

Instrumental Pedal Tutorial

Overview

This tutorial outlines the process to build a model for a bike pedal and crank and how to assign force data from analog signals. This tutorial assumes your human model is already built. Follow this tutorial to create a model before continuing [Building 6DOF Model](#). This tutorial will have two parts, the first part will show how to create the model for the bike pedal and crank, and the second will show the force data processing.

The following c3d files can be downloaded as a starting point:

Bike Model c3d Starter

Full completed CMZ can be downloaded here: [Bike Model CMZ Complete](#)

Segments

Landmarks

Muscles

Subject Data / Metrics

Landmarks

Functional

Digitizing

Landmark Name	Segment	ML	AP
LHIP			
RHIP			
RHJC	LAB	0.0	0.0
LHJC	LAB	0.0	0.0
RTROCH_CORR			
LTROCH_CORR			
LAB_ORIGIN	LAB	0.0	0.0
LAB_X	LAB	0.1	0.0
LAB_Y	LAB	0.0	0.1
LAB_Z	LAB	0.0	0.0
MLPED1			
B_LAB_Z	LAB	0.0	0.0
B_LAB_Y	LAB	0.0	0.1
B_LAB_X	LAB	0.1	0.0
B_LAB_Z_R	LAB	0.0	0.0
B_LAB_Y_R	LAB	0.0	0.3
B_LAB_X_R	LAB	0.1	0.0
MRPED1			
CRANK 7	LAB	0.0	0.0

Add New Landmark

Copy Selected Landmark

Modify Selected Landmark

Copy & Reflect Landmark

Delete Selected Landmark

Rename/Prefix Landmark

Edit Landmark Properties

Build Model

Segments | Landmarks | Muscles | Subject Data / Metrics

Landmarks | Functional | Digitizing

Name:

Define Orientation Using:

Starting Point (Reference)

☐ Targets and/or Landmark:
Ending Point (On a line)
*Lateral object (On a plane)
*Project From (Projection onto a line or plane)
* = Optional

☒ Existing Coordinate System

Landmark Offset from Start Point (Reference) or Segment Origin

☐ Offset to Existing Calibration Target or Landmark

☒ Offset Using the Following ML/AP/AXIAL Offsets
X
Y
Z
☐ Offset by Percent (1.0 = 100%) (Meters when not checked)

☐ Calibration Only Landmark (Not generated for assigned motion file(s))

Apply

Build Model

Close Tab

Undo Changes

Copy to Clipboard

Paste from Clipboard

To build a new segment select the **Segment** tab input your desired segment name, ensure the segment type is **Visual 3D**, select **Create**, input the definitions for the segments shown below, and select **Apply**.

Segments
Landmarks
Muscles
Subject Data / Metrics

Segments
Segment Properties
IK Constraints
Example

Segment Name
Example
☐ Kinematic Only

Segment Type
Visual 3D
Create

Double-Click Segment to View/Edit

Segment Name	Segment Type	Calibrated	Kinetic
LAB	Visual 3D	N/A	YES
Right Thigh	Visual 3D	YES	YES
Right Shank	Visual 3D	YES	YES
Right Foot	Visual 3D	YES	YES
Left Thigh	Visual 3D	YES	YES
Left Shank	Visual 3D	YES	YES
Left Foot	Visual 3D	YES	YES
RtFoot	Visual 3D	YES	NO
LftFoot	Visual 3D	YES	NO
Thorax/Ab	Visual 3D	YES	YES
Left Crank	Visual 3D	YES	NO
Bike_Lab	Visual 3D	YES	NO
Right Crank	Visual 3D	YES	NO
R_Bike_Lab	Visual 3D	YES	NO
Left Pedal	Visual 3D	YES	NO
Right Pedal	Visual 3D	YES	NO

Algorithm for Computing Pose
Inverse Kinematics

Apply
Build Model

View Segment
Duplicate Selected Segment

Delete Selected Segment
Copy & Reflect Selected Segment

Rename/Prefix Selected Segment

Segments
Landmarks
Muscles
Subject Data / Metrics

Segments
Segment Properties
IK Constraints
Example
Example_2

Define Proximal Joint and Radius

Lateral
Joint Center
Medial

None
None
None

Radius (Meters)

