Insert a Force Vector

base model: models/RevoluteJoint.mdl

final model: models/ForceVector.mdl

Explanations

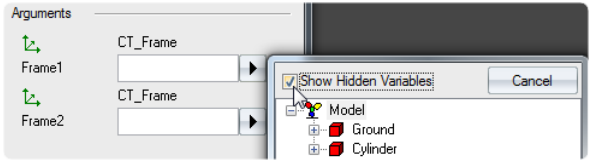
- Similar to joints a force in alaska is an interactor and connects two bodies via two Frames
- If the force should model an external force, `Ground.BFR` is used as one of the two Frames
- Most force-elements apply the defined force in the second argument of the force template `"Frame2"`
- In this tutorial we model a vector-valued force, which is constant in the local Frame of application
- We use the modeled pendulum in `RevoluteJoint.mdl` and want to apply a force at the lower end of the pendulum

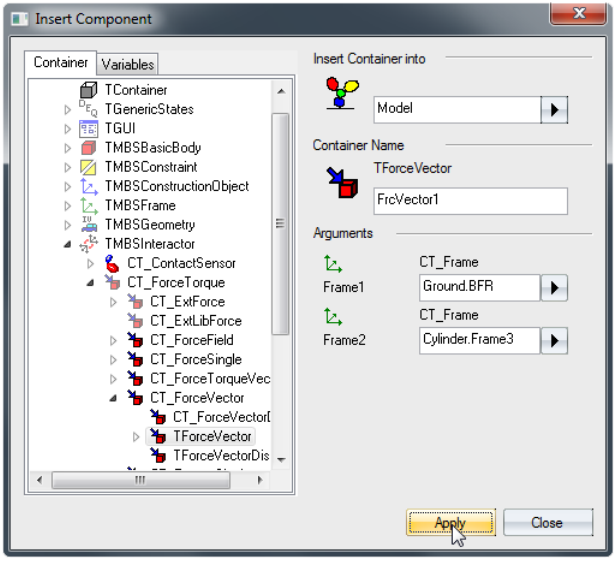
Insert a ForceVector

Open the base model `RevoluteJoint.mdl` in alaska/ModellerStudio

1. Insert a **`TFrame`** to the `Cylinder` and name it `"Frame3"`
2. Move `Frame3` to the lower end of the `Cylinder` by setting its position variable *`Pos`* to $\{0, 0, -0.5\}$
3. Insert one instance of the template **`TForceVector`** from location *`TMBSInteractor`* $\rightarrow$ *`CT_ForceTorque`* $\rightarrow$ *`CT_ForceVector`* $\rightarrow$ *`TForceVector`* to the `"Model"`
4. Name it `"FrcVector1"` and specify the arguments
   1. Frame1: `Ground.BFR`
   2. Frame2: `Cylinder.Frame3`

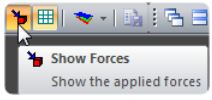
- To be able to find the BFR of a body, check the Show Hidden Variables checkbox




Insert Component
Container
Variables
TContainer
TGenericStates
TGUI
TMBSBasicBody
TMBSConstraint
TMBSConstructionObject
TMBSFrame
TMBSGeometry
TMBSInteractor
CT_ContactSensor
CT_ForceTorque
CT_ExtForce
CT_ExtLibForce
CT_ForceField
CT_ForceSingle
CT_ForceTorqueVec
CT_ForceVector
CT_ForceVector(
TForceVector
TForceVectorDis
Insert Container into
Model
Container Name
TForceVector
FrcVector1
Arguments
CT_Frame
Frame1
Ground.BFR
CT_Frame
Frame2
Cylinder.Frame3
Apply
Close

