



Altair

HyperWorks

Altair MotionView 2019 Tutorials

MV-1090: Creating a Dataset using MDL

MV-1090: Creating a Dataset using MDL

In this tutorial, you will learn how to:

- Create a dataset to specify the start time, mid time, end time, and the force magnitude
- Include dataset definition in the analysis definition
- Vary the magnitude and time of the impulse torque

In many occasions, it is convenient to provide inputs or change the design variables of a model through a single interface. The variables could be in the form of real numbers, integers, or strings. Or it could also be the name of a file. Having a dataset enables such a modeling scenario.

A dataset is a collection of user-defined variables whose values are used or referred by another entity within MDL. Datasets are either created using the MDL language or the graphical user interface.

This exercise will focus on creating a dataset using MDL. A dataset is defined using a `*DefineDataSet()` - `*EndDefine()` block, which is similar to other definition based entities such as Systems and Analyses. The definition is then instantiated using the `*DataSet()` statement.

Exercise: Defining a Dataset

Step 1: Create a dataset definition.

The following steps illustrate how to create a dataset definition.

Refer to the *MDL Language Reference* online help for the correct syntax for the MDL statements you choose to use.

1. In a new document in a text editor, create the `*DefineDataSet()` and `*EndDefine()` block. You will create data members belonging to the dataset between these statements.

The data members that you need to define in the dataset are:

- Starting time
- Mid time
- End time
- Force magnitude

As all the data members are real numbers, we will use `*Real()` to define them.

Use one `*Real()` statement to define one data member.

You can also define other types of members such as: integers, strings, options, or a file (as applicable to your model).

2. Save the file in the working directory as `dataset.mdl`. Your file should look like this:

```
*DefineDataSet(ds_def_force)
  *Real(start_time, "Starting Time")
  *Real(mid_time, "Mid Time")
  *Real(end_time, "End Time")
  *Real(force_magnitude, "Force Magnitude")
*EndDefine()
```

Step 2: Include the dataset in the analysis definition and instantiate.

The dataset will be included in the analysis definition by using the `*Include()` statement. The dataset entities will be incorporated in the expression for torque.

1. In a text editor, open the analysis definition file created in tutorial [MV-1080](#). Include the dataset definition in it by using the `*Include()` statement **before** the `*DefineAnalysis()` statement.

The syntax is:

```
*Include("dataset.mdl")
```

2. Instantiate this dataset definition using the `*DataSet()` statement within the `*DefineAnalysis()` block and after the `*Attachment()` statement.

The syntax for the `*DataSet()` statement is:

```
*DataSet(ds_name, "ds_label", ds_def, [optional arguments])
```

where

- `ds_name` is the variable name of the dataset.
- `ds_label` is the label of the dataset.
- `ds_def` is the variable name of the existing dataset definition.
- `optional arguments` are arguments that are passed as attachments (if any)

Instantiate the dataset by choosing a suitable variable name and label. The `ds_def` should be the same as the variable of the dataset definition used in the `*DefineDataset()` statement.

```
*DataSet(ds_force, "Force Data", ds_def_force)
```

3. Set the default values of the data members in the dataset by using the `*SetReal()` statement within the `*DefineAnalysis()` block.

The syntax for the `*SetReal()` statement is:

```
*SetReal(real_name, real_value)
```

where

- `real_name` is the variable name of the data member for which the value is being set.
- `real_value` is the value of the data member.

As the data member is a part of the dataset, the correct form of referring to the variable name of the real entity is `ds_name.real_name`.

For example, the `*SetReal()` statement for start time would be:

```
*SetReal(ds_force.start_time, 0.3)
```

4. Set the default values of all the data members used in the dataset definition.

Include the lines below after the `*DataSet()` statement:

```
*SetReal(ds_force.start_time, 0.3)
*SetReal(ds_force.mid_time, 0.31)
*SetReal(ds_force.end_time, 0.32)
*SetReal(ds_force.force_magnitude, 10)
```

5. The `*SetForce()` statement in the analysis definition looks like:

```
*SetForce( force_1, EXPR, `step(TIME, .3,0, .31,10) +
step(TIME, .31,0, .32,-10)`)
```

6. Change the appropriate values in the `*SetForce()` statement by incorporating the dataset members. The idea is to use the dot operator to browse through the model hierarchy and access the dataset values (for example, use `ds_force.start_time.value` to access the start time value from the dataset). This is illustrated in the following statement:

```
*SetForce(force_1, EXPR, `step(TIME, {ds_force.start_time.value}, 0,
{ds_force.mid_time.value},
{ds_force.force_magnitude.value}) + step(TIME,
{ds_force.mid_time.value}, 0, {ds_force.end_time.value}, -
{ds_force.force_magnitude.value})`,0,0)
```

The expressions within the curly braces (`{}`) get processed by Tempex in MotionView and get evaluated to the appropriate values defined in the dataset.

The analysis definition file should look as below:

```
*Include("dataset.mdl")

*DefineAnalysis( def_ana_0,j_att )

  *Attachment(j_att, "Joint Attachment", Joint, "Select joint to apply
torque")

*DataSet(ds_force, "Force Data", ds_def_force)
```

```

*SetReal(ds_force.start_time, 0.3)

*SetReal(ds_force.mid_time, 0.31)

*SetReal(ds_force.end_time, 0.32)

*SetReal(ds_force.force_magnitude, 10)

*ActionReactionForce( force_1, "Torque", ROT, j_att.b1, j_att.b2,
j_att.origin, Global_Frame )

*SetForce(force_1, EXPR, `step(TIME, {ds_force.start_time.value}, 0,
{ds_force.mid_time.value}, {ds_force.force_magnitude.value}) +
step(TIME, {ds_force.mid_time.value}, 0, {ds_force.end_time.value}, -
{ds_force.force_magnitude.value})`)

*Output( o_force, "Input Torque", FORCE, FORCE, force_1, Global_Frame)

*EndDefine()

```

7. Save the above work in a new analysis definition file named `analysis_dataset.mdl`.

Step 3: Change the dataset parameters and run the analysis.

1. In MotionView, load the triple pendulum model created in tutorial *MV-1080*.
2. Delete the existing analysis (if any) by right-clicking on the **Analysis** in the **Project Browser** and clicking **Delete**.
3. Click on **Model** in the **Project Browser**.

The **Systems/Assembly** panel is displayed.

4. From the **Import/Export** tab, import the new analysis definition file `analysis_dataset.mdl` following similar steps used to import the analysis in the earlier exercise (*MV-1080*).
5. From the **Project Browser**, expand the **Datasets** folder and select **Force Data**.

You will see the dataset with the **Labels** and **Values** for all of the members in the dataset.

Label	Variable	Type	Value
Starting Time	start_time	Real	0.3000
Mid Time	mid_time	Real	0.3100
End Time	end_time	Real	0.3200

6. Change the **Starting Time** to 0.5, **Mid Time** to 0.55, **End Time** to 0.6 and the **Force Magnitude** to 15.

-
7. Solve the model.
 8. Compare the `Input Torque` in the plot window with that of the earlier analysis.
You can now change the force parameters easily through the dataset graphical user interface and re-run your analysis.
 9. Save your model as `triplependulum_dataset.mdl`.