

Loading a TKC Assembly and Running a Design Study

Table of Contents

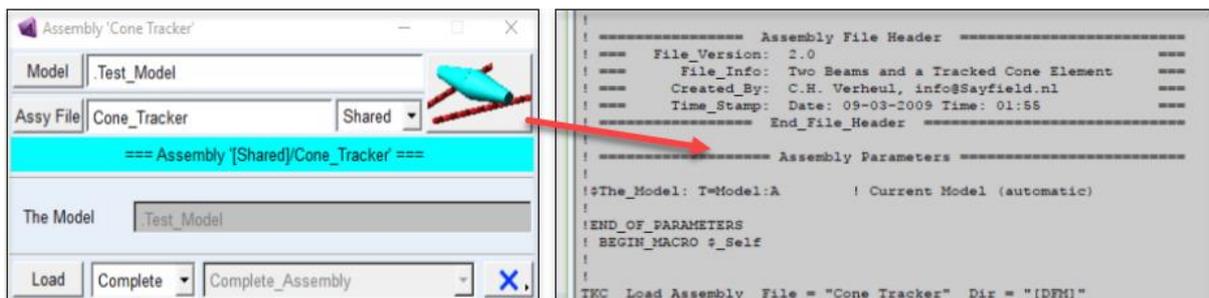
- 1: Introduction
- 2: Opening the Assembly Loader Dialog
- 3: Loading the *Cone_Tracker* Assembly
- 4: Running a first design Run
- 5: Change model parameters and perform new Run
- 6: Comparing Animations of the Runs

1: Introduction

This TKC tutorial will describe the use of the TKC Assembly loader to load an Assembly, perform simulations, change design variables and compare the results.

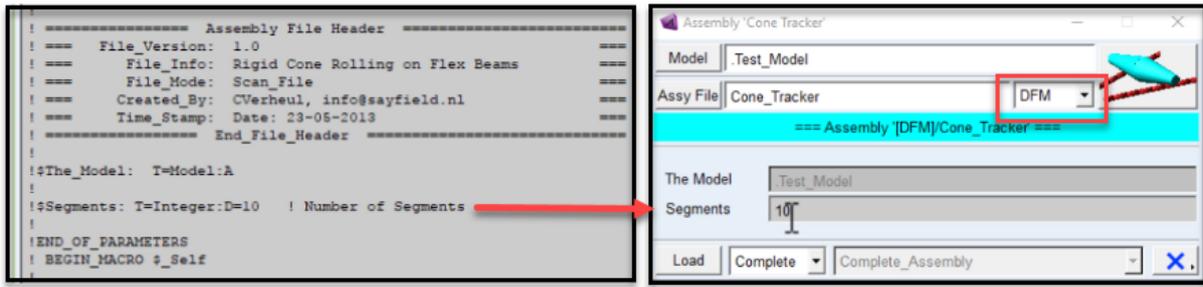
2: Opening the Assembly Loader Dialog

Opening the Assembly loader is done by clicking on the Assembly button on the main dialog, the Assembly loader dialog is a generic dialog that can access Assemblies from TKC libraries and Toolkits. Assemblies in the *Shared* library of TKC typically load Assembly files from the original Toolkits. The *Cone_Tracker* file shown in the image below loads the *Cone_Tracker* Assembly from the *DFM* toolkit. The right side of the image shows the code of the *Shared Cone_Tracker* assembly macro file.



The Toolkit Name *DFM* (for *Discrete Flexibility Models*) stores the original *Cone_Tracker* Assembly. To check the code of an Assembly, the **.mac_assy* file can be opened in a text editor or by clicking on the *Assy_File* button on the Assembly Loader dialog. The *ideal* Assembly file is a documented and readable file not bigger than two or three pages. The bitmap shown above illustrates that this Assembly defines two hollow-round discrete red flexible struts and a cyan two-sided cone body rolling over the struts.

