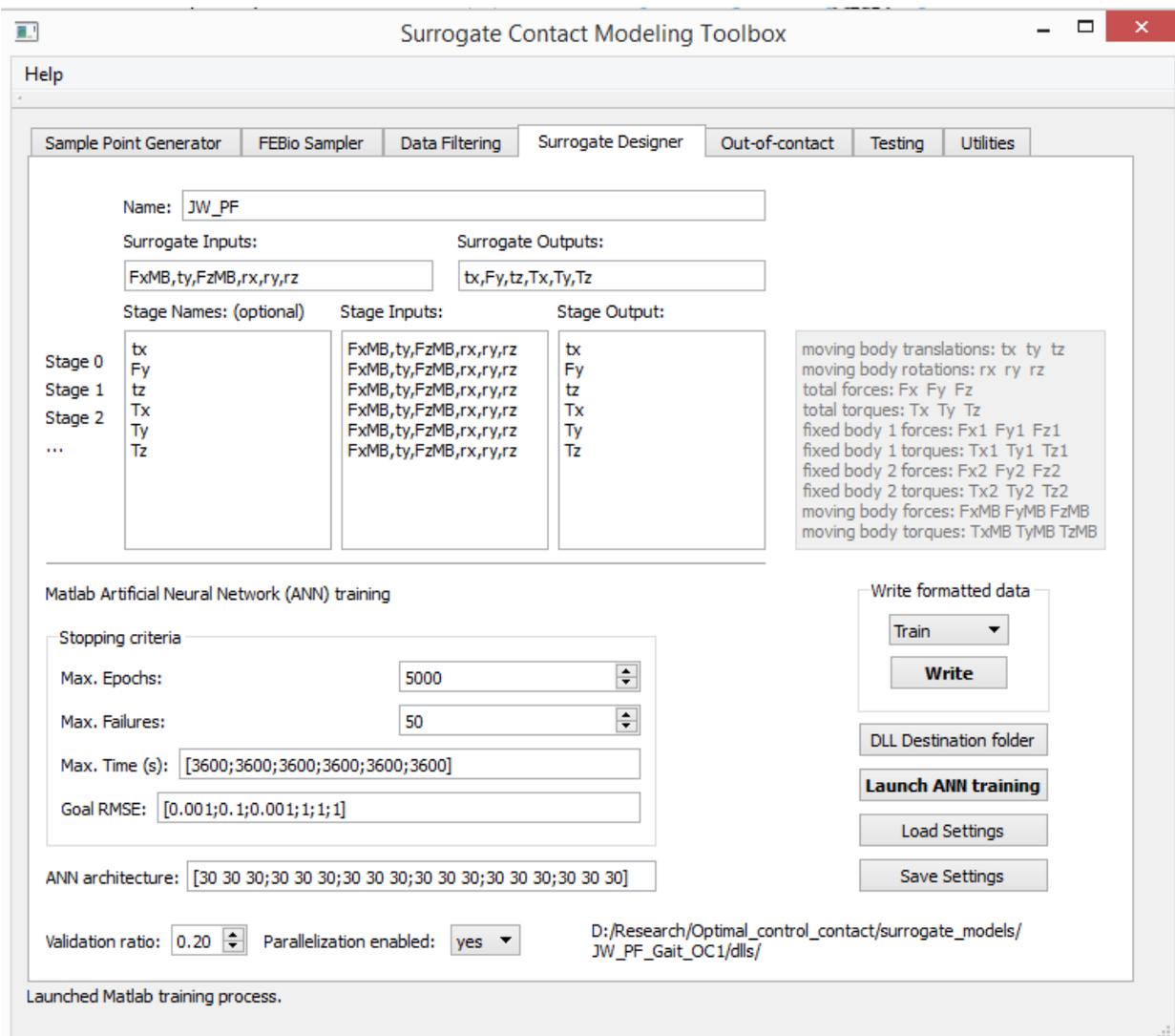


Figure 12. Screenshot of surrogate designer.

## 8.1 Set inputs and outputs

- Set name of the surrogate model.
- Set the inputs to the surrogate model. These will typically be pose parameters. An example would be: tx,ty,tz,rx,ry,rz
- Set the outputs to the surrogate model. These will typically be loads such as: Fx,Fy,Fz,Tx,Ty,Tz
- Set the inputs and single output to each of the surrogate model stages. Each quantity in a stage input must be in either the surrogate model input or as an output in a previous stage. The stage names are optional.

To the right, there is a list of possible inputs and outputs. When specifying inputs and outputs make sure to have the quantities of interest separated by a comma with no spaces, or any other characters.