Define Distal Joint and Radius

Lateral
Joint Center
Medial

None
None
None

Radius (Meters)

Extra Signal to Define Orientation (if needed)

Location
None

Select Tracking Signals

☐ Use Calibration Signals for Tracking

☐ LHIP
☐ LTROCH_CORR
☐ MLPED1
☐

☐ RHIP
☐ LAB_ORIGIN
☐ B_LAB_Z
☐

☐ RHJC
☐ LAB_X
☐ B_LAB_Y
☐

☐ LHJC
☐ LAB_Y
☐ B_LAB_X
☐

☐ RTROCH_CORR
☐ LAB_Z
☐ B_LAB_Z_R
☐

Depth (Meters):

6DOF Cutoff Frequency
0.0

Close Tab
Guess Properties
Apply
Build Model

The first set of **landmarks** define a virtual position for the cranks

Landmark	Starting Point	Existing Coordinate System	ML Offset	AP Offset	AXIAL Offset
VRCRKAXIS	RCRKAXIS	LAB	-0.0045	0	0
VLCRKAXIS	LCRKAXIS	LAB	0.0045	0	0

The following set of **landmarks** define virtual positions from the pedal markers in the LAB

Landmark	Starting Point	Existing Coordinate System	ML Offset	AP Offset	AXIAL Offset
VRPED1_X	RPED1	LAB	-0.3	0	0
VRPED1	RPED1	LAB	-0.6	0	0
VRPED2	RPED2	LAB	-0.3	0	0
VRPED3	RPED3	LAB	-0.5	0	0
VLPED1_X	LPED1	LAB	0.3	0	0
VLPED1	LPED1	LAB	0.6	0	0
VLPED2	LPED2	LAB	0.3	0	0
VLPED3	LPED3	LAB	0.5	0	0

VRPED3_0	RPED3	LAB	-0.2	0	0
VLPED3_0	LPED1	LAB	0.2	0	0
VRPED2_X1	RPED2	LAB	-0.07	0	0
Landmark	Starting Point	Ending Point	Project From		
MRPED1	RPED1	VRPED1_X	VRCRKAXIS		
MLPED1	LPED1	VLPED1_X	VLCRKAXIS		

The following set of **landmarks** are for the Bike Lab Coordinate systems.

Landmark	Starting Point	Existing Coordinate System	ML Offset	AP Offset	AXIAL Offset
B_LAB_Z	LCRKAXIS	LAB	0	0	0.1
B_LAB_Y	LCRKAXIS	LAB	0	0.1	0
B_LAB_X	LCRKAXIS	LAB	0.1	0	0
B_LAB_Z_R	RCRKAXIS	LAB	0	0	0.1
B_LAB_Z_R	RCRKAXIS	LAB	0	0.3	0
B_LAB_Z_R	RCRKAXIS	LAB	0.1	0	0

We can now define our first **segments**.

Segment	Proximal Joint	Distal Joint	Extra Signal	Tracking Signals
Right Pedal 1	Lateral: VRPED3_0 Radius:0.07	lateral: RPED2 Radius:0.07	Medial: RPENDANT	MRPED1, RPED2, RPED3
Left Pedal 1	Lateral: VLPED3_0 Radius:0.07	lateral: LPED2 Radius:0.07	Medial: LPENDANT	MLPED1, LPED2, LPED3
Bike_Lab	Joint Center: B_LAB_Z Radius:0.07	Joint center: LCRKAXIS Radius:0.01	Anterior: B_LAB_Y	B_LAB_Z, B_LAB_Y, B_LAB_X, LCRKAXIS
R_Bike_Lab	Joint Center: B_LAB_Z_R Radius:0.07	Joint center: RCRKAXIS Radius:0.01	Anterior: B_LAB_Y_R	B_LAB_Z_R, B_LAB_Y_R, B_LAB_X_R, RCRKAXIS
AXIS	Joint Center: RCRKAXIS Radius:0.1	Joint center: CRANK_Z Radius:0.01	Medial: LCRKAXIS	Check Use Calibration signals for Tracking

