



TIRE ANALYSIS SOFTWARE

Help File
Version 1.0.14

By



►Seminars ►Consulting ►Software Development

8801 E. Hampden Ave.
Suite 210
Denver, CO 80231 (USA)

Phone: 303-752-1562
Fax: 303-368-0121

engineering@optimumg.com

www.optimumg.com

Welcome

Thank you for purchasing OptimumT the new benchmark in tire model fitting and analysis. This help file contains information about all the functions and features of OptimumT. Included in this help file are:

- Information on how to use OptimumT
- Special Features in OptimumT
- Tips and tricks on how to use OptimumT efficiently

Feedback

OptimumT is a continually developing program and we give high regard to any suggestions, comments, complaints or criticisms that OptimumT users might have. Please contact us at engineering@optimumg.com and we will work to improve OptimumT based on your feedback.

Features Coming Soon

The following is a list of features that will be available in future updates. Please contact us if you have any suggestions on other features that should be included in OptimumT in the future.

- Additional Tire Models
- Import and Export from TYDEX Files
- Automated Reports
- Embeddable Live Graphs

Contents

Welcome	2
Introduction	8
Installation Requirements	9
License	11
1 Screen Layout	13
1.1 Tire Project Tree	14
1.2 Data Entry Form	16
1.3 Worksheets	17
2 Raw Tire Data	19
2.1 Importing Data	21
2.2 Data Cropping	23
2.2.1 Data Cropping Templates	25
2.3 Data Collapsing	26
2.4 Exporting Data	31
3 Tire Models	32
3.1 Manually Input Model	32
3.2 Import and Export Models	35
3.2.1 Export Templates	36

3.2.2	Excel Export Templates	37
3.3	Fitting Models to Raw Data	38
3.3.1	Pure Lateral Model	39
3.3.2	Aligning Torque Model	50
3.3.3	Model Scaling Factors	53
3.3.4	Pure and Combined Longitudinal Model	58
3.3.5	Combined Lateral Model	61
3.3.6	Additional Models	63
3.4	Model Fitting Order	64
3.5	Model Coefficient Form	66
3.6	Adjusting Models	67
3.7	Creating Coefficient Boundaries From an Existing Model	69
4	Graphing	70
4.1	Standard Graphs	70
4.1.1	Types of Standard Graphs	70
4.1.2	Setting up a Standard Graph	73
4.1.3	Graph Outputs	74
4.1.4	Graph Inputs	76
4.1.5	Plotting Raw Data	80
4.1.6	Copy Graph Inputs from Collapsed Data	81
4.1.7	Linking Graphs	83
4.1.8	Coloring Graphs	84
4.1.9	Graph Templates	87
4.2	Report Graphs	88
4.3	Graph Area Options	89
4.3.1	Display Legend	90
4.3.2	Axis Zero Cross	92
4.3.3	Data Selection Cursor	93

4.3.4	Copy and Print Graphs	94
4.3.5	Export Graph Data	95
5	Custom Models	97
5.1	Introduction to Custom Models	97
5.2	Creating Custom Models	97
5.2.1	Software Requirements	98
5.2.2	Creating the Custom Model Project	98
5.2.3	Referencing the Custom Model Template	99
5.2.4	Creating a Coefficient File	101
5.2.5	Creating a Calculation File	104
5.2.6	Common Programming Errors	105
5.3	Importing Custom Models in OptimumT	106
5.4	The Custom Model Fitting Process	107
5.5	Custom Model Import and Export	109
6	Additional Features	110
6.1	OptimumT Add-in	110
6.1.1	Using the Add-in with Excel	111
6.1.2	Matlab COM Add-in	117
6.1.3	Using the COM Addin in Programs	119
6.2	Template Manager	121
6.3	Project Backups	122
6.4	Error Evaluation	122
6.5	Lookup Table Export	123
7	Tips and Tricks	127
7.1	Plot All Data	127
7.2	Large Tolerance Graph Inputs	129
7.3	Override Default Name in Legend	129

7.4	Changing Units	131
7.5	Preview Model Coefficient Change	132
7.6	Hide Axis Values	133
7.7	Importing Multiple Data Files	133
8	References	134
8.1	Example Files	134
8.2	Coordinate Systems	134
8.3	Units	137
8.4	Fiala Model	138
8.5	Harty Model	138
8.6	Brush Model	140
8.7	Pacejka Models	141
8.7.1	Pacejka '96	142
8.7.2	Pacejka 2002	142
8.7.3	Pacejka 2002 with Inflation Pressure Effects	142
8.7.4	Magic Formula 5.2	142
8.7.5	Pacejka 2006	142
8.8	Pacejka Coefficients	143
8.9	Pacejka Scaling Factors	146
	Additional Information	148

Introduction

Through OptimumG's consulting and seminar services, it identified tire data analysis and model fitting as an important, but often neglected area of vehicle dynamics. Therefore, OptimumT was created from OptimumG's extensive knowledge and experience in testing, analysis, and development of vehicle tires and suspension systems. OptimumT is a convenient and intuitive software package that allows users to perform advanced tire data analysis, visualization, and model fitting.

Pacejka '96, 2002, and 2006 tire models are available in OptimumT. Simpler Brush and Fiala models are also included. The models can be manually inputted, imported, or fit to raw tire data. The model fitting procedure is very fast and efficient partially due to the data processing tools incorporated into the software. All aspects of the tire models can be easily compared to the raw tire data to ensure accuracy. OptimumT also allows adjustment of the tire models if necessary and includes scaling factors for the Pacejka models. These features ensure that the tire models created will accurately correlate to data collected from the road or race track. These tire models, as well as the processed data, can be exported from OptimumT to a variety of file formats.