*New:* In version 1, the net loads applied on the moving body can also be used as inputs or outputs. To use these, just add “MB” to the load name, e.g., FxMB, FyMB etc. These loads are expressed in the global coordinate system (also the fixed coordinate system) but applied to the origin/COM of the moving body mesh.

## 8.2 Set neural network training criteria

Set the stopping criteria for the neural network training. There are tooltips available for help with the formatting. Simply hover the cursor on the field of interest.

Specify the ANN architecture, validation ratio, and parallelization option. Next, click on “DLL Destination Folder” to specify to which folder should the DLLs generated by Matlab will be copied.

Make sure to save the settings as a stageMap XML object for future reference.

## 8.3 Write formatted data

Once all properties have been specified in the designer, and having generated both a set of training data and a set of testing data, we transcribe the sample points to tables using the “Write” button. Once both sets of sample points have been transcribed into .train and .test files, the “Launch ANN training” button can be pressed. This will lead to a dialog asking for the folder where the GenerateANNs.m file is located. This folder must be the same where the .test and .train files are located. After selecting the folder, Matlab will be launched in automation mode. Training can be monitored in the Matlab command window. Once training is over, the testing and training errors for the neural network corresponding to each stage will be displayed.

**IMPORTANT NOTE:** There should be NO BLANK SPACES in the names of the folders or files. Use underscores if needed.

Check for the DLLs in the destination folder. The DLLs together with the settings previously saved settings file (stageMap XML file) make up the surrogate contact model.

## 9 Surrogate Model Tester



The tester allows users to compute the discrepancy between the testing data and the surrogate contact model. The root-mean-squared errors and maximum absolute errors may be calculated on a per-domain basis.