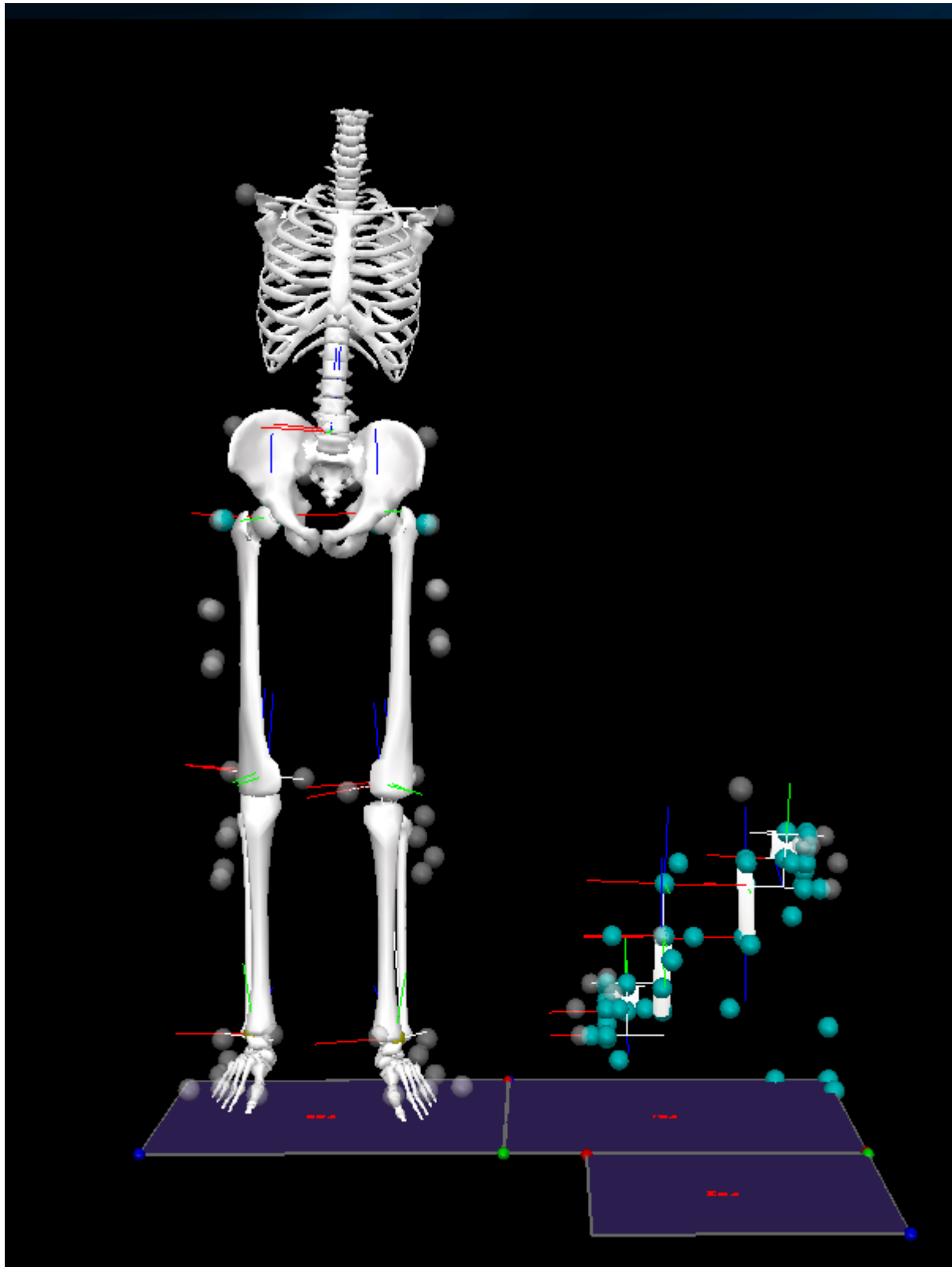
We can now define the next group of **Landmarks** built using the segments created above.