OptimumT is also a very powerful data visualization tool. It has the capability to display tire models and raw tire data in a user friendly but extremely powerful graphing utility. It features 2D and 3D graphing of over 30 different tire parameters. The ability to create surface plots and crossed line graphs allows the user to be able to easily create sophisticated graphs, like tire friction ellipses. The graphs can be further enhanced by including the effect of many different parameters including vertical load, inclination angle, and inflation pressure. These graphs can be easily copied or directly printed from the OptimumT interface to be included in other projects or reports. The data from these graphs can also be exported for further analysis.

Installation Requirements

Hardware Requirements

Processor

- Intel® Pentium 4™, Intel® Xeon™, and Intel® Core™
- AMD® Athlon™, AMD® Opteron™, and AMD® Turion™

Memory

- Minimum: 512MB RAM
- Recommended: 1GB RAM or more
- Virtual Memory recommended to be twice the amount of RAM

Storage

- Free space of at least 100MB. Includes installation of OptimumT® and required software components (see below)

Display Adapter

- Minimum: Microsoft® DirectX® 9.0c capable graphics card with 32MB RAM
- Recommended: DirectX® 9.0c capable NVIDIA® GeForce® or ATI® Radeon® with 128MB RAM or higher

Display Unit

- Minimum: 15" Screen with resolution of 1024 x 768 pixels
- Recommended: 19" Screen with resolution of 1280 x 1024 pixels

Other

- Mouse or other pointing device

Software Requirements

Operating System

- Microsoft® Windows® XP (32 or 64bit) or Microsoft® Windows® Vista (32 or 64bit) or Microsoft® Windows® 7 (32 or 64bit)

Required Software Components

- Microsoft® .NET® Framework 2.0 or higher

Other

- Microsoft® Excel™ version 10 or higher for import or export of data

License

Moving License

If the license needs to be moved to another computer this can be done through OptimumT. The two computers will both need to have an internet connection.

To move a license to another computer:

- Launch OptimumT.
- From the project screen select the Help menu and click Deactivate License.
- Click Yes in the Deactivate License message box.
- Click Deactivate in the Web Activation message box.

The "Online Deactivation was successful" message box will appear once the license has been deactivated. OptimumT can now be installed on another computer using the same license key.

License Viewer

The license viewer shows the status of the current license and allows the user to enter a new time license. Once OptimumT is installed the license viewer can be accessed through the Windows start menu. It will be located in the OptimumT folder. A new OptimumT time license can be activated by pressing the Activation Key button in the lower left corner of the license viewer. Then the new license key can be inputted and activated.

License Agreement

OptimumT is protected by copyright and other intellectual property laws and treaties. OptimumG owns the title, copyright, and other intellectual property rights in the Software. The Software is licensed, not sold. Purchase of OptimumT does not grant you any rights

to trademarks or service marks of OptimumG. For more information regarding the rights and limitations of the OptimumT License please refer to the End-User License Agreement (EULA).

Chapter 1

Screen Layout

When OptimumT is opened the project screen will initially appear as shown in Figure 1.1. From this screen the user can either choose to open an existing project or create a new project. Once one of these is selected the primary OptimumT screen layout will be displayed.

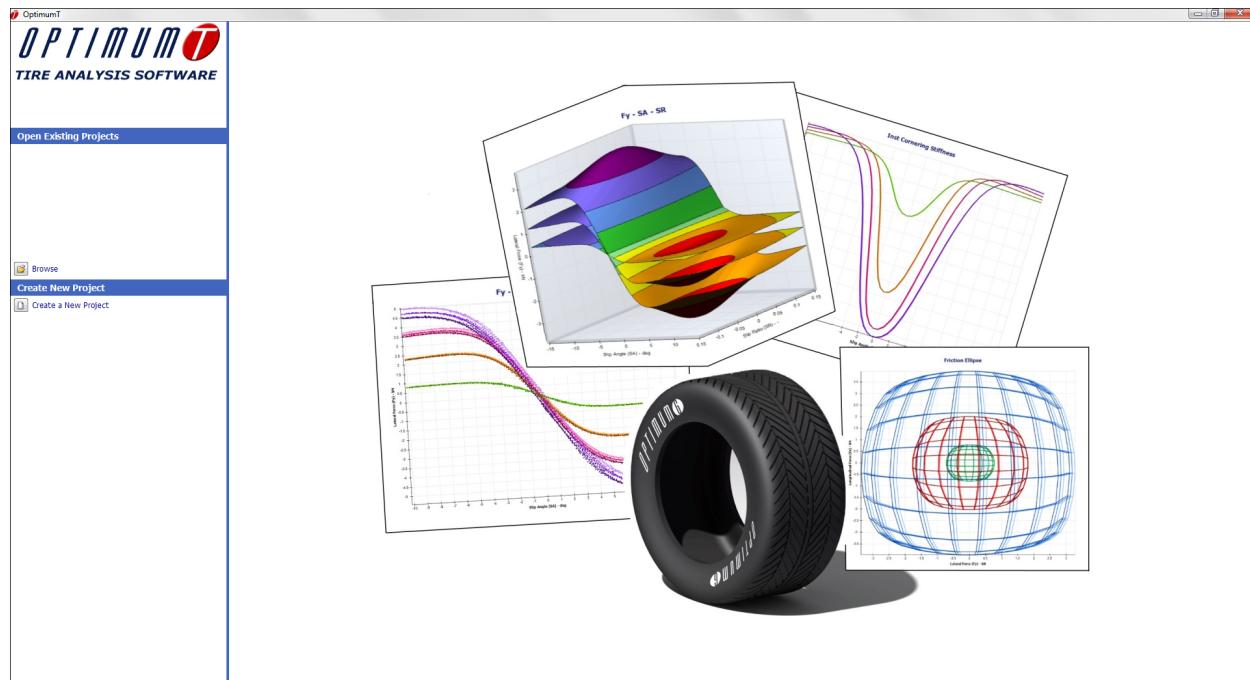


Figure 1.1: Project Screen

The OptimumT screen layout is shown in Figure 1.2. The screen is divided into three basic sections:

- The tire project tree

- The data entry form
- The worksheets and the associated worksheet tabs

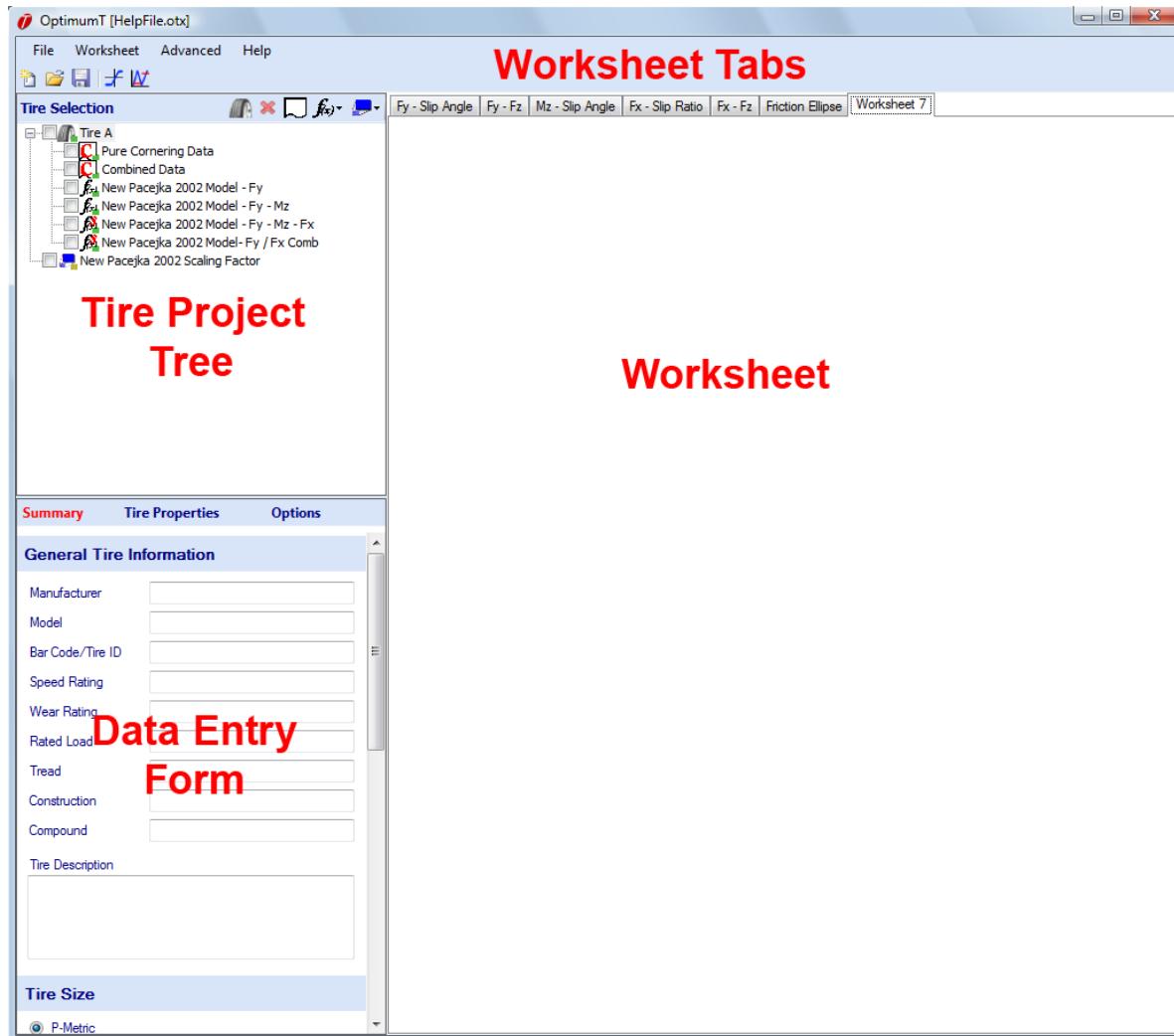


Figure 1.2: Screen Layout

1.1 Tire Project Tree

The tire project tree contains all the raw data, tire models, and scaling factors contained within the OptimumT project. Raw data and tire models are organized by the tire item. The tire items can be seen more closely in Figure 1.3.

