Resources Mass Spring Damper Printable

Table of Contents

◄ **Previous:** **Mass Spring Damper** | **Next: Chapter 2:** **Getting Started** ►

Mass Spring Damper Tutorial Introduction

The Open META-CyPhy tools have been developed by ISIS-Vanderbilt to deliver an integrated capability using open source tools for:

- Compositional design synthesis at multiple levels of abstraction;
- Design trade space exploration and metrics assessment with structural and information-based metrics of system complexity;
- Formal semantic integration of models across multiple physical and cyber domains, and
- Probabilistic verification of system correctness with respect to realistic context models providing correct-by-construction capability.

This tutorial will teach you how to build, test, and share designs using the Open META-CyPhy tools. The designs you will be creating are mass-spring-damper systems. This system is composed of a mass suspended from a spring, which uses a damper to gradually slow movement of the mass. The damper serves to discourage the mass from oscillating when it moves and helps it return more quickly to a state of equilibrium.

This tutorial can be broken roughly into three stages: preparation, building, and testing.

- In the preparation phase, you will download and install the tools and ready them for use.
- In the building phase, you will construct the design and connect the components, then enhance the design so that it contains multiple possible configurations of components. After that, you will build an environment in which you can test the design.
- In the testing phase, you will simulate the design and its environment using various remote and local services.

Finally, there are instructions on using the repository to analyze your simulation results, score your design against a set of predefined requirements, and share your design with others.

◄ **Previous:** **Mass Spring Damper** | **Next: Chapter 2:** **Getting Started** ►

◄ **Previous: Chapter 1:** **Mass Spring Damper Introduction** | **Next: Chapter 3:** **Using the Component Library** ►

Chapter 2: Getting Started

Step 1: Launching

Start GME from the Start menu by clicking **Start** → **All Programs** → **GME** → **GME** as shown in Figure 2.1.

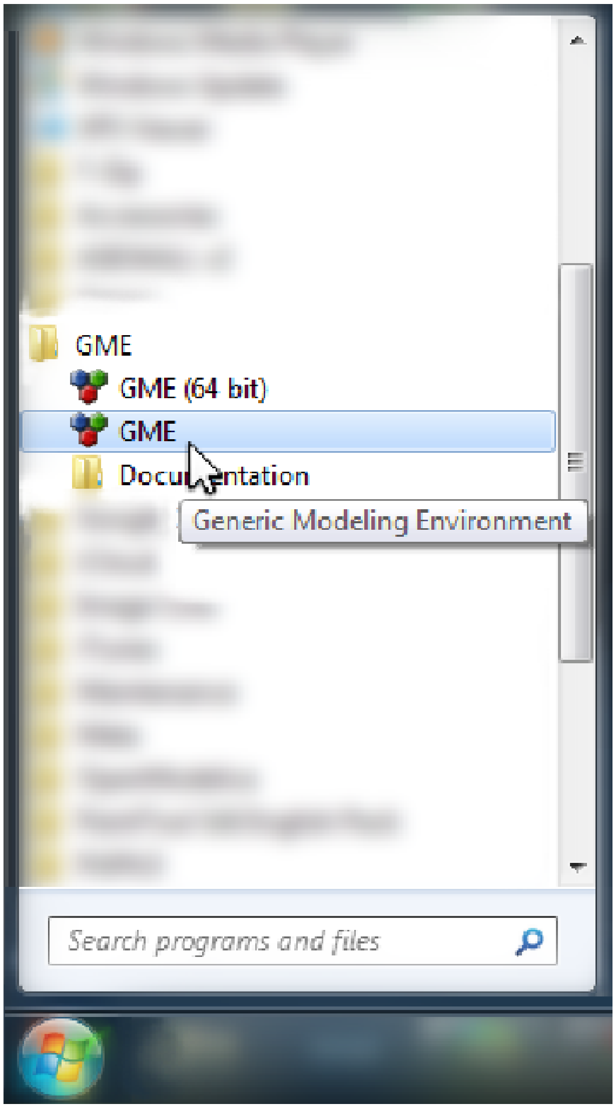


Figure 2.1: Launching GME from the Start Menu

Close the dialog box asking you to start or open a project.

Step 2: Creating a Project

Download the files required for this tutorial here. **Download the zip file and extract its contents**. Contents included are the GammaMSDTutorial_Starter_File.xme, and a folder called "Post_Processing".

In GME, **go to File → Import XML. Browse to the location of the files you extracted and import MSDTutorial.xme.**

Select Next as shown in Figure 2.2 to create a new project file.

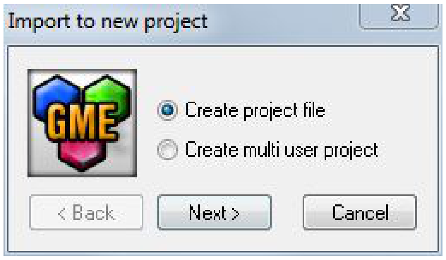


Figure 2.2: Create Project File

Save the MSDTutorial as a MGA file. **Be sure the extracted Post_Processing folder that came in the zip file is stored in the same directory that you saved the MGA file.** This is required in order for your design in the tutorial to be properly simulated and scored.

After a moment, a toolbar should appear in the top section of the GME screen. Figure 2.3 shows what GME should look like after you expand MassSpringDamper.

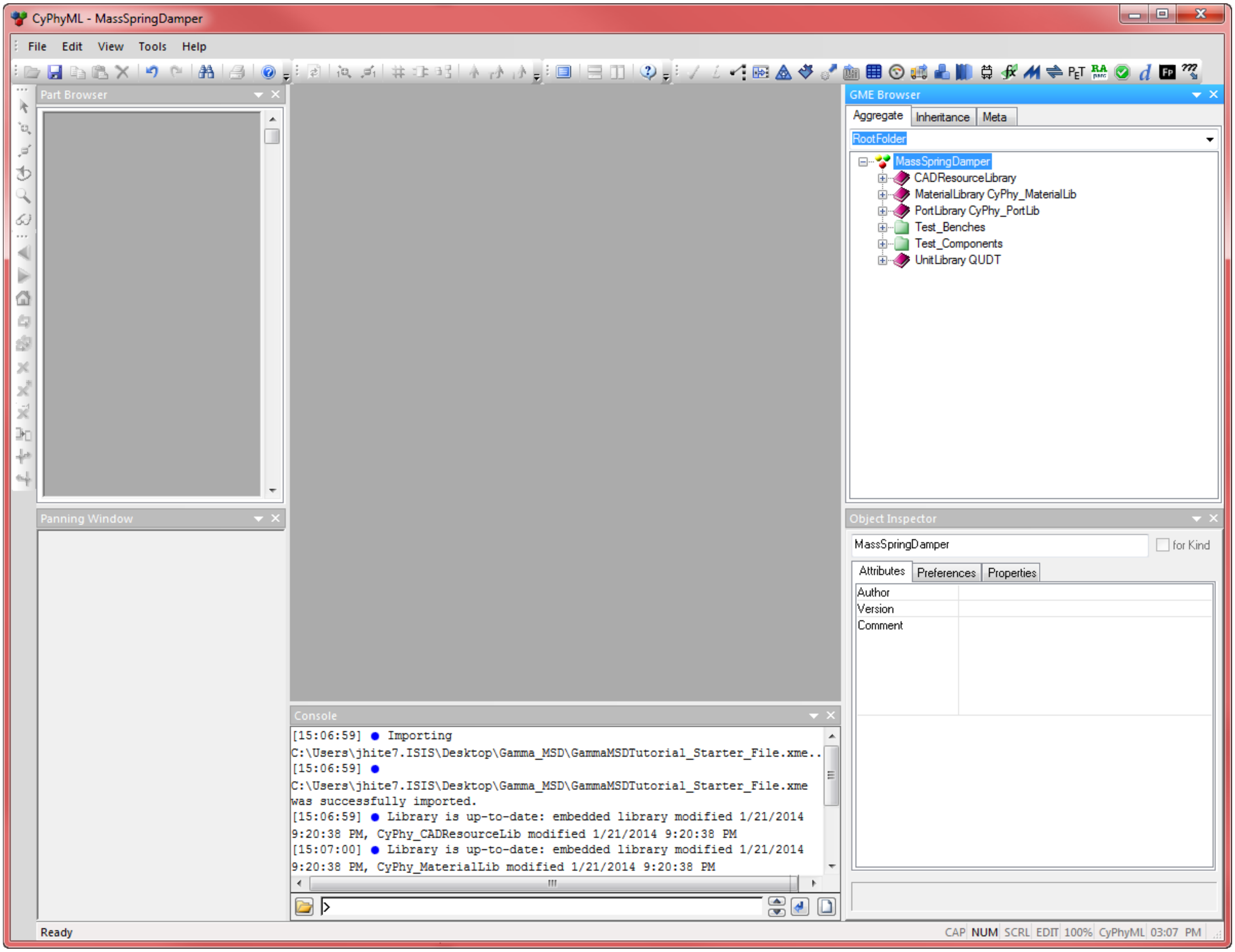


Figure 2.3: Create Project File

Saving Your Project

There are two save formats supported by the GME and the Open Meta-CyPhy tools. These formats are .xme and .mga. The .xme format is an xml that can be interpreted by GME to reproduce a project in GME. When GME interprets the .xme file it creates a .mga binary file. This file can also be opened by GME.

The binary .mga can be saved by clicking the blue floppy disk icon in the tool bar or by going to File -> Save Project or File -> Save Project As. These options work in a similar manner to other programs. This format is easily read by GME but is difficult to migrate between versions of the Open Meta-CyPhy tools.

The .xme format can be created from a project by going to File -> Export XML. This file format is easier to port between versions of the tools, but takes extra time to import into GME because it must be converted into a .mga file first. This conversion process does not modify the original file in anyway. It is recommended that you back any files to a .xme format before updating tools.

◄ **Previous: Chapter 1:** Mass Spring Damper Introduction | **Next: Chapter 3:** Using the Component Library ►

◄ **Previous: Chapter 2: Getting Started** | **Next: Chapter 4: Creating a Component Assembly** ►

Chapter 3: Using the Component Library

Component information is available for the individual Mass, Spring, and Damper components in **Appendix A: MSD Component Documentation.**

The component exchange on the Vehicle Forge website is a tool that you can use to retrieve components for design in Open META-CyPhy. All curated components exist here and are free for anyone to use. These components will reflect components that can either be produced using iFAB or obtained off the shelf.

The process is only two steps, which are explained in more detail below:

1. Visit the Component Exchange on VehicleForge and download the appropriate component.
2. Import the downloaded component into GME.

Step 1: Find and download the appropriate component

First, **log into VehicleForge**. Next, **click on the gears icon in the VehicleForge header bar** as shown in Figure 3.1. This will take you to the component library.

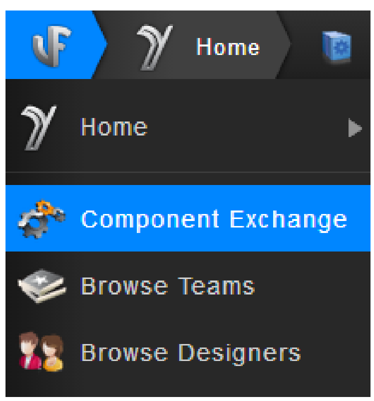


Figure3.1: Access to the component exchange in VehicleForge.

In this example we will be using the Mass components, so **select Mass from the FANG Components list.** Figure 3.2 shows the appropriate selections:

Figure 3.2: List of Mass, Spring, Damper Components Available for Download

Click "**Mass**" in the second column (the bracketed number indicates how many components are in the category). A list of the available components will appear below, as seen in Figure 3.3.

Showing 10 results of 3

Name	m	Density
Mass_Gold	[0.25 to 0.75]	0.0000193
Mass_Steel	[0.065 to 0.31]	0.00000785
Mass_Titanium	[0.025 to 0.15]	0.000004506

Figure 3.3: List of Fang Mass Component Types Available for Download

Click the component you wish to download, which is in this case "Mass_Gold".

You'll be brought to a screen where you can view detailed data about the component. On the right sidebar you will see several options as shown in Figure 3.4.

Click **Download Component**. Zip files downloaded from here contain the component along with any other files required to run the component from Open META-CyPhy.

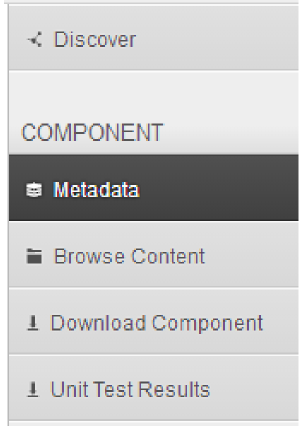


Figure 3.4: META Component Options

For more information on the component exchange, please see the correspoding section in the **Vehicle Forge Guide.**

Step 2: Import Components with the Component Importer

Open or return to GME, then **find the component importer in the tool bar and select it. Browse for the zip file that you downloaded for that component. Once you have selected the component press Open**

Note: You will not need to unzip the components downloaded from the exchange to import them.

The GME browser should contain a new folder called “Imported Components.” Inside this folder is the Mass_Gold from the component exchange. This process can be repeated for any components you feel will be necessary for your design. You can import multiple components at the same time by selecting all of them.

Repeat this process for Damper_10, Damper_6, Damper_2, Mass_Steel, Mass_Titanium, Spring_Tungsten, Spring_Steel, and Spring_Titanium. The GME interface should look like Figure 3.5 below.

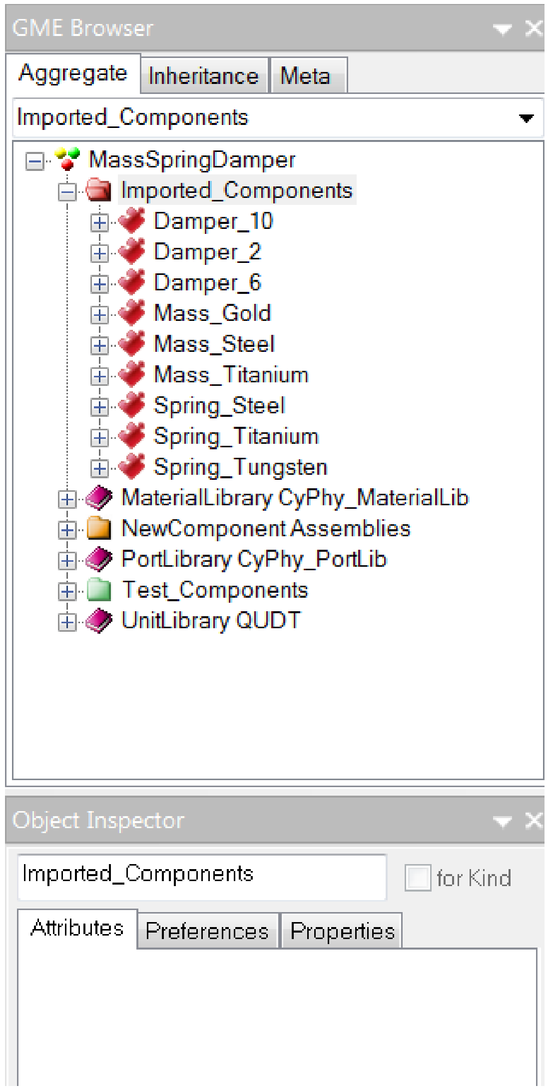


Figure 3.5: All 9 components imported into GME

◄ **Previous: Chapter 2: Getting Started** | **Next: Chapter 4: Creating a Component Assembly** ►

◄ **Previous: Chapter 3: Using the Component Library** | **Next: Chapter 5: Creating a Design Space Assembly** ►

Chapter 4: Creating a Component Assembly

Introduction

This chapter guides you through the process of making a simple component assembly in Open META-CyPhy. Three components are used: Mass, Spring, and Damper.

These three components will be connected to make a component assembly, referred to in this document as MassSpringDamper (MSD).

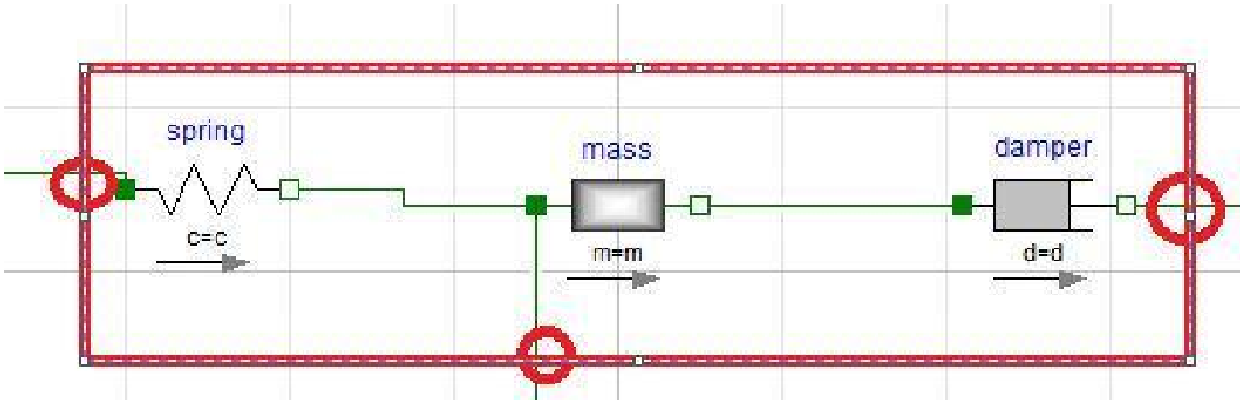


Figure 4.1: Mass-Spring-Damper Diagram

Chapter Overview

1. Create a folder to hold a component assembly, and then create the component assembly in the folder.
2. Copy components from the library into the assembly.
3. Make solid-modeling connections between the components.
4. Add parameters and properties.
5. Make dynamic connections between the components.

Key Vocabulary

A **component** is an individual model in Open META-CyPhy.

A **component assembly** is a collection of models in Open META-CyPhy that interact with each other as a unit.

A **components folder** is a type of folder that is designed to hold components and other general component-related folders.

A **component assembly folder** is a type of folder that is designed to hold component assemblies.

A **parameter** is a value in Open META-CyPhy given to a component that is meant to be edited (i.e. the fuel flow in a carburetor).

A **property** is a value in Open META-CyPhy given to a component that is not meant to be edited (i.e. the material of a block).

A **port** is an interface on a component that is used to connect the component to other components.

The **Solid Modeling** aspect holds CAD models in the Open META-CyPhy language.

The **Dynamics** aspect holds Modelica models in the Open META-CyPhy language.

The **Job Manager** allows for configurations to automatically run in other programs. It also allows for easy access to the results folders for each configuration run.

See **Appendix A: CyPhy Reference Guide** for further information on key vocabulary seen above and other common words used in OpenMeta-CyPhy.

Procedure

Step 1: Create Folders and Create a Component Assembly

A component assembly is a way to organize components in preparation for conversion into a design space. This will represent only one of the possible configurations, or arrangements, that can be generated by the design space.

1.1: Create a Component Assembly Folder

The first step in creating a component assembly in GME is to insert a new component assembly folder in the GME Browser.

In the GME Browser, **right-click on "MassSpringDamper" and select Insert Folder > Component Assemblies** as shown in Figure 4.2. **Name this folder "MyComponentAssemblies".**

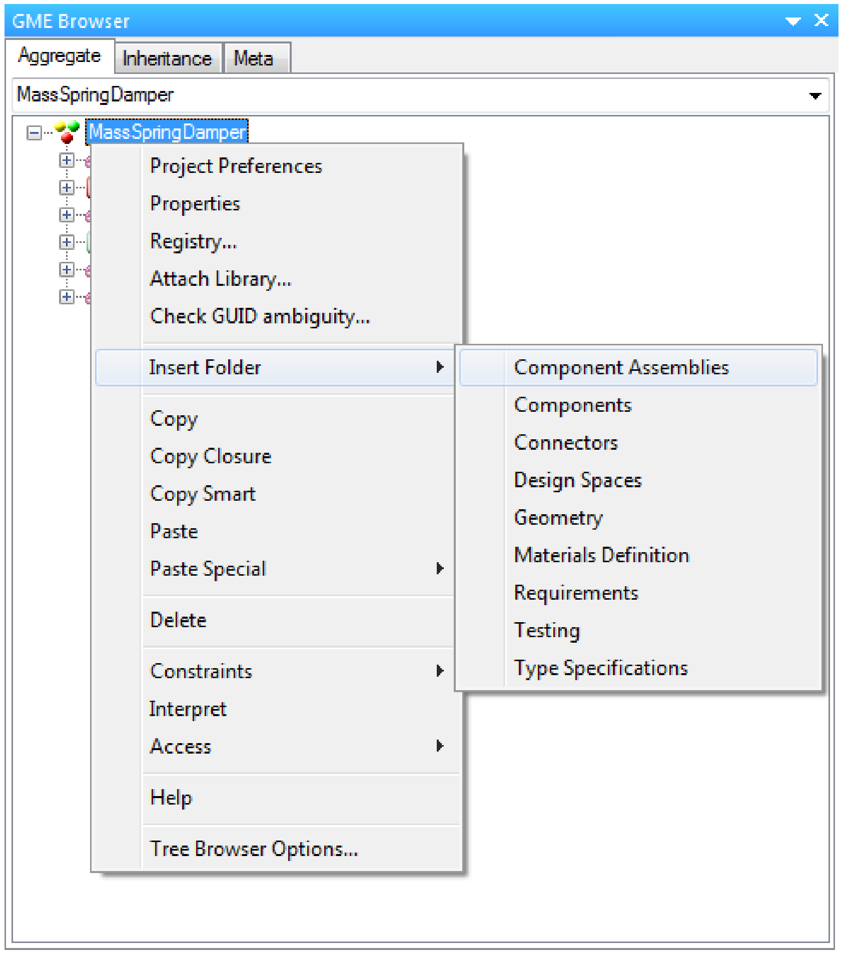


Figure 4.2: Create a Component Assembly Folder

1.2: Create Component Assembly Model

Now that the new component assembly folder is created, the next step is to insert a Component Assembly Model into this folder.

Right click on the MyComponentAssemblies folder and select Insert Model > Component Assembly. Name this component assembly "MyMassSpringDamper" as shown in Figure 4.3.

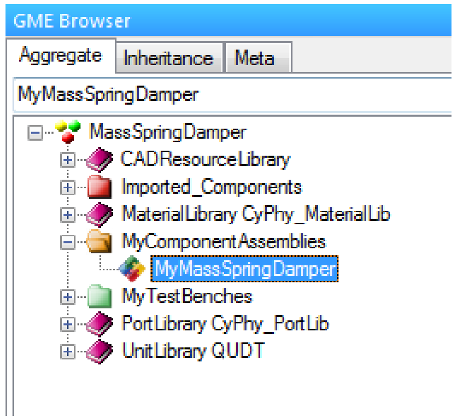


Figure 4.3: Creating a Component Assembly Model inside a Folder

Step 2: Copying Components into Component Assembly

In this step, we introduce pasting a component as a reference. The difference between pasting as a reference and a plain paste is that a plain paste will create an instance of the component; this means that any changes made to the original component will not be propagated to the copied instance. However, when a component is pasted as a reference the changes in the base model will affect the copy. **Unless stated otherwise, you should always paste as a reference.**

2.1: Copy/Pasting Components into a Component Assembly

Double click on the newly created component assembly model (not the folder) titled "MyMassSpringDamper." This will open a blank window in the Editing Area.

In the GME Browser, locate the first component to be added to the Editing Area.

To do this, **expand the Imported Components folder** by clicking on the plus sign next to it. **Right click on Damper_2 and select Copy** as shown in Figure 4.4.

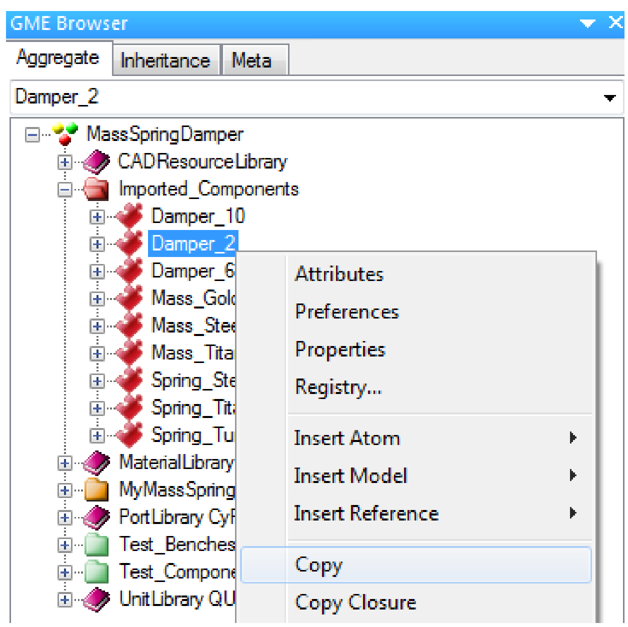


Figure 4.4: Copying a Component

After copying Damper_2, **right click in the open Editing Area. Select Paste Special > As Reference**. Damper_2 will now be located in the Editing Area as shown in Figure 4.5.

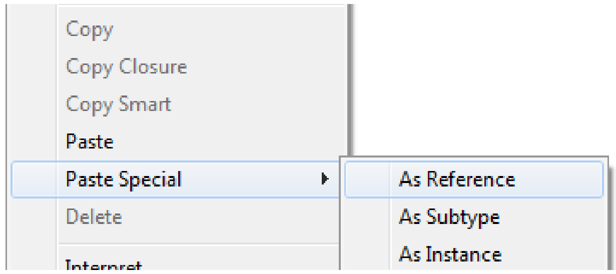


Figure 4.5: Copy -> Paste as reference in to editing area

Next, **add the components named Mass_Steel and Spring_Steel to the Editing Area** by repeating what you did for Damper_2 as seen in 4.6.

Figure 4.6: Mass_Steel, Spring_Steel, and Damper_2 in Editing Area

Step 3: Make Connections between Components

3.1: Enable Connect Mode

Once the three components are located inside of the Editing Area, the next step is to enable Connect Mode. Connect Mode can be enabled either by **pressing CTRL+2** or **selecting the button highlighted in Figure 4.7 on the left-hand tool bar**.

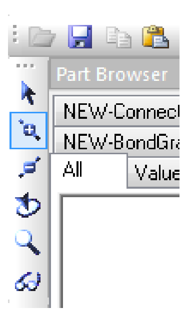


Figure 4.7: Connect Mode

3.2: Connecting Modeling Ports

After Connect Mode has been enabled, the components need to be connected to each other. First **click on the puzzle piece labeled Threaded_Pin on the damper component**, then **connect it to the puzzle piece labeled INSIDE_HOLE on the spring component**. This will create a connection between these two components, as shown in Figure 4.8. (Note that the ports on your component may be in a different order. This is only visual. Ensure that you connect correctly based on the port NAME.) If you accidentally select the wrong port, simply press Esc and the port will be deselected. **CTRL+3** enables Disconnect Mode and allows you to delete any previous connections.

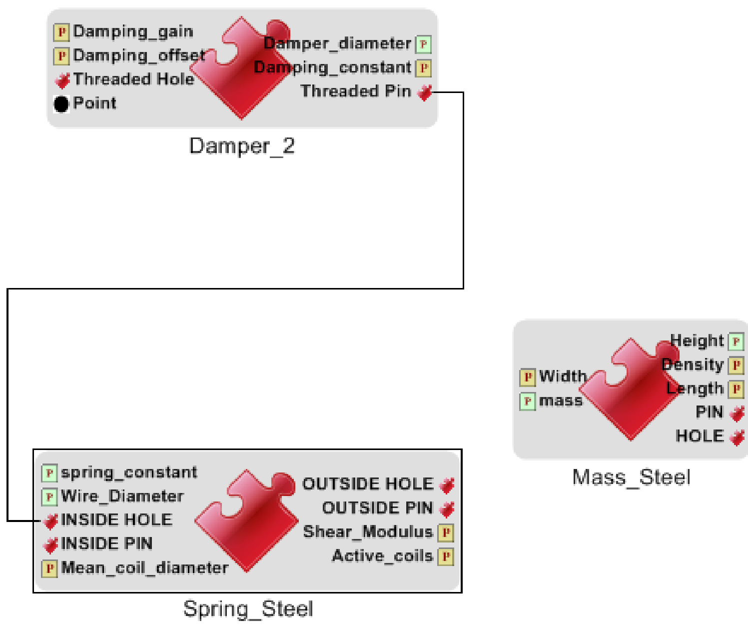


Figure 4.8: Connecting Ports

Next, **connect the following ports as shown in Figure 4.9.** OUTSIDE_PIN on the spring component should be connected to HOLE on the mass component, and Threaded_Hole on the damper to INSIDE_PIN on the spring.

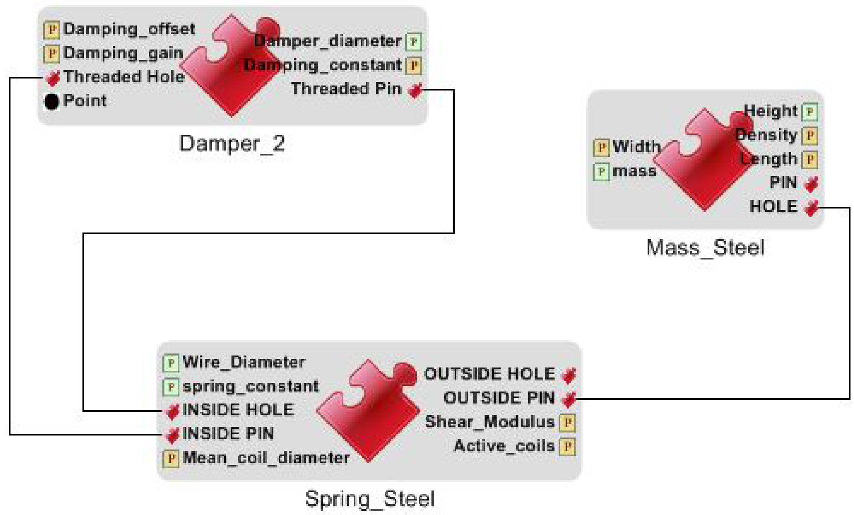


Figure 4.9: Connecting Interfaces between the Spring_Steel and Mass_Steel

Step 4: Add Parameters and Properties

Properties and parameters are ways of displaying values that define components. Properties are values that are fixed for a given component, such as the mass of a particular engine. Parameters on the other hand are values that can be varied within a range like the length of an adjustable drive-shaft. Together these values help to define a component in Open META-CyPhy.

A property or parameter can only ever have one incoming connection, but can have several outgoing connections. Those connections are also direction specific as denoted by the double arrow drawn when a connection is made.

4.1: Adding Parameters to the Work Space

Select Edit Mode with CTRL+1 or the  **button.** Then, **locate the parameter icon** in the Part Browser to the left of the Editing Area seen in Figure 4.10.

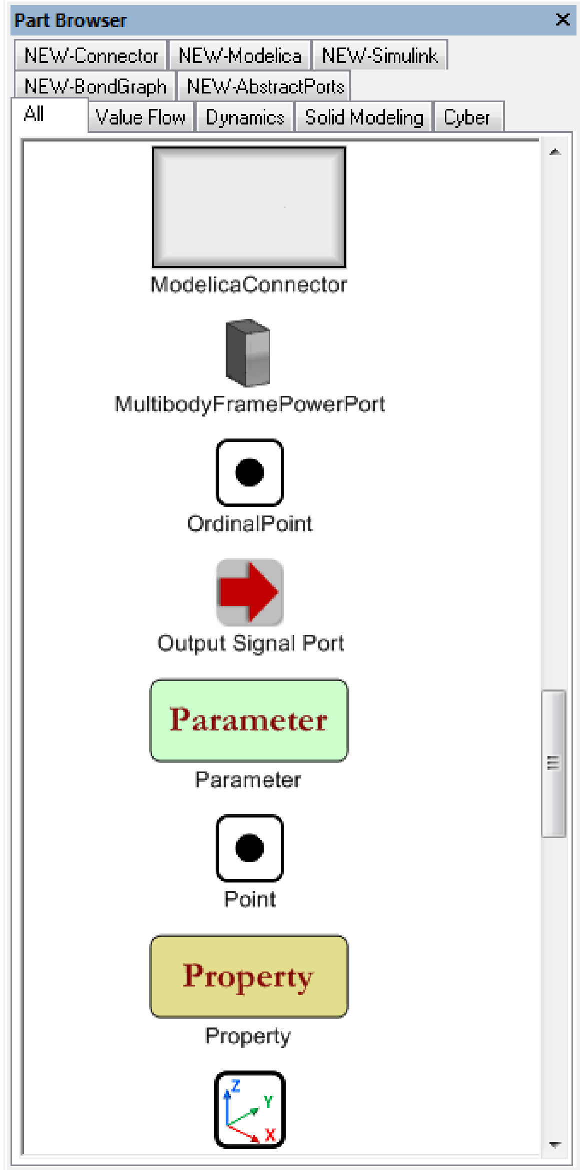


Figure 4.10: Parameter in the GME Browser

Drag and drop this icon into the Editing Area.