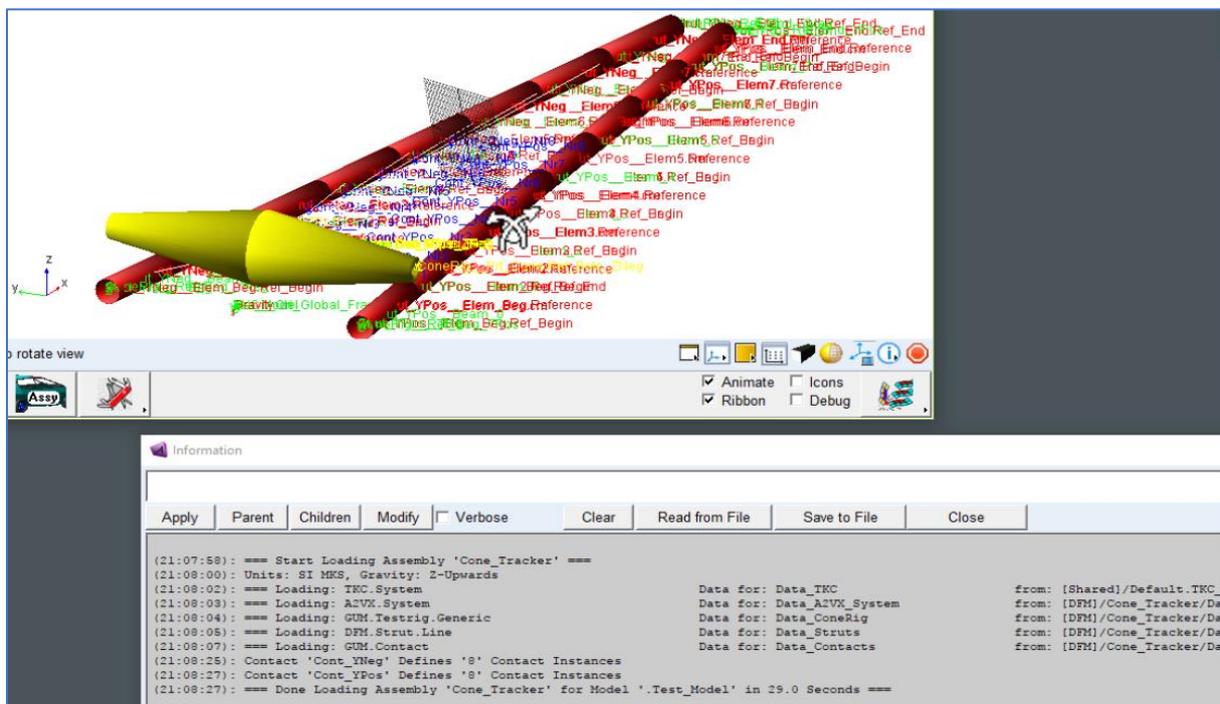
In TKC, parameters of ADAMS/View macro files automatically define input fields on the generic macro dialog defined. For the *DFM Cone_Tracker* Assembly shown below, a *Segments* field is defined in the Assembly dialog with a default value of 10. This value represents the number of discrete segments of the two red tracks. By changing this field value, changes to model topology, like number of parts and contacts, are automatically performed when loading the assembly.



Note: In the *Shared Cone_Tracker* shown above, no *Segments* value is passed in the Load_Assembly command. Thus, models created from this Assembly will always have the default Segments=10.