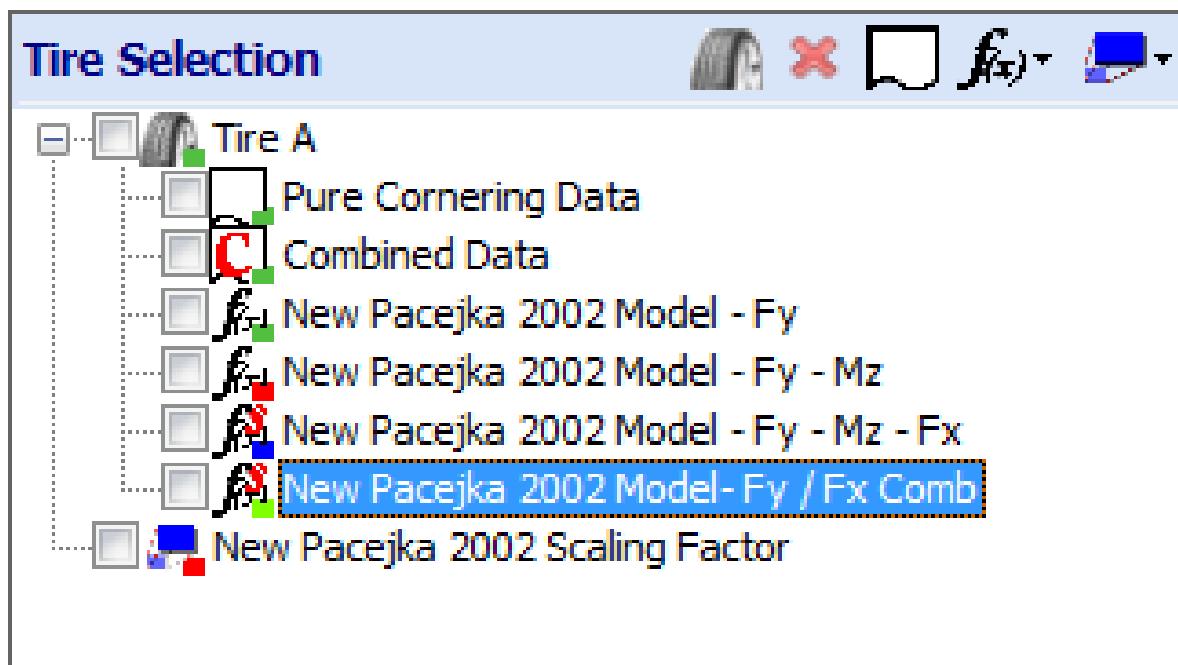


Figure 1.3: Project Tree

The item type in the project tree can be identified by the icon to the left of the item. In Figure 1.3, the first item in the list is a tire. The tire item is like a folder that contains other items in the project. Both raw data and tire models must be associated with a tire item, so this must be the first item added to a new project. This allows data and models from the same tire or construction to be grouped together.

Raw Data

In Figure 1.3, the first two items in the project tree under Tire A are raw data. The icon for the second item, Combined Data, has a large red "C" in it to indicate that the collapsed data is being used. Data collapsing will be explained in section 2.3.

Tire Models

The third and fourth items in this project tree are tire models. The next two items are also tire models but the red "S" in the upper right corner of the icon indicates that a scaling factor has been applied to them.

Scaling Factors

The final item is a Pacejka scaling factor. These allow the model to be adjusted without making any changes to the model coefficients. Scaling factors are discussed in more detail in section 3.3.2.

Like the tire item when any of the raw data, tire models, or scaling factors is clicked additional information and functionality related to the selected item will appear in the data entry form. The small color squares in the lower right corner of the icons indicate the color in which the data or model will be displayed when it is graphed. To graph a specific tire model or raw data set, check the box next to the item in the project tree (graphing is covered in section 4). By right clicking on an item you can rename, delete, or copy it as well as perform other operations on it that will be discussed in later sections.

To add items to the project, the buttons above the project tree can be used. Figure 1.4 shows these buttons. The three buttons furthest to the right will only be enabled when an item in the project tree is selected.

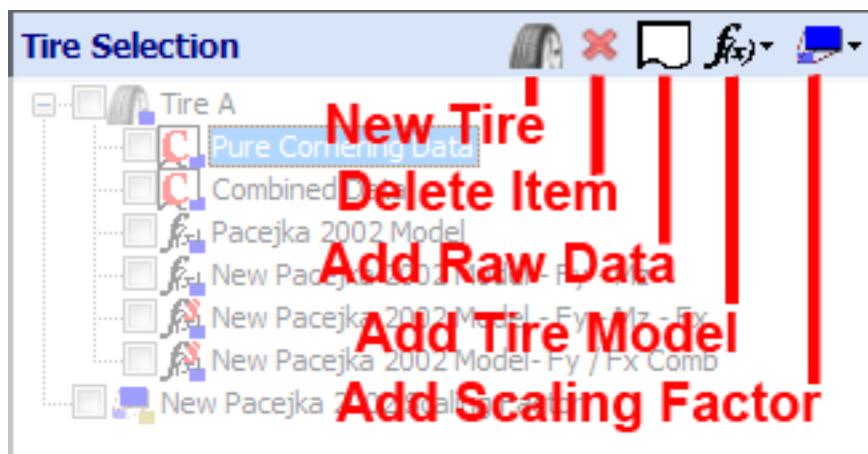


Figure 1.4: Project Tree Buttons

1.2 Data Entry Form

When an item in the project tree is clicked on, a data entry form corresponding to the project item will appear in the data entry area. Note that the name of the project item or its icon must be clicked on. Clicking on the checkbox next to the item in the project tree will not bring up the data entry form, but will change whether or not the item is to be graphed.

Figure 1.5 shows an example of the data entry form. This information appears when a tire item is selected in the project tree. Information about the tire size, manufacturing, and testing procedure can be stored here.