Using the Object Inspector Browser (in the lower right hand corner) **change the name of the parameter from "parameter" to “spring_constant.”**

Add an additional parameter icon so that the Editing Area matches the diagram in Figure 4.11.

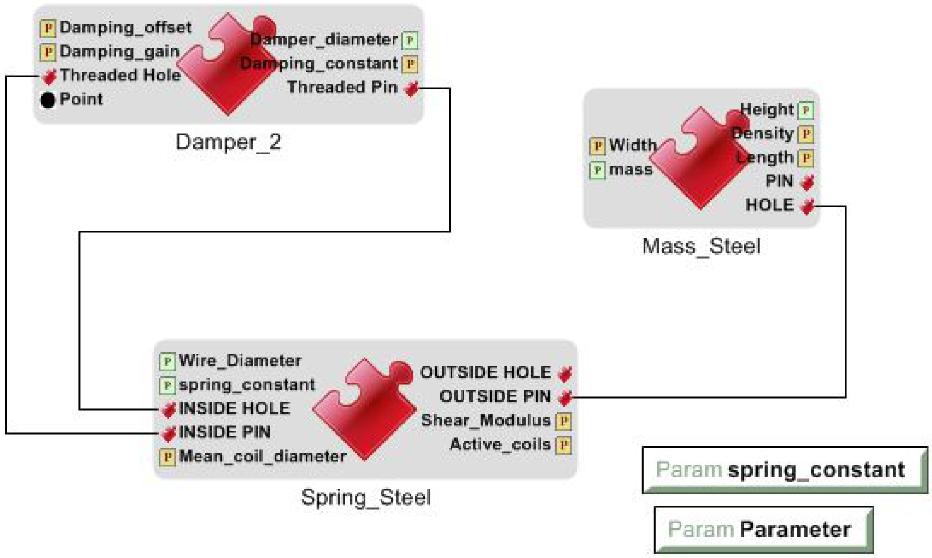


Figure 4.11: Adding Parameter to the Component Assembly

4.2: Defining Parameters

Now that there are two parameter icons in the Editing Area, the next step is to give these icons values.

Ensuring that Edit Mode is selected (CTRL+1), click on one of the parameter icons in the Editing Area.

Use the Object Inspector Browser to the right of the Editing Area to fill out the parameter values as shown in Figure 4.12.

To set the value of the parameter select the attributes tab in the Object Inspector and enter the value of the parameter in the value field.

Set the value of the spring_constant parameter to 1000, and change the Data Type to "Float". "Data Type" is distinct from "Type".

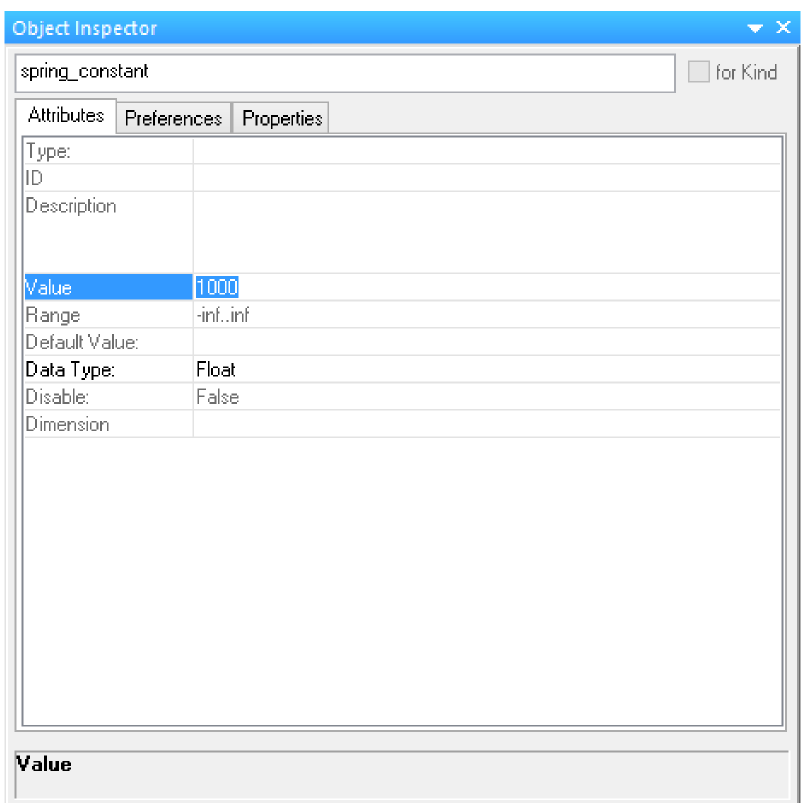


Figure 4.12: Setting Values for the "Spring_Constant" Parameter

Rename the second parameter "mass," and give it the value 0.5 as shown in Figure 4.13 (.5 and 0.5 are the same).

Set Data Type to Float.

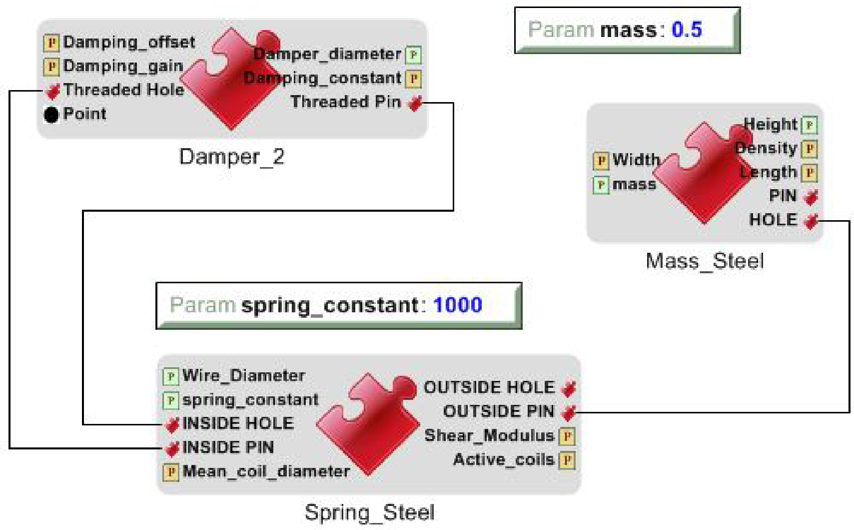


Figure 4.13: Setting Value of the "mass" Parameter

4.3: Connecting Parameters to Components

Enable Connect Mode. Then, **connect the spring_constant Parameter to the Spring_Steel component, and mass Parameter to the Mass_Steel component.**

Make sure that the arrow points AWAY from the parameter and TO the port.

The Editing Area should resemble the diagram shown in Figure 4.14. (Component, port, and connection visual layout may be different, but the connections must match!)

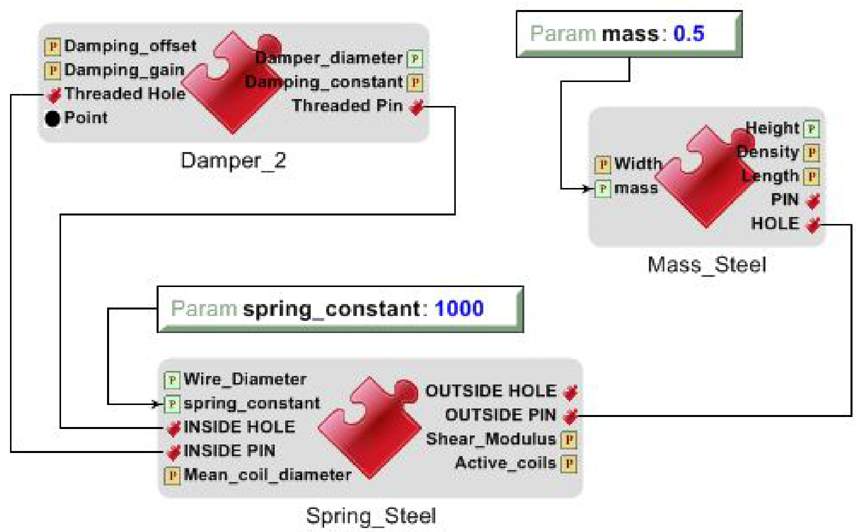


Figure 4.14: Connecting Parameters to Components

4.4: Adding Units to Parameters

Open META - CyPhy is able to use units to help make sense of all the properties and parameters that make up a component.

To add units to the model, find **UnitLibrary QUDT** in the GME Browser as shown in Figure 4.15.

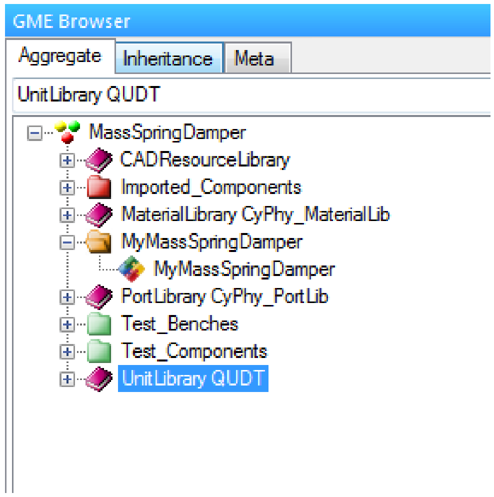


Figure 4.15: UnitLibrary QUDT

Next, **expand the selections until the Units folder is fully expanded** as shown in Figure 4.16.

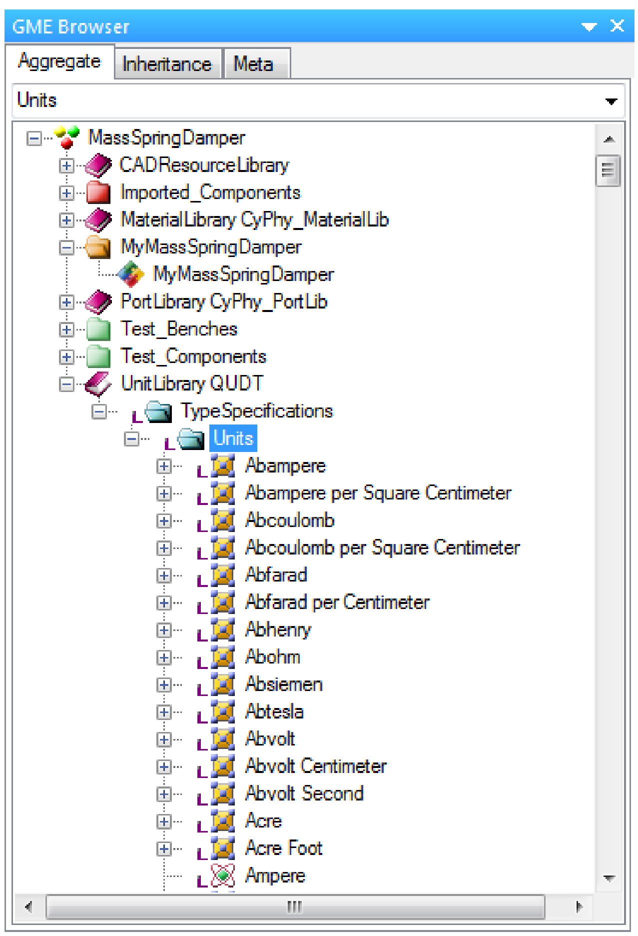


Figure 4.16: Units Folder Expanded

Now, **browse for the unit Kilogram; simply drag and drop the unit on the mass parameter.** You can use either Kilogram object. One is an atom used to derived other units such as Newtons. The other is a derived the derived unit put in the library for consistency.

The units for the spring_constant are **Newtons per meter**. **Drag and drop on spring_constant.** Once this is complete, your model will resemble Figure 4.17.

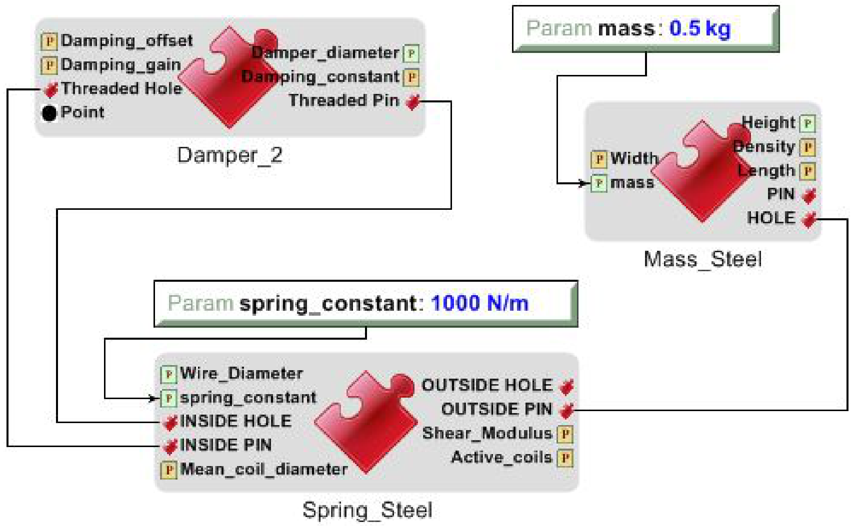


Figure 4.17: Mass and Spring_Constant parameter with units

4.5: Adding a Property

We will now add a property to the Editing Area. **Drag and drop a Property (not Parameter) block** from the Part Browser and **name it "Height"** as shown in Figure 4.18.

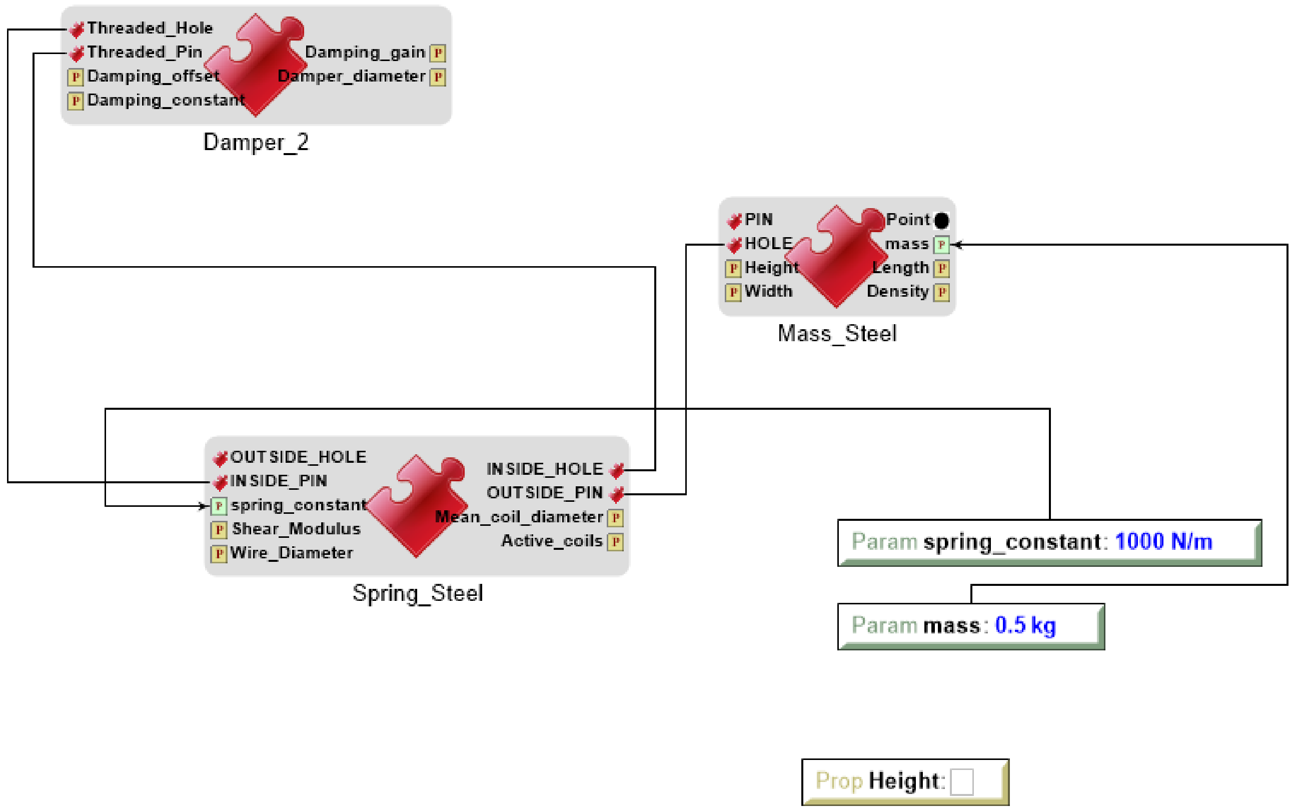


Figure 4.18: Adding a Property

Enable Connect mode and connect the "Height" property in the Mass_Steel component to the "Height" property. ***Make sure that the arrow points from the port to the Property.***

Also, **add the Meter unit to this property** as you did with kg and N/m (either Meter unit will work). Your screen should now resemble Figure 4.19.

Note: This allows the height value to flow to a higher level in the model where we can perform other functions on it. These will be discussed in Chapter 5.

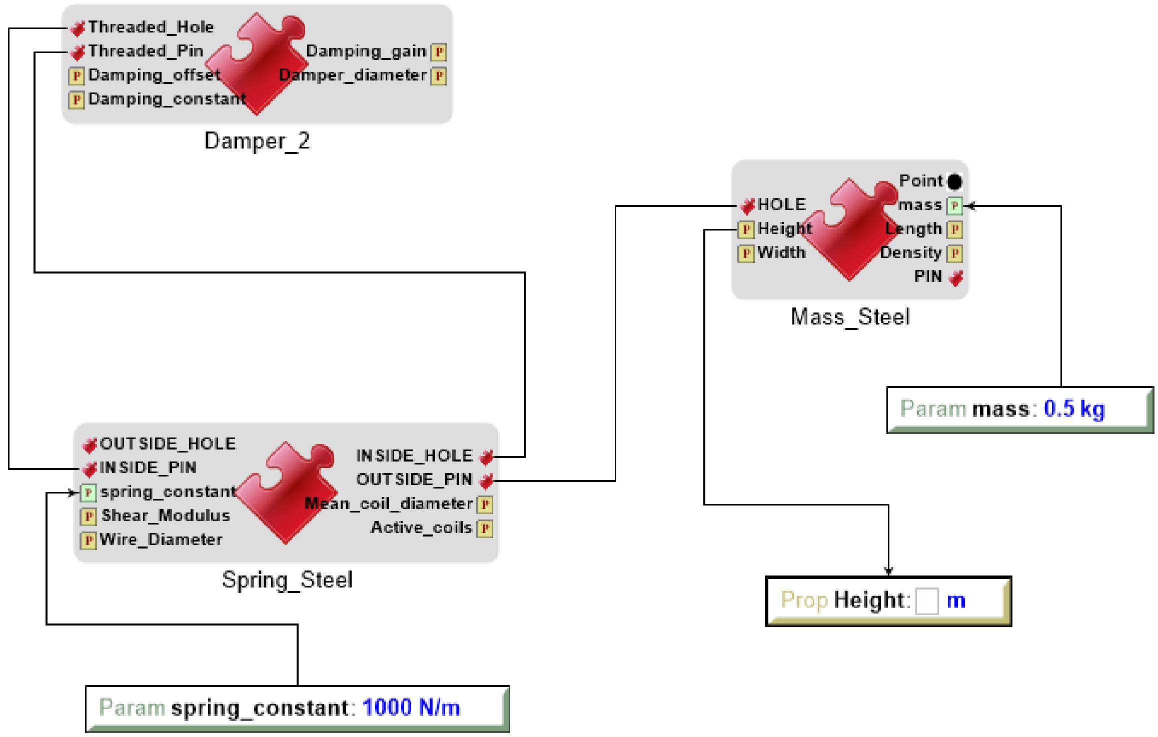


Figure 4.19: Connecting Formulas and Properties

Step 5: Connectors

This step shows how to add connectors to a component assembly. This same method can also be done with adding connectors to design spaces, seen in the next chapter. These connectors represent the interaction points between different components, subsystems, and systems.

5.1: Adding Connectors

The addition of connectors enables the model to respond to outside signals.

In Edit mode, locate the PIN connector within the Mass by double-clicking the Mass_Steel component. It should match Figure 4.20.

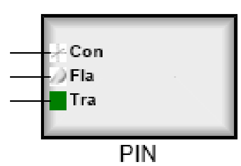


Figure 4.20: PIN Connector

Copy the PIN connector and paste it (not as a reference) within the MyMassSpringDamper component assembly. To get back to the component assembly, click on the MyMassSpringDamper tab at the top of the Editing Area.

Do the same with the OUTSIDE_HOLE from the Spring.

Copying and pasting the connector from the already created component ensures that the connector is built exactly the same way.

5.2: Connecting to Components

Enable Connect Mode. Connect the newly added Connector icons to the components as shown in Figure 4.21. (Component, port, and connection visual layout may be different, but the connections must match!)

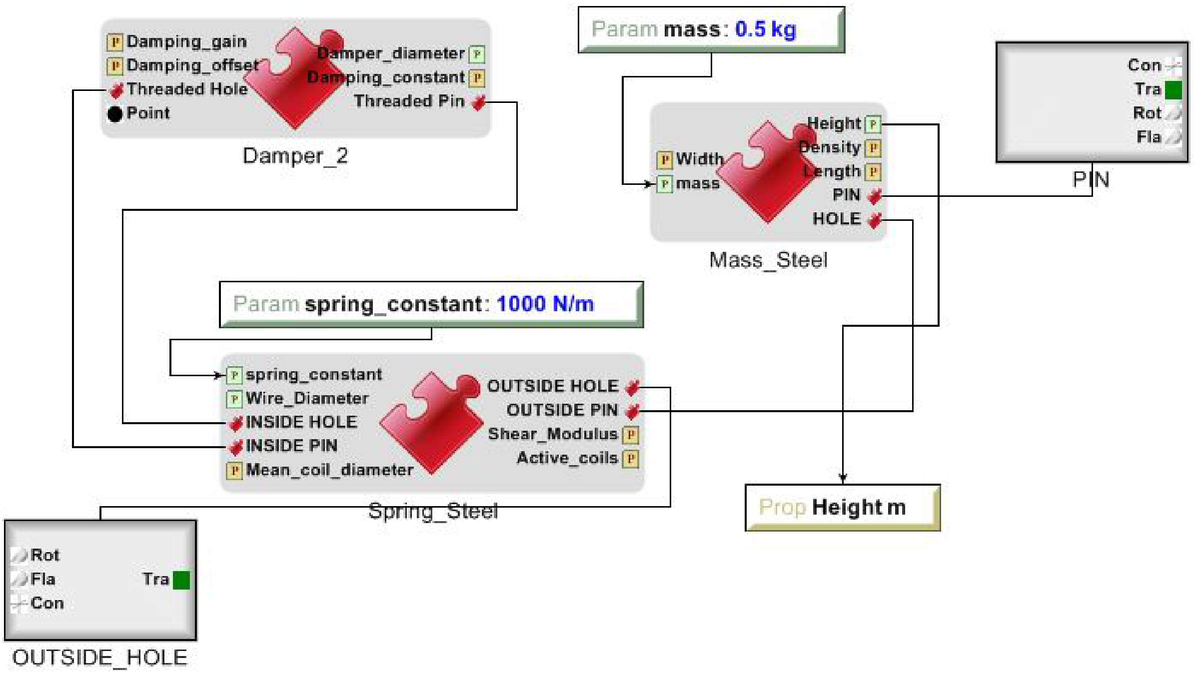


Figure 4.21: Connecting the Connectors

You have now successfully created a component assembly in GME. The next chapter will cover how to transform this component assembly into a design space assembly, which will make it possible for the simulator to quickly swap out the parts of your design.

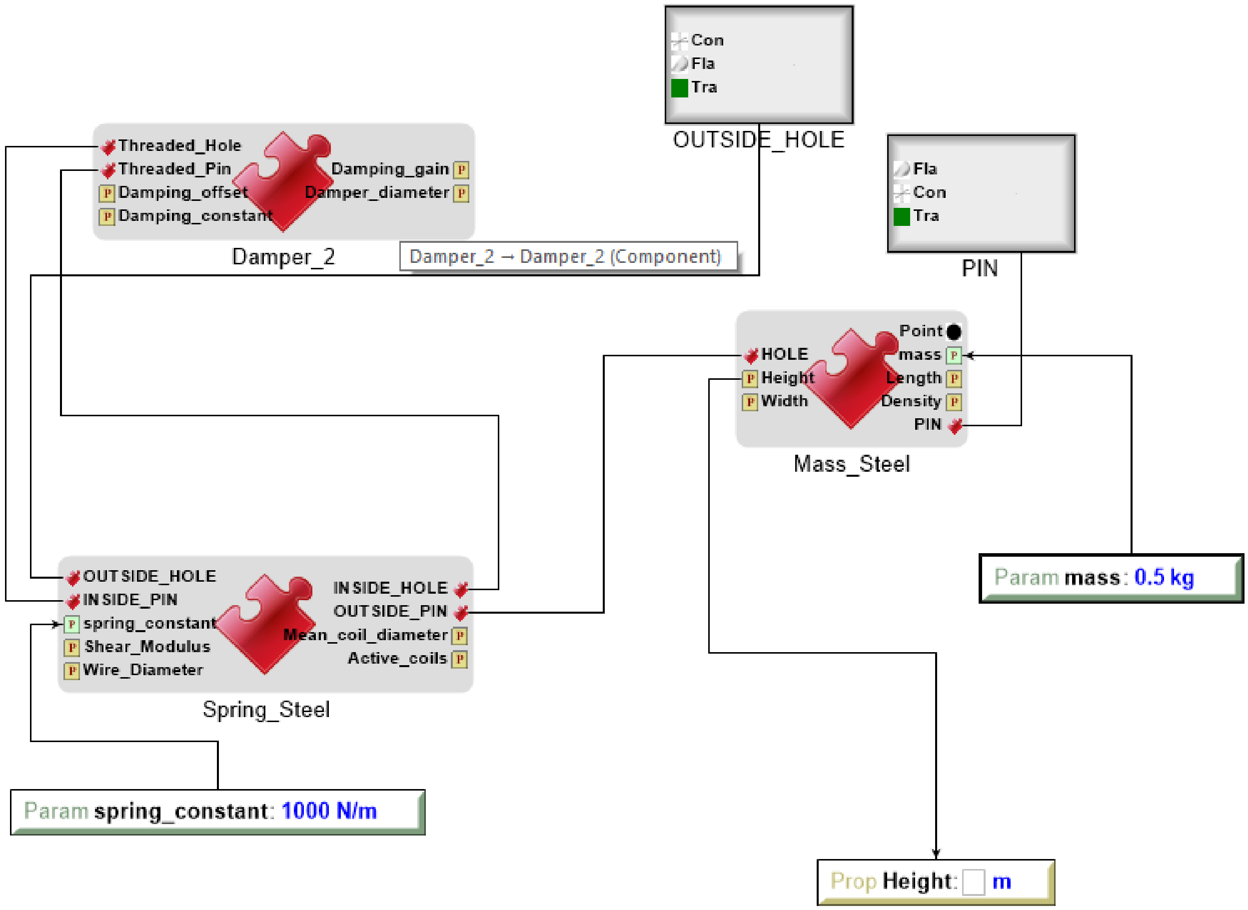


Figure 4.22: Final Component Assembly

◄ **Previous: Chapter 3:** Using the Component Library | **Next: Chapter 5:** Creating a Design Space Assembly ►

◄ **Previous: Chapter 4:** Creating a Component Assembly | **Next: Chapter 6:** Creating a Test Bench ►

Chapter 5: Creating a Design Space Assembly

Introduction

In the previous section of this tutorial, you successfully created a functioning MassSpringDamper assembly. However, this assembly represents only a single possible configuration for the mass spring damper.

When you test this assembly, you will only get results pertaining to a mass-spring-damper system with a steel spring, steel mass, and Damper_2. This is not useful when you need comparative data.

While it's possible to assemble different configurations of the test bench by hand, it can prove incredibly time-consuming. In this chapter, you will add features to the assembly that allow for the automatic assembly of every possible configuration of the model.

To do this, you will first create a “Design Space Assembly”. Then, in the next chapter, you will learn how to create a test bench for gathering data.

Chapter Overview

1. Convert your assembly into a Design Space assembly.
2. Try adding components manually.
3. Try adding components automatically using CLM_Light.
4. Add constraints to the assembly.

Procedure

Step 1: Getting Started

The first step in creating a Design Space Assembly is to **open the “MyMassSpringDamper” assembly.**

Once you have opened the assembly and it appears in the Editing Area, make sure no component in the Editing Area is selected.

After this, **click on the "Design Space Refactorer" button** shown in Figure 5.1:

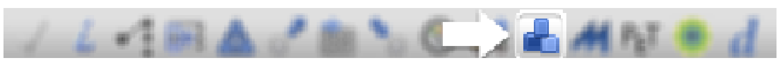


Figure 5.1: Design Space Refactorer button on the CyPhy-META toolbar

After clicking on this button, a new folder should appear in the GME browser as shown in Figure 5.2:

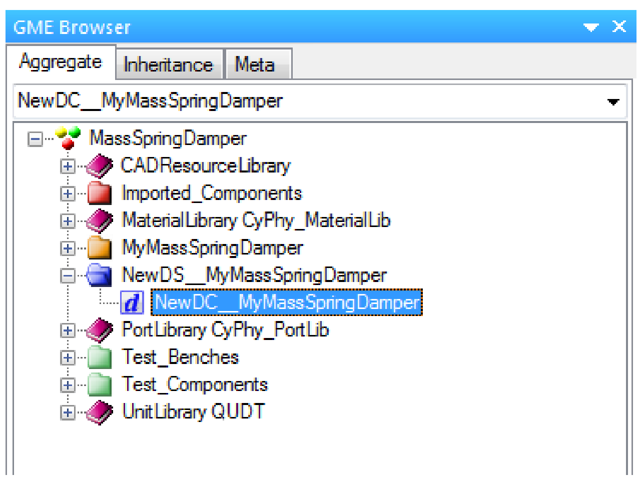


Figure 5.2: Design Space Folder and Icon

Double-click the design space named “NewDC__MyMassSpringDamper” in the “NewDS__MyMassSpringDamper” folder to open it.

Within, there should now be an assembly that looks similar to the image in Figure 5.3.

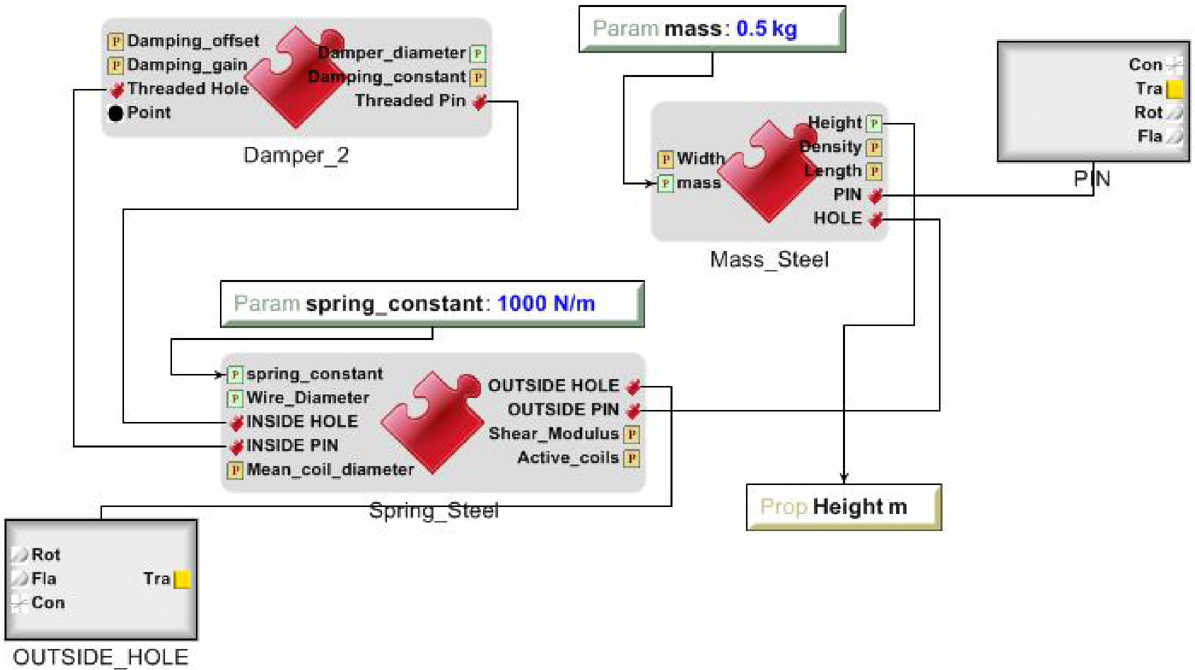


Figure 5.3: Initial Design Space for NewDC__MyMassSpringDamper

Step 2: Creating a Design Container for Components

Now with the "NewDC__MyMassSpringDamper" design space open in the Edit Area, **click on the Damper_2 component icon in the GME Browser, ensuring that Edit Mode (CTRL+1) is on.**

After this, **click on the "Design Space Refactorer" button again.** After clicking on that button, a pop-up window will appear that looks like the image in Figure 5.4:

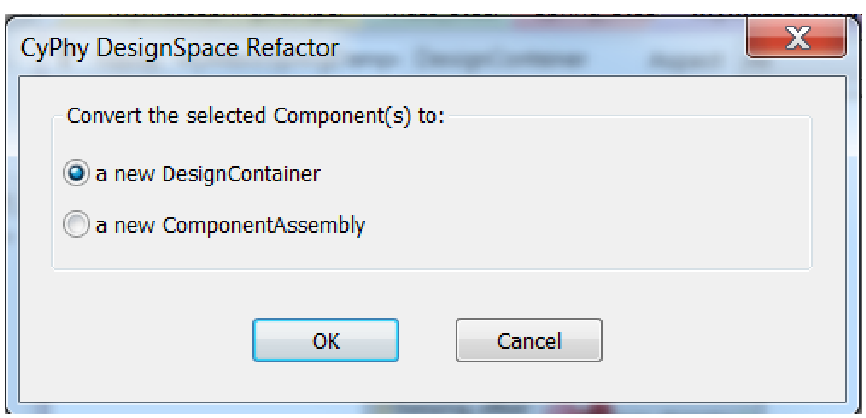


Figure 5.4: Desert Configuration to Assembly Pop-Up

Select the option to convert the selected component to **"a new DesignContainer"** in this screen and **click OK**. Now the icon for the damper component should look like Figure 5.5.