Landmark	Starting Point	Existing Coordinate System	ML Offset	AP Offset	AXIAL Offset
VRPED1_Z	VRPED1	Right Pedal 1	0	0.21	0
VRPED2_Z	VRPED2	Right Pedal 1	0	0.21	0
VRPED3_Z	VRPED3	Right Pedal 1	0	0.21	0
VLPED1_Z	VLPED1	Left Pedal 1	0	0.21	0
VLPED1_Z	VLPED2	Left Pedal 1	0	0.21	0
VLPED1_Z	VLPED3	Left Pedal 1	0	0.21	0
MRPED1_Z	MRPED1	Right Pedal 1	0	0.21	0
MLPED1_Z	MLPED1	Left Pedal 1	0	0.21	0
VRPED1_Z_X	VRPED1_Z	Right Pedal 1	-0.38	0	0
VRPED2_Z_X	VRPED2_Z	Right Pedal 1	-0.38	0	0
VLPED1_Z_X	VLPED1_Z	Left Pedal 1	-0.38	0	0
VLPED2_Z_X	VLPED2_Z	Left Pedal 1	-0.38	0	0

We now have all the required Landmarks to define the final 4 **segments**.

Segment	Proximal Joint	Distal Joint	Extra Signal	Tracking Signals	
Right Pedal	Joint Center: VRPED1_Z_X Radius:0.038	lateral: VRPED2_Z_X Radius:0.038	Lateral: VRPED1	MRPED1_Z, RPED2, RPED3	0.01
Left Pedal	Joint Center: VLPED1_Z_X Radius:0.038	lateral: VLPED2_Z_X Radius:0.038	Lateral: VLPED1	MLPED1_Z, LPED2, LPED3	0.01
Right Crank	Joint Center: RCRKAXIS Radius:0.01	lateral: MRPED1 Radius:0.01	Lateral: RPED1	MRPED1, RCRKAXIS, RPED1	0.01
Left Crank	Joint Center: LCRKAXIS Radius:0.01	lateral: MLPED1 Radius:0.01	Lateral: LPED1	MLPED1, LCRKAXIS, LPED1	0.01

You should now have all the landmarks and segments built for your model and you human and bike model should resemble the following.



Force Processing

Force data from bikes is often imported using analog signals not force plates. Data requires processing to interpret. The following steps will show the command pipeline used for this process. The completed pipeline can also be downloaded here. Not there are values specific to this data such as variable names and lever arms based on pedal size that will need to be changed based on the specific data.

The original force data is very noisy so a low-pass filter is used to smooth the data. The command **Lowpass_Filter** can be under **Signal Filter** press » to add to your pipeline.

```
Lowpass_Filter
/SIGNAL_TYPES= ANALOG
! /SIGNAL_FOLDER=ORIGINAL
! /SIGNAL_NAMES=
! /RESULT_FOLDER=PROCESSED
! /RESULT_SUFFIX=
! /FILTER_CLASS=BUTTERWORTH
/FREQUENCY_CUTOFF=6.0
/NUM_REFLECTED=6
! /NUM_EXTRAPOLATED=0
/TOTAL_BUFFER_SIZE=20
! /NUM_BIDIRECTIONAL_PASSES=1
;
```

Once smoothed the **Evaluate_Expression** command can be used to determine the Force in each pedal before using the **Transform_Data_Coordinate_System** to convert force values to the Lab.