If the raw data, tire model, or scaling factor items are selected the information in the data entry area will change to reflect these items. The tire model coefficients are also contained in this form. If a graph is clicked on, the graph setup form for that specific graph will appear in the data entry area. The data entry forms will be discussed in more detail in the chapters corresponding to these specific items.

Summary	Tire Properties	Options
General Tire Information		
Manufacturer	<input type="text"/>	
Model	<input type="text"/>	
Bar Code/Tire ID	<input type="text"/>	
Speed Rating	<input type="text"/>	
Wear Rating	<input type="text"/>	
Rated Load	<input type="text"/>	
Tread	<input type="text"/>	
Construction	<input type="text"/>	
Compound	<input type="text"/>	
Tire Description	<input type="text"/>	
Tire Size		
<input checked="" type="radio"/> P-Metric <input type="text"/> / <input type="text"/> R <input type="button" value="▼"/> <input type="text"/>		

Figure 1.5: Tire Item Data Entry Form

1.3 Worksheets

The worksheet is the area where graphs can be added to the project. Creating graphs will be covered in section 4. An unlimited number of worksheets can be included in a project. To add a new worksheet, click on the *Worksheet* menu and select *New Worksheet*.

Alternatively, worksheets can be added by right-clicking on the worksheet or the worksheet tabs and selecting *New Worksheet*.

Worksheets can be renamed by right-clicking on the worksheet and choosing *Rename Worksheet*. A worksheet can be deleted (and all the graphs on that worksheet) by choosing *Delete Worksheet* from the menu that appears when you right-click on a worksheet. Note that you cannot undo a delete operation.

Chapter 2

Raw Tire Data

The raw data item in the project tree contains the imported data and provides some tools for manipulating the data. These tools allow the data to be quickly and conveniently viewed and fitted to tire models.

The raw data form, shown in Figure 2.1, is displayed when a raw data item in the project tree is clicked on. This form has a place to store comments about the test data contained in the item. It also contains the *Data Cropping* and *Data Collapsing* tools. The *Data Cropping* tool allows user to easily view and eradicate erroneous or undesired data from the raw data. The *Data Collapsing* tool removes hysteresis from the data and separates the data into sets depending on the conditions that the tire was tested at. Once the data has been collapsed a summary of the separated data sets will appear in the table in the raw data form. The *Model Fitting* tool in this section allows you to fit the raw or collapsed data to a tire model. Tire model fitting will be covered in more detail in section 3.

The Options button at the top of the raw data form allows you to add another data file, export the tire data, or access the *Data Cropping* tool. A more detailed description of these operations is included in this section. The Data Comments allow the user to enter notes and information about the data. This information will be saved with the raw data in OptimumT.

Summary Raw Data Page Options

Data Comments

Override default name in legend

Collapse Data

Use Collapsed Data

 Selected Test Cases Sort by Tolerance

Use Row	Vertical Load	Inclination Angle	Speed	Pressure	Data Points

Vertical Load Unit: N Inclination Angle Unit: deg Speed Unit: m/s Pressure Unit: bar

Model Fitting

Pacejka 2006

The steady-state model given in the second edition of 'Tire and Vehicle Dynamics' by Hans Pacejka in 2006. This model is similar to the 2002 Pacejka model, but the pure lateral and aligning torque models are modified. An additional parameter is also included in both the combined lateral and combined longitudinal model.

Figure 2.1: Raw Tire Data Form Before Collapsing

2.1 Importing Data

When the *Add Raw Data* button above the project tree is clicked on, an open file window will appear. OptimumT can open either a .rtd file, which is the OptimumT binary format for tire data, or a CSV or ASCII file with .csv or .dat file extensions. Files with no extension are assumed to be CSV/ASCII files.

If a .rtd file is opened, the data will be automatically imported in the correct format and coordinate system. No further actions are required by the user.

If a CSV or ASCII file is selected a dialog box as shown in Figure 2.2 will open. In this box the the file properties can be specified. The character that separates the columns in the file and then the character that represents the decimal point should be selected. Multiple column separators can be selected if different file formats are to be used.

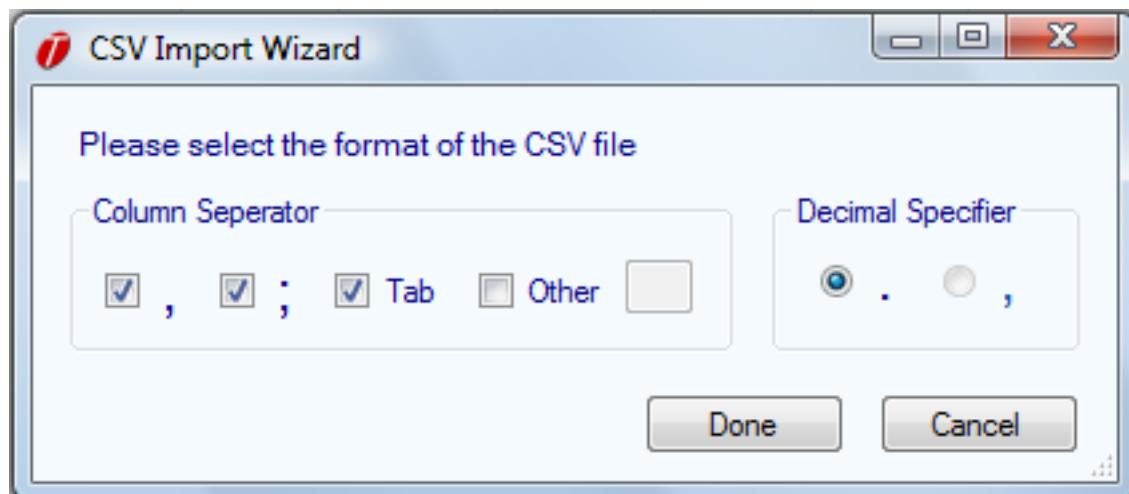


Figure 2.2: CSV Import Options

If a CSV or ASCII file is opened, the data import wizard, shown in Figure 2.3, will appear. In this window the user specifies what quantity each column of raw data contains (i.e. SA, SR, Fx, etc) and the unit for that quantity. At the bottom of the dialog box, default values for quantities that are missing from the data can be specified. For example, if inflation pressure was not recorded in the test, the user could manually enter a constant inflation pressure to be included in the data. The coordinate system that the data was collected also needs to be specified.