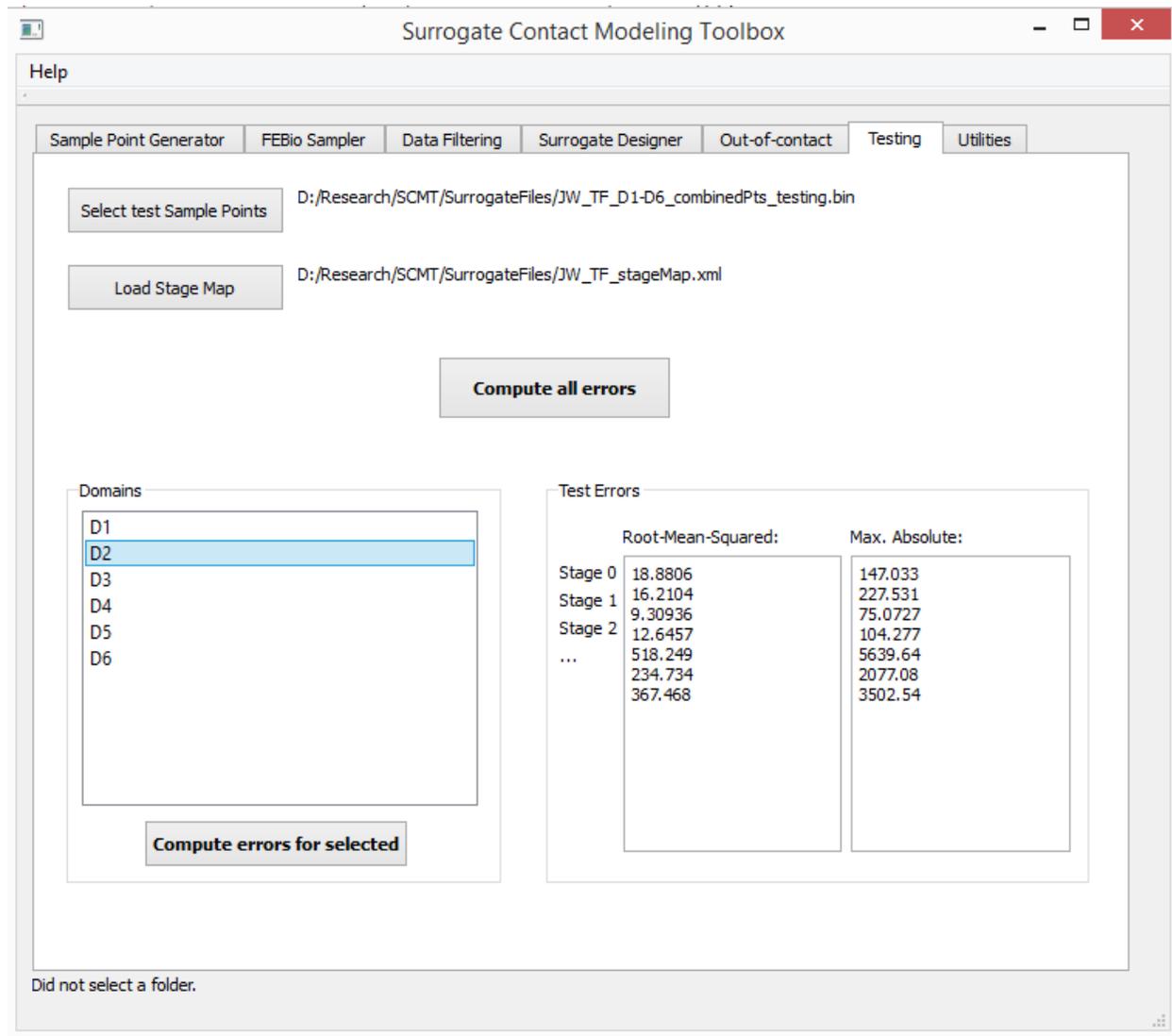


Figure 13. Screenshot of tester tool.

## 9.1 Load files

- Load the sample points for testing as .bin
- Load the stageMap (or surrogate model) as .xml. This is the same file saved as settings in the Surrogate Designer tool. Make sure that both the XML file and the DLLs are in the same folder as the SCMT executable.

## 9.2 Compute errors

Errors may now be computed either by clicking on the “Compute all errors” button which calculates the errors in all domains, or by selecting a domain from the list and clicking on the “Compute errors for selected.”