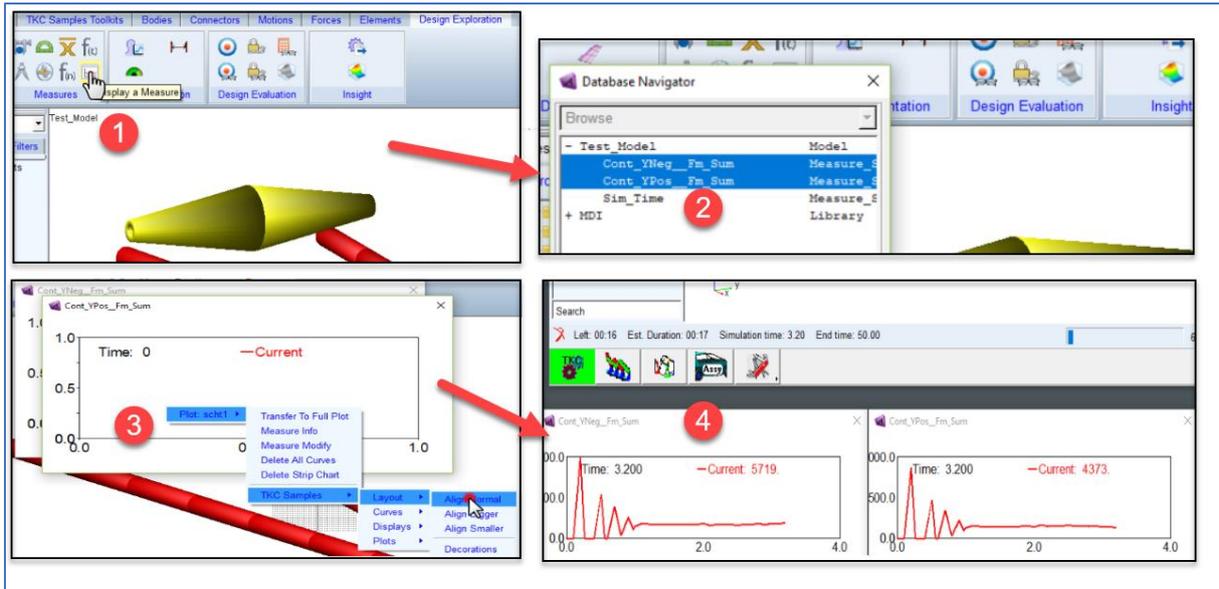
3: Loading the Cone_Tracker Assembly

While loading Assemblies, TKC shows info and warnings on the loading process and displays the time required to generate the model structure. We now have a model build-up from a TKC Assembly file.

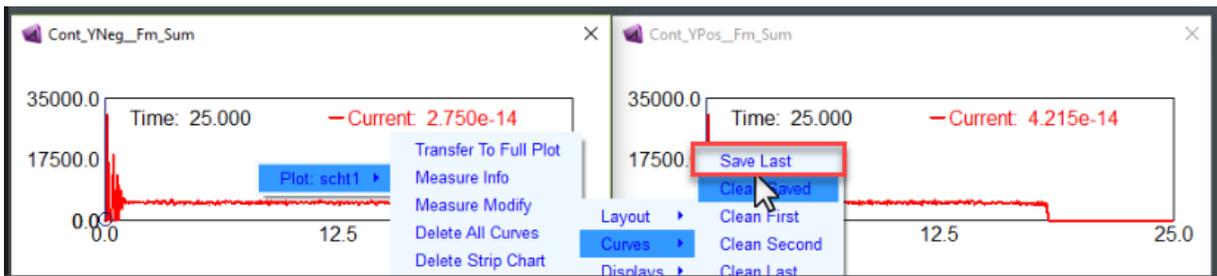


4: Running a first design Run

Next, a first simulation will be performed after which a model parameter is changed to run the model again and compare the results. Comparing the results will be done by evaluating signal measures, so we will display show some relevant measures. Showing measures is done by clicking the display measure button (1) and selecting the desired measures (2). Strut contact force measures are selected, and the displays are aligned using the TKC measures pop-up dialog (3). Picture (4) shows the measure displays with interactive simulation results plotted to the simulation display.



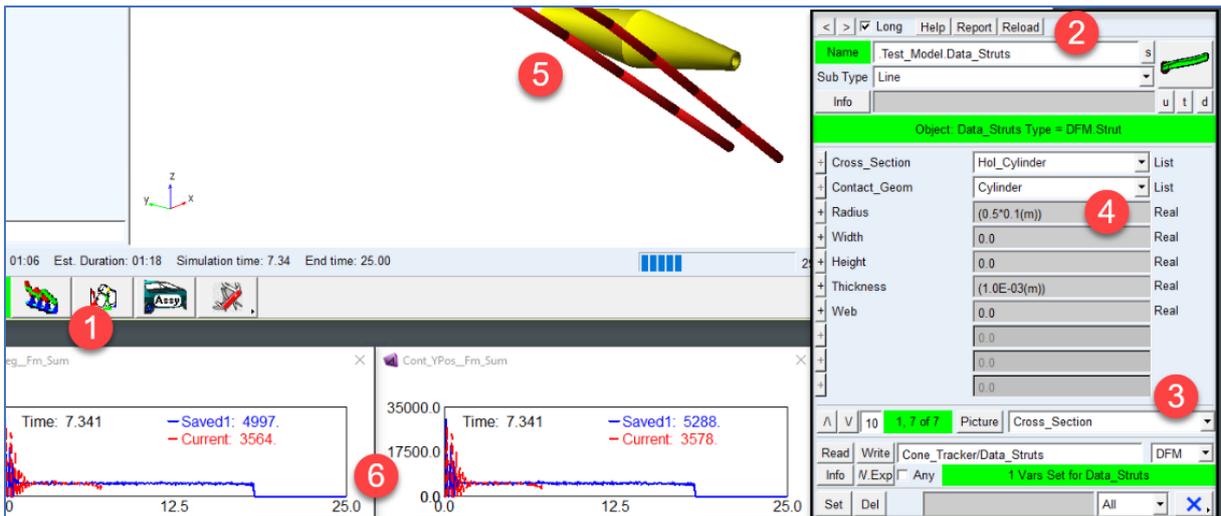
The results of the first run will be stored as *Run_1* and the curves of the measures will be saved to visually compare to the remaining runs of the design study. For this, we open the measures pop-up dialog in any measure display shown and select the *-Curves*, *-Save Last* button to save the last curves for all measure displays.