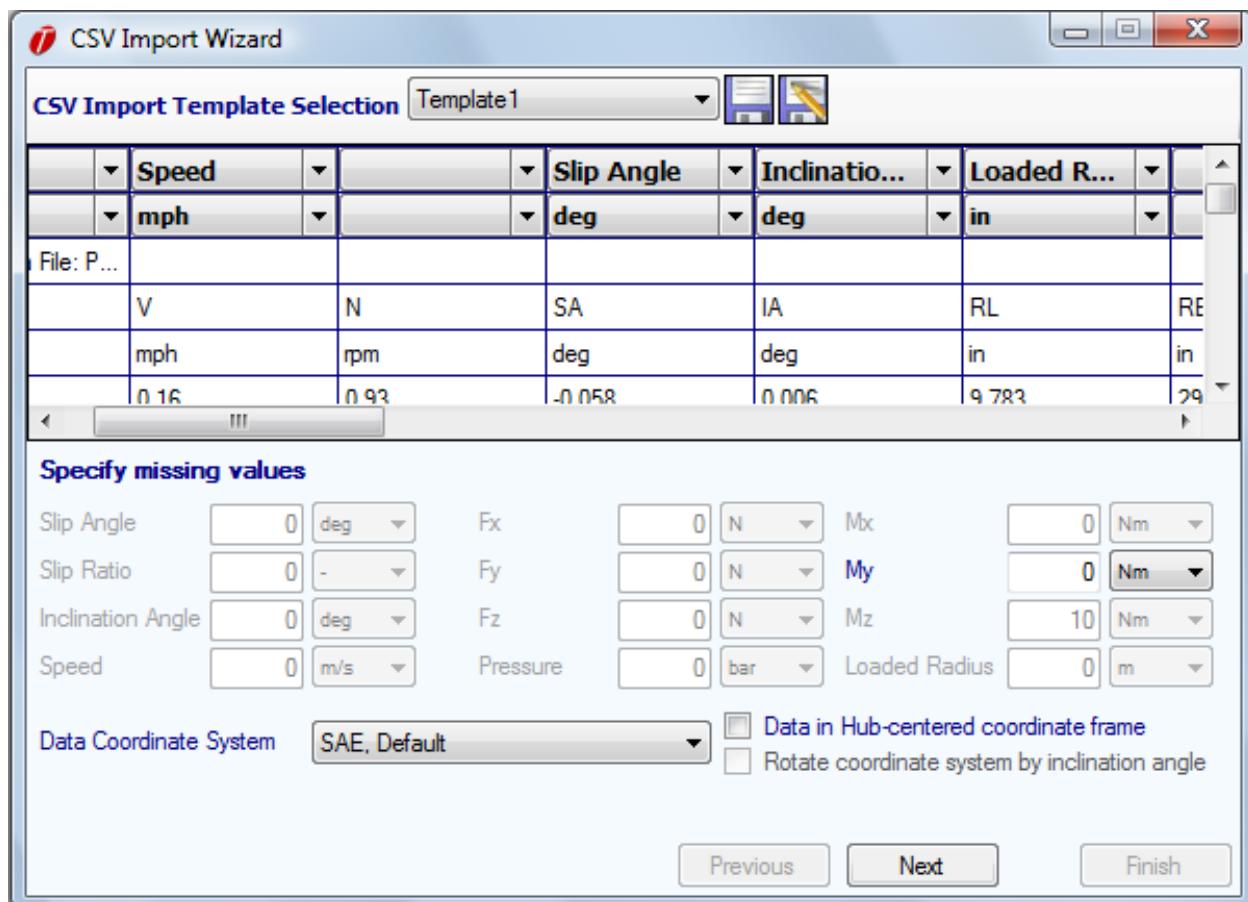


Figure 2.3: Data Import Wizard

This process can be automated by using the import template feature at the top of Figure 2.1. The template to be used is chosen through the *CSV Import Template Selection* dropdown box. It can be seen in the figure that currently Template 1 is selected. New import templates can easily be created and saved. First the data quantities, units, and coordinate system are specified. Then to save this as an import template, click the *Save as New Template* button. Import templates can also be modified and saved by using the *Save Template* button.

Typically tire data provides the force and moments at the center of the tire contact patch. However sometimes, especially with wheel force transducer data, it will be the force and moments at the wheel center. Therefore the *Data in Hub-centered coordinate frame* checkbox should be selected. Then OptimumT will transform the force and moment data to the center of the contact patch. If this is the case the *Rotate coordinate system by inclination angle* can also be selected. This should be done if the coordinate system that the force and moments were measured in are fixed to the wheel. In this case the vertical force would be in the same direction as the inclination angle and not perpendicular to the ground. Therefore if this option is selected OptimumT will transform the forces and moments to a ground fixed coordinate system.