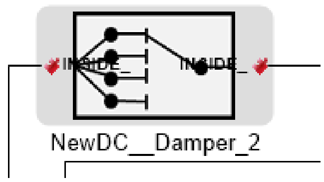


Figure 5.5: Design Container Icon

A Design Container was just created. **Double click on the Design Container.** The Editing Area should now look like Figure 5.6:

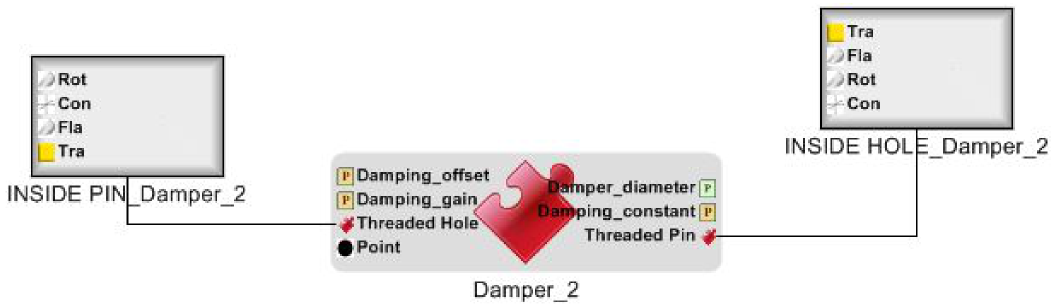


Figure 5.6: Editing Design Container

Step 3: Adding Components to Design Container - Manually

Alternative components can be added so that the Editing Area contains all 3 of the different dampers. The components should be located in the "Imported Components" folder in the GME browser.

Locate and copy the Damper_6 component. Then, right click on the Editing Area and choose Paste Special > As a Reference.

Repeat this process for Damper_10. The Editing Area should now look like Figure 5.7:

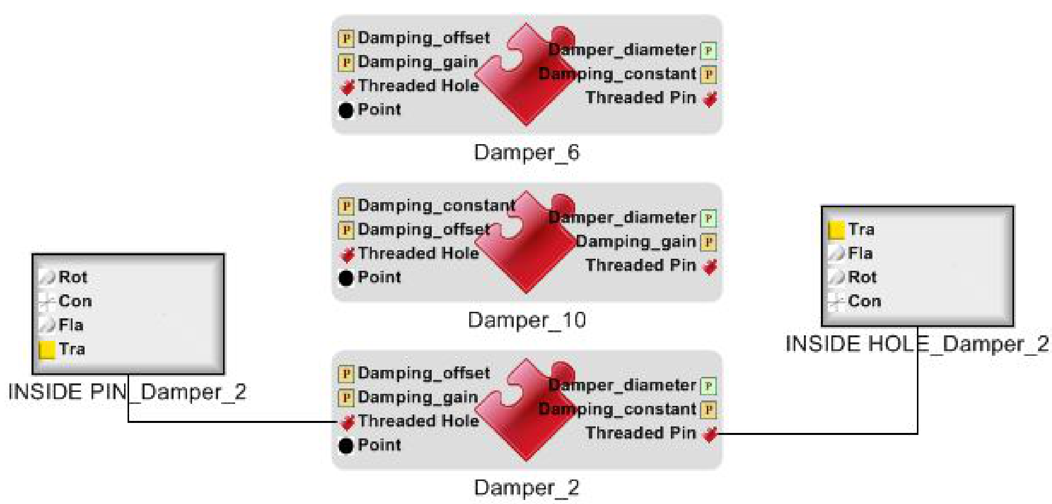


Figure 5.7: Adding Alternate Damper Components

Now that all the alternative components are in the Editing Area, it’s time to connect them like the original component. Connections are made in the same way as explained in the assembly tutorial.

Make sure that connect mode is enabled (select Connect Mode in the Mode bar or press Ctrl+2). After doing this, **click on two objects that need to be connected**, such as the **Threaded_Pin** port on **Damper_6** and the **INSIDE HOLE_Damper_2**, and connect them.

Make sure the alternative components are connected exactly as the original component is connected. When finished, it should look like the image in Figure 5.8.

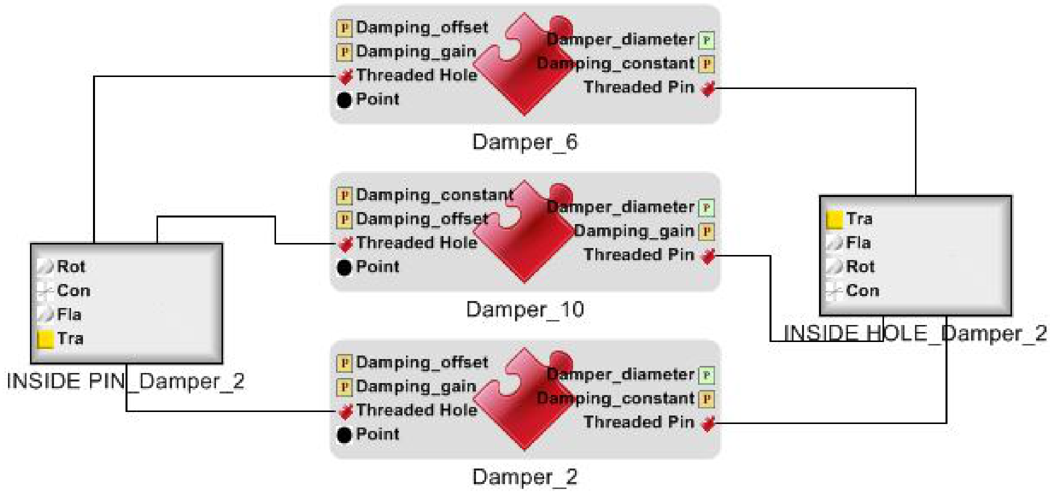


Figure 5.8: Alternate Damper Components Connected

Step 4: Adding Components to Design Container - Automatically

The **CLM_light** tool can be used to quickly add components downloaded from the component library to the design container. This tool is not necessary if you connect the components manually, but saves a lot of work and minimizes the chance of errors due to missing connections.

To use this tool, **return to the design space by clicking the NewDC_MyMassSpringDamper tab, then select Mass_Steel and create a design container as explained in Step 2 above. Then, double click on the design container.** The editor should look similar to Figure 5.9.

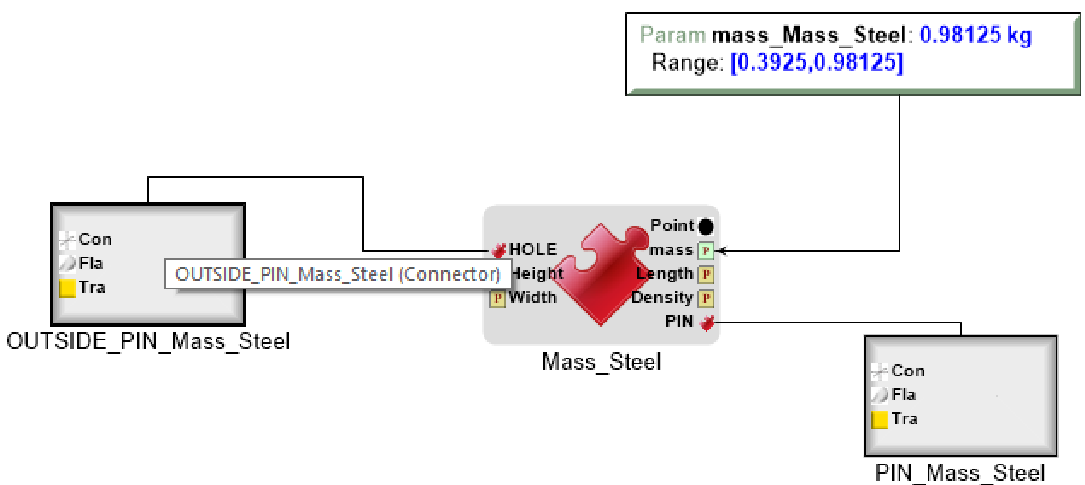


Figure 5.9: Inside Mass Design Container with One Mass Reference

Now **select the Mass_Steel component reference and click on the CLM_light tool** (Figure 5.10) from the tool bar. This opens a menu with all the components in the project that are available to be connected.

Figure 5.10: CLM_light tool Interface

Now, to select which components you wish to add to the design space, **click on the button/box to the upper left, just beside the component's name** (do not select the Name, RootNodeKind, or Path fields).

Both Mass_Gold and Mass_Titanium are used as an example in Figure 5.11. You can use this tool to add multiple components at once by clicking the top left corner to select all or by holding Ctrl and clicking on the rows that correspond to the components you want. It does not make a difference if Mass_Steel is also checked in the Component Selection box.

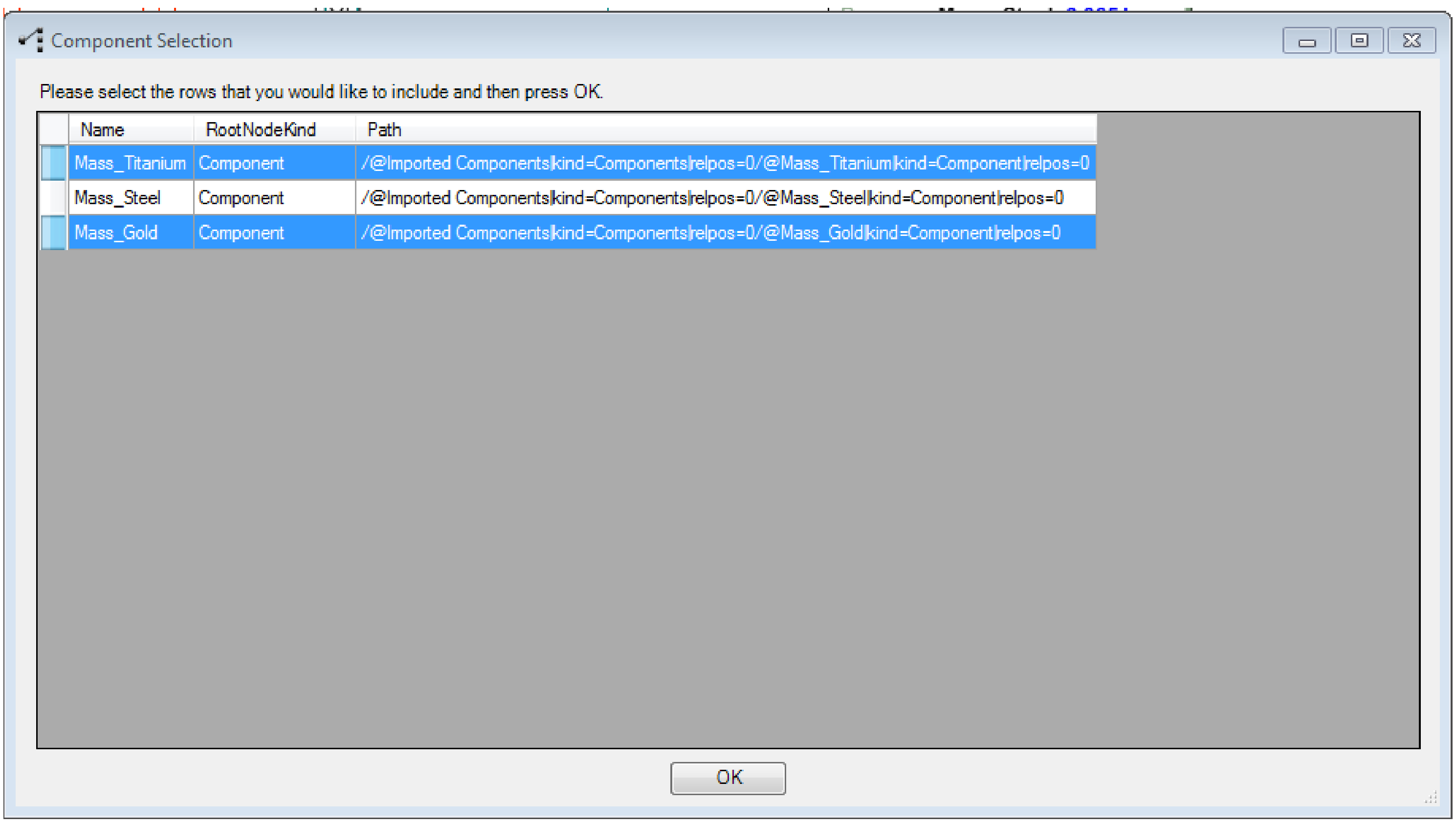


Figure 5.11: Selecting a component to enter using the CLM_light tool.

Click "OK" and any selected components that were not present in the design container before will be added to the design space as seen in Figure 5.12.

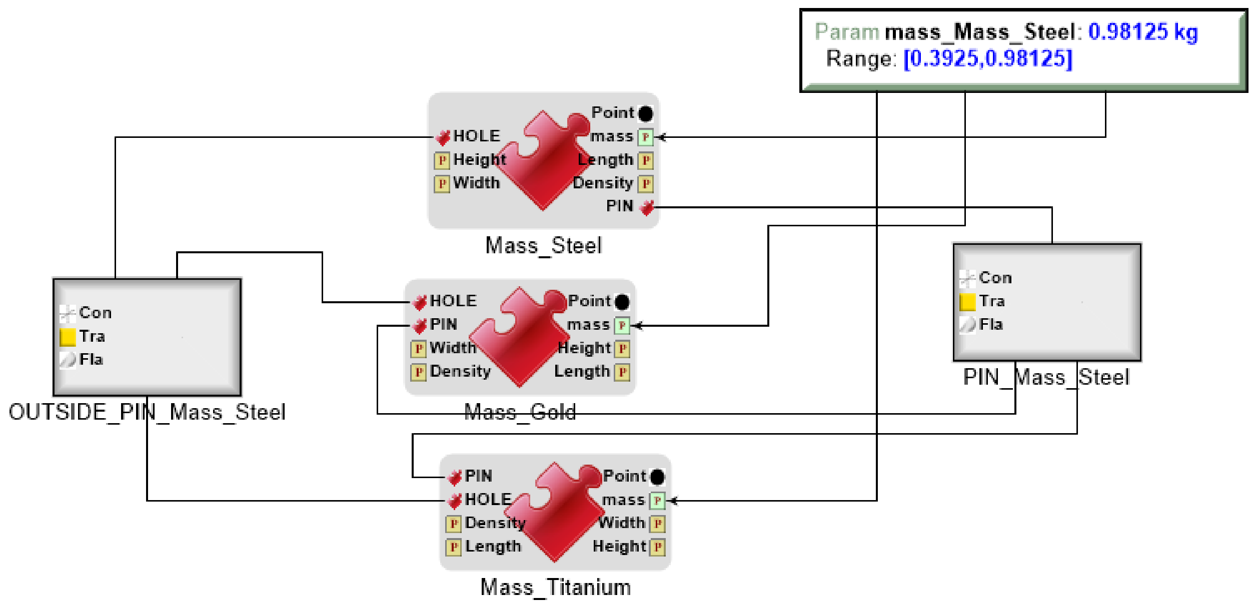


Figure 5.12: Mass_Gold and Mass_Titanium Added to the Design Container

Now create a design container using Spring_Steel and use either the manual or automatic method for adding components to populate the design container.

Now that each type of component has its own Design Container, “NewDC__MyMassSpringDamper” in the design space folder should resemble Figure 5.13:

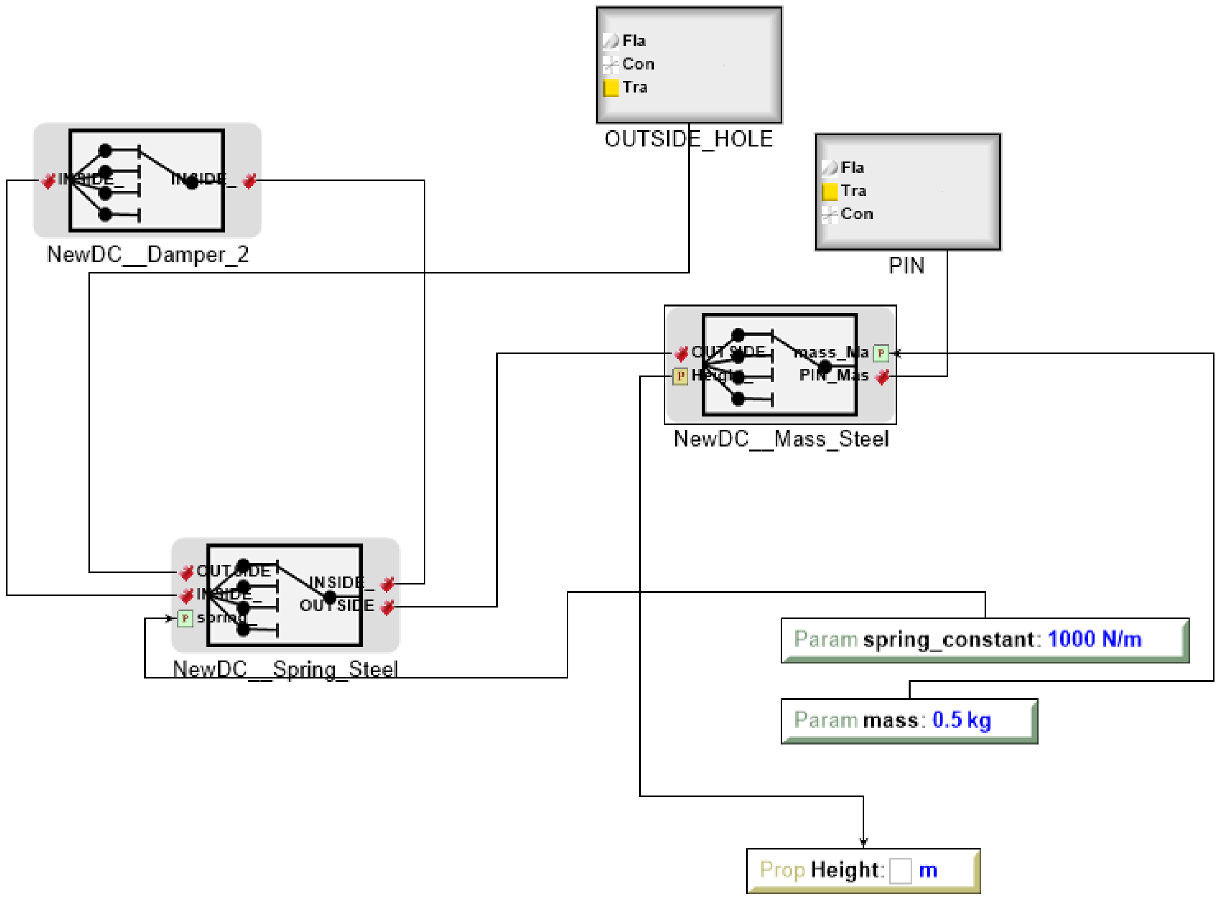


Figure 5.13: Design Space for NewDC__MyMassSpringDamper

Return to the design space, and change the names of the three design containers to "Mass," "Spring," and "Damper." The names reflect how these components are now design containers; each container contains three separate components instead of being the individual Damper_2, Mass_Steel, and Spring_Steel components.

The final Design Space assembly should look something like Figure 5.14:

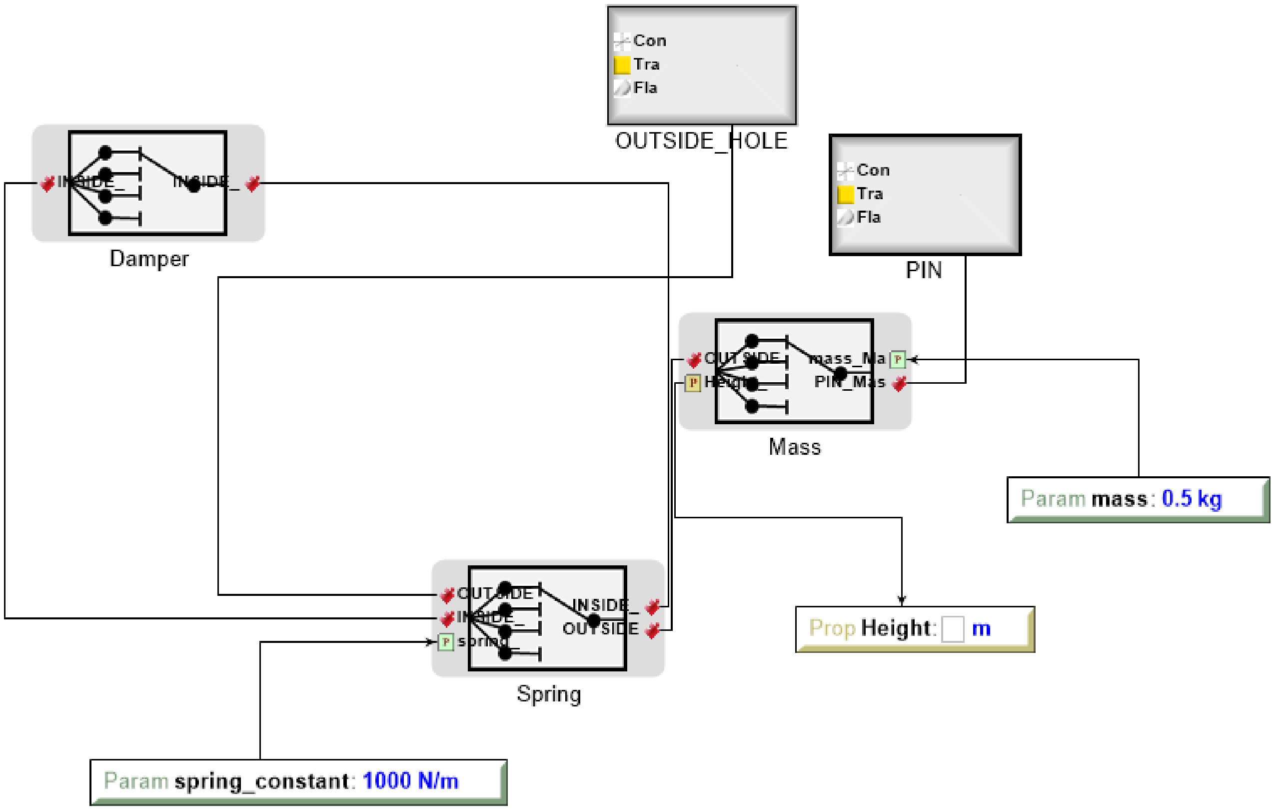


Figure 5.14: All Alternate Components

Now you can generate every configuration possible with the alternate components. In this case, the 3 different dampers, 3 masses, and 3 springs combine to form 27 unique configurations.

Step 5: Constraints

There are multiple uses of constraints, but their main use is to limit the options available and thus reduce the number of possible configurations. This limitation is necessary because some components are compatible only with certain components.

For example, installing a special type of shocks in a car's suspension may limit the number of different A-arms that can be used in the suspension. This situation represents a constraint that has been put on the design space.

5.1: Adding Visual Constraint

To add a constraint, **open your Design Space assembly and switch to the DesignSpaceAspect view of the Mass Spring Damper model (click the DesignSpace tab in the Parts Browser).** In the Part Browser, **find the VisualConstraint icon (Figure 5.15) and drag it into the workspace.**

Figure 5.15: VisualConstraint

For this exercise, we will assume that the Mass_Steel and Spring_Steel are only compatible with each other. Therefore, they must be "connected" in the VisualConstraint so that when using the Design Space Exploration Tool (Chapter 7), the Mass_Steel and Spring_Steel components can be used together but not with other alternatives.

To do this, **go into the Mass *design container* (not the GME Browser) created in the previous step, and copy the Mass_Steel component reference. Next, click the NewDC__MyMassSpringDamper tab. Then, move inside the VisualConstraint by double-clicking the icon in the**

Editing Area, and paste the Mass_Steel as a reference.

Add the Spring_Steel component the same way.

Finally, **connect them using the Connect Tool by clicking the middle of the components instead of any particular ports. Connect the mass and the spring in both directions.**

When this step is finished, the final model should look like Figure 5.16.

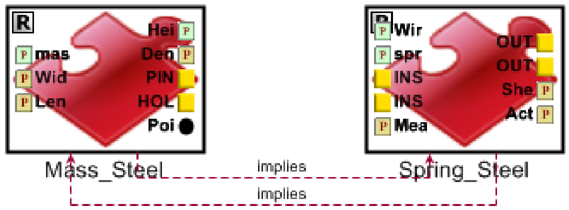


Figure 5.16: Inside VisualConstraint Icon

5.2: Property Constraints

Return to the NewDC__MyMassSpringDamper design space by clicking the NewDC__MyMassSpringDamper tab and then go into the design space aspect view. Drag in a property constraint from the Parts Browser.

Rename this object "Maximum_Height_Constraint". This will be used to limit the maximum height of the mass. It should look like Figure 5.17.

Figure 5.17: PropertyConstraint

Now go into connection mode and **connect the Maximum_Height_Constraint to the Height property** as shown in Figure 5.18. After making the connection, **return to Edit mode.**

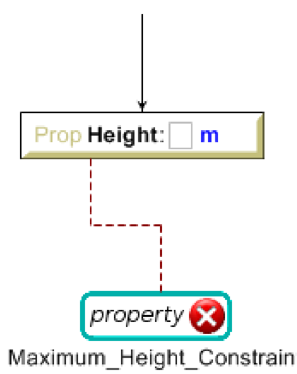


Figure 5.18: Connecting Maximum_Height_Constraint to Height Property

Select the Maximum_Height_Constraint by clicking on it once. In the Object Inspector, under the Attributes tab, **change the TargetValue to 0.04 and the TargetType to “Must Be Less Than”** as shown in Figure 5.19.

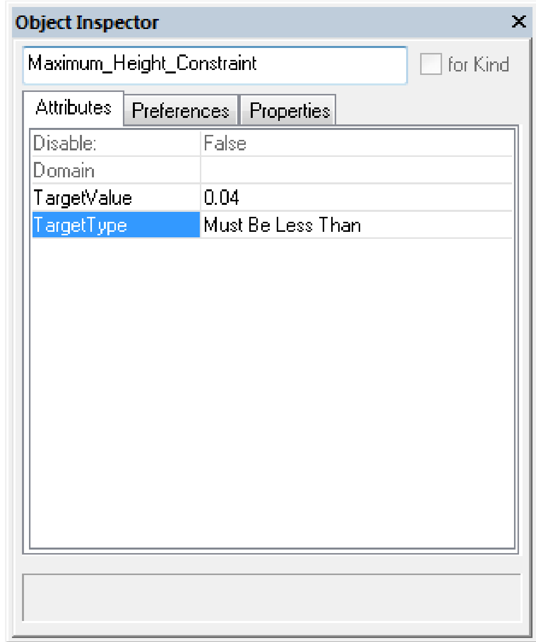


Figure 5.19: Editing the Constraint

You will need to **add one addtional property constraint called Minimum_Height_Constraint. This constraint needs to be set to "Must Exceed" for the Target Type and the TargetValue should be 0.02.** Figure 5.20 shows these settings below. Connect the Minimum_Height_Constraint to the Height property.

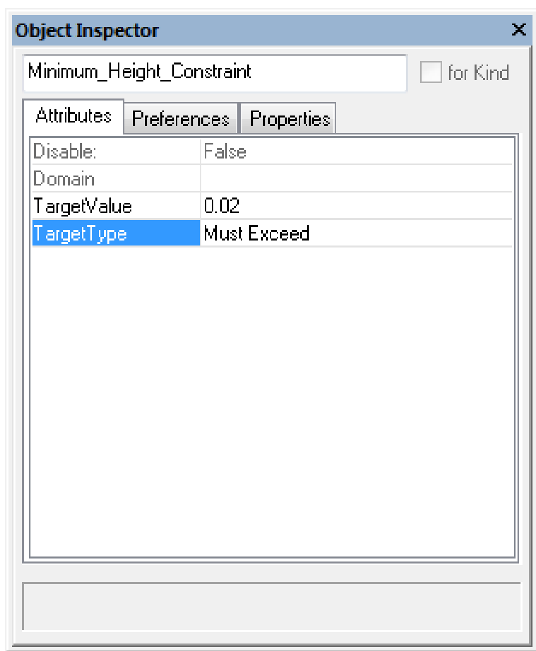


Figure 5.20: Editing Minimum_Height_Constraint

These constraints are now set up and are able to be applied to the system when DESERT is run in Chapter 7.

Your workspace should now resemble Figure 5.21:

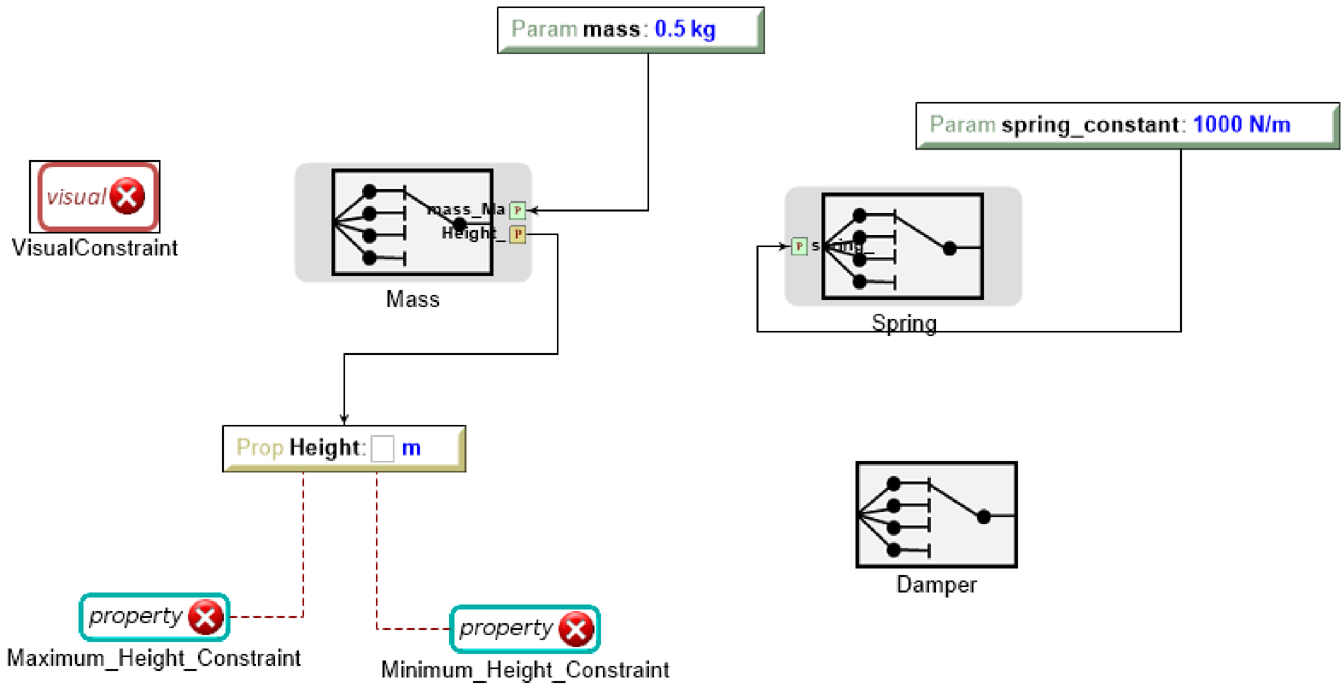


Figure 5.21: Final Workspace Image

◄ **Previous: Chapter 4: Creating a Component Assembly** | **Next: Chapter 6: Creating a Test Bench** ►

◄ **Previous: Chapter 5: Creating a Design Space Assembly** | **Next: Chapter 7: The Design Space Exploration Tool (DESERT)** ►

Chapter 6: Creating a Test Bench

Introduction

At this point, you have been introduced to the creation of a Design Space component assembly. This section will outline how to create a test bench to gather data from this model. This data will be analyzed further using the tools discussed in chapter 9 and and the scoring function discussed in chapter 10 to determine which design best meets our requirements. There are two key types of test benches: Dynamics Test Benches and CAD Test Benches.