# 10 Utilities

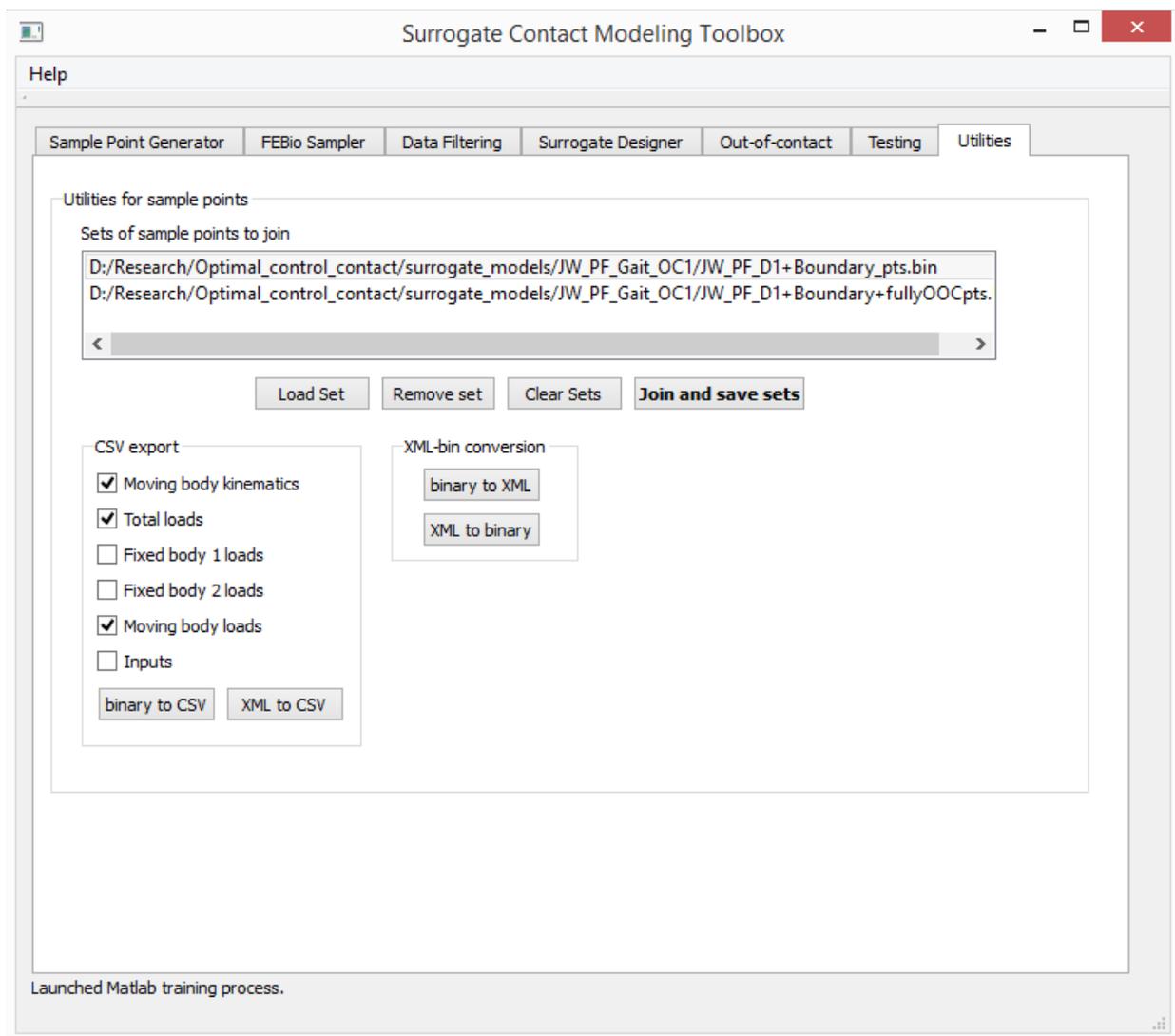


Figure 14. Screenshot of utilities tool.

### **10.1 Joining sets of sample points**

To join sets of sample points (in .bin format) click on the “Load Set” button. Once all the paths for the files being added are listed, click of “Join and save set.”

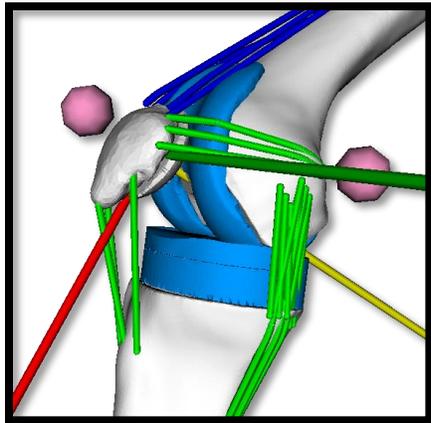
### **10.2 Exporting files as comma separated values (CSV)**

Both XML and BIN files containing sample points may be written to CSV. This utility is mainly for exploring the data in a third party program such as excel or Matlab. Check the boxes for the quantities to be transcribed.

### **10.3 Conversion between XML and BIN**

Sample points may be converted from XML to binary and vice-versa using this tool. Click on the appropriate button to load and save the files. Be careful when converting binary to XML as the memory requirements of XML files can be exceedingly large for large data sets.

# 11 Plugin for OpenSim

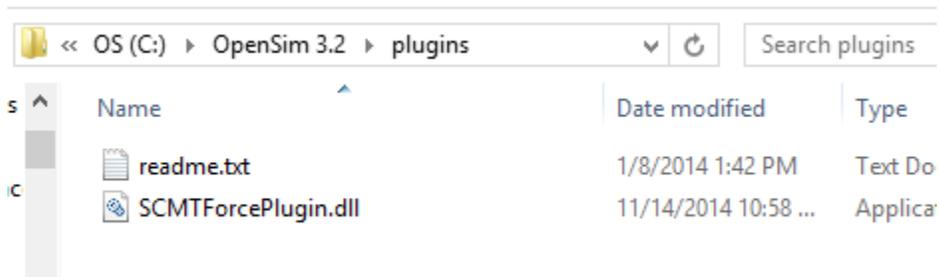


Included with SCMT is a plugin for OpenSim that allows surrogate contact models to be used as Force model components. This lets users to model and simulate musculoskeletal systems with embedded surrogate contact models.

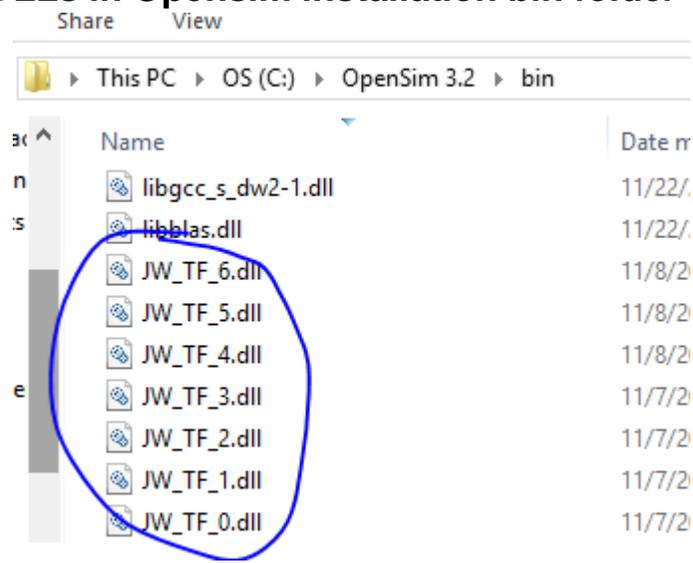
## 11.1 What does it do

The OpenSim plugin takes the relative poses and orientations between two bodies, and feeds them into the surrogate contact model. The computed contact forces are then applied to the bodies automatically during simulation. For this to work, the surrogate model outputs must be all the 6 spatial loads expressed in the fixed body coordinate system. Alternatively, if there are two fixed bodies, the outputs can also be the loads that add up to the net load. e.g., instead of Fx, you could have Fx1 and Fx2 as outputs, because they add up to Fx.