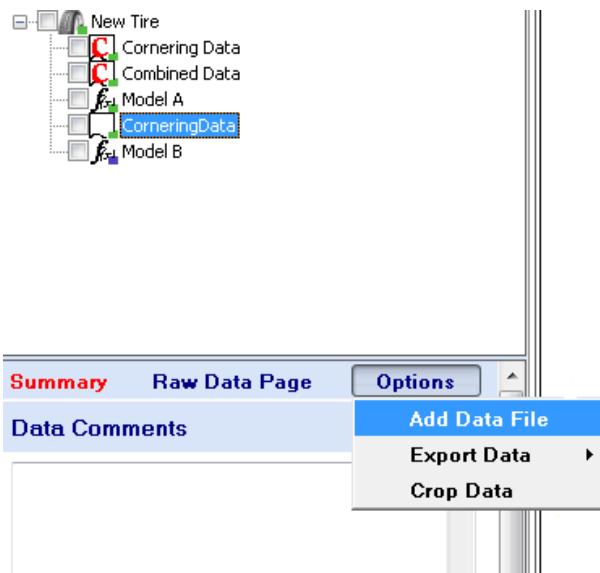


Figure 2.4: Adding more data to a data set that has already been imported

Once all of the column definitions have been assigned pressing the Next button will show a preview of the data to be imported. If the data to be imported is correct, clicking on Finish will import the data into OptimumT.

Generally different sets of raw data should be imported into OptimumT as separate files. However multiple test files can be imported and combined in OptimumT. This can be achieved in two ways. If the files are in exactly the same format, simply select both files at the same time (use the shift or ctrl keys to choose more than one file). The import wizard will appear for the first file and the other files will be imported with exactly the same settings.

If the files are in different formats, multiple import into the same data item is done by importing the first data set as would normally be done. After this is completed, click on the data set in the project tree. Then select Add Data in the Options button in the upper right corner of the raw data form (see Figure 2.4). This will open the same CSV Import Wizard that was used previously. Follow the same steps as before and the data will be added to the previously imported set.

2.2 Data Cropping

Data cropping allows the user to remove any unwanted or unnecessary data. Often in tire tests the data from conditioning or warm up procedures is included in the data file. This data can be easily removed in OptimumT. The raw data can be cropped by selecting *Crop Data* under the *Options* button on the top of the raw data form. This will open the *Crop Data* window as shown in Figure 2.5. The raw data to be cropped is displayed in the graph. You can select what properties of the raw data are graphed by selecting the checkboxes on

the left. In the figure slip angle is represented by the blue line and the inclination angle by the red line. By clicking on the boxes to the right of the properties the color of the property can be changed. The values to the right of this correspond to the data values at the location of the black line on the graph. At the bottom the units of the data and the axis ranges can be changed.