Chapter Overview

1. Create a folder to house the test bench.
2. Create a test bench in the folder.
3. Add your assembly to the test bench.
4. Add other test components.
5. Add parameter, property, and metric components.
6. Connect the test components to your assembly.

7. Add SolverSettings.
8. Add Workflow Definition
9. Add a Post-Processing component.

Procedure

Step 1: Creating a Test Bench Folder

Before you get started creating test benches, you will need to create a folder to store the test benches in. This step will walk you through that procedure.

A testing folder can be created by **scrolling over the project name, MassSpringDamper, in the GME browser window, right clicking, and following the path Insert Folder > Testing**, as seen in Figure 6.1. This creates a folder type which can support test objects, such as the test bench itself.

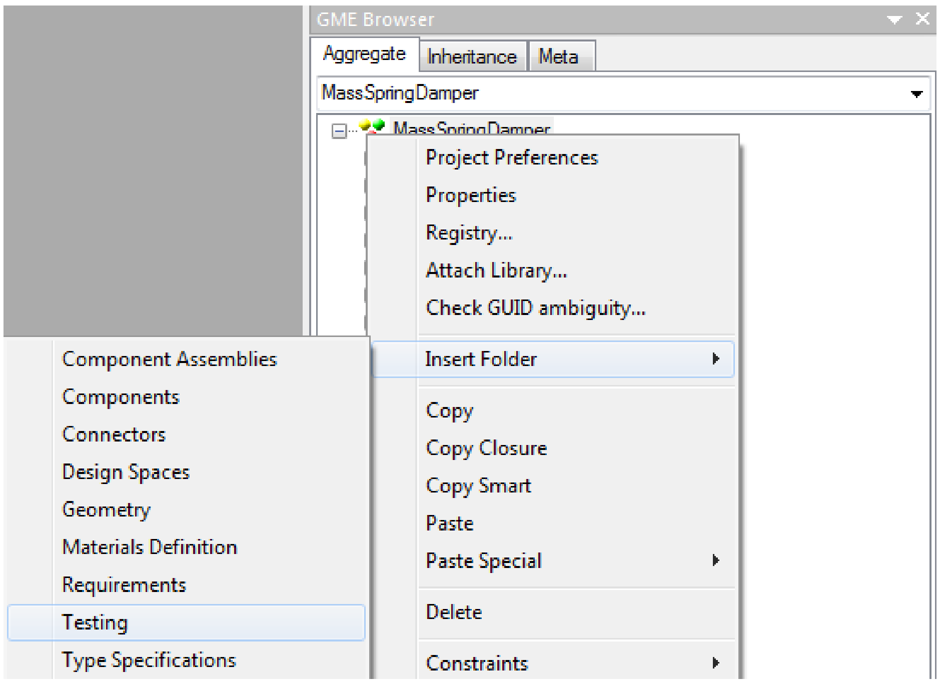


Figure 6.1: Inserting a Testing Folder

Name the folder "MyTestBenches".

Step 2: Insert Test Bench

Ensure that edit mode is enabled (Ctrl+1) and then select and **right-click the name of the Testing Folder** that was just created.

Similar to the steps for inserting an assembly, inserting a test bench into GME requires right clicking on the newly created testing folder and following the extension **Insert Model > Test Bench**.

This creates a test bench object which will be able to hold the system under test and the parts needed to interact with it.

This model can be renamed by **highlighting, and then specifically selecting, the title with the left mouse button**. In this case, name the model "**System_Dynamics_OM**".

Figure 6.2 shows where to access the test-bench insert function on the right-button menu.

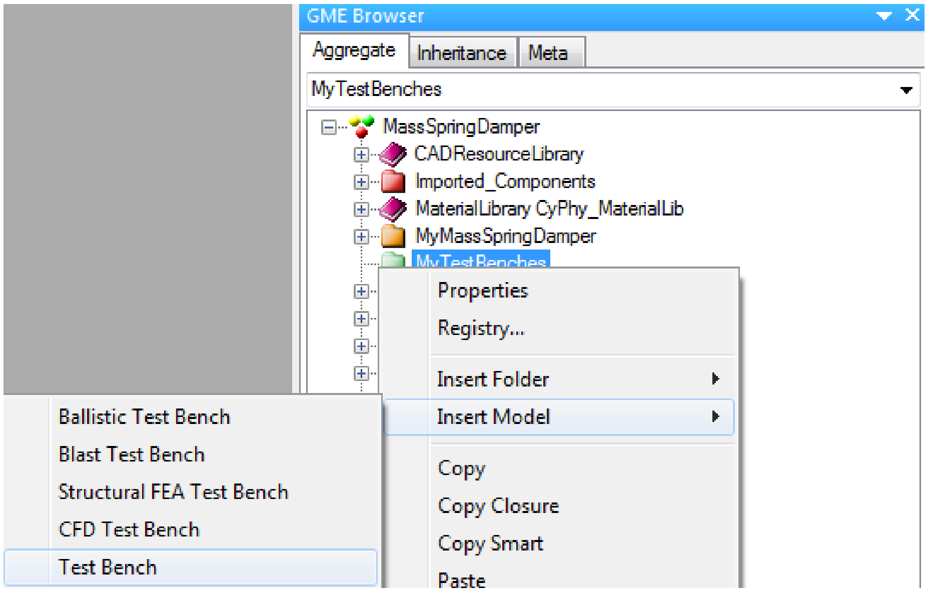


Figure 6.2: Inserting a Test Bench Model

To open the test bench, **double click on the newly created System_Dynamics_OM test bench in the GME browser**. This will open up the initially blank test bench.

Step 3: Adding a System under Test

3.1: Copy the Design Space Assembly

The test bench will be designed around the Design Space assembly, which will serve as our object under test - the principle assembly from which data will be gathered.

Go to the GME Browser window and, as shown in Figure 6.3, **copy the Design Space Assembly** that you created in the previous section. **Be sure that the object, not the folder, is selected** before moving on to the next step.

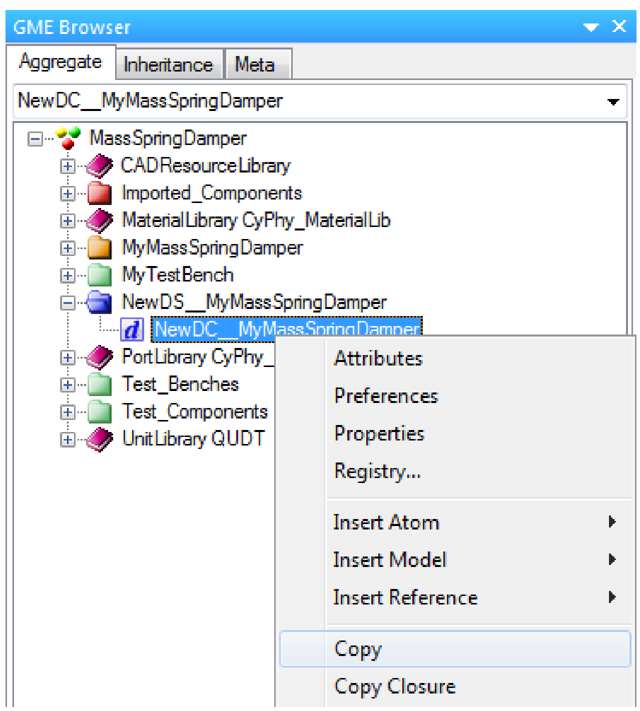


Figure 6.3: Copying MyMassSpringDamper

Step 3.2: Paste the Assembly in the Test Bench

Now, **right click on the workspace of the test bench** and follow the extension **Paste Special > As Reference** as shown in Figure 6.4. This will cause a “**Select Reference Role Type**” menu to pop up asking what type of object is being pasted.

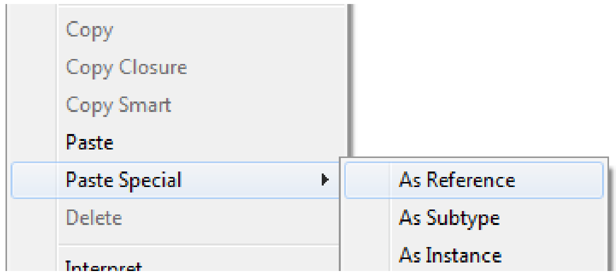


Figure 6.4: Pasting Design Space as Reference

Choose “**TopLevelSystemUnderTest**” and click OK, as seen in Figure 6.5.

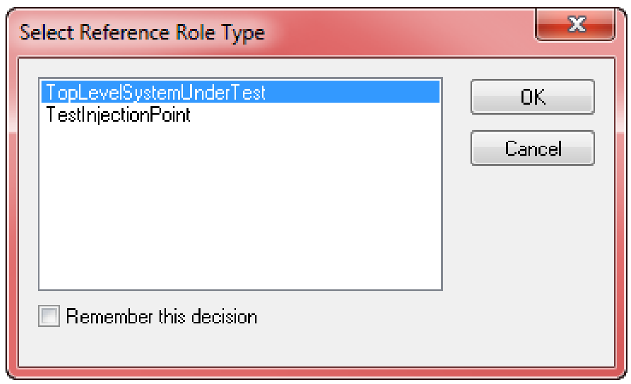


Figure 6.5: Top Level System Under Test Pop-Up

You have now added the design space for the Mass Spring Damper system to the test bench as the top level system under test. All that remains now is to populate the test bench with the other components needed for testing.

Step 4: Adding Test Components

Adding test components is very similar to the process of adding the assembly in Step 3.

To find test components, simply **expand the test components folder, expand the "Test Components" folder on the next level**, and **select the AccelSensor test component to insert** as shown in Figure 6.6.

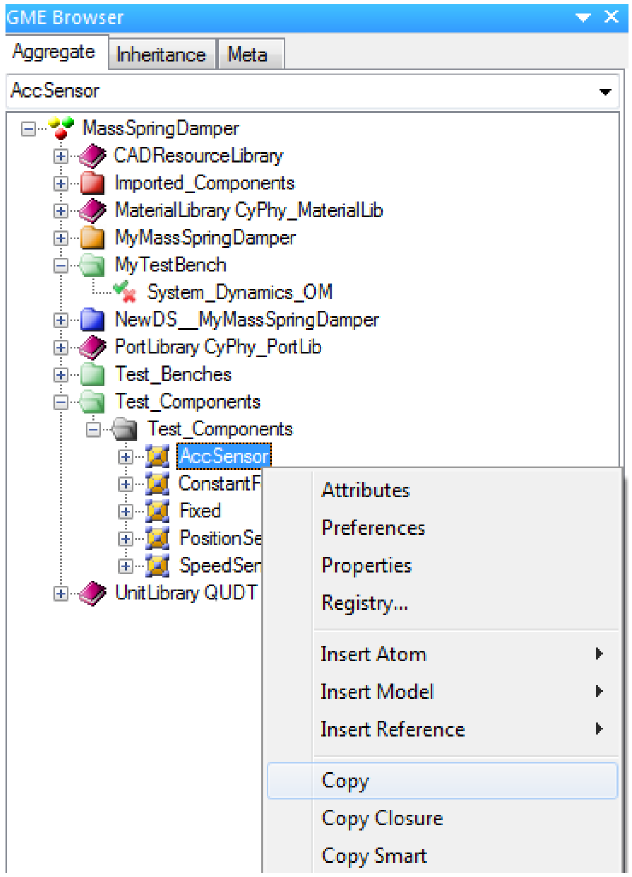


Figure 6.6: Selecting a Test Component

First **copy the AccelSensor test component**, and then **right-click on the workspace of the test bench** and follow the extension **Paste Special > As an Instance (NOT as Reference)**. This time, choose "**Test Component**". **DO NOT simply drag and drop.**

Then, **repeat the process** for the following test components:

- PositionSensor
- SpeedSensor
- Fixed
- ConstantForce

Once you have pasted them all, the screen should look like Figure 6.7.

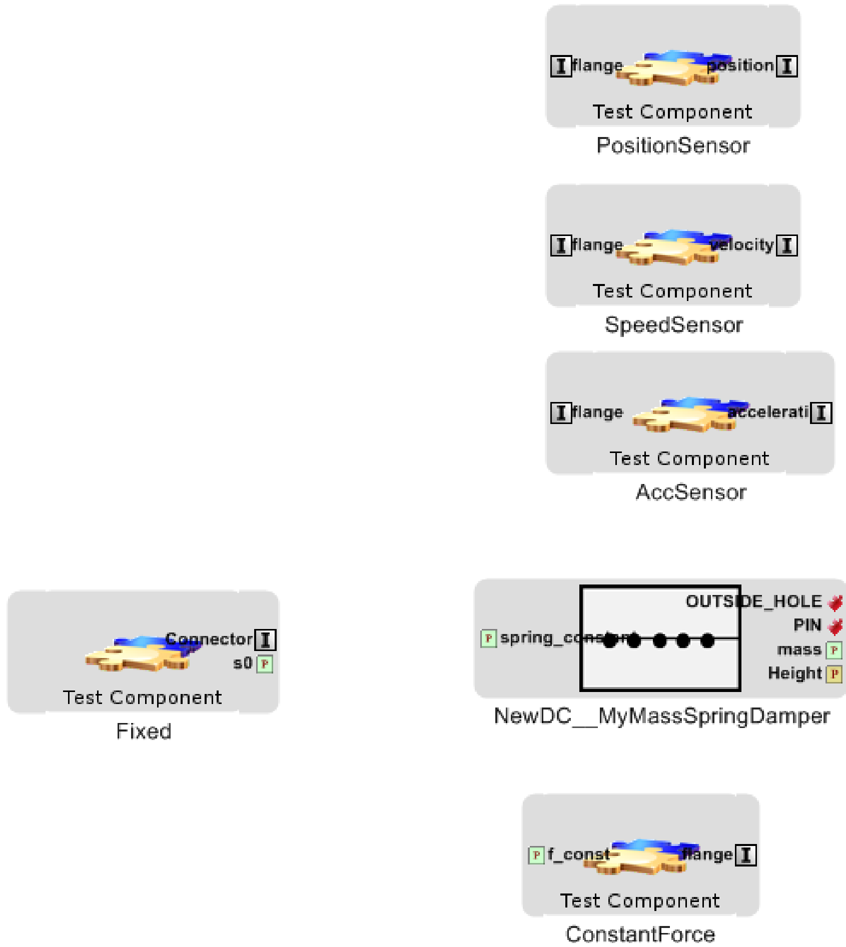


Figure 6.7: System_Dynamics_OM Test Bench with Design Space and Test Components

Step 5: Adding Metrics Blocks

In this step, you will add a metric to the test bench. Metrics are are outputs from your analysis that you wish to compare across designs. An example of a metric in automotive design is the maximum speed of the vehicle.

Since this is a test bench **the only inputs will be parameters and properties, while the only outputs are metrics** - the rest of the system is there to feed information into these metric and parameter ports. For this test bench, no inputs are expected from other test benches so there will only need to be metrics added to the test bench.

While in the **"all" view** in the Part Browser, the metric part is about halfway down the menu, as shown in Figure 6.8.

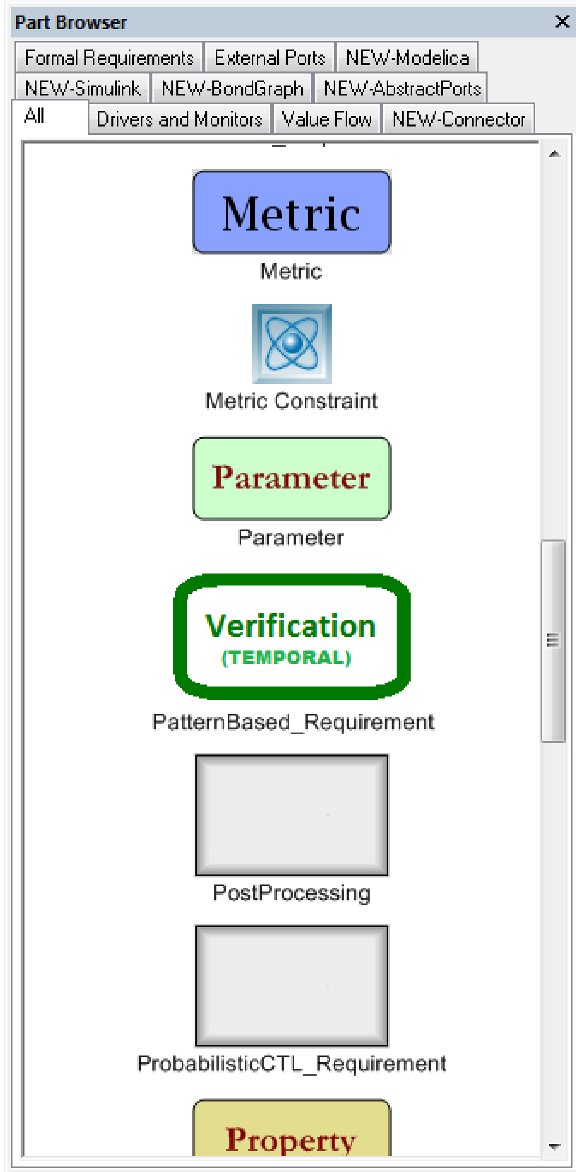


Figure 6.8: Adding Metrics to Test Benches

We will require **three metric parts. Drag them into your workspace**.

Change the names of these metrics to "**Time_to_Zero**", "**Stopped_Moving**", and "**Final_Position**". These three metrics will remain unconnected.

Add the unit "seconds" to Time_to_Zero metric and "meters" to Final_Position metric. When you have added the metrics, the test bench should look similar to Figure 6.9.

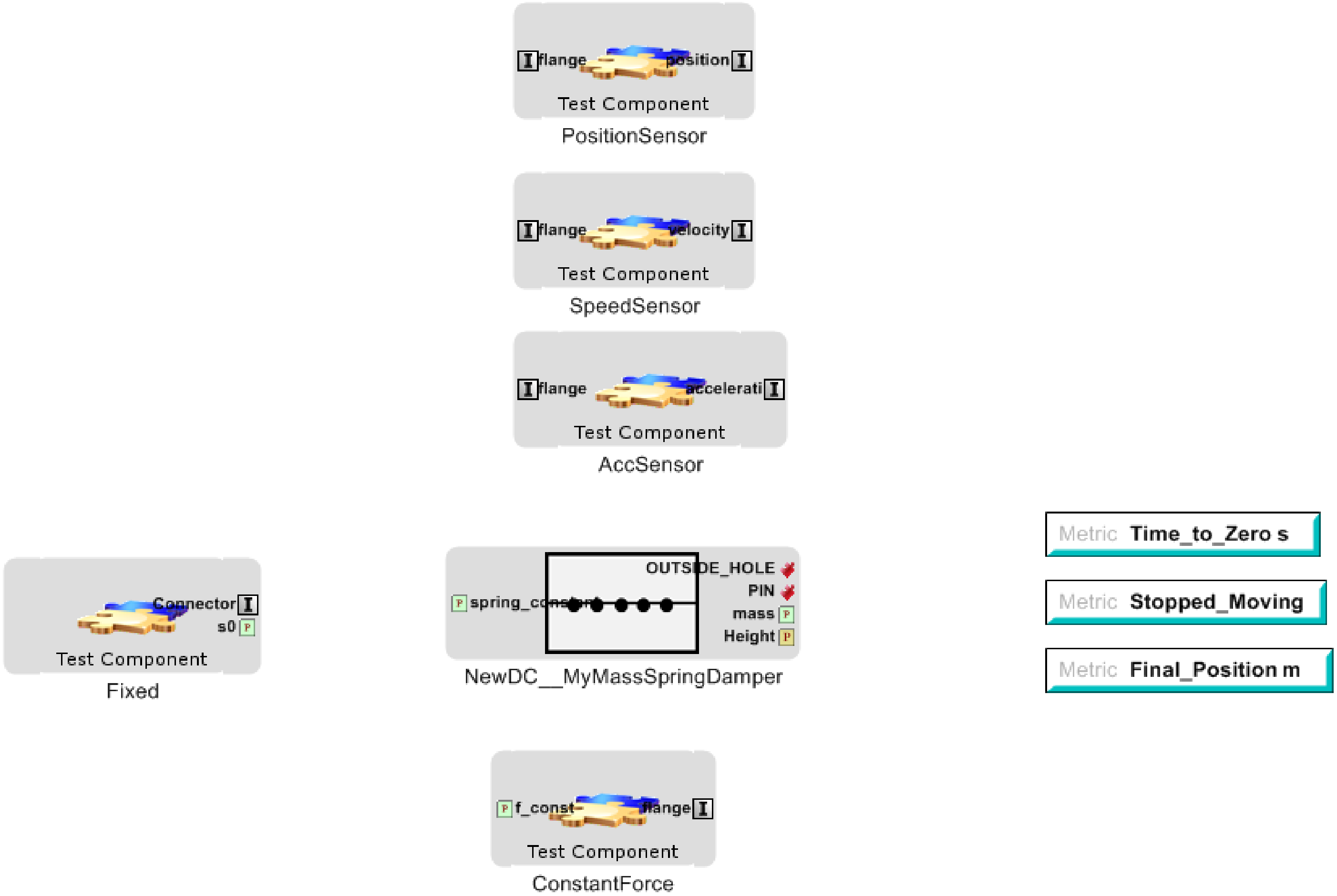


Figure 6.9: System_Dynamics_OM Test Bench with Metric

Step 6: Drawing the Connections

Now that the system under test, the test components and metrics parts have been gathered, it is time to connect them into one functioning test bench. The fixed component indicates that two objects are fixed in relation to each other at that location and a moving frame, which in this system describes the damper and spring components.

Connecting the parts of a test bench is just like connecting the parts of the assembly you made earlier.

Simply **switch from edit to connect mode (Ctrl+2)** and **draw the connections between the correct ports to complete the test bench**, which when finished should look similar to Figure 6.10.

The flange of the AccSensor, ConstantForce, PositionSensor, and SpeedSensor test components connects to PIN of MyMassSpringDamper. This is where we are applying the constant load of 2N (the default for the ConstantForce block) to the mass component. Also we are taking the readings of position, speed and acceleration from this flange.

The flange of the Fixed test component connects to the OUTSIDE_HOLE of MyMassSpringDamper, which indicates that two objects are fixed in relation to each other at that location and a moving frame, which in this system describes the damper and spring components.

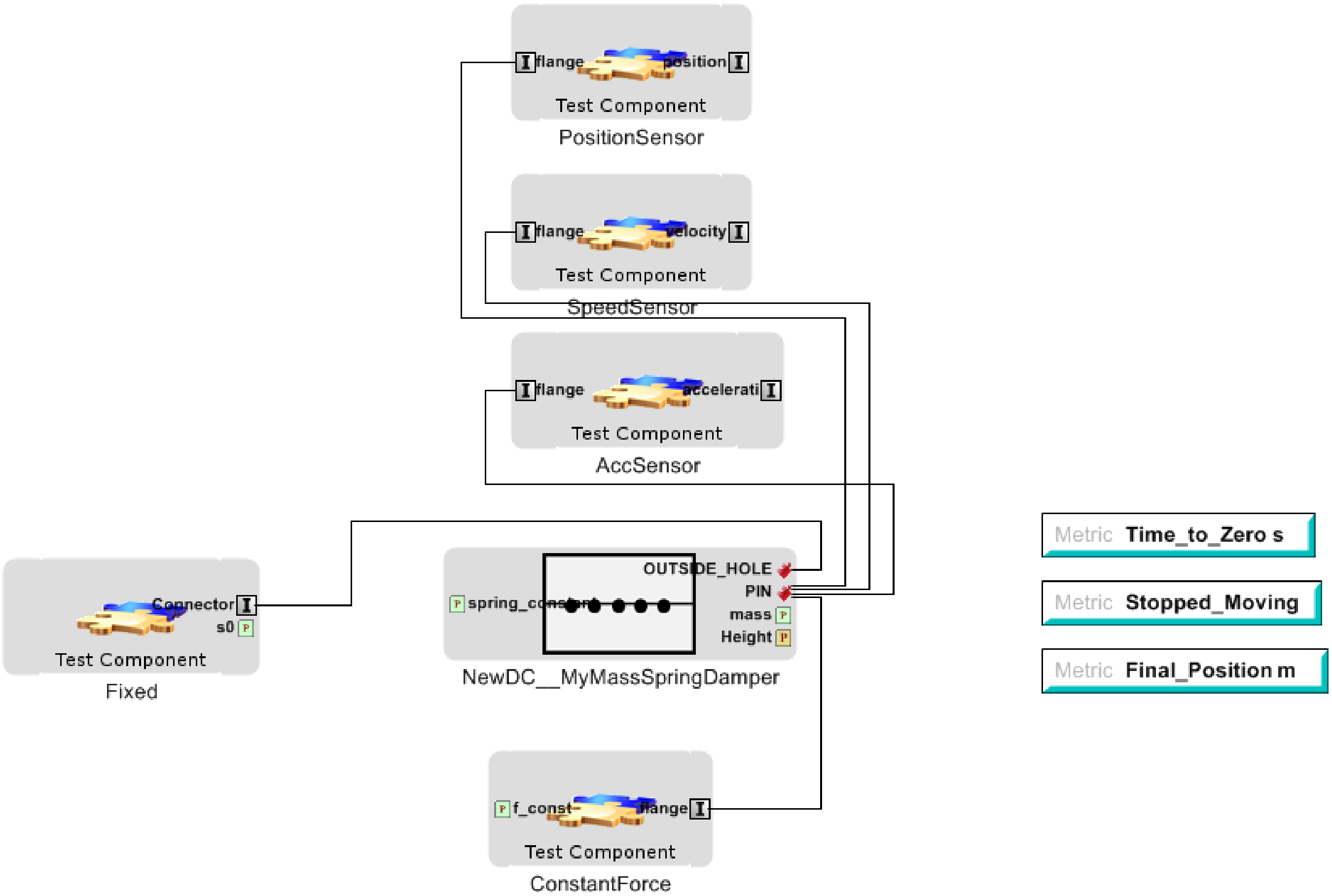


Figure 6.10: Connecting Test Components to Design Space

You've now provided all of the external factors needed to simulate a test of the system.

Step 7: Adding Test Settings

In order to have control over the simulation settings, the **SolverSettings icon**, seen below, should be added to the workspace. Simply drag and drop as before. Figure 6.11 shows what the solver settings icon should look like.

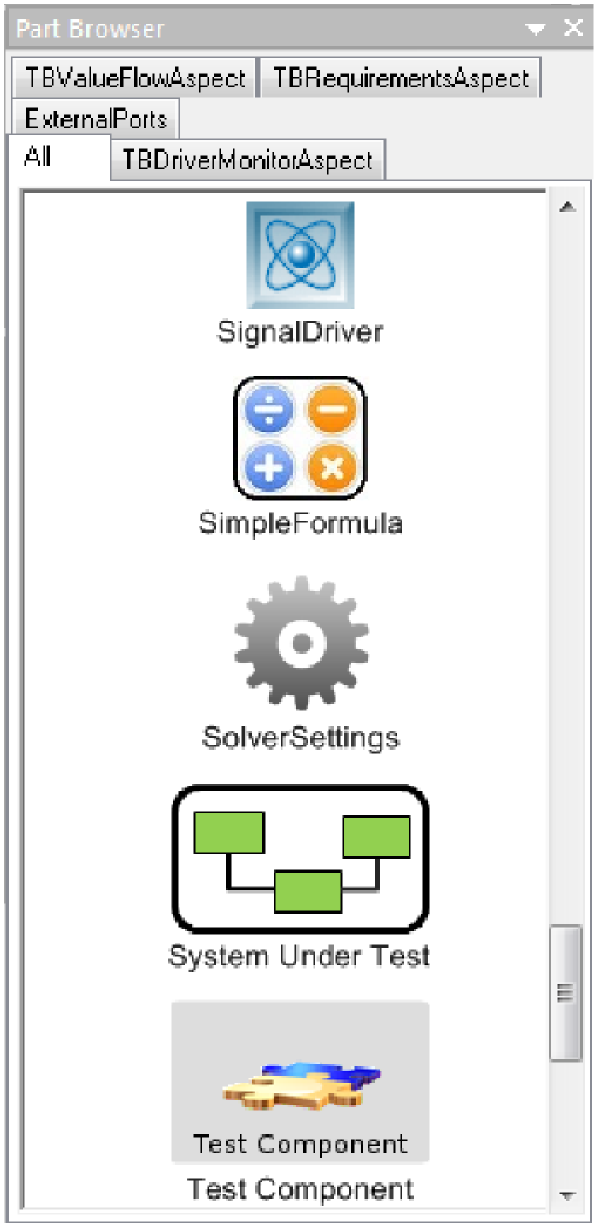


Figure 6.11: Adding Solver Settings

SolverSettings doesn't need to be connected to anything else – once you have added it to the workspace, all you have to do is **left click on the SolverSettings object and change its attributes in the Object Inspector browser** to reflect the testing duration, tolerance, and number of intervals you would like the test to run.

As defined, the test will run for 100 seconds. **Set the Tool Selection attribute to OpenModelica (latest).** Figure 6.12 shows what the attributes of the solver setting should look like.

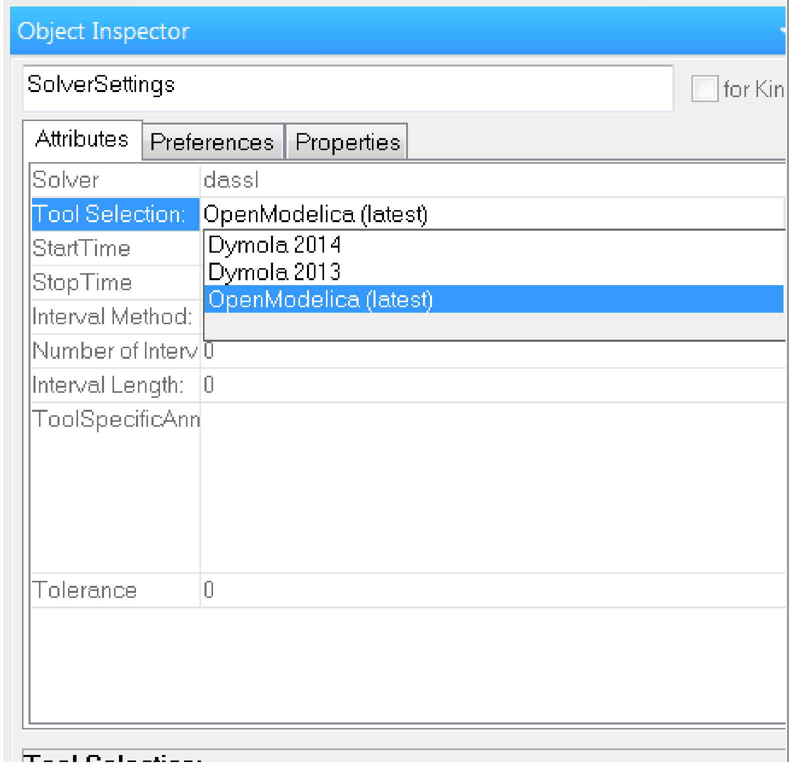


Figure 6.12: Solver Settings used for Test Bench

Open META-CyPhy is not a simulator and doesn't evaluate the test bench internally. Instead, it composes a simulation for a Modelica tool and exports the test to an external simulator process.

A workflow definition will tell Open META-CyPhy where to export the model.

Step 8: Creating a Workflow Definition

A workflow definition is an object that tells Open META-CyPhy's Master Interpreter (Chapter 8) which interpreter to use. The steps below will show how to implement the OpenModelica workflow definition.

These steps can be used for all other interpreters, such as the CyPhy2CAD interpreter.

Right click on the Test Benches folder that contains the **System_Dynamics_OM** and go to **Insert Folder > Workflow Definitions**. **Right click** on the new folder and select **Insert Model > Workflow**.

Name the new workflow Modelica.