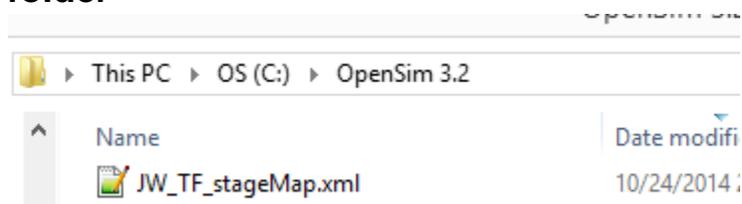
## 11.2 Place the plugin file in the plugin folder



### 11.3 Place the DLLs in Opensim installation bin folder



### 11.4 Place the surrogate/stageMap XML file in the Opensim installation folder



### 11.5 Setup your force component in the osim file

Once the plugin has been loaded within the GUI, check the XML browser for "SCMTForce." You can then copy that into the osim file under ForceSet → objects.

```

<!--Forces in the model.-->
<ForceSet>
  <objects>
    <SCMTForce name="JW_TF_SCM">
      <!--Flag indicating whether the force is disabled or :
      <isDisabled>false</isDisabled>
      <!--Name of fixed body.-->
      <body_1>tibial_tray</body_1>
      <!--Name of moving body.-->
      <body_2>femoral_component</body_2>
      <!--Location of contact coordinate system origin in b
      <location_body_1>0 0 0</location_body_1>
      <!--Orientation of contact coordinate system in body
      <orientation_body_1>0 0 0</orientation_body_1>
      <!--Location of contact coordinate system origin in b
      <location_body_2>0 0 0</location_body_2>
      <!--Orientation of contact coordinate system in body
      <orientation_body_2>0 0 0</orientation_body_2>
      <!--Name of surrogate contact model to load.-->
      <stageMapFile>JW_TF_stageMap.xml</stageMapFile>
      <!--Scale factor: unit of length of OpenSim divided b
      <scaleFactor>1000</scaleFactor>
      <!--Switch to report the loads directly calculated fr
      <reportContactMeasures>true</reportContactMeasures>
      <!--Switch to report the loads applied to parent and
      <reportBodyLoads>true</reportBodyLoads>
    </SCMTForce>
  </objects>
</ForceSet>

```

Figure 15. Place the SCMTForce in the osim file under the ForceSet objects.

## 12 Surrogate Model Use in C++

Include libraries and use std namespace;

```
#include "LoadSaveFunctions.h"  
using namespace std;
```

Snippet of code to load model:

```
// create default surrogate model  
StageMap surrogate;  
// load the xml file into a StageMap (surrogate) object  
xml::loadStageMap(surrogate, "testSurrogate.xml");  
//dynamically load the neural networks  
surrogate.LoadANNs();
```

Snippet of code to run model:

```
// run surrogate model with vector<double> inputs  
vector<double> outputs = surrogate.Run(inputs);
```

## 13 Surrogate Model Use in Matlab

A matlab file called “compileSurrogateMex.m” is provided to help with the compilation settings. After creating a mex file you may use it in the following manner:

If you have a single surrogate model to evaluate named “surrogate1.xml” you would first load it:

```
SCMT_mex('surrogate1.xml');
```

Then use it for one evaluation for six inputs, `inputs = [in1 in2 in3 in4 in5 in6]`:

```
outputs = SCMT_mex(inputs);
```

If evaluating multiple outputs simultaneously, you can use a matrix as an input:

```
inputMatrix = ([in11 in12 in13 in14 in15 in16; in21 in22 in23 in24 in25 in26];
```

```
outputMatrix = SCMT_mex(inputMatrix);
```

If evaluating multiple surrogate models at once, load them both in one call:

```
SCMT_mex('surrogate1.xml', 'surrogate2.xml');
```

Get all outputs simultaneously:

```
[outputMatrix1, outputMatrix2] = SCMT_mex(inputMatrix1, inputsMatrix2);
```