Force

/Right

```
Evaluate_Expression
/EXPRESSION=
VECTOR((-1.0*ANALOG::PROCESSED::Force.Fx_2),((1.0*ANALOG::PROCESSED::Force.Fy_3)+(1.0*ANALOG::PROCESSED::Force.Fy_4)),((1.0*ANALOG::PROCESSED::Force.Fz_3)+(1.0*ANALOG::PROCESSED::Force.Fz_4)))
/SIGNAL_TYPES=ANALOG
! /SIGNAL_FOLDER=PROCESSED
/SIGNAL_NAMES=Force.Fx_2+Force.Fy_3+Force.Fy_4+Force.Fz_3+Force.Fz_4
! /SIGNAL_COMPONENTS=
! /RESULT_TYPES=DERIVED
/RESULT_FOLDERS=FORCE_RIGHT
/RESULT_NAME= FORCE
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
;
```

```
Transform_Data_Coordinate_System
/SIGNAL_TYPES= DERIVED
/SIGNAL_FOLDER=FORCE_RIGHT
/SIGNAL_NAMES= FORCE
! /FROM_SEGMENT_CS= Right Pedal
/TO_SEGMENT_CS= LAB
/RESULT_TYPES= DERIVED
/RESULT_FOLDERS= FORCE_LAB_RIGHT
```

```

!/RESULT_NAME=
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=TRUE
/ORIENTATION_ONLY=TRUE
! /USE_DATA_RATE=FALSE
;

=====
/Left
=====

Evaluate_Expression
/EXPRESSION=
VECTOR((-1.0*ANALOG::PROCESSED::Force.Fx),((1.0*ANALOG::PROCESSED::Force.Fy)
+(1.0*ANALOG::PROCESSED::Force.Fy_2),((1.0*ANALOG::PROCESSED::Force.Fz)+(1.0
*ANALOG::PROCESSED::Force.Fz_2)))
/SIGNAL_TYPES=ANALOG
! /SIGNAL_FOLDER=PROCESSED
/SIGNAL_NAMES=Force.Fx+Force.Fy+Force.Fy_2+Force.Fz+Force.Fz_2
! /SIGNAL_COMPONENTS=
! /RESULT_TYPES=DERIVED
/RESULT_FOLDERS=FORCE_LEFT
/RESULT_NAME= FORCE
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
;

Transform_Data_Coordinate_System
/SIGNAL_TYPES= DERIVED
/SIGNAL_FOLDER=FORCE_LEFT
/SIGNAL_NAMES= FORCE
! /FROM_SEGMENT_CS= Left Pedal
/TO_SEGMENT_CS= LAB
/RESULT_TYPES= DERIVED
/RESULT_FOLDERS= FORCE_LAB_LEFT
!/RESULT_NAME=
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=TRUE
/ORIENTATION_ONLY=TRUE
! /USE_DATA_RATE=FALSE
;

```

In order to change the force assignments the **Center Of Pressure** and **Freemoment** for each pedal are also required. First we must compute the moment, followed by the **COP** and finally the **Freemoment**.

Moment

Note that the decimal coefficients are lever arm measurements that will change based on pedal size.

```

=====
/Right
=====

```

```
Evaluate_Expression
/EXPRESSION=VECTOR((0.02845*DERIVED::FORCE_RIGHT::FORCE::Y),((0.01945*ANALOG
::PROCESSED::Force.Fz_4-0.01945*ANALOG::PROCESSED::Force.Fz_3)-0.02845*ANALO
G::PROCESSED::Force.FX_2),((0.01945*ANALOG::PROCESSED::Force.Fy_4-0.01945*AN
ALOG::PROCESSED::Force.Fy_3)))
/RESULT_NAME=MOMENT
/RESULT_TYPE=DERIVED
/RESULT_FOLDER=FORCE_RIGHT
;
```

=====

/Left

=====

```
Evaluate_Expression
/EXPRESSION=VECTOR((0.02845*DERIVED::FORCE_LEFT::FORCE::Y),
((-0.01945*ANALOG::PROCESSED::Force.Fz +
0.01945*ANALOG::PROCESSED::Force.Fz_2) -
0.02845*ANALOG::PROCESSED::Force.FX),((0.01945*ANALOG::PROCESSED::Force.Fy
- 0.01945*ANALOG::PROCESSED::Force.Fy_2)))
/RESULT_NAME=MOMENT
/RESULT_TYPE=DERIVED
/RESULT_FOLDER=FORCE_LEFT
;
```