The GME Browser Tree should look like Figure 6.13:

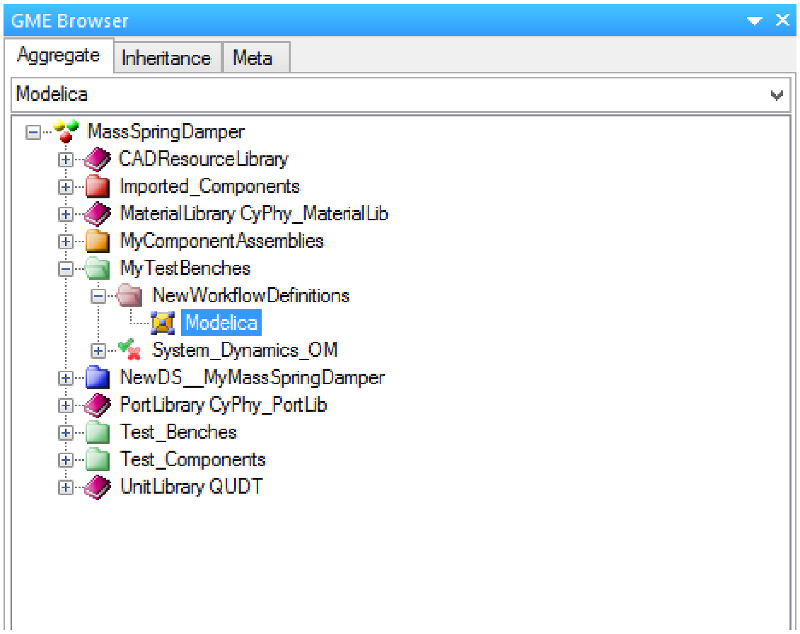


Figure 6.13: GME Browser

Double-click on the OpenModelica workflow. This will open a blank workspace with two objects in the Part browser. **Drag the Task object shown in Figure 6.14 into the workspace.**

Figure 6.14: Work flow Icon

The window shown in Figure 6.15 will then appear, prompting for which kind of interpreter to use. Choose "**CyPhy2Modelica_v2**".

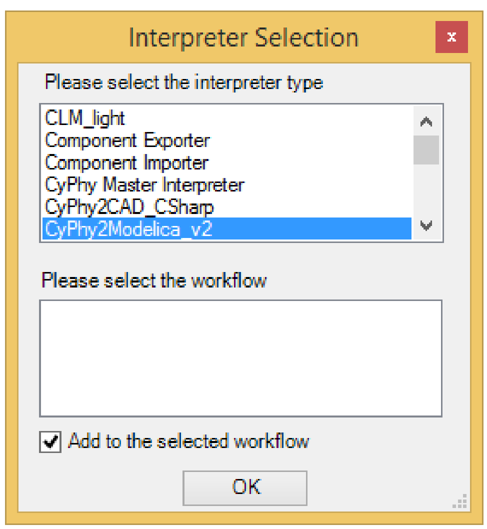


Figure 6.15: Interpreter Selection

Now return to the test bench. Next, **Copy the OpenModelica workflow** you created from the GME browser, then **right click and Paste Special > As Reference** in the workspace.

This will paste the workflow into the test bench. Now the test bench will run a Modelica simulation using OpenModelica when the Master Interpreter is used on it.

Step 9: Post Processing

It's possible to gather a great deal of information simply by running a simulation, but some data can only be obtained by processing data that is produced after a test is run.

To perform this function, python scripts are used to analyze the data gathered through the simulation. You can learn more about the python programming language here.

To allow post processing to occur, drag in a post processing block from the Part browser as shown in Figure 6.16.

Figure 6.16: Post Processing Block

Next, **left click the PostProcessing block**. In the object inspector, **change the ScriptPath to go to: Post_Processing/MSDPostProcessing.py** as shown in Figure 6.17. This assumes that the post processing folder provided is in the same folder as the .mga file generated when the .xme file was opened.

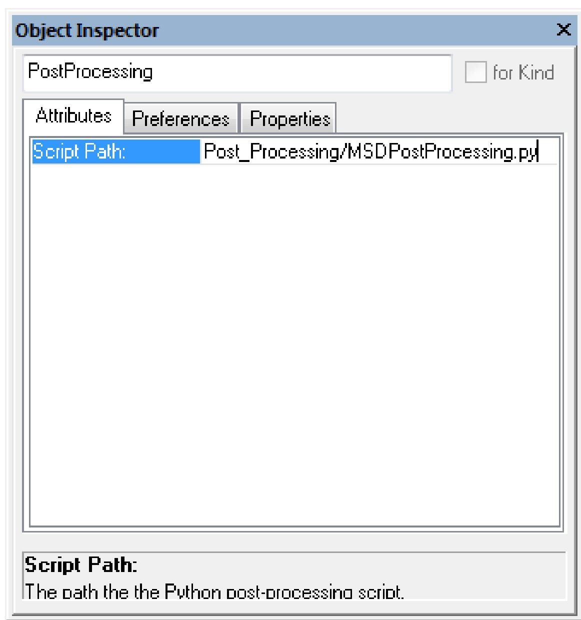


Figure 6.17: Settings for Post Processing

Figure 6.18 should look similar to your final workspace.

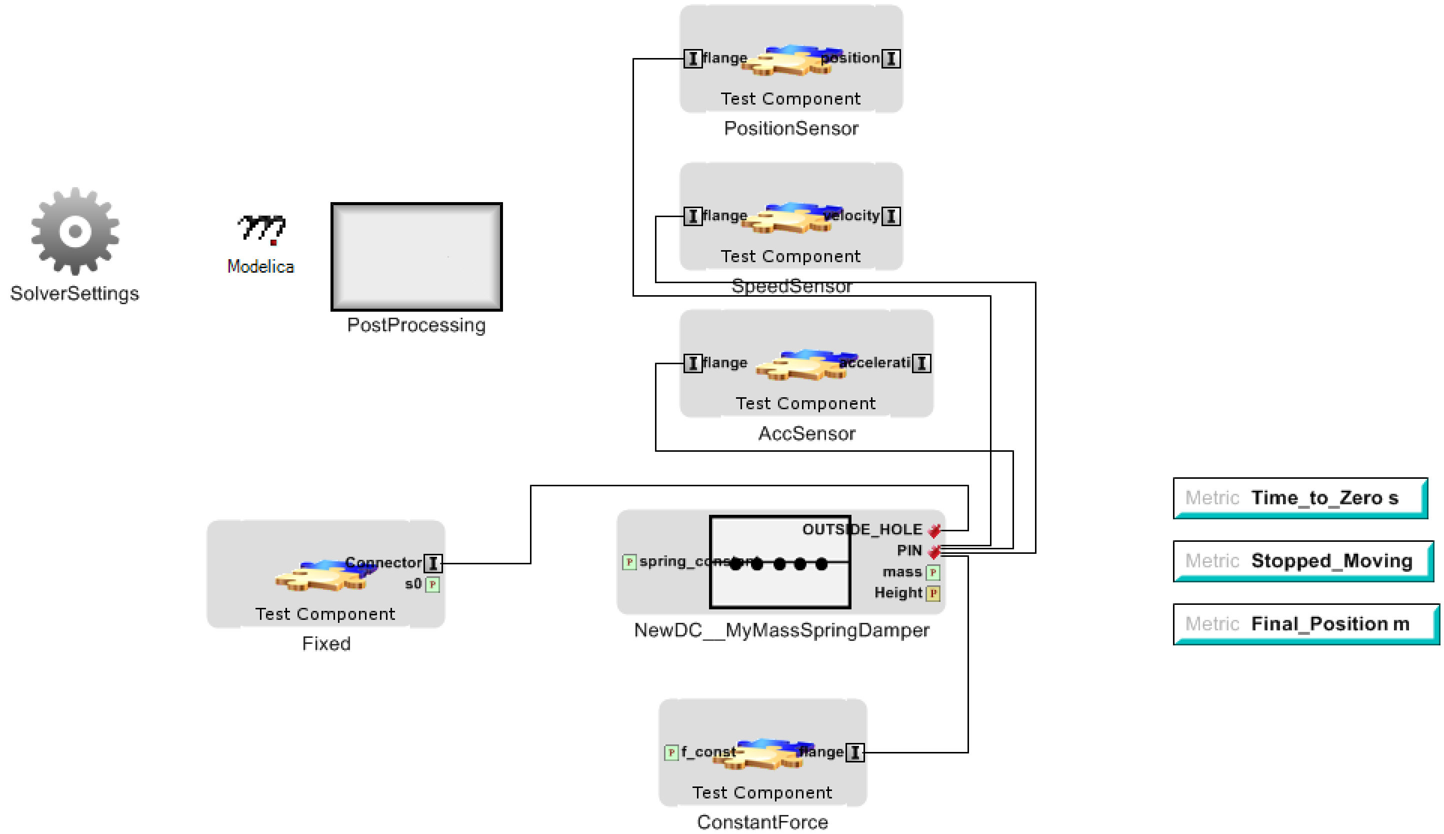


Figure 6.18: Final Workspace for Test Bench

◄ **Previous: Chapter 5: Creating a Design Space Assembly** | **Next: Chapter 7: The Design Space Exploration Tool (DESERT)** ►

◄ **Previous: Chapter 6: Creating a Test Bench** | **Next: Chapter 8: The Master Interpreter** ►

Chapter 7: The Design Space Exploration Tool (DESERT)

Introduction

This section of the tutorial explores the use of Design Space assemblies through the assembly made for the mass spring damper in the second section of this tutorial.

The document explains how to use the Design Space Exploration Tool (DESERT) and the Master Interpreter to run simulations on the “DesignSpace MyMassSpringDamper” model using alternate component configurations, which are best defined as different arrangements of parts.

Note: This section is incredibly important for making practical use of the Open META-CyPhy tools. DESERT and the Master Interpreter make it easy to compare results between designs that have different component configurations. For example, testing how “MyMassSpringDamper” behaves with a tungsten spring instead of a steel spring only takes a few steps using the DESERT tools.

Design Spaces

A design space is defined as the realm of design possibilities given a set of constraints. For example, installing special shocks in a car’s suspension limits the types of A-arms that can be used.

This situation represents a constraint that has been put on the design space. Open META-CyPhy uses the idea of design space to help the user develop the best possible design.

Procedure

Step 1: Running the DESERT Simulation

The DESERT simulator is a tool in GME that is able to assemble many different configurations for a given system in a very short amount of time.

To proceed, navigate to the toolbar at the top of the page and **click on the DESERT icon** as shown in Figure 7.1.

Figure 7.1: DESERT Icon

The following window, the "Top Design Space Selector" will pop up as shown in Figure 7.2.

Click on "**NewDC_MyMassSpringDamper (Compound)**" to run DESERT on the full ("Compound") component assembly (**NOT one of its individual components**). Click OK.

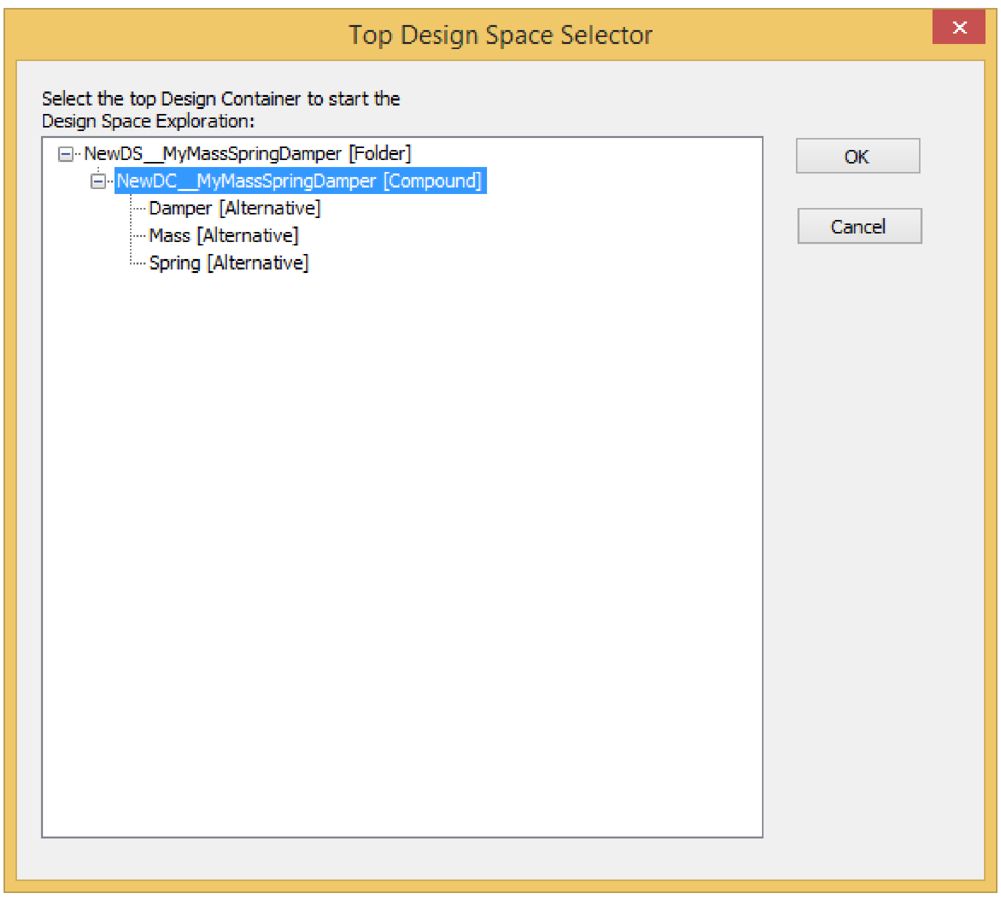


Figure 7.2: Design Space Selection

This causes the Open META-CyPhy DESERT Tool window to appear, as seen in Figure 7.3. As you can see, this window contains constraints which you can apply to the system.

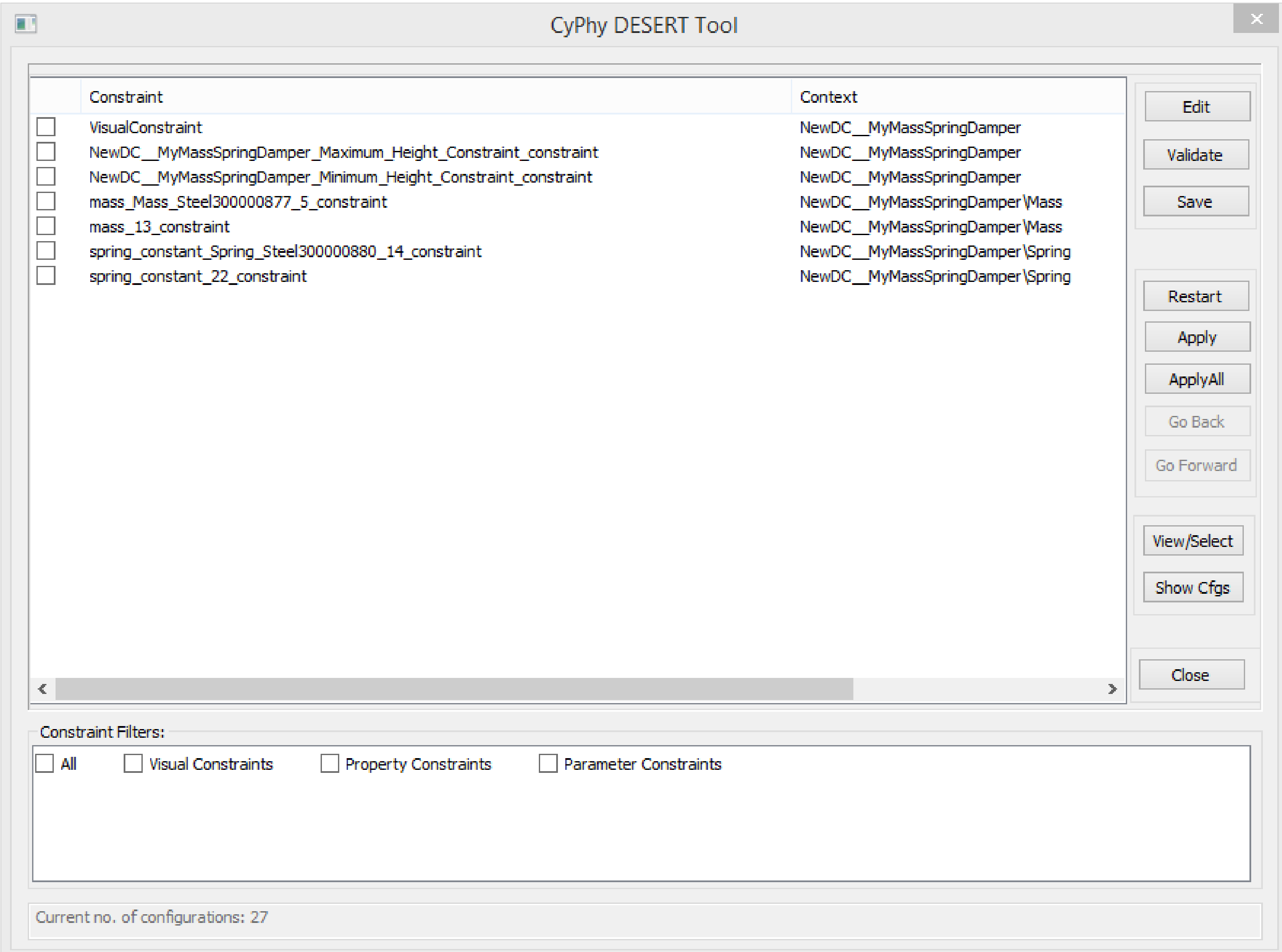


Figure 7.3: Design Space Constraints

Constraints are restrictions on the number of designs that can be made.

Note:

- The values used to determine whether a part violates a constraint are based on the parameter values set in the design space.
- “View/Select” can be clicked before clicking “Show Cfgs”. This displays all of the available components and can be used to manually decrease the number of configurations which will be tested for compliance with the constraints.

The constraints to the “NewDC__MyMassSpringDamper” model are determined by Open META-CyPhy based on the visual constraint and property constraints you created in Chapter 5.

Click “Apply All,” and then click **“Show Cfgs,”** or **configurations** (Figure 7.3).

After clicking “Show Cfgs,” a window called “Desert Configuration Dialog” will pop up (Figure 7.4). This window is used to choose which designs to analyze.

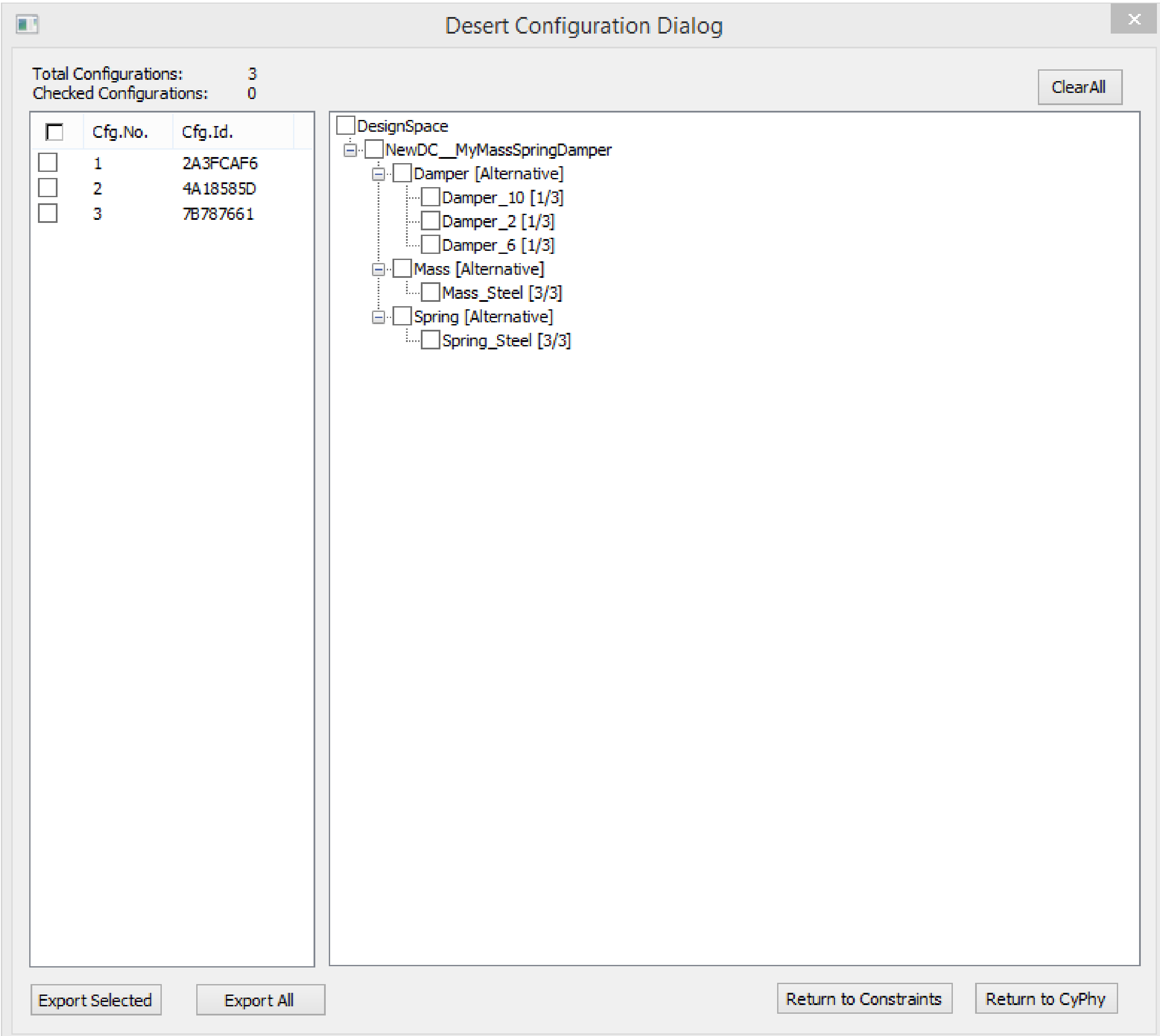


Figure 7.4: The Desert Configuration Dialog

Of the 27 possible configurations of components, the constraints allow only 3. For example, the gold mass was eliminated because it fell below the allowed range of 0.2m to 0.5m for the Height Parameter.

You can simulate any number of configurations. For the purpose of this demonstration, we will export all of the configurations. **Click "Export All." After the program has finished exporting, close the window/click Return to CyPhy.**

In the next chapter you will run the test bench you will learn how to execute test benches.

◄ **Previous: Chapter 6: Creating a Test Bench** | **Next: Chapter 8: The Master Interpreter** ►

◄ **Previous: Chapter 7: The Design Space Exploration Tool (DESERT)** | **Next: Chapter 9: The Dashboard** ►

Chapter 8: The Master Interpreter

Introduction

The Master Interpreter is a tool in Open META-CyPhy that allows the user to generate simulation files from multiple interpreters using the workflow definition created previously.

Chapter Overview

The process of using the Master Interpreter is as follows:

1. Run the Master Interpreter.

Procedure

Step 1: Running the Master Interpreter

Ensure that you are still in your test bench window. Click on the icon for the Master Interpreter in the toolbar as shown in Figure 8.1.

Figure 8.1: Master Interpreter Icon

Check the box "Post to META Job Manager". Be sure to select which configurations you wish to run from those that have been exported. Figure 8.2 shows all the boxes that should be selected. **Click OK.**

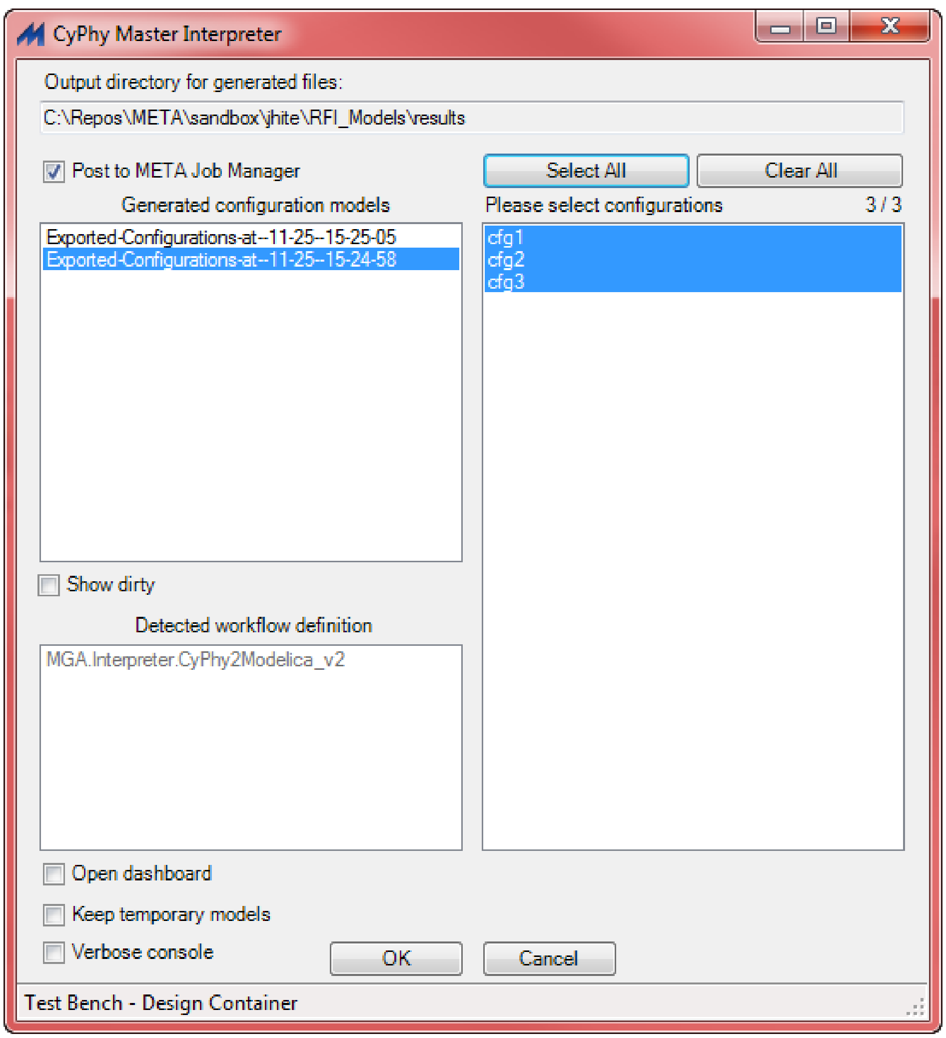


Figure 8.2: Open META-CyPhy Master Interpreter

The next window will look like Figure 8.3. This screen allows you to select which Modelica libraries you will use to simulate your design. The Modelica Standard Library (MSL) 3.2 is already included and therefore you will not need to load a package for this tutorial. **Click OK.**

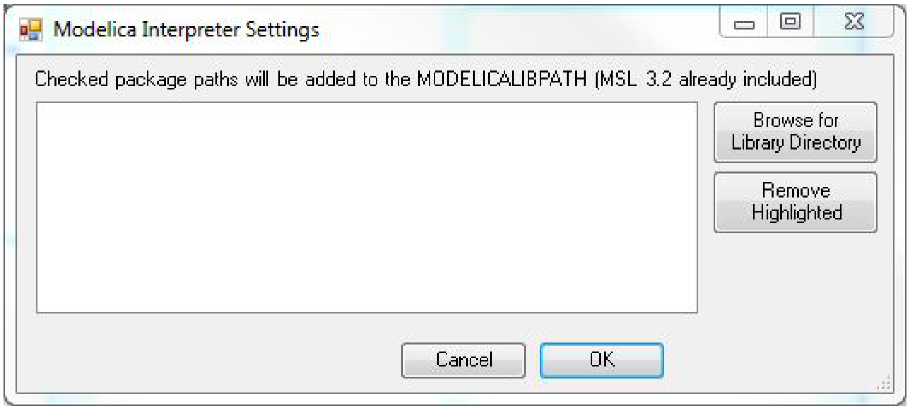


Figure 8.3: Modelica Interpreter Settings

A new window for the Job Manager will appear as shown in Figure 8.4 after a minute or two of loading.

There are two execution modes the Job Manager can run: Remote execution and local execution. Remote execution will send the information for the simulation to servers on VehicleForge for analysis and then download the results to your computer once the analysis is complete. Local execution runs the simulation directly on the users machine.

Make sure the Remote Execution box is not checked as seen in Figure 8.4. Click Save and the simulations will run locally.

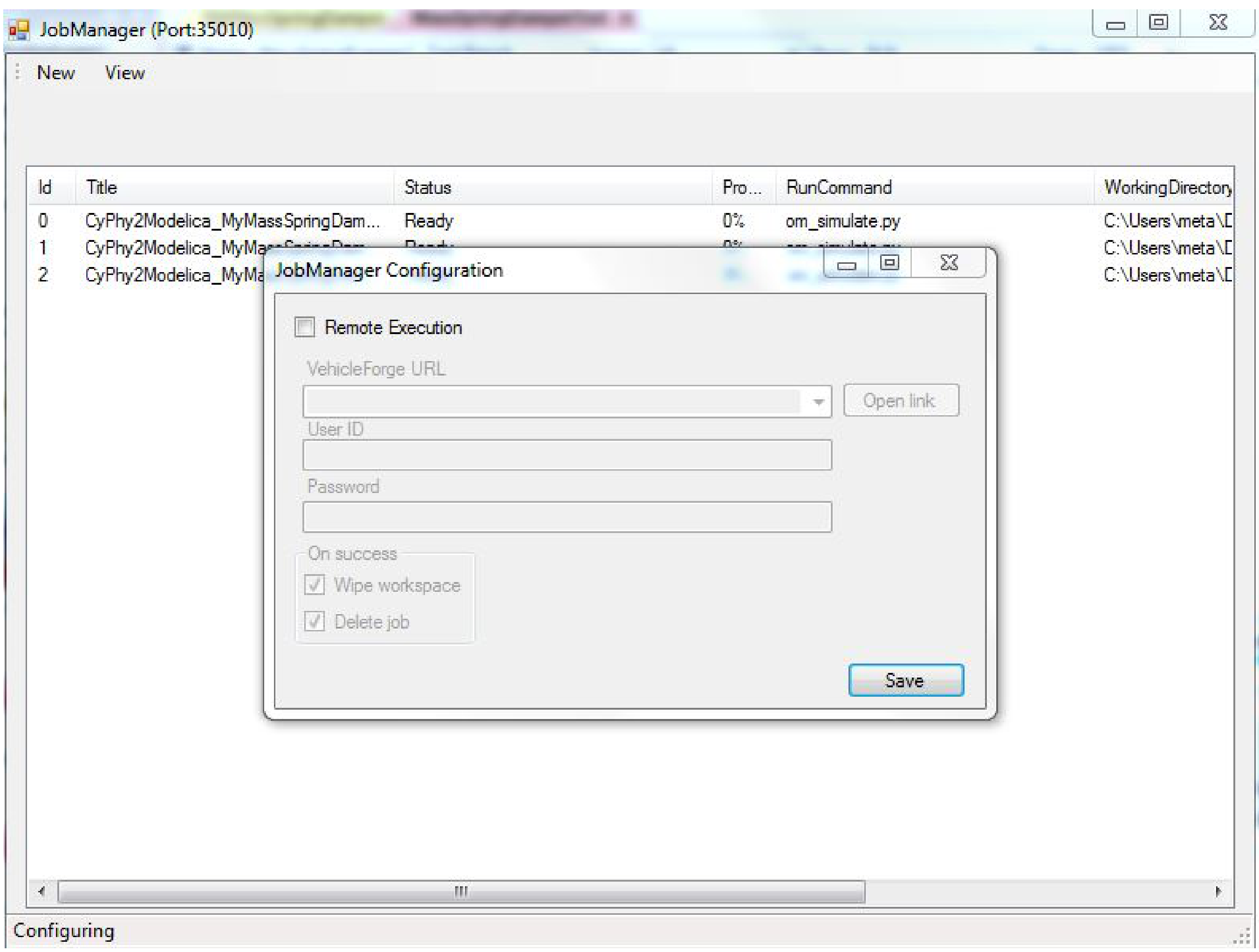


Figure 8.4: Job Manager

After a few moments, the three jobs you ran should change from blue to green and the status should change from Running to Succeeded as shown in Figure 8.5. If this does not happen, go through and check that you created the model, per the tutorials specifications.