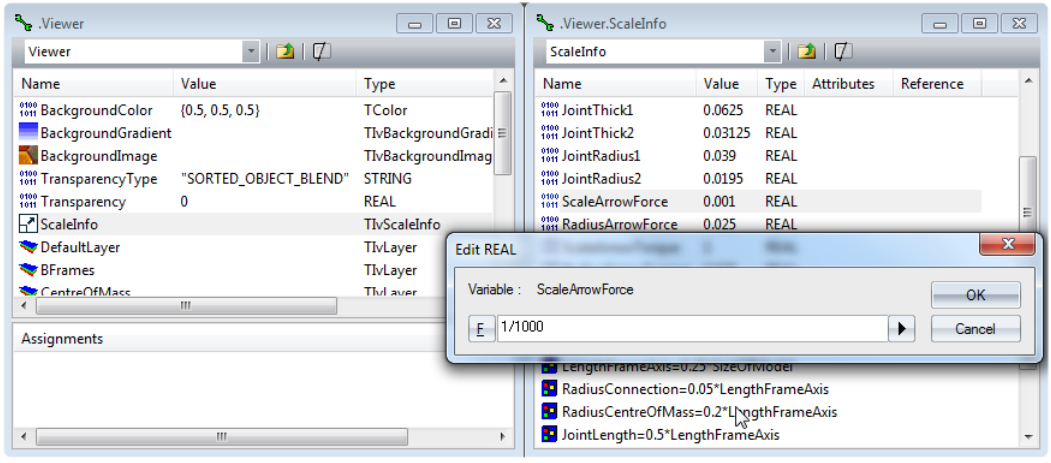
Value of applied force

1. Double-click on `FrcVector1` to open its "`Component View`" and double-click on the variable *`Force`* to change the value of the applied force
2. We define a constant force of 650 N in the 2-axis of `Frame3` by setting the value to $\{0, 650, 0\}$
3. If the `Viewer` is opened hardly anything can be seen, as the graphic visualization of the force is too large
4. For a better visualization you can deactivate the visualization of all Force-elements in the `Viewer` using the symbol



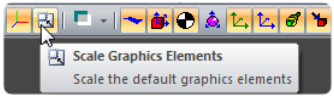
Visualization of the force

5. Alternatively all visualizations within the `Viewer` can be scaled
6. To scale Force-arrows open the "`Component View`" of the `Viewer`
7. Note: A double-click on the `Viewer` will always open the so-called "`Special View`", to open the "`Component View`" you can:
   1. Select the `Viewer` within the "`Model Tree`" and press F4
   2. Or right-click the `Viewer` and use the menu entry Open Component View
8. The Viewer contains the container `ScaleInfo`, which sets scaling properties for graphic visualizations of almost all elements without given geometric properties (e. g. frames, joints, force and torque arrows, ...)
9. Open the `ScaleInfo` by double-click and change the variable *`ScaleArrowForce`* to 1/1000

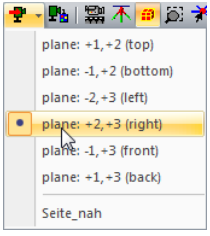

.Viewer
Viewer
Name
Value
Type
BackgroundColor
{0.5, 0.5, 0.5}
TColor
BackgroundGradient
TIvBackgroundGradi
BackgroundImage
TIvBackgroundImag
TransparencyType
"SORTED_OBJECT_BLEND"
STRING
Transparency
0
REAL
ScaleInfo
TIvScaleInfo
DefaultLayer
TIvLayer
BFrames
TIvLayer
Assignments
.Viewer.ScaleInfo
ScaleInfo
Name
Value
Type
Attributes
Reference
JointThick1
0.0625
REAL
JointThick2
0.03125
REAL
JointRadius1
0.039
REAL
JointRadius2
0.0195
REAL
ScaleArrowForce
0.001
REAL
RadiusArrowForce
0.025
REAL
Edit REAL
Variable :
ScaleArrowForce
OK
F
1/1000
Cancel
RadiusConnection=0.05*LengthFrameAxis
RadiusCentreOfMass=0.2*LengthFrameAxis
JointLength=0.5*LengthFrameAxis

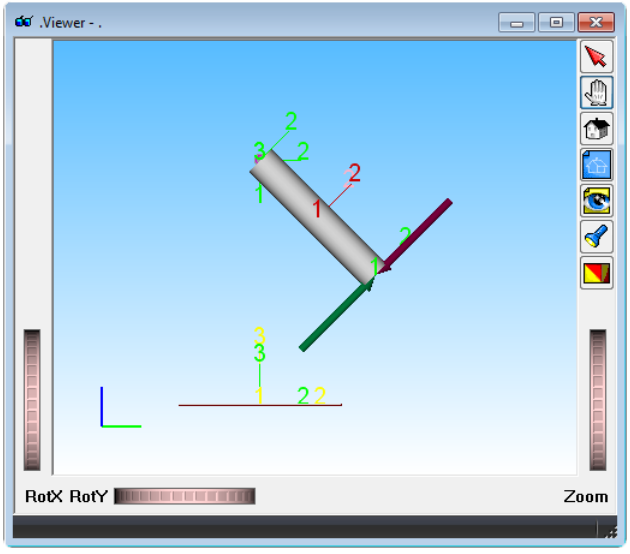
Visualization of the force

10. Now you can refresh the `Viewer` by pressing the toolbar symbol



11. Or you can close and reopen the `Viewer` window
12. To reset the scene in the `Viewer` to a nice view, you can use preset cameras, e. g. use plane +2, +3 within the toolbar:




.Viewer - .
RotX RotY
Zoom

Check

13. Now you can run `Assembly` and `Integration` task

- The pendulum will oscillate due to the initial angle and the applied force
- Note: The force always stays perpendicular to the pendulum
  This is due to the definition – as the value of the force is interpreted in its second argument `"Frame2"`, which lies on the pendulum
- If the force should be constant in the global coordinate system, the value of the vector has to be converted from the global coordinate system to `Frame2` using a relative rotation
  This is done in the next tutorial

Next tutorial

For a Force vector, which acts constant in the global coordinate system: Transform a Vector Using a Rotation
Or jump to the tutorial of a torque, which models a Insert a Spring-Damper