Center Of Pressure

=====

/Right

=====

```
Evaluate_Expression
/EXPRESSION=VECTOR((-1 * DERIVED::FORCE_RIGHT::MOMENT::Y /
DERIVED::FORCE_RIGHT::FORCE::Z), (DERIVED::FORCE_RIGHT::MOMENT::X /
DERIVED::FORCE_RIGHT::FORCE::Z), (0.0*analog::processed::Force.Fx_2))
/RESULT_NAME=COP
/RESULT_TYPE=DERIVED
/RESULT_FOLDER=FORCE_RIGHT
;
```

```
Transform_Data_Coordinate_System
/SIGNAL_TYPES=DERIVED
/SIGNAL_NAMES=COP
/SIGNAL_FOLDER=FORCE_RIGHT
/FROM_SEGMENT_CS=Right Pedal
/TO_SEGMENT_CS=Lab
/RESULT_TYPE=Derived
/RESULT_FOLDER=FORCE_LAB_RIGHT
```

```

/RESULT_SUFFIX=
/ORIENTATION_ONLY=FALSE
! /USE_DATA_RATE=FALSE
;

=====
/Left
=====

Evaluate_Expression
/EXPRESSION=VECTOR((-1*DERIVED::FORCE_LEFT::MOMENT::Y /
DERIVED::FORCE_LEFT::FORCE::Z), (DERIVED::FORCE_LEFT::MOMENT::X /
DERIVED::FORCE_LEFT::FORCE::Z), (0.0*analog::processed::Force.Fx))
/RESULT_NAME=COP
/RESULT_TYPE=DERIVED
/RESULT_FOLDER=FORCE_LEFT
;

Transform_Data_Coordinate_System
/SIGNAL_TYPES=DERIVED
/SIGNAL_NAMES=COP
/SIGNAL_FOLDER=FORCE_LEFT
/FROM_SEGMENT_CS=Left Pedal
/TO_SEGMENT_CS=Lab
/RESULT_TYPE=Derived
/RESULT_FOLDER=FORCE_LAB_LEFT
/RESULT_SUFFIX=
/ORIENTATION_ONLY=FALSE
! /USE_DATA_RATE=FALSE
;

```

Freemoment

```

=====
/Right
=====

Evaluate_Expression
/EXPRESSION=VECTOR((0.0*analog::processed::Force.Fx_2), (0.0*analog::processe
d::Force.Fy_3), (Derived::FORCE_RIGHT::MOMENT::Z -
Derived::FORCE_RIGHT::FORCE::Y * Derived::FORCE_RIGHT::COP::X +
Derived::FORCE_RIGHT::FORCE::X * Derived::FORCE_RIGHT::COP::Y))
/RESULT_NAME=FREEMOMENT
/RESULT_TYPE=DERIVED
/RESULT_FOLDER=FORCE_RIGHT
;

Transform_Data_Coordinate_System
/SIGNAL_TYPES=DERIVED
/SIGNAL_NAMES=FREEMOMENT