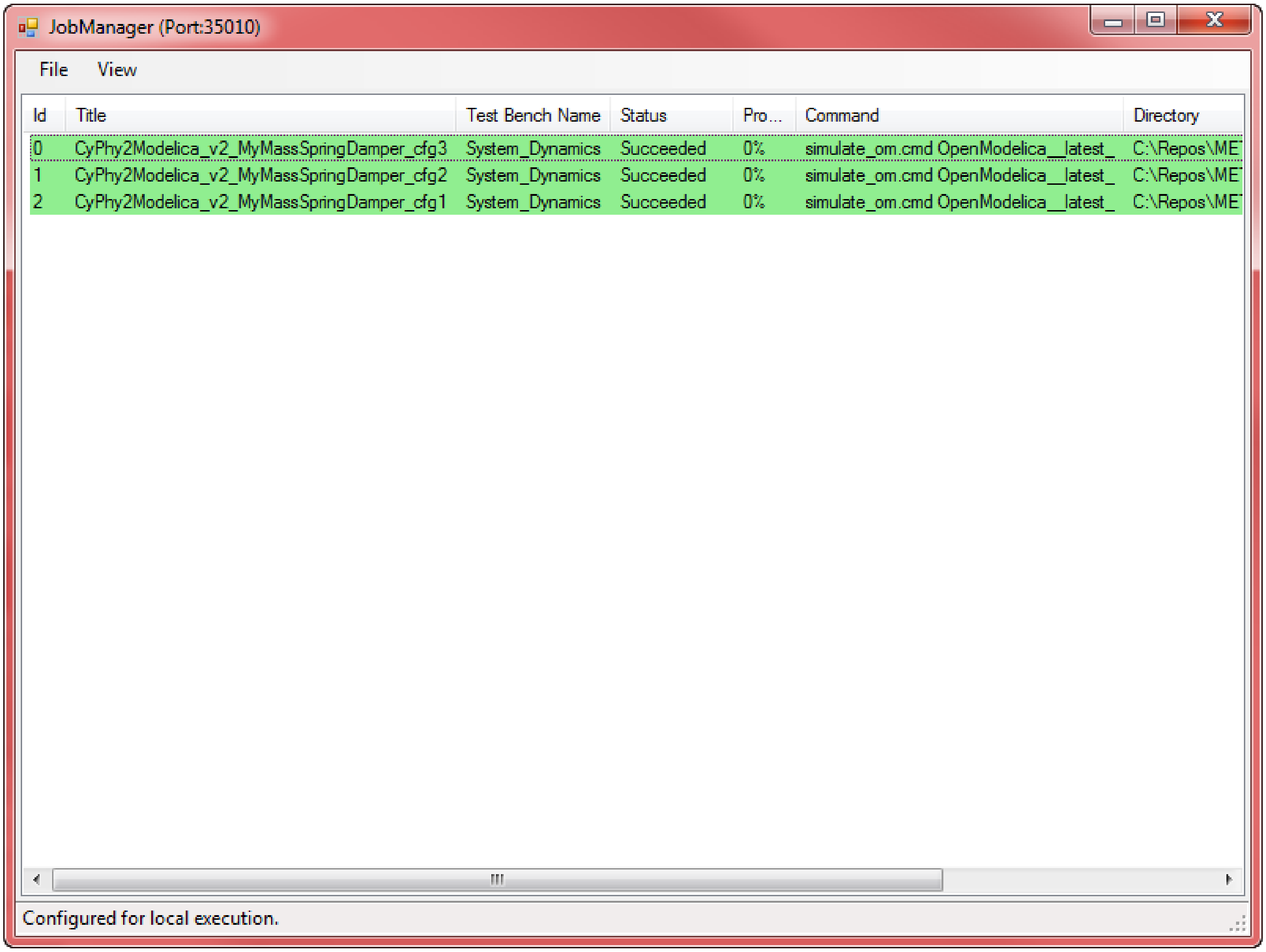


Figure 8.5: Job Manager showing successful job runs

One additional test bench has been provided for you to execute before moving on to scoring. This is the System_Cost test bench. This test bench builds the CAD assembly of the model and uses the manufacturing data provided with the components to estimate of the cost and manufacturing Lead time for your Mass Spring Damper system.

Inside the Test_Benches folder you will see a folder called System_Cost. You will need to copy and paste as reference your NewDC__MyMassSpringDamper Design Space like you did in Chapter 6 Step 3. No additional connections are required. When you are finished your test bench should look similar to Figure 8.6.

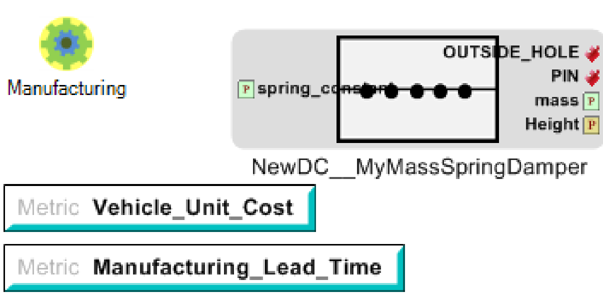


Figure 8.6: System_Cost test bench

Run the master interpreter again and select all three configurations as above in Figure 8.3. A new window will appear after the master interpreter. The next window that will appear is the CAD options window as seen in Figure 8.7. This window is used to manage what CAD files will be used and what output formats will be exported once a CAD assembly is created. The native Creo format is always generated when a CAD assembly is composed for analysis.

Make sure the "Use Project Manifest" button is checked as well as the step files button and click Ok to continue as shown in Figure 8.7.

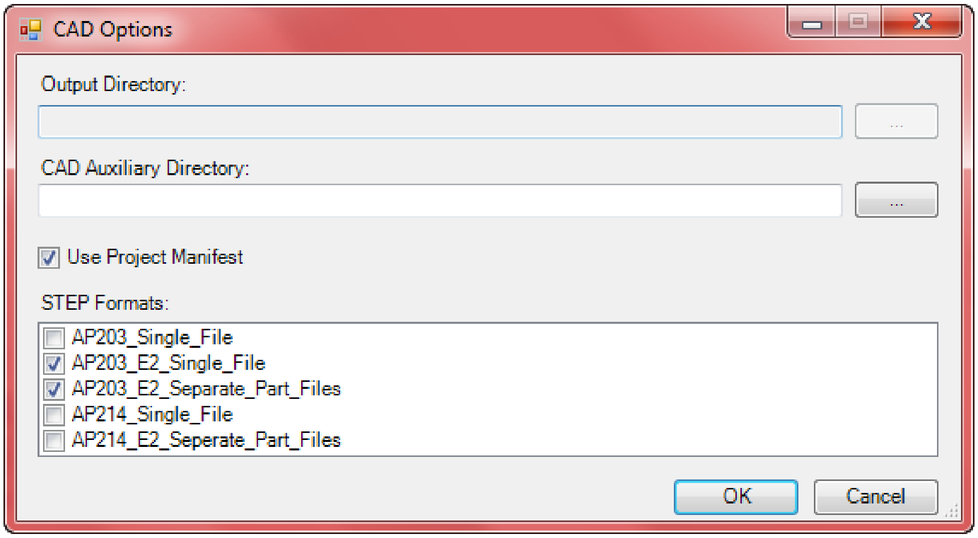


Figure 8.7: CAD Options Window

The "Use Project Manifest" option tells the interpreter to use the CAD files provided with the components when building the assembly. This will post the System_Cost test benches to the job manager for analysis.

Once you have run the System_Dynamics and System_Cost test benches for all three configurations successfully you can proceed on to the next chapter for getting your results posted to VehicleForge for further analysis and scoring.

◄ **Previous: Chapter 7: The Design Space Exploration Tool (DESERT)** | **Next: Chapter 9: The Dashboard** ►

◄ **Previous: Chapter 8: The Master Interpreter** | **Next: Chapter 10: Scoring** ►

Chapter 9: The Design Space Analyzer

Using the SVN Repository using TortoiseSVN

These instructions will guide you through your first commit to your team's SVN using TortoiseSVN. If you don't have a team, please see the Vehicle Forge Guide for more instructions.

First, **find the repository on VehicleForge for your team that you wish to use to house your MSD dataset.** The default repository is called Design. Please see the VehicleForge Guide for more information regarding navigatoin on your team's project space. Click on the repository. Once you have done this, **select the settings you wish for this repository check out to have, as shown in Figure 9.1; (HTTPS) is required to check out and commit designs.**

Figure 9.1: Read/Write permissions for repository

Each team can decide for themselves who has these permissions. Next, **click on the tortoise button** **to the right of the repository URL line**. You are likely to encounter a warning from your web browser that you need to use an external application (TortoiseSVN). Once you have read the message, continue forward. This will prompt the TortoiseSVN checkout screen to open as shown in Figure 9.2.

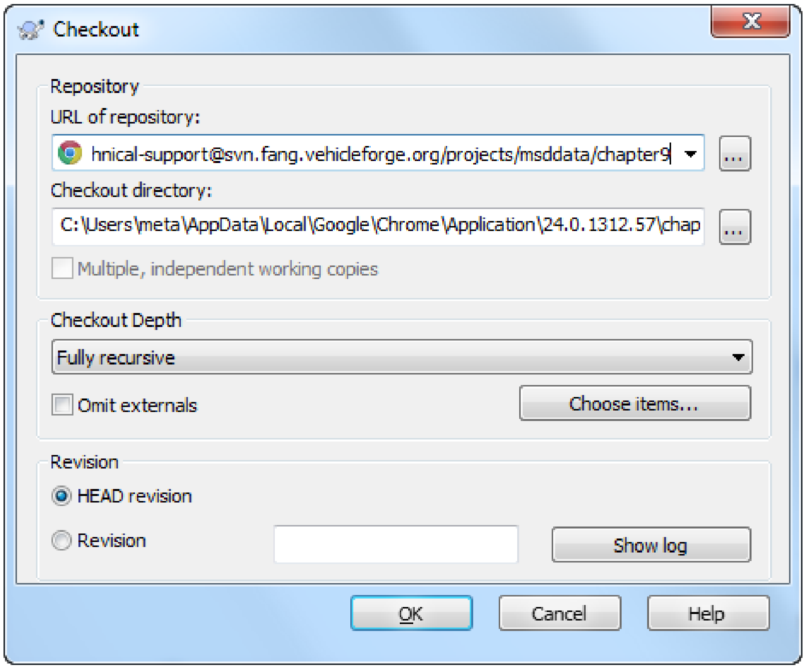


Figure 9.2: TortoiseSVN Repository Checkout Screen

In the **Checkout directory** box shown in Figure 9.2, **browse for the folder on your local machine you wish to house this repository**. The URL is automatically filled out based upon the repository you are currently in and the permissions you select. This will check out all the files in the repository to the folder you selected. **Click OK once you have selected the folder you want to use for your repository.**

A window will appear asking for a username and password as shown in Figure 9.3.

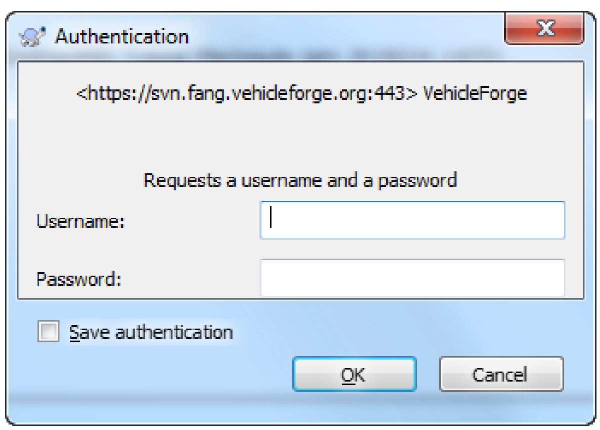


Figure 9.3: Request for VehicleForge Username and Password

Enter your VehicleForge credentials and the SVN will be linked to the folder you chose.

Once you have done this, **return to the folder where you originally saved the files for this exercise in Chapter 2 Run the export_for_dashboard_scoring.py script by double clicking it.** A command window will appear as shown in Figure 9.4.

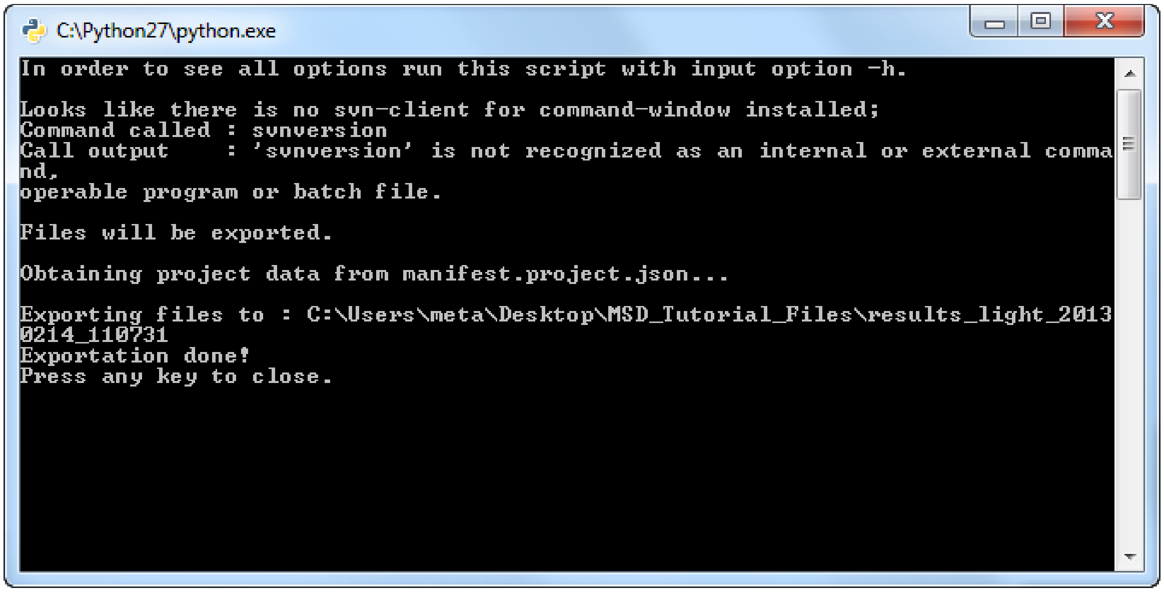
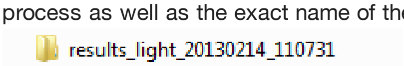


Figure 9.4: Command Window after running export_for_dashboard_scoring.py script

This script will pull out all the files directory that are required for scoring and the dashboard and save them in a folder called results_light that is one level up; part of this folder's name is also a time stamp when it was created. For example, if the msd files were in a folder on the desktop, the folder created by the export_for_dashboard_scoring.py file will be created on the desktop. This command window will show the progress of the copying process as well as the exact name of the folder created. **Locate this folder which will have a name similar to this:**



. Move the results_light folder to the folder you choose previously to use for the SVN check out.

Once you have done this, **right click on the results_light folder and follow the path TortoiseSVN > Add as shown in Figure 9.5.** This will bring up a dialog box asking which files should be added.

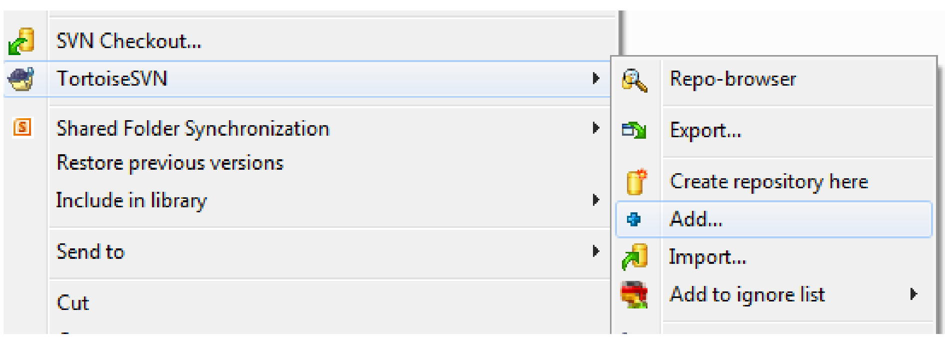


Figure 9.5: SVN Add

Make sure to **select the all files and click OK.** This will tell SVN which new files need to be committed to the repository.

Now, **right click on the results_light folder again**; two new options will appear: SVN Update and SVN Commit, as shown in Figure 9.6. Update pulls files from the repository and saves them on your local drive. SVN commit will take files from the local drive and post them to the repository. **Select SVN commit.**

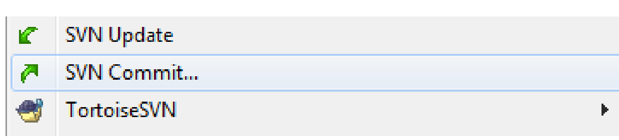


Figure 9.6: SVN Commit

A dialog box will appear as shown in Figure 9.7. The top of this dialog box displays messages about the files being committed, while the bottom shows a list of all files that could be posted to the repository up. The files we added before will automatically be selected as files to commit. **Write a message describing your commit. You should do this so that the other members of your team know what changes were made. Click OK.**

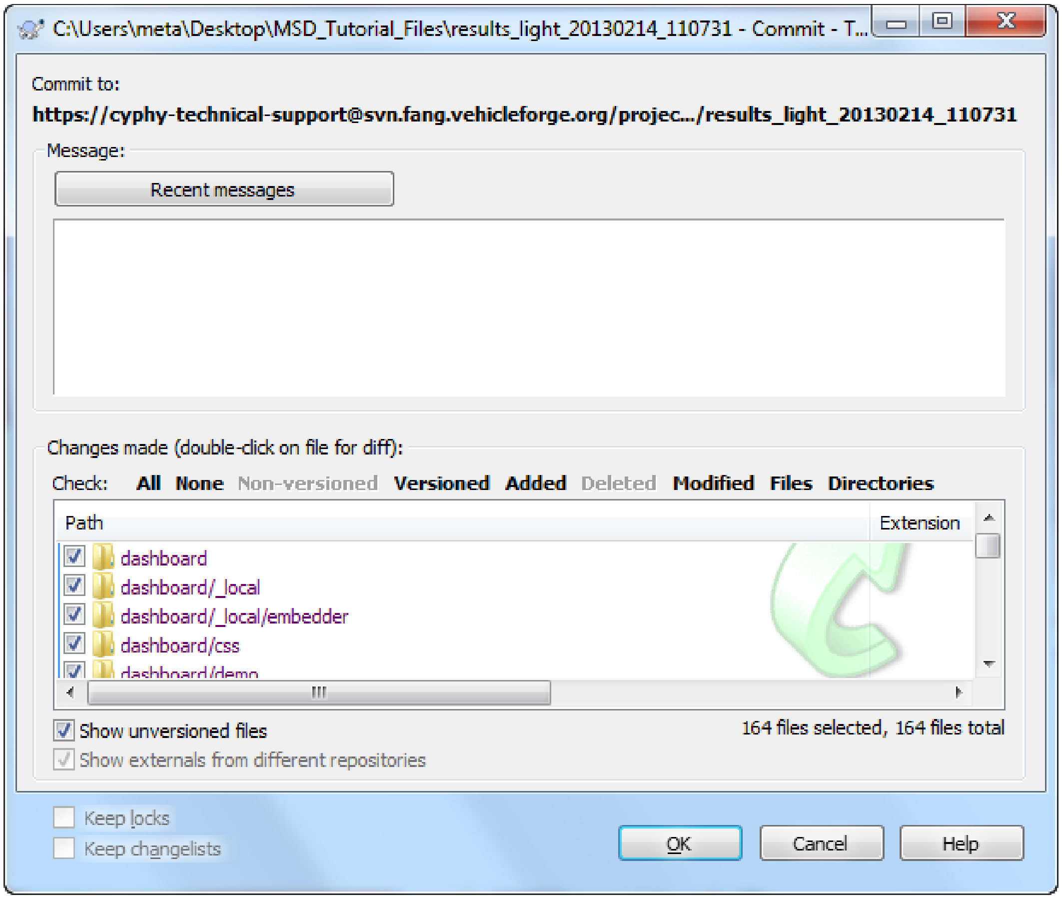


Figure 9.7: TortoiseSVN Commit Dialog Box

You will once again be asked for your username and password. After you enter your credentials again the files will begin to be uploaded to the SVN. Once the commit is complete, as shown in Figure 9.8 you will be able to utilize the Dashboard and Scoring functions on VehicleForge.

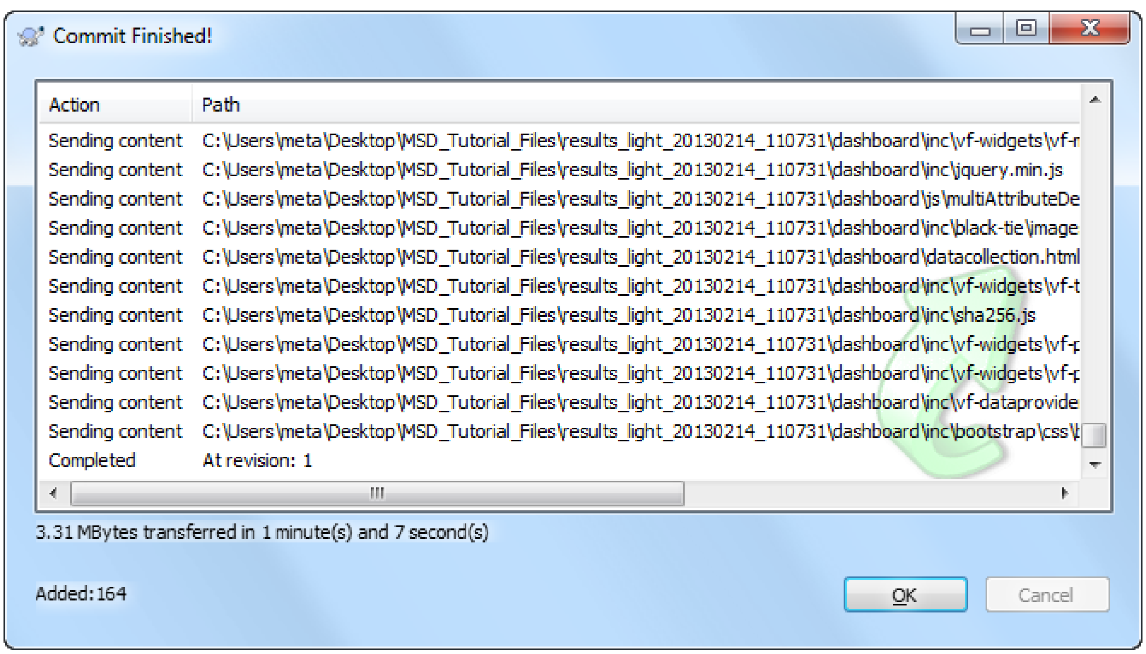


Figure 9.8: Completed SVN Commit

Design Space Analyzer

Overview

The design space analyzer is a systems engineering visual analytics tool. It converts predefined data sets into a collection of interactive analytical visualizations for the purpose of enabling or enhancing a user’s capacity for cognitive reasoning based on perceptual principles. The two primary benefits being:

- An increased conceptual understanding of the data being visualized.
- And an increased transparency as to how one should react to the information embedded in the data.

The tool is organized into tabs corresponding to specific common systems engineering design activities. The visibility of a tab is dependent on the data available.

NOTE:
This section is intended to show you the tools available for analyzing designs once you have run the simulations. Not all the images below are from the Mass Spring Damper models.

Featured Tabs

The availability of various features is dependent on the online status of the application and the availability of data. Below is an enumerated list of each feature along with the conditions for the feature to be enabled.

Feature	**Specific Data Required**	**Selections Required**
Requirements Analysis	Requirements definition	
Design Space	Successfully completed testbench results	
Design Space Analysis	Successfully completed testbench results	At least two variables and one design selected
Parametric Design Analysis	PET analysis results	At least one response and two variables selected
Probabilistic Certificate of Correctness	PCC analysis results	
Formal Verification	Verification Results	
Multi-Attribute Decision Analysis	Successfully completed testbench results	At least one variable
Surrogate Modeling Performance	Surrogate summary results	

Getting Started

The actual procedure for starting the design space analyzer depends on the operating environment. The META tools provides a convenient option to start the design space analyzer after successfully running a test bench. The design space analyzer can also run within the scope of VehicleForge. Simply use the Project Navigator on your team's project page to create custom virtual datasets to be read into the design space analyzer. You can follow the directions found in the VehicleForge Guide in the Design Space Analysis section

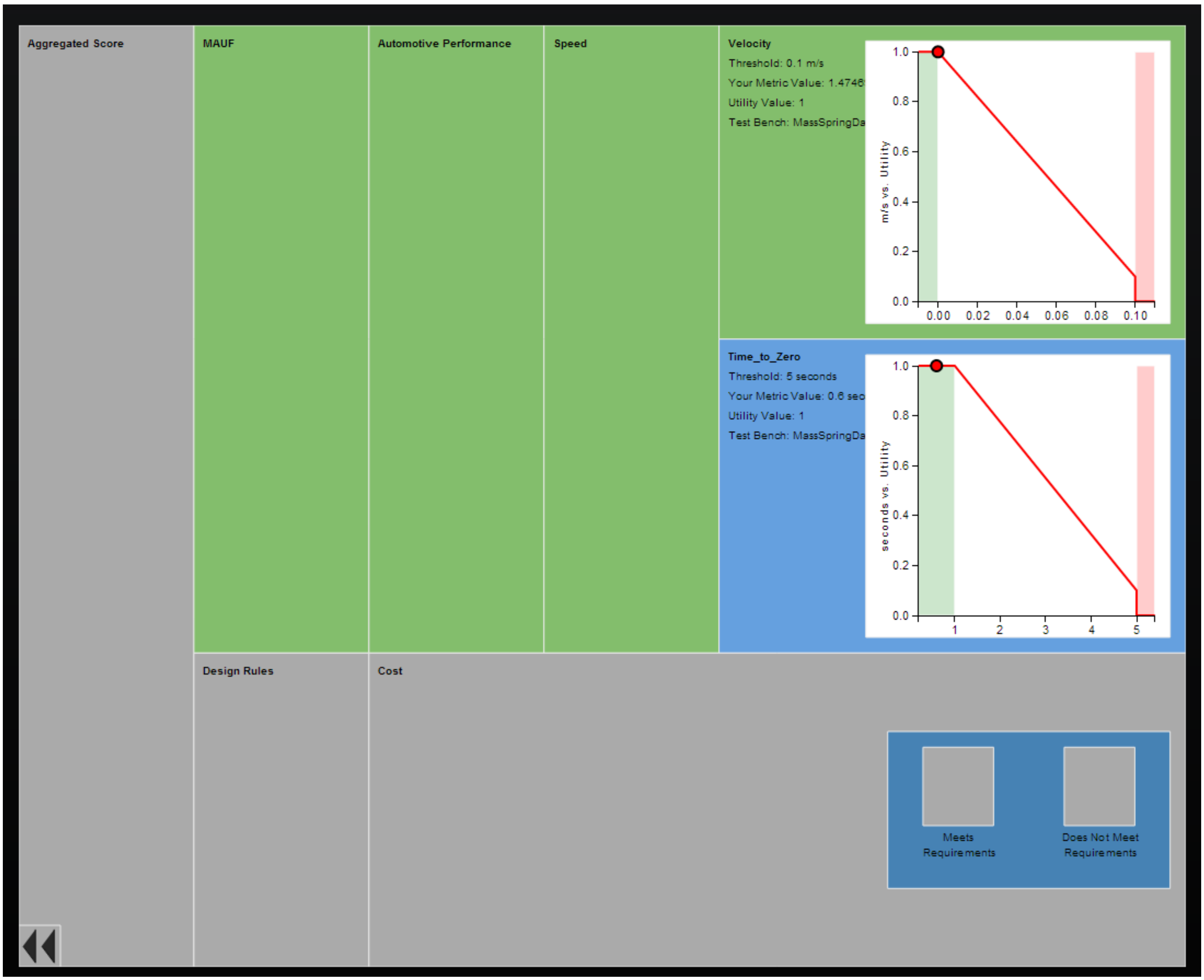
Tabs

As it was mentioned above each tab focuses on a specific design activity. However, the tabs do have integrating elements which tie them together. In the section below, each of these tabs, along with their integrating elements, will be explored.

Requirements Analysis

The Requirements Analysis tab allows rapid analysis of a configuration in order to analyze which design metrics do and do not meet required values. The results are presented in two tables and a stoplight diagram.

The tables shown are the design info box and the requirements list which provide, respectively, all of the data about the design from the test benches and all of the data from the requirements definition. The stoplight diagram shows the requirements tree graphically and is color coded to demonstrate where a design does and does not meet the requirements. Specifically, grey indicates a requirement for which there is no data provided to compare the design, red indicates a requirement in which the threshold is not met, green represents a design in which the threshold is met, but the objective is not, and blue represents a design in which the objective is met. Higher level colorings are indicative of their weakest child.

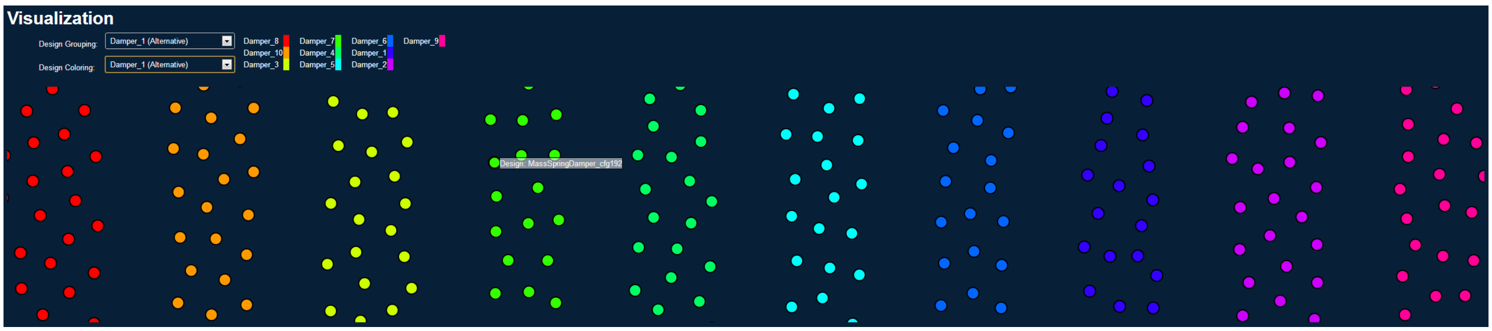


The top two panels are the design info box and the requirements list. These two tables are both controlled in the same manner. First, the configuration whose information should be displayed is selected via the drop down menu labeled "configuration". Changing the configuration field will change all configuration fields in the tab. The number of entries in the table can be controlled via the second drop down menu labeled "show X entries", in which X is the number of entries selected to be shown. If not all entries in a given category are displayed, the list of entries can be browsed by clicking the right and left arrows located at the bottom right corner of the table. The tables can be sorted in either ascending or descending order by clicking the up and down arrows at the right hand side of each column header. A bidirectional arrow indicates no sorting preference, an upward arrow indicates an ascending sort, in which low numbers are shown above high numbers for numeric fields, and early letters are shown before later letters in alphabetic fields. A downward arrow indicates a descending sort. Finally, typing search terms in the search box separated by spaces will filter results to show only rows that contain all of the listed search terms.

There are three methods of controlling the stoplight diagram: configuration selection, zooming, and checkbox usage. Configuration selection is done in the same manner as it is in the tables. Zooming is one of the primary functions of the stoplight diagram and allows analysis of the components of a specific design category. It is done by clicking on the category of choice which then zooms the view-able area to show only that category and the children of that category. This allows increased detail to be shown, including data on the requirements and the utility curves for each metric. The stoplight diagram can be zoomed one level up by clicking on the left edge of the diagram, and fully zoomed out by clicking the double arrow button in the lower left-hand corner of the diagram. Checkbox usage is straight forward: click on a box to select that option. A large 'X' will appear in an active box. Selecting a different box will deselect the current box. Clicking on a selected box will also deselect it and will change the metric completeness to 'undefined'.

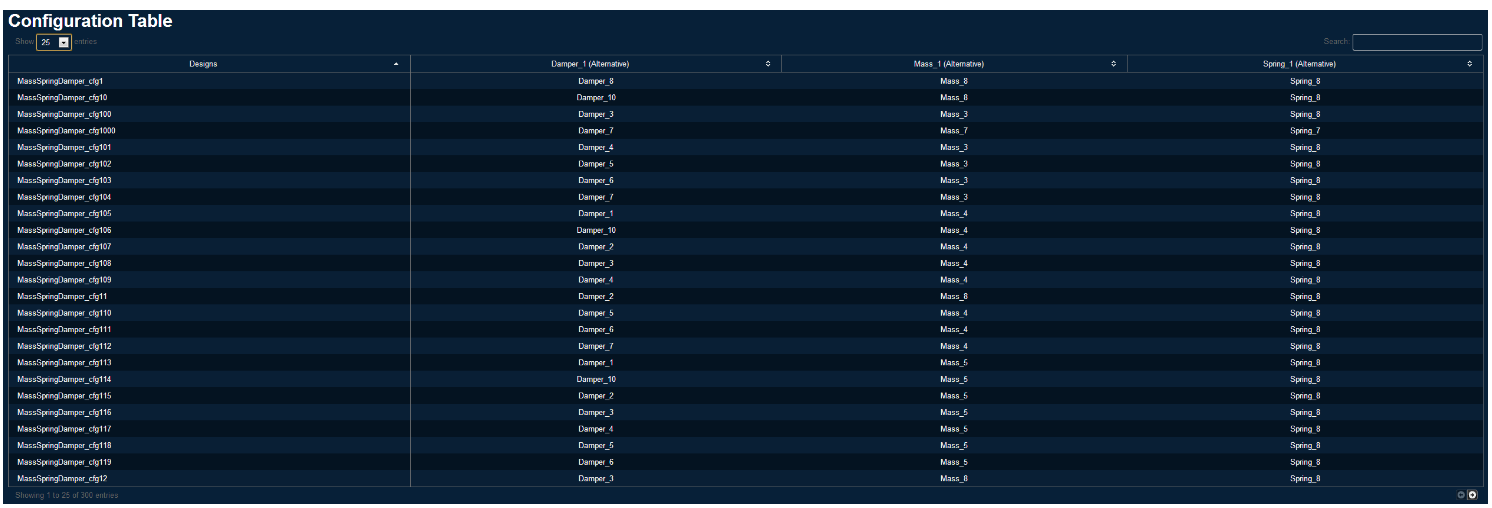
Design Space

The design space tab shows each candidate design along with their configuration. A configuration is a collection of components utilized within the design. At the top of the tab, each design is illustrated with a colored circle where the user is able to specify grouping and color coding for visualization purposes.