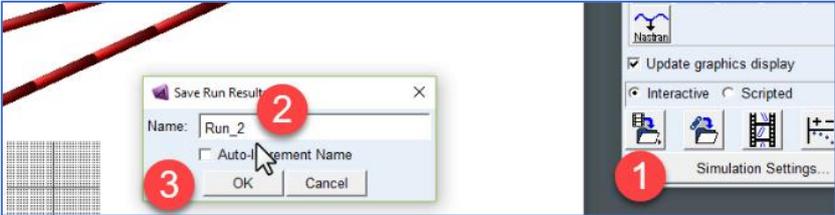
5: Change model parameters and perform new Run

A *typical* TKC method to change the strut radii is demonstrated below by: (1) clicking the parameter set selector to open the parameter dialog for the two struts, (2) setting the *group* selection list to *Cross_Section* (3) to show only a selection of 7 of all design variables in *.Test_Model.Data_Struts* and, (4) setting the struts *Radius* to the expression $(0.5*0.1(m))$. This expression allows users to change variables and still show the original values.

Note that the full name of the struts radius design variable is *.Test_Model.Data_Struts.Radius*.



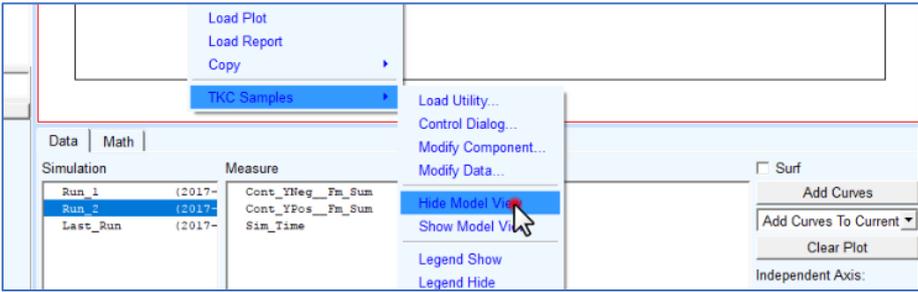
The animation of the second run shows the changed motion of the model (5), and the curves of the response of *Run_2* are added as *Current* curves to the measure displays (6).



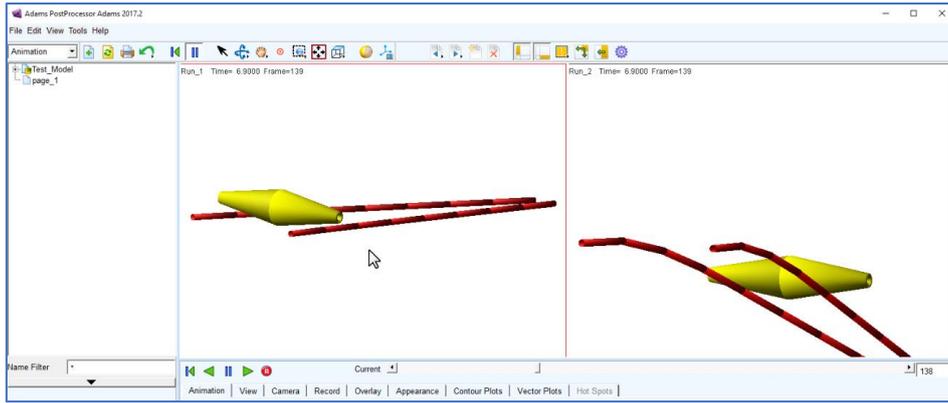
To store the results of the second run: (1) open the *Save Run Results* dialog, (2) change the run Name in the dialog and (3) save the results of the second run as *Run_2*.

6: Comparing Animations of the Runs

The animations of the two runs are compared in the ADAMS/View Post Processor (ADAMS/PPT) dialog. The image below shows that TKC also provides a dedicated ADAMS/PPT pop-up menu.



As measure displays in the model view slow down animations in ADAMS/PPT, the pop-up menu is used here to close the modelling view. Other typical features of the PPT pop-up menu are to open dialogs, manipulate plots legends and other plot properties in ADAMS/PPT. Animating the results of *Run_1* and *Run_2* side-by-side shows the amount of softening due to the change in radius. Note that the strut radii in the two animations seems identical which is caused by the fact that only one model is used to animate both simulations.



To show the actual strut radius in each animation, unique models must be used to animate *Run_1* and *Run_2*. This is standard ADAMS behaviour, and can be prevented by adjusting the radius in a copy of the *Cone_Tracker* model in the ADAMS session. An animation of the results can be exported to file by clicking on the red **Record** button in the ADAMS/PPT environment.