```

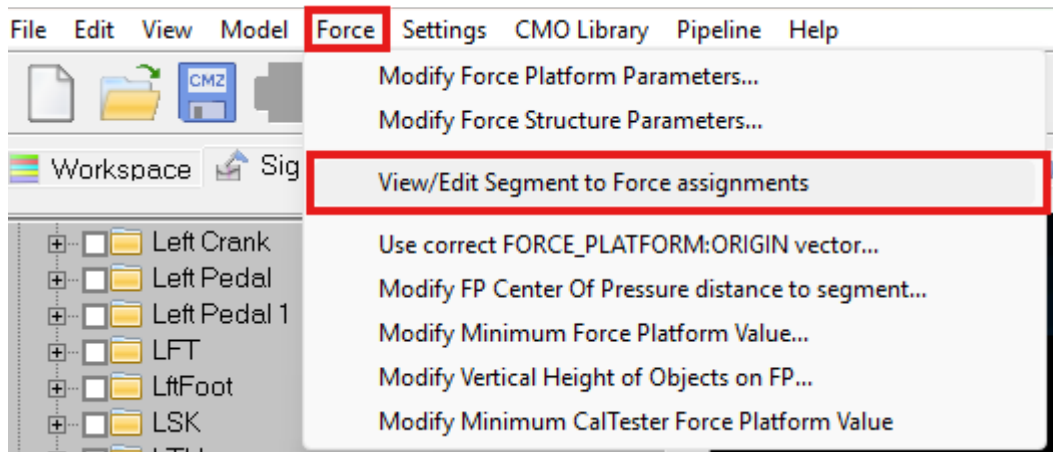
```
/SIGNAL_FOLDER=FORCE_RIGHT
/FROM_SEGMENT_CS=Right Pedal
/TO_SEGMENT_CS=Lab
/RESULT_TYPE=Derived
/RESULT_FOLDER=FORCE_LAB_RIGHT
/RESULT_SUFFIX=
/ORIENTATION_ONLY=TRUE
! /USE_DATA_RATE=FALSE
;

=====
/Left
=====
Evaluate_Expression
/EXPRESSION=VECTOR((0.0*analog::processed::Force.Fx),(0.0*analog::processed::Force.Fy),(Derived::FORCE_LEFT::MOMENT::Z - Derived::FORCE_LEFT::FORCE::Y *
Derived::FORCE_LEFT::COP::X + Derived::FORCE_LEFT::FORCE::X *
Derived::FORCE_LEFT::COP::Y))
/RESULT_NAME=FREEMOMENT
/RESULT_TYPE=DERIVED
/RESULT_FOLDER=FORCE_LEFT
;

Transform_Data_Coordinate_System
/SIGNAL_TYPES=DERIVED
/SIGNAL_NAMES=FREEMOMENT
/SIGNAL_FOLDER=FORCE_LEFT
/FROM_SEGMENT_CS=Left Pedal
/TO_SEGMENT_CS=Lab
/RESULT_TYPE=Derived
/RESULT_FOLDER=FORCE_LAB_LEFT
/RESULT_SUFFIX=
/ORIENTATION_ONLY=TRUE
! /USE_DATA_RATE=FALSE
;
```

Modifying Force Assignments

With all values calculated the force assignments for the Right and Left feet can be set. Select **Add/Modify Segment Assignment To Force** under the **Force** tab in the upper toolbar.



Select **Add Segment Assignment** and enter the following values. Change the end frame to the last frame in your specific trial.

The screenshot shows the 'Add/Modify Segment Assignment to Force' dialog box. The dialog has a title bar with a close button. It contains three columns for signal assignment: 'Force Signal', 'Center of Pressure Signal', and 'Freemoment Signal'. Each column has a 'Type' dropdown (all set to 'DERIVED'), a 'Folder' dropdown (all set to 'FORCE_LAB_RIGHT'), and a 'Name' dropdown (set to 'FORCE', 'COP', and 'FREEMOMENT' respectively). Below these is a checkbox 'Automatically assign segment and frames' which is unchecked. A 'Segment' dropdown is set to 'Right Foot'. There are input fields for 'Start Frame' (1), 'End Frame' (309), and 'Inside Frame'. At the bottom, there are two lists: 'Defined Events' (containing LOFF, LON, ROFF, RON) and 'Event Sequence' (empty). Between these lists are '>>' and '<<' buttons. Below the 'Event Sequence' list is another set of '>>' and '<<' buttons and a text label 'Exclude sequences containing these events'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Add/Modify Segment Assignment to Force

Force Signal

Type

DERIVED

Folder

FORCE_LAB_LEFT

Name

FORCE

Center of Pressure Signal

Type

DERIVED

Folder

FORCE_LAB_LEFT

Name

COP

Freemoment Signal

Type

DERIVED

Folder

FORCE_LAB_LEFT

Name

FREEMOMENT

☐ Automatically assign segment and frames

Segment

Left Foot

Start Frame

1

End Frame

309

Inside Frame

Defined Events

LOFF
LON
ROFF
RON

>>
<<

Event Sequence

Exclude sequences containing these events

OK

Cancel

From:
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:tutorials:modeling:bicycle_model&rev=1781544227

Last update: 2026/06/15 17:23