The example above shows 300 designs being grouped and color coded by the damper used within the mass-spring-damper system.

Following just below the Configuration table shows in detail which components are used within each design



Configuration Table

Show 25 entries Search:

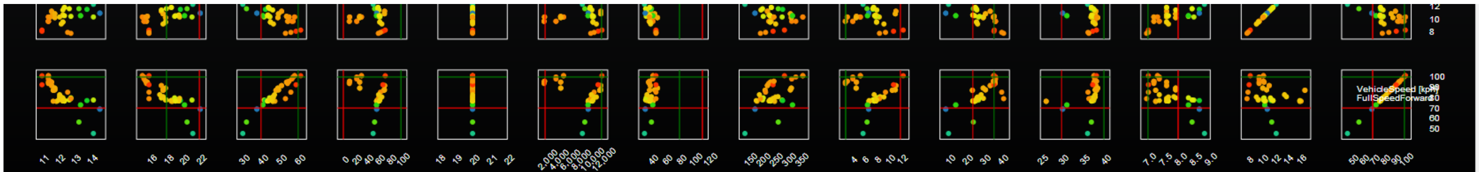
Designs	Damper_1 (Alternative)	Mass_1 (Alternative)	Spring_1 (Alternative)
MassSpringDamper_cfg1	Damper_8	Mass_8	Spring_8
MassSpringDamper_cfg10	Damper_10	Mass_8	Spring_8
MassSpringDamper_cfg100	Damper_3	Mass_3	Spring_8
MassSpringDamper_cfg1000	Damper_7	Mass_7	Spring_7
MassSpringDamper_cfg101	Damper_4	Mass_3	Spring_8
MassSpringDamper_cfg102	Damper_5	Mass_3	Spring_8
MassSpringDamper_cfg103	Damper_6	Mass_3	Spring_8
MassSpringDamper_cfg104	Damper_7	Mass_3	Spring_8
MassSpringDamper_cfg105	Damper_1	Mass_4	Spring_8
MassSpringDamper_cfg106	Damper_10	Mass_4	Spring_8
MassSpringDamper_cfg107	Damper_2	Mass_4	Spring_8
MassSpringDamper_cfg108	Damper_3	Mass_4	Spring_8
MassSpringDamper_cfg109	Damper_4	Mass_4	Spring_8
MassSpringDamper_cfg11	Damper_2	Mass_8	Spring_8
MassSpringDamper_cfg110	Damper_5	Mass_4	Spring_8
MassSpringDamper_cfg111	Damper_6	Mass_4	Spring_8
MassSpringDamper_cfg112	Damper_7	Mass_4	Spring_8
MassSpringDamper_cfg113	Damper_1	Mass_5	Spring_8
MassSpringDamper_cfg114	Damper_10	Mass_5	Spring_8
MassSpringDamper_cfg115	Damper_2	Mass_5	Spring_8
MassSpringDamper_cfg116	Damper_3	Mass_5	Spring_8
MassSpringDamper_cfg117	Damper_4	Mass_5	Spring_8
MassSpringDamper_cfg118	Damper_5	Mass_5	Spring_8
MassSpringDamper_cfg119	Damper_6	Mass_5	Spring_8
MassSpringDamper_cfg12	Damper_3	Mass_8	Spring_8

Showing 1 to 25 of 300 entries

Design Space Analysis

The design space analysis tab provides a multidimensional view of each design. Within this tab, each metric of each design can be compared, providing a valuable understanding of the explored design space. If a requirements definition is available this tab can also visualize the space as it is constrained by the requirements. The tab also allows the user to introduce more dimensionality through color.


Requirements Analysis
Design Space
Design Space Analysis
Parametric Design Analysis
Multi-Attribute Decision Analysis
! Limit Violations Summary
! File Load Summary
Refresh
Color
Settings
Help
Parallel Axes Plot
Acc20kph [s] FullSpeedForwardTemp
Acc40kph [s] FullSpeedForward
AverageSpeed [kph] FullSpeedForward
AverageSpeed [kph] HighwayCruise
CoolantTemperature [C] FullSpeedForwardTemp
DistanceCovered [m] HighwayCruise
EngineTemperature [C] HillClimbSand
ExhaustTemperature [C] FullSpeedForwardTemp
FuelConsumed [L] HighwayCruise
MaxVehicleSpeed [kph] HillClimbSoil
MaxVehicleSpeed [kph] HillClimbConcrete
ReverseAcc10kph [s] FullSpeedReverse
TimeToMaxReverseSpeed [s] FullSpeedReverse
VehicleSpeed [kph] FullSpeedForward
Multivariate Plot

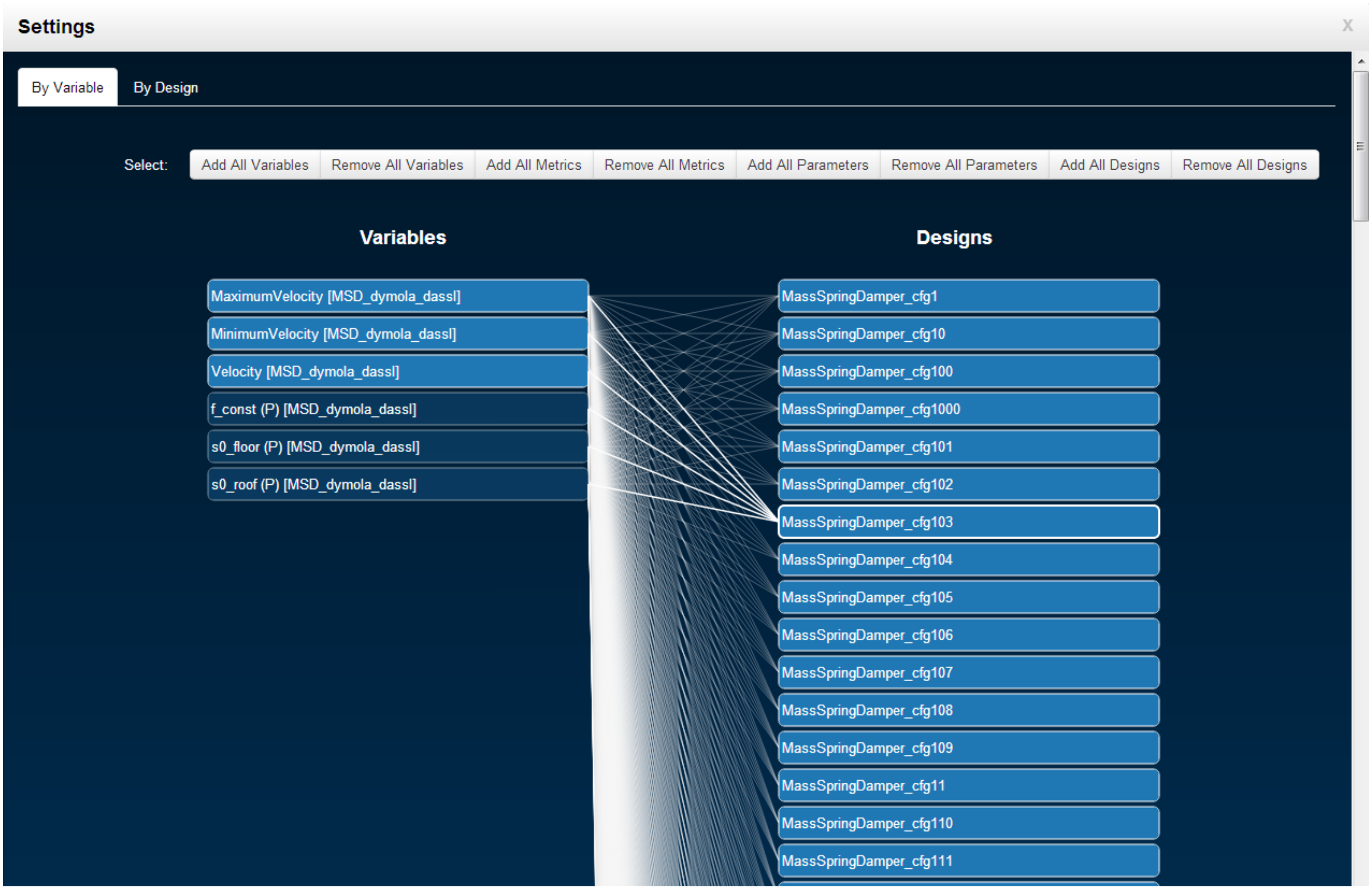


Selected Designs

Design ID	Acc20kph [s]	Acc40kph [s]	AverageSpeed [kph]	AverageSpeed [kph]	CoolantTempe [C]	DistanceCover [m]	EngineTemper [C]	ExhaustTempe [C]	FuelConsumed [L]	MaxVehicleSp [kph]	MaxVehicleSp [kph]	ReverseAcc10k [s]	TimeToMaxRe [s]	VehicleSpeed [kph]
design_v4_cfg1	14.2	21.1	41.8	62.5	20	10400	39.5	281	8	27.3	37.2	8.52	10.4	77.8
design_v4_cfg10	11.5	15.5	55.2	11.1	20	1850	37.7	249	3.28	35.7	38.2	7.2	10.4	95.7
design_v4_cfg11	11.4	15.6	54.4	10.6	20	1770	37.1	229	2.96	36.8	37.6	7.2	9.72	93.6
design_v4_cfg12	11.4	15.6	54.7	10.6	20	1770	37.1	239	2.96	36.8	37.6	7.2	9.72	95
design_v4_cfg13	11.4	15.1	58.2	31.5	20	5250	42.4	361	5.44	36	37.6	7.32	10.4	101
design_v4_cfg14	11.6	17.4	51.2	71.5	20	11900	38.7	224	11.3	34	38.1	7.62	11.3	91.6
design_v4_cfg15	11.4	15.1	58.2	31.5	20	5250	42.5	361	5.44	36	37.6	7.32	10.4	101
design_v4_cfg16	11.6	17.4	51.2	71.5	20	11900	38.7	224	11.3	34	38	7.62	11.3	91.6
design_v4_cfg17	11.4	15.3	56.8	23.4	20	3900	42.3	318	4.19	37	37.8	7.38	9.72	97.9
design_v4_cfg18	11.6	15.7	53.9	28.4	20	4740	36.3	236	4.77	36.3	37.9	7.62	10.5	94.9
design_v4_cfg19	11.4	15.3	57.4	23.4	20	3900	42.3	334	4.19	37	37.7	7.38	9.72	99.8
design_v4_cfg2	14.6	22.1	37.9	57.8	20	9640	31.6	176	6.81	8.72	30.2	8.7	11	69.1
design_v4_cfg20	11.6	15.7	53.9	28.4	20	4740	36.3	236	4.81	36.3	37.9	7.62	10.5	94.9
design_v4_cfg21	11.8	17.6	49.5	71.2	20	11900	44.9	313	11.2	31.3	37.9	7.92	10.9	87.5
design_v4_cfg22	11.8	17.6	49.6	71.2	20	11900	44.8	313	11.2	31.3	37.9	7.92	10.9	87.5
design_v4_cfg23	11.8	17.9	49.2	28.3	20	4720	48.8	317	4.72	30	38.5	7.92	10.3	89.2
design_v4_cfg24	11.8	17.9	49.2	28.7	20	4780	48.7	316	4.77	30	38.5	7.92	10.3	89.2
design_v4_cfg25	11.7	17.6	48.2	70	20	11700	37.6	215	10.7	23.8	38.2	7.26	16.9	87.1
design_v4_cfg26	11.7	17.6	48.2	70	20	11700	37.6	215	10.7	23.9	38.2	7.26	16.9	87.1
design_v4_cfg27	14.2	21.2	30.3	51.7	20	8610	30.5	137	5.88	10.3	36.1	8.87	12.4	45.9
design_v4_cfg28	13.3	20.5	32.3	54.2	20	9040	40.3	266	6.54	24.4	37	8.46	11.7	56.8
design_v4_cfg29	12.4	18.6	44.1	60.1	20	10000	37.9	239	7.62	25.5	37	7.62	11.9	77.3
design_v4_cfg3	13.4	20.2	41.1	57.6	20	9610	42.1	321	6.61	14.4	31.2	8.34	10.6	73.1
design_v4_cfg30	12.3	19.5	43.8	61.4	20	10200	37.9	231	7.99	24.1	37.1	7.62	10.7	76.9
design_v4_cfg31	13.8	19.7	42.5	59.6	20	9940	39.3	283	7.53	26.2	37.5	8.7	11.6	77.3
design_v4_cfg32	12.3	17.4	46.4	61.8	20	10300	33.4	199	8.15	26.4	37.7	7.92	11.5	79.8
design_v4_cfg33	12.1	17.6	46.6	63.4	20	10600	40.3	265	8.69	28.6	37.6	7.86	14	81
design_v4_cfg34	12.7	18	44.8	59.6	20	9940	32.1	205	7.5	26.1	37.5	8.46	11.7	77.9
design_v4_cfg35	12.4	18.6	44.1	60.1	20	10000	37.9	239	7.62	25.5	37	7.62	11.9	77.3
design_v4_cfg36	11.9	18.7	46.5	57.7	20	9620	42.6	252	6.83	25.1	26.3	7.26	16.1	76.5
design_v4_cfg37	11.7	18.6	47.6	58	20	9660	37.2	184	6.96	25.1	26.3	7.26	16.3	76.7
design_v4_cfg38	12	18.6	46.3	58	20	9670	33.9	181	6.9	25.8	37.5	7.38	16.1	77.3
design_v4_cfg39	12.1	19.6	44.9	63	20	10500	39.1	246	8.55	30.1	36.9	7.62	10.4	79.4
design_v4_cfg4	12.8	19	44.7	63.7	20	10600	32.9	208	8.3	27.9	37.5	8.34	10.5	79.9
design_v4_cfg40	12.3	19.5	43.8	61.4	20	10200	37.9	231	7.99	24.1	37.1	7.62	10.7	76.9
design_v4_cfg41	14.6	22.1	37.9	57.8	20	9640	31.6	176	6.81	8.72	30.2	8.7	11	69.1
design_v4_cfg42	13.4	20.2	41.1	57.6	20	9610	42.1	321	6.61	14.4	31.2	8.34	10.6	73.1
design_v4_cfg43	12.3	17.9	45.2	63.2	20	10500	38.3	235	8.11	26.2	37.4	7.62	9.6	78
design_v4_cfg44	11	15.9	61.5	72.6	20	12100	46.6	290	12.5	36.3	38.2	7.02	8.34	101
design_v4_cfg45	11	15.9	59	68.5	20	11400	46.6	246	10.9	36.1	37.3	7.02	8.16	92.4
design_v4_cfg46	12.6	15.9	56.2	64.3	20	10700	46.7	217	9.58	40.5	36.2	7.02	7.98	85.3
design_v4_cfg47	12.8	15.9	53.3	60.7	20	10100	46.7	197	8.59	41.6	36.4	6.96	7.83	79.1
design_v4_cfg5	12.4	19.1	43.6	62.1	20	10400	38	235	7.77	25.1	37.1	7.68	10.6	76.8
design_v4_cfg6	12.2	17.6	46.8	66.7	20	11100	39.2	256	9.32	26.8	37.9	7.68	9.3	80.7
design_v4_cfg7	12.3	17.9	45.2	63.2	20	10500	38.3	235	8.11	26.2	37.4	7.62	9.6	78
design_v4_cfg8	12.1	17.2	47.4	61.1	20	10200	34	193	7.71	28.8	37.9	7.38	13.7	82.9
design_v4_cfg9	11.5	15.5	55.2	11.1	20	1850	37.5	249	3.28	35.7	38.2	7.2	10.4	95.7

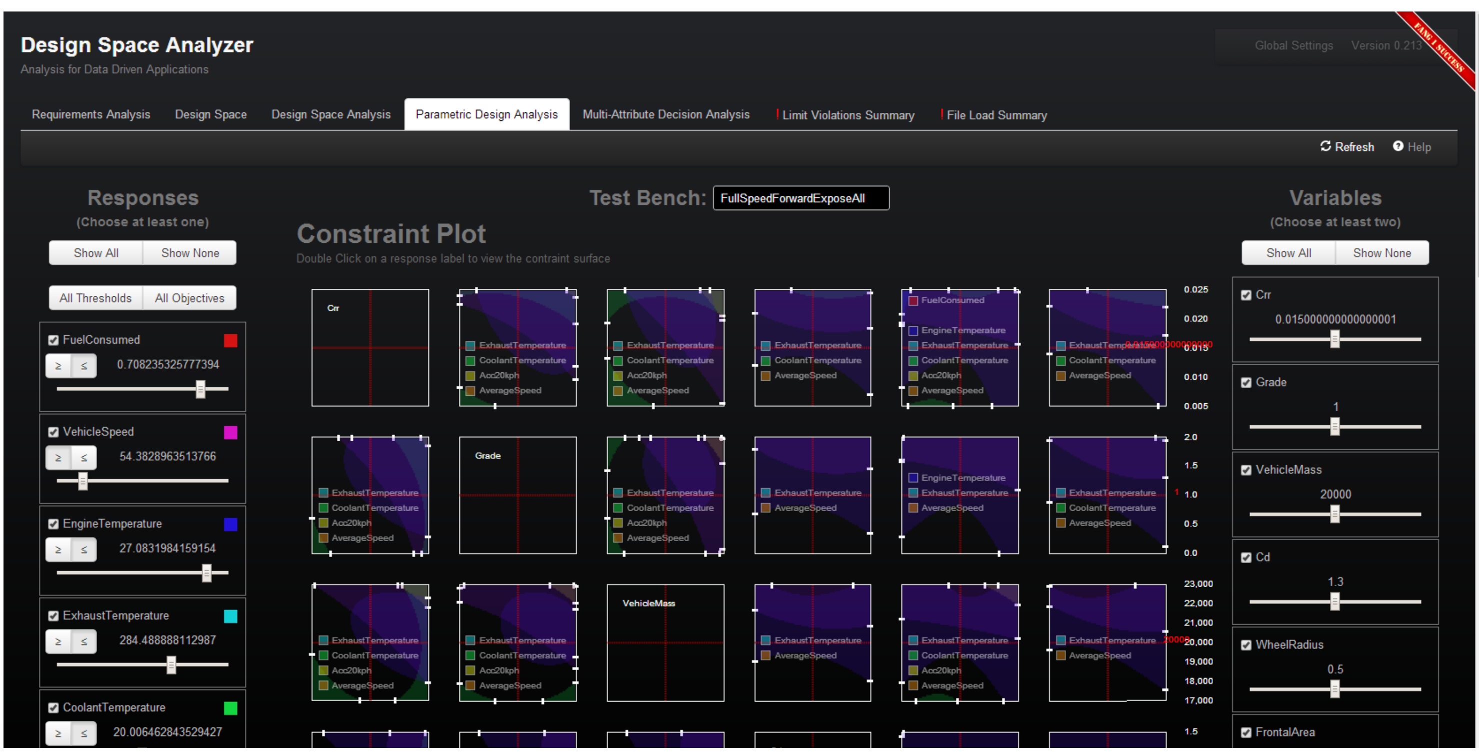
The designs can be color coded with respect to scores, requirements, MADA results, PCC results, or limit violations. Each color coding allows the user to see how the various types of requirements, constraints and analyses push designs to particular parts of the design space.

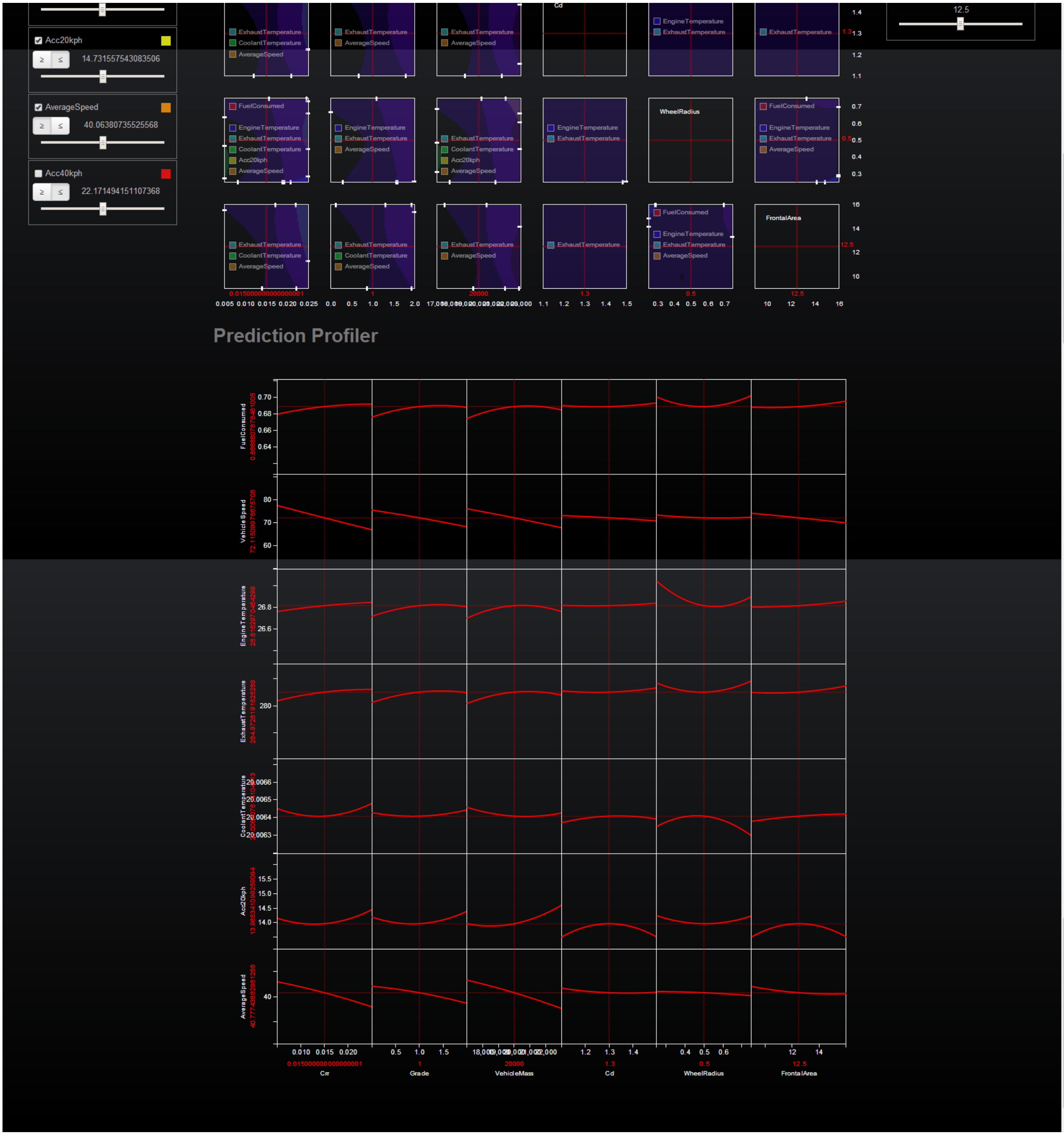
Of course it is possible to have a mixed dataset where not every design has been run through the same test benches, in which case the tool allows the user to see which designs can be displayed together as shown below.



Parametric Design Analysis

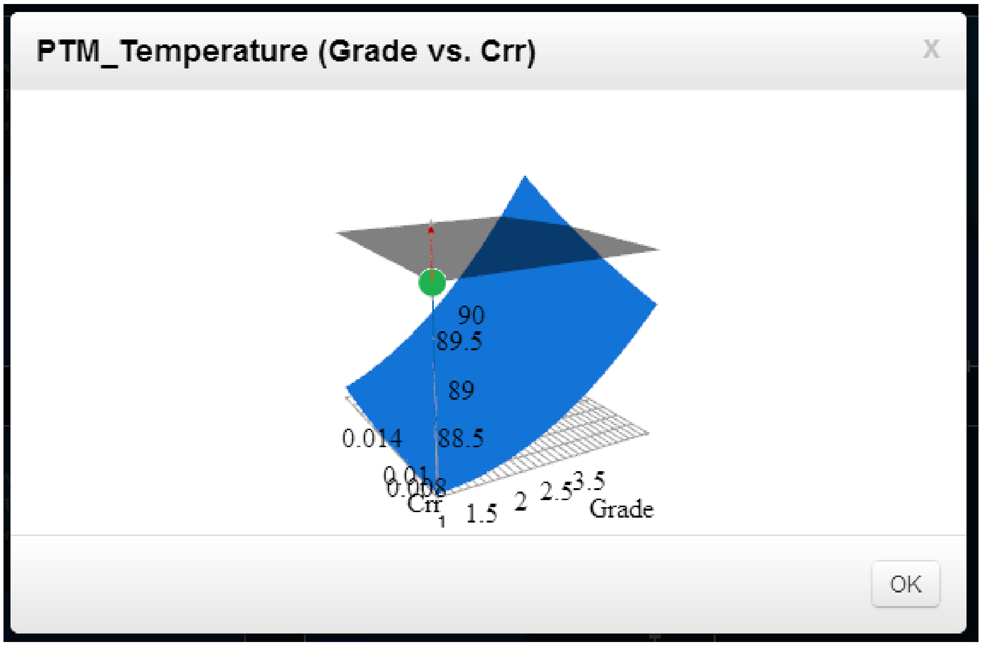
If the user has performed a PET analysis the parametric design analysis tab will be available. This tab specializes in "what if" scenarios. To start, the user must select at least two variables and one response. At the top of the tab is a constraint plot illustrating the design space after constraints initially based on the requirements definitions are applied. The user is able to explore the space in terms of the constraints visualizing the effects of relaxing or restricting each constraint.



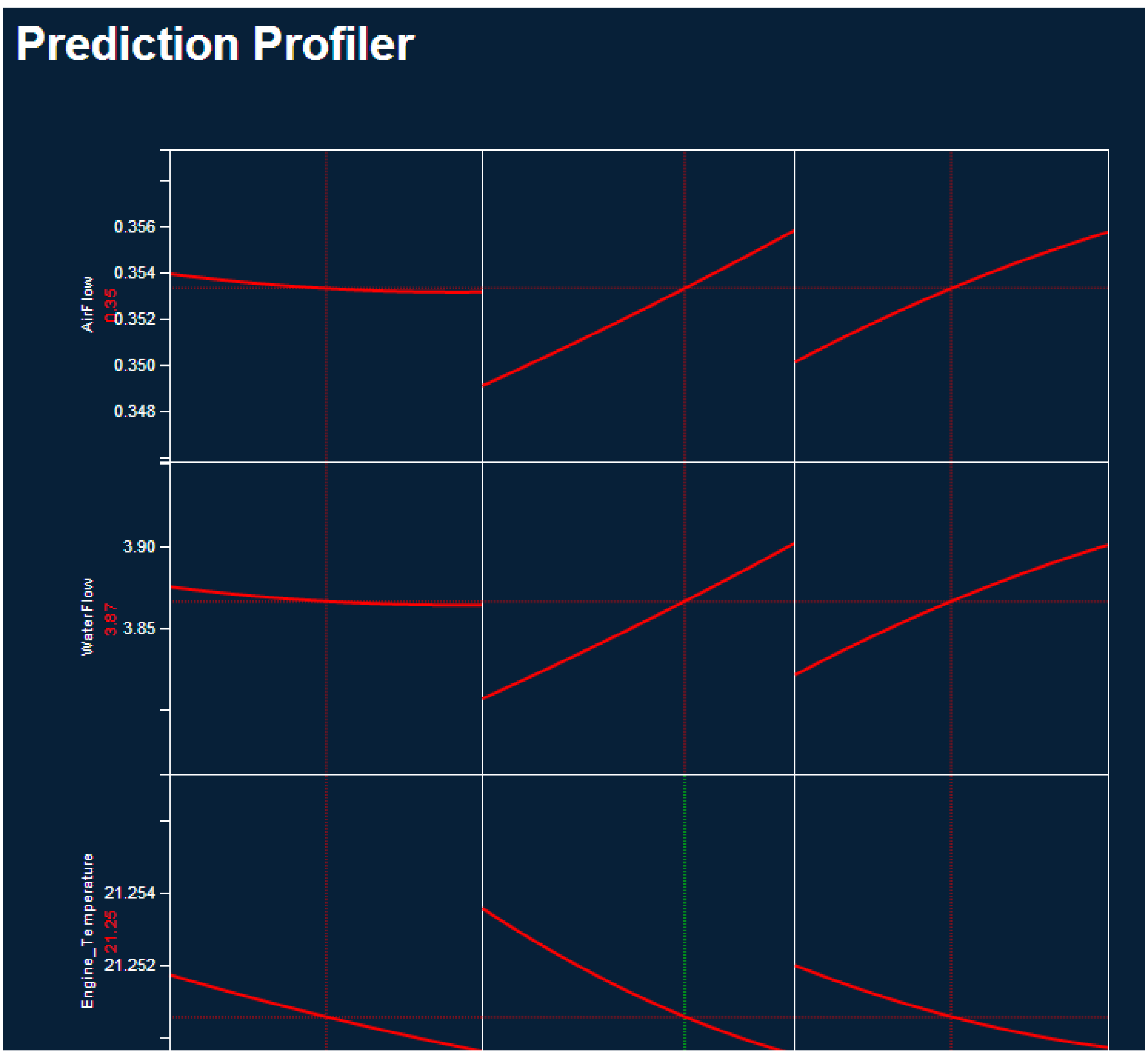


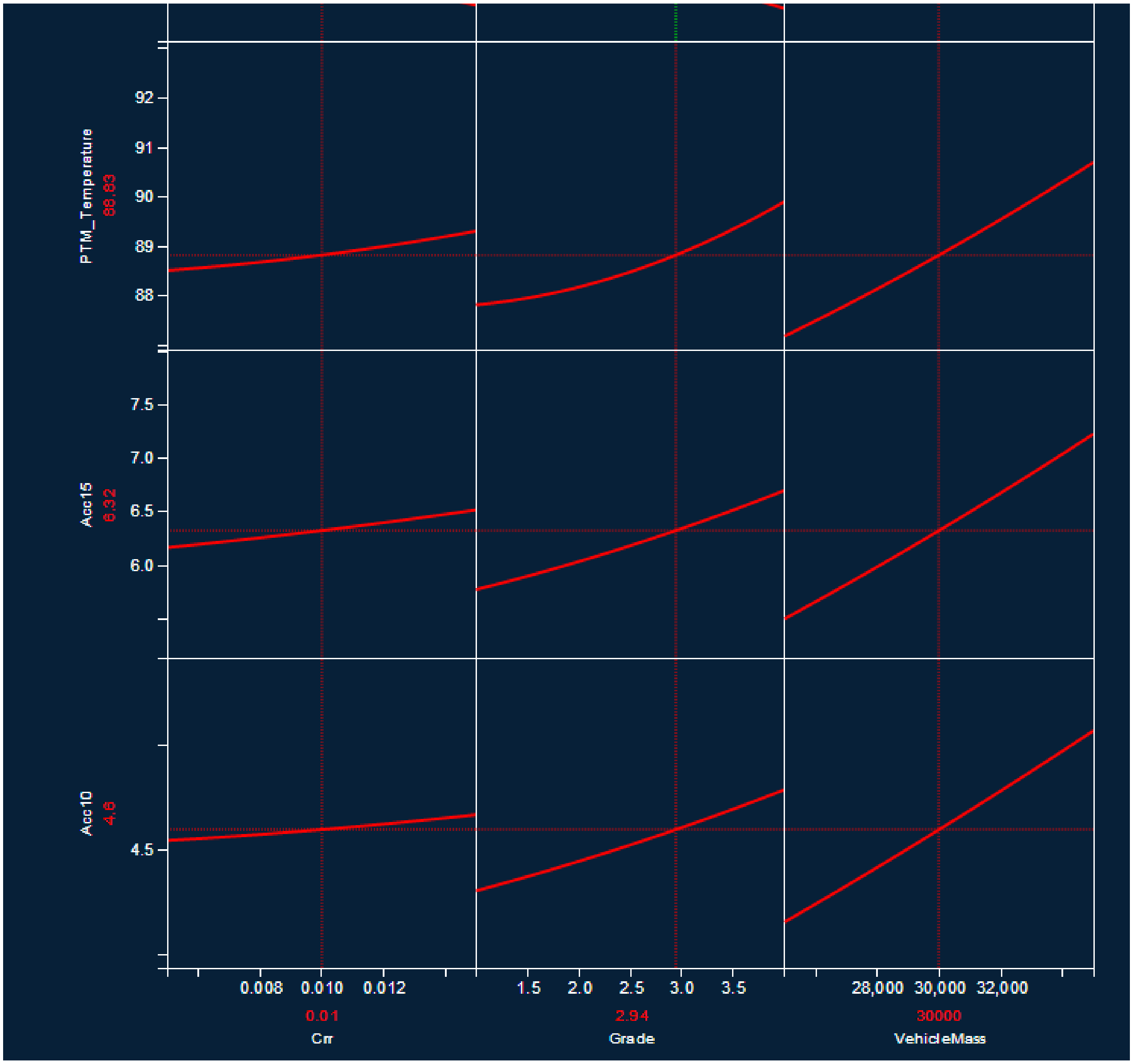
The colored regions within the constraint plot are regions which are not feasible. Thus, the visualization shows the design space as if the constraints where encroaching upon it. Both the slider bars and the red dotted cross hairs may be used to change the parameter values. Since the responses are dependent on the parameters they will update after each parameter change. This visualization provides insight into opportunities to optimize against known constraints.

In order to see the slope or sensitivity of each constraint, the user can double-click on the constraint name to show a three-dimensional visualization. The constraint in marked at the intersection of the response surface and the flat constraint plane. The 3D view can be rotated to view the surfaces from different angles. The blue arrow indicates which side of the constraint plane is feasible.



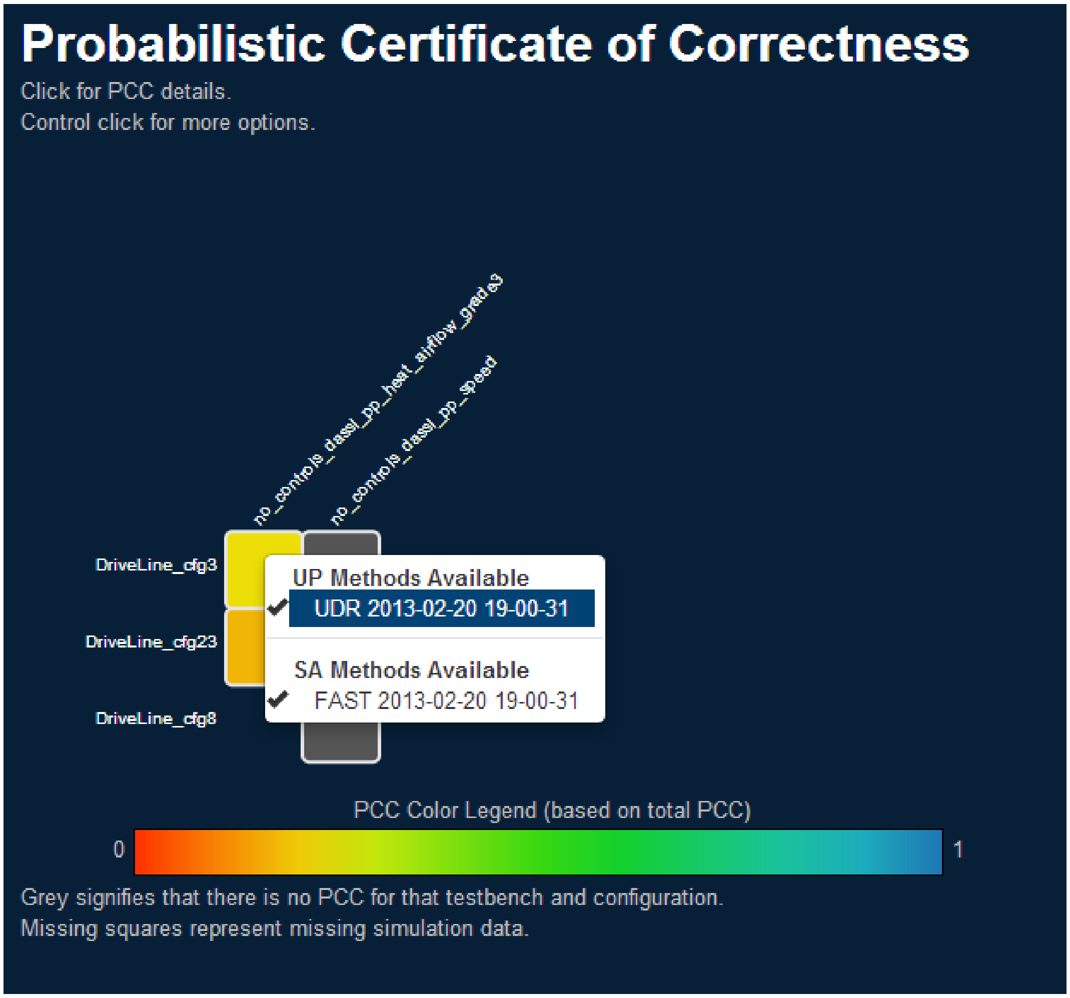
Below the constraint plot is the prediction profiler. This tool allows the user to quickly review the relationships between the parameters and the responses. The user can dial-in specific parameter values and instantly see the resulting response values by dragging the vertical cross hairs. The curves show the partial derivatives at the specific parameter values. The visualization also shows the maximum and minimum response values.



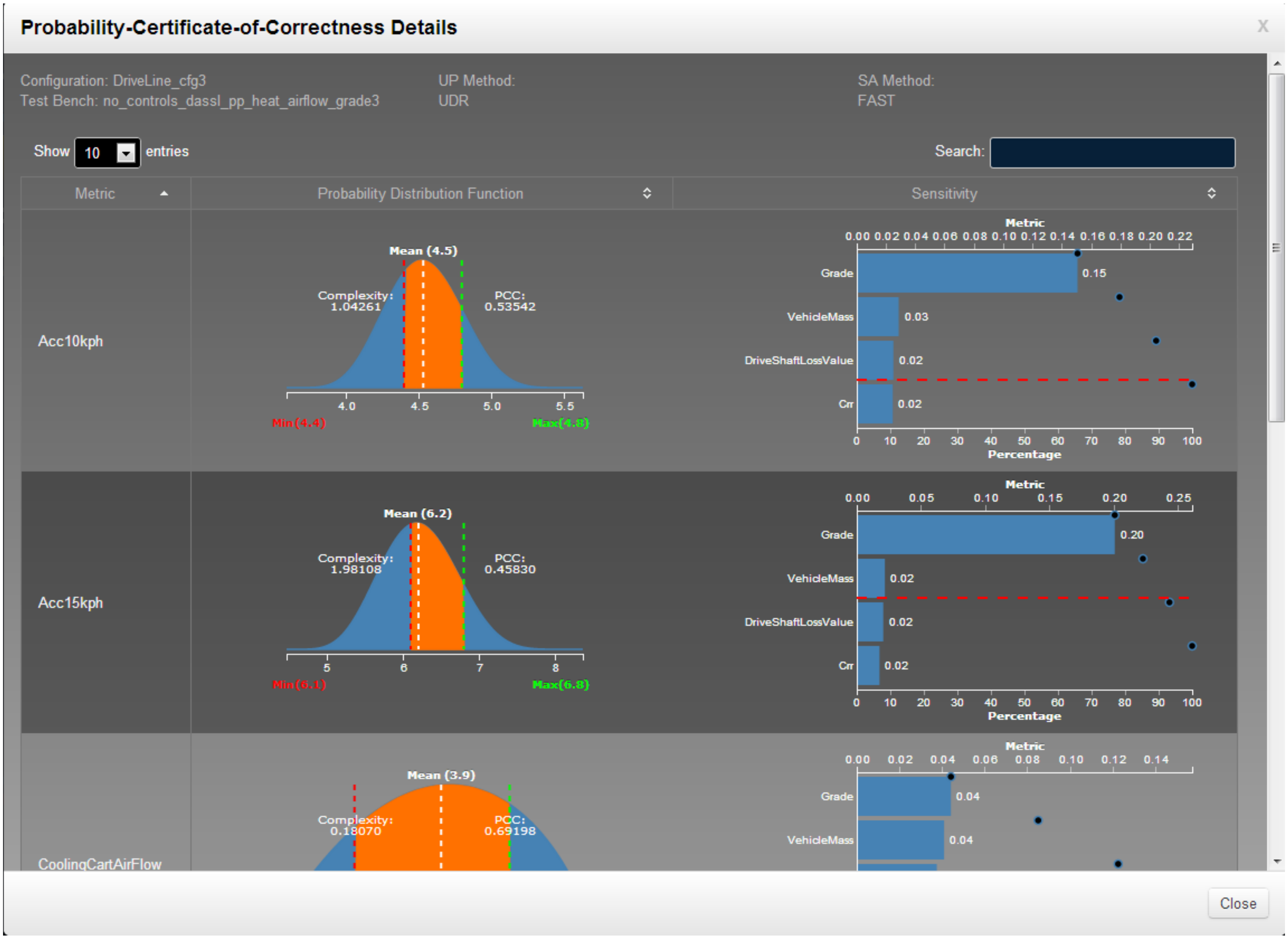


Probabilistic Certificate of Correctness

If the data set contains the results of a PCC analysis then the probabilistic certificate of correctness (PCC) tab will be available. This tab provides a measure of robustness with respect to upper and lower limits of the metrics. The overall robustness of a design is displayed within a heat map. The designs are automatically sorted by the first column. The user can change the column order by dragging the columns into a new order.



If the data set contains multiple sets of results the result sets are listed in a context menu which is available by ctrl-left-clicking a box in the heat map. A standard left-click will present the user with a more detailed view of the PCC results.



On the first column to the left is a list of the metrics from the selected test bench. The next column shows the resulting metric distributions from the uncertainty propagation analysis. The last column shows the sensitivity of each parameter to the metric specified in the same row. The probability distribution function also indicates the PCC index indicating the percentage of the results which fall between the defined limits. The complexity index is a measure of how complex the metric result is. Thus, the PCC index is a measure of the probability that the metric meets the requirements. The complexity index is a measure of level of interdependence within the metric calculation. The probability density function (PDF) provides a visualization of the distribution of the results of each metric.

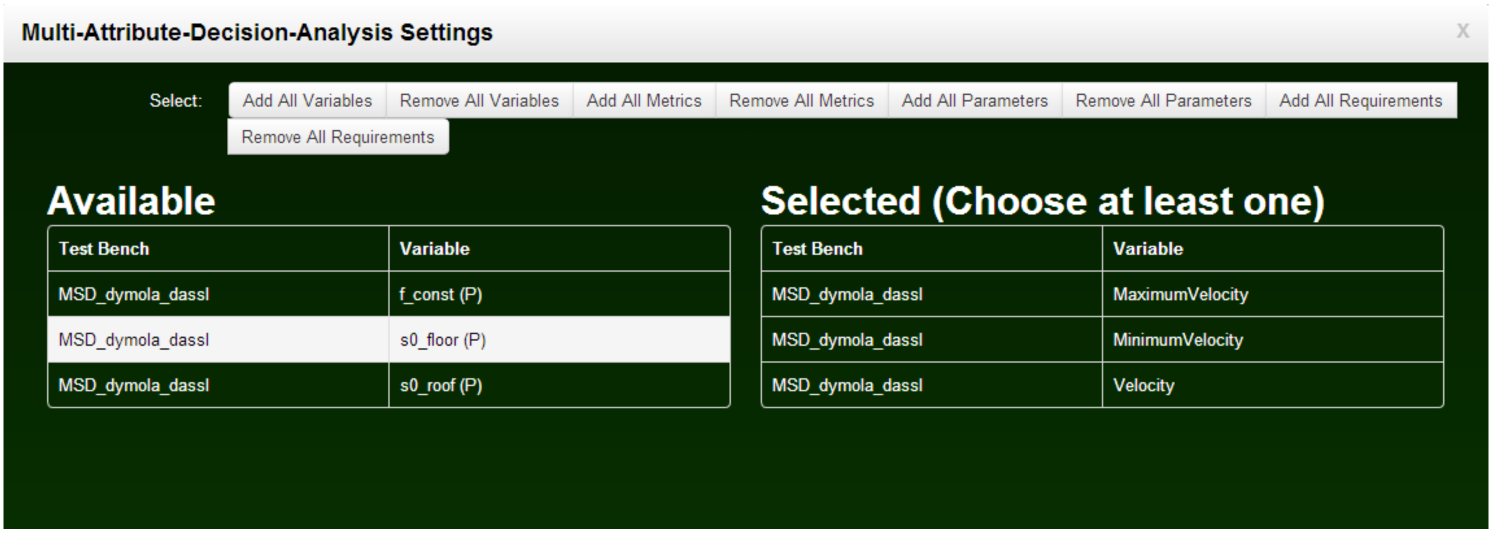
Sensitivity analysis is accomplished using a Pareto plot, which provides a visualization of which parameters control the most variability in reaching the metric goals. Pareto analysis uses the Pareto principle, which states that 80% of the variability of a design is controlled by 20% of the parameters. The red dashed line separates the top, most influential parameters (80% of the variability) from the bottom, less impactful parameters (20% of the variability).

Multi-Attribute Decision Analysis

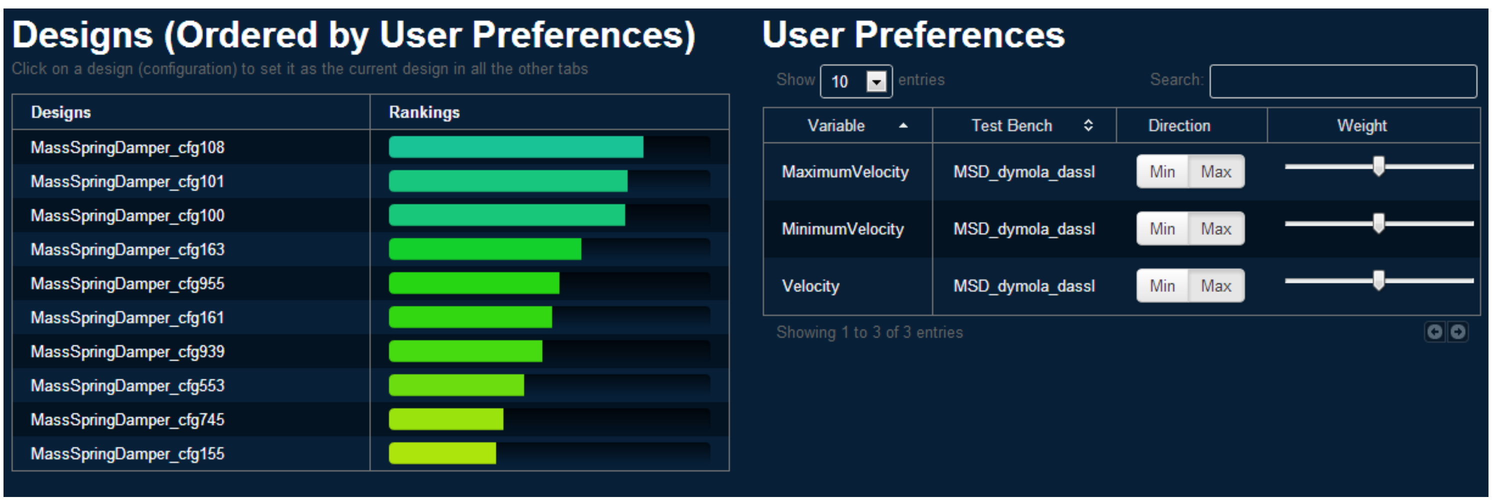
The Multi-Attribute-Decision-Analysis tab allows a set of designs (configurations) to be quickly ranked with respect to a user defined preference. The results are based on a modified technique called TOPSIS.

If this tab is opened without a valid set of preferences defined by the user, the settings dialog is automatically opened, giving the user an opportunity to select which variables to provide preferences for. All of the available variables (or variables without preferences) are listed in the left-hand plane of the dialogue. All of the selected variables (or variables with preferences) are listed in the right-hand plane. Clicking on a variable within either plane will toggle the variable status as having a preference or not having a preference. The multi-attribute decision analysis algorithm will ignore all variables without preferences. Each design is ranked relative to each other within the currently loaded dataset.

At the top of the dialog several shortcuts have been provided to help the user quickly select a subset of the available variables. Of course, all variables will either add or remove all of the variables. All metrics will add or remove variables which are outputs of test benches. All parameters will add or remove all variables which are inputs to test benches. Finally, all requirements will add or remove all variables which are associated with a requirement. Please note that the all requirements shortcut requires a valid requirements definition. Close the dialog box in order to finalize a selection of variables. The dialog box can always be reopened by pressing the settings button within the menu bar of the design space analyzer.



The body of the multi-attribute decision analysis tab is divided into a left and right plane. The left plane contains the ordered rankings of all of the designs (configurations) with respect to the user's preferences. The blue horizontal bars indicate relative closeness of a given design to the positive ideal design and the negative ideal design as defined by the TOPSIS technique. Thus, a design with a blue bar that spans its entire column is mathematically equivalent to the positive ideal design. A design with no bar is mathematically equivalent to the negative ideal design.



The right plane within the body of the multi-attribute decision analysis tab contains all of the variables with preferences. By default, each variable with a preference is set to a preference of 50%. A preference is defined by its direction and weighting. The direction indicates whether or not a variable should be maximized or minimized. The weighting indicates the relative importance of a variable compared to all other variables with preferences.

Limit Violations Summary

The limit violation summary tab shows occurrences of designs where maximum or minimum limits have been exceeded. The presence of a red exclamation point in the tab title indicates that limit exceeded errors were detected during the file load process. The table shows test benches and designs that produce the limit error. The name of the limit exceeded is displayed along with the predefined limit value and the actual value generated when the design was run through the test bench.

◄ **Previous: Chapter 8: The Master Interpreter** | **Next: Chapter 10: Scoring** ►

◄ **Previous: Chapter 9: The Dashboard** | **Next: Appendix A: MSD Component Documentation** ►

Chapter 10: Scoring

Prerequisite: Design Submission

Once you have committed your design successfully so the new Mass Spring Damper dataset is uploaded in the SVN repository (see Chapter 9:The Dashboard for details), your new design will be automatically submitted for scoring.

Step 1: Check Score of Submitted Design

In order to check the score or scoring status, first go to your team page and **click the Scoring Submissions button in the Analysis Tools menu on the right hand side.** . Please see the VehicleForge Guide for more information about how to navigate to your team's project space.

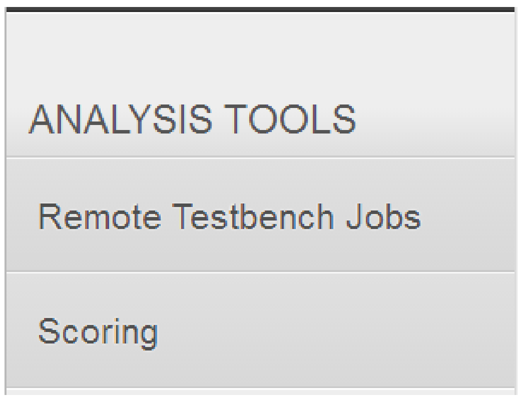
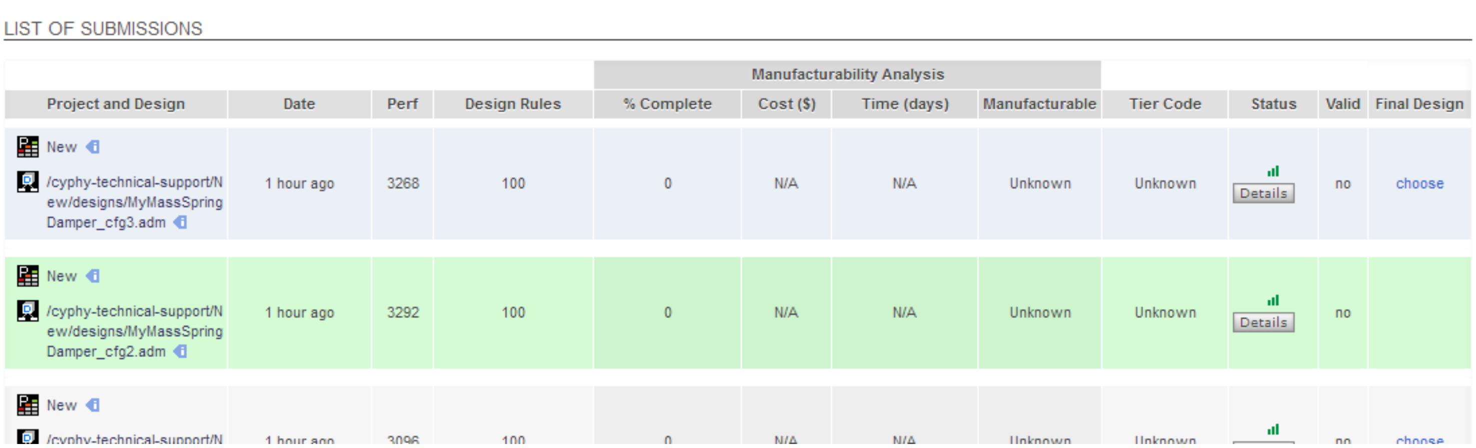


Figure 1: Analysis Tools Menu

Once a design is uploaded and submitted for scoring, the submission is added to the top of the LIST OF SUBMISSIONS table. You'll see your submission entry and the Details button in the Status column similar to Figure 2. Click the Details button to see any errors or warnings in the scoring.



LIST OF SUBMISSIONS

				Manufacturability Analysis							
Project and Design	Date	Perf	Design Rules	% Complete	Cost ($)	Time (days)	Manufacturable	Tier Code	Status	Valid	Final Design
New /cyphy-technical-support/New/designs/MyMassSpringDamper_cfg3.adm	1 hour ago	3268	100	0	N/A	N/A	Unknown	Unknown	Details	no	choose
New /cyphy-technical-support/New/designs/MyMassSpringDamper_cfg2.adm	1 hour ago	3292	100	0	N/A	N/A	Unknown	Unknown	Details	no	
New /cyphy-technical-support/N	1 hour ago	3096	100	0	N/A	N/A	Unknown	Unknown		no	choose

Figure 2: Scoring Status

For the MSD example, the requirements for MSD design are all in the category of Programatic and Performance, which is referred to as Perf in the columns of the table in Figure 2. At this point, you can disregard the scores in all other columns but the Perf column. If nothing goes wrong, the Status icon turns to green and you'll see a score.

You can see the MSD requirement at the VehicleForge front page by clicking the Design Requirements menu on the right hand side of the screen. The

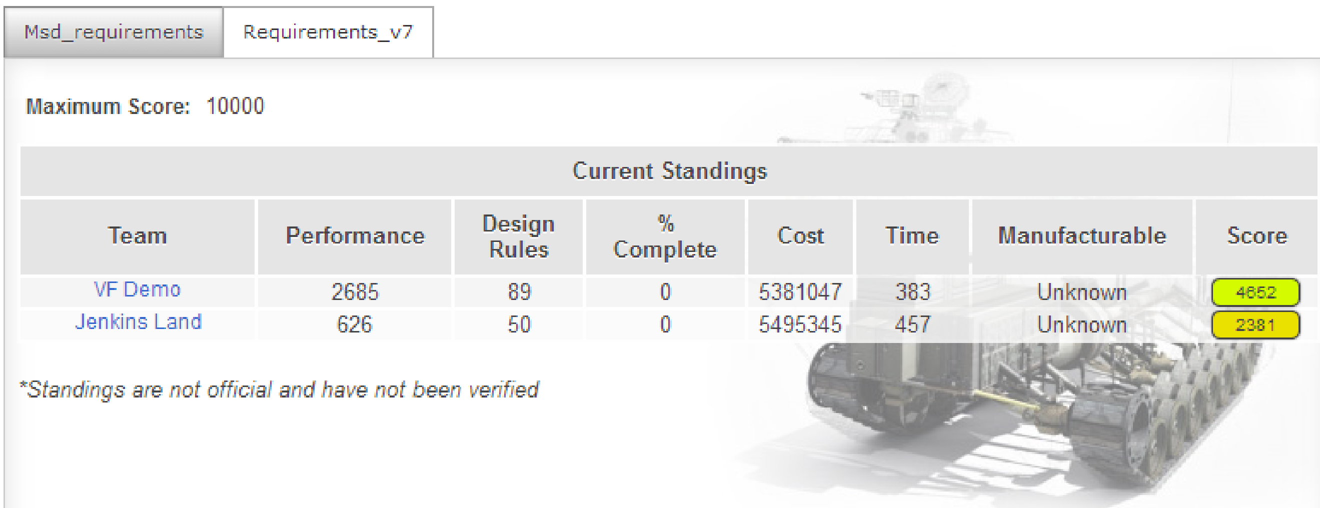
Design Requirements page also displays the maximum score you can get.

We recommend that you complete the exercise as written and receive a non "0" score in the Perf (Programmatic and Performance) column in the scoring submissions section, indicating that you have completed it successfully. At that point you are encouraged to explore variations to your design as you learn the CyPhy tools and improve your score.

Step 2: Select Final Design

If there are multiple design configurations with score available, you can now choose your team's final design from the LIST OF SUBMISSION table. To select one, find the design in the table and click the **choose** hyperlink in the Final Design column similar to Figure 2.

In the actual competition, the Leaderboard will be updated to show your team's current position in the competition based on your final design selection. The Leaderboard example is shown in Figure 3 which can also be found on the competition homepage. For the MSD example design, the Leaderboard won't be provided.


Msd_requirements | Requirements_v7

Maximum Score: 10000

Current Standings							
Team	Performance	Design Rules	% Complete	Cost	Time	Manufacturable	Score
VF Demo	2685	89	0	5381047	383	Unknown	4652
Jenkins Land	626	50	0	5495345	457	Unknown	2381

*Standings are not official and have not been verified

Figure 3: Leaderboard

◄ **Previous: Chapter 9: The Dashboard** | **Next: Appendix A: MSD Component Documentation** ►

◄ **Previous: Chapter 10: Scoring** | **Next: Appendix B: System Dynamics** ►

Appendix A: Mass Spring Damper Component Documentation

Mass

The mass component represents a frictionless rigid body that moves in one dimension with a parametric mass. This component has two interfaces, Hole and PIN what type of connections that interface represents. The component also has a parameter height that is driven by the selected mass of the system as well as the density of the mass; the width and length are constants.

Interface Summary

The interface summary below detains what types of interfaces are contained within each connection. These connectors define what types of external interactions are compatible with the component at that interface. For instance, a component could have an interface for HydraulicPower port twhich would be incompatible with a TranslationalFlange port.

Similarly the Structural Interface Definitions for two connectors must match for an interface to be compatible. A COMMON_INTERFACE structural interface definition is not compatible with a UNCOMMON_INTERFACE structural interface definition.

Connector Name	Structural Interface Definition	Power Interface
Hole	COMMON_INTERFACE	TranslationalFlange port
PIN	COMMON_INTERFACE	TranslationalFlange port

Properties and Parameters Summary

The table below summarizes the properties and parameters for the Mass components.

Name	Definition	Units	Property or Parameter
Mass	Mass of the system	kg	Parameter
Density	Density of the material used for the mass block	kg/m3	Property
Width	Width of the mass block	m	Property
Length	Length of the mass block	m	Property
Height	Height of the mass block	m	Parameter

CAD

The models for each mass is a solid block with parametric height and a lenght x width of 50 mm x 50 mm.

File format/language

Creo v2.0, Data Code M070

Modelica

This component is represented by the modelica standard library mass component. It is a frictionless, rigid body with two interfaces, flange_a and flange_b.

Governing equation

- m*a = flange_a.f + flange_b.f
- v = der(s)
- a = der(v)

where s is the position of the mass, v is the velocity of the mass, a is the acceleration of the mass, m is the mass of the mass, flange_a.f is the force applied at flange_a , and flange_b.f is the force on flange_b.

Domains

1-D translational

Spring

The spring component is a linear spring that contains a parametric spring constant and wire diameter and acts in dimension. This component class has four interfaces, INSIDE_PIN, INSIDE_HOLE, OUTSIDE_PIN, and OUTSIDE_HOLE. The inside interfaces are for the damper component which is properly sized to fit within the spring, while the outer interfaces can support both damper and mass connections without interference. The wire diameter for this component is affect by the choose spring constant and the shear modulus of the material of the spring; the mean coil diameter and number of active coils are constants for this spring class.

Interface Summary

The interface summary below detains what types of interfaces are contained within each connection. These connectors define what types of external interactions are compatible with the component at that interface. For instance, a component could have an interface for HydraulicPower port twhich would be incompatible with a TranslationalFlange port.

Similarly the Structural Interface Definitions for two connectors must match for an interface to be compatible. A COMMON_INTERFACE structural interface definition is not compatible with a UNCOMMON_INTERFACE structural interface definition.

Connector Name	Structural Interface Definition	Power Interface
INSIDE_HOLE	COMMON_INTERFACE	TranslationalFlange Port
INSIDE_PIN	COMMON_INTERFACE	TranslationalFlange Port
OUTSIDE_HOLE	COMMON_INTERFACE	TranslationalFlange Port
OUTSIDE_PIN	COMMON_INTERFACE	TranslationalFlange Port

Properties Summary

The table below summarizes the properties and parameters for the spring components.

Name	Definition	Units	Property or Parameter
Active Coils	Number of active coils for the spring	N/A	Property
Mean_Coil_Diameter	The average diameter of the coils	m	Property
Shear_Modulus	The shear modulus of the spring material	Pa	Property
Spring_constant	The spring constant of the spring	N/m	Parameter
Wire_Diameter	The diameter of the wire that the spring is made.	M	Parameter

CAD

The spring model is a helical sweep with a parametric wire thickness. It also has adapters at the ends for the damper and mass components to attach.

File format/language

Creo v2.0, Data Code M070

Modelica

This component is represented by the modelica standard library spring component in the mechanics.translational library. It is a linear 1-D spring with a parametric spring constant and two interfaces, flange_a and flange_b.

Governing Equation

$f = c$(s_rel - s_rel0)

Where f is the force on each flange, c is the spring constant, s_rel is the displacement from the initial position and s_rel0 is the initial position of the spring.

Domains

1-D Translation

Damper

The damper component is represented by velocity dependent damper with a parametric diameter. This component has two interfaces: Threaded_Pin and Threaded_Hole. The damper component has a parameteric damping constant as well as a parametric diameter. The damping_gain and damping_offest for this model are constants.

Interface Summary

The interface summary below detains what types of interfaces are contained within each connection. These connectors define what types of external interactions are compatible with the component at that interface. For instance, a component could have an interface for HydraulicPower port twhich would be incompatible with a TranslationalFlange port.

Similarly the Structural Interface Definitions for two connectors must match for an interface to be compatible. A COMMON_INTERFACE structural interface definition is not compatible with a UNCOMMON_INTERFACE structural interface definition.

Connector Name	Structural Interface Definition	Power Interface
Threaded_Hole	COMMON_INTERFACE	TranslationalFlange Port
Threaded_Pin	COMMON_INTERFACE	TranslationalFlange Port

Properties Summary

The table below summarizes the properties and parameters for the spring components.

Name	Definition	Units	Property or Parameter
Damping Constant	Damping constant of the damper	N*s/m	Property
Damping_offset	The average diameter of the coils	N*s/m	Property
Damping_Gain	The shear modulus of the spring material	N/A	Property
Spring_constant	The spring constant of the spring	mm	Parameter

CAD

The model for the damper is a series of extrusions. The larger cylinder has a parametric diameter.

File format/language

Creo v2.0, Data Code M070

Modelica

This component is modeled by the modelica standard library damper component from the Mechanics.Translational library. It is a linear, 1-D, velocity dependent damping element with two interfaces, Flange_a and Flange_b

Governing Equation

$f = d$v_rel;

where f is the force applied at each flange, d is the damping constant and v_rel is the difference in the velocity between flange_a and flange_b

Domains

1-D Translational

◄ **Previous: Chapter 10: Scoring** | **Next: Appendix B: System Dynamics** ►

◄ **Previous: Appendix A: MSD Component Documentation** | **Home:** Mass Spring Damper ►

Appendix B: System Dynamics Test Bench

1.0: Purpose:

Dynamics analysis allows for a design to be analyzed in an environment defined by the test bench. This allows for analyzing and comparing the performance of different parameter values, components choices, and architecture options throughout the design process to minimize design revisions late in the design.

In AVM, Dymola and OpenModelica are used to analyze the performance of a design in a multi-domain space. The models are able to support multiple levels of fidelity in a trade for compute time vs. accuracy of the generated results. OpenModelica will be used for the MSD tutorial provided with the RFI.

System Dynamics Test Bench

Requirements Tested

- **Stopped Moving:** Final state of the mass
- **Final Position:** Final Position of the mass after the test is run
- **Time to Zero:** time required for the mass to come to a complete stop

Theory of Operation

The design is analyzed by applying a load of 2 N is applied to the Pin port of the mass component while holding the Outside_Hole of the spring in a constant position. This position, velocity, and acceleration of the mass are then measured and post-processed to calculate the final position at the end of the simulation, the time required for the mass to stop moving and if the mass stopped moving.

Test Bench Structure

The test bench structure is covered in depth in **Chapter 6** of the MSD tutorial.

Metrics

- **Time_to_Zero:** The time required for the Mass to stop moving. If the mass does not stop moving within the simulation time, the value will be set to 10000 s.
- **Stopped_Moving:** This metric measure if the Mass stopped moving during the simulation. It is measured as a True/False binary
- **Final_Position:** This metric measures the final position of the mass away from its initial position. Positive means the spring compressed and negative is the spring extending.

◄ **Previous: Appendix A: MSD Component Documentation** | **Home:** Mass Spring Damper ►