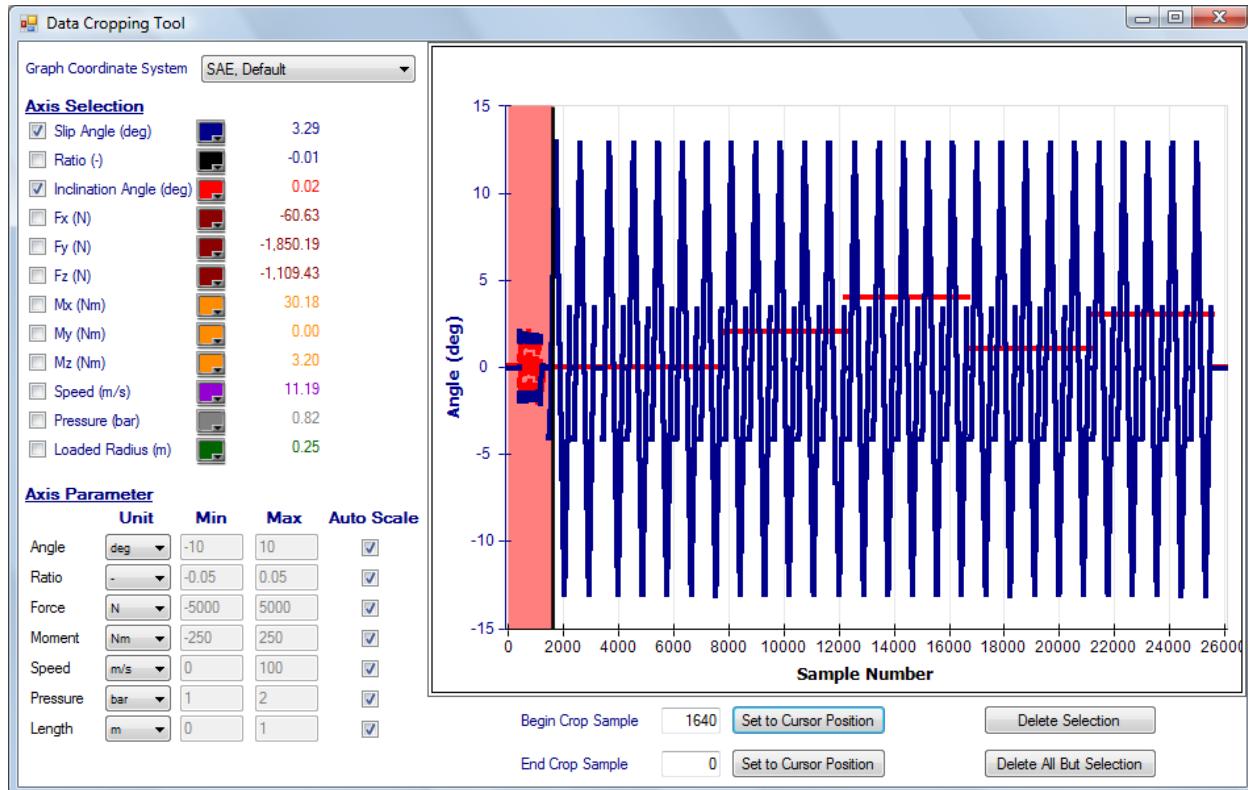


Figure 2.5: Data Cropping Tool

As can be seen at the beginning and end of the run extra data exists that is not necessary. The data to be removed can be selected by entering the beginning and ending sample numbers in the *Begin Crop Sample* and *End Crop Sample* textboxes at the bottom of the window. Alternatively the vertical black line on the graph can be dragged to the point where the data should be cropped. Pressing the corresponding *Set* button defines the beginning or ending sample number. The background of the selected data will be pink. Then the *Delete Selection* or *Delete All But Selection* buttons can be clicked to remove the selected data. Figure 2.6 displays the data from Figure 2.5 after it has been cropped. As can be seen only the relevant test data remains. When the Crop Data is closed the program will return to the primary OptimumT screen with cropped data.

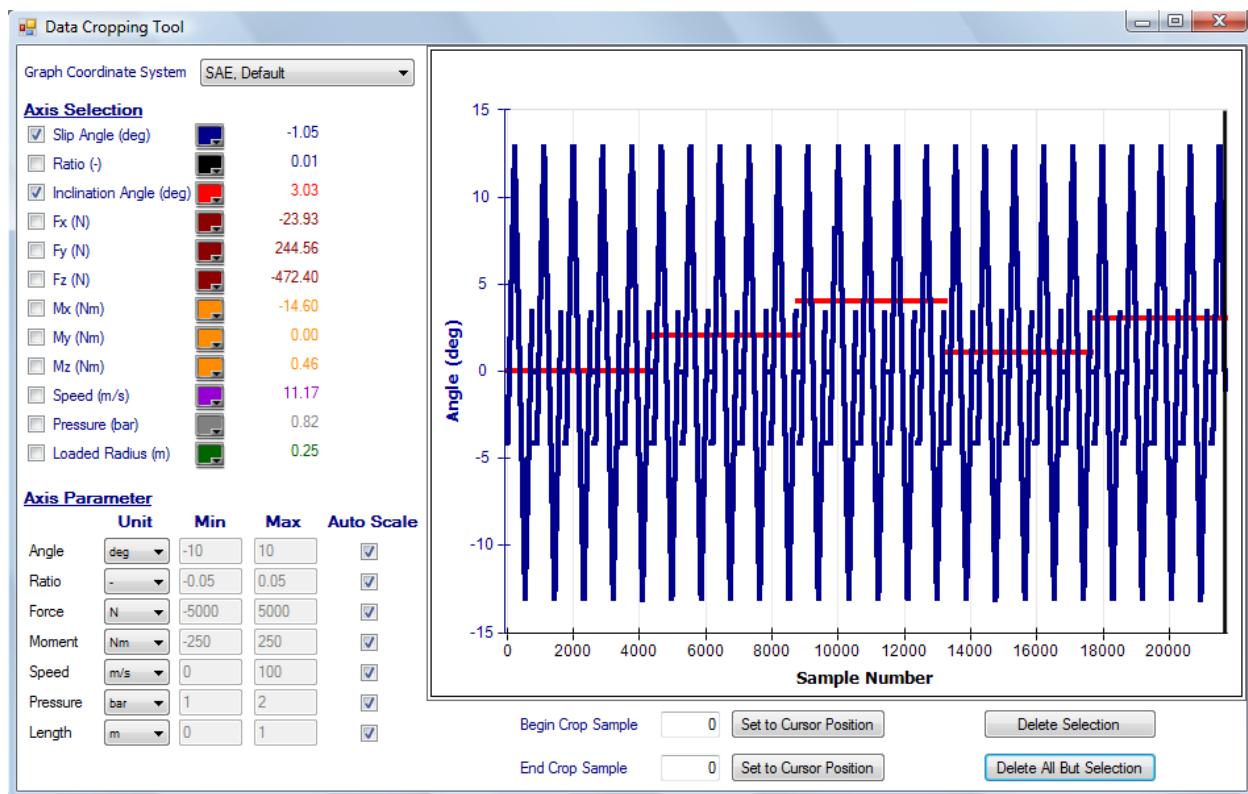


Figure 2.6: Cropped Data

2.2.1 Data Cropping Templates

To simplify the task of repetitively cropping data, templates can be set up. Crop templates record the beginning and ending sample number of each section deleted. The same actions can be applied to another data file. The file on which a template is used must be at least as long as the largest ending sample number in the template. A warning will be displayed if the crop template was created for a data file with a different number of samples than the one its being applied to.

To record a crop template, simply crop a data file as usual. Once this is done, click on the "Save As" button next to Crop Template. Enter a name for the template and click on Save.

To use a crop template, open the Data Cropping Tool (see Section 2.2) and choose the desired crop template from the Crop Templates List.

2.3 Data Collapsing

The raw data is collapsed to remove hysteresis and variance from the test and make it easier to identify the tire test conditions. Collapsing the data allows tire models to be fit much more quickly. The following figures demonstrate this tool. Figure 2.7 shows an example of raw data before it is collapsed and Figure 2.8 shows the same data collapsed. The data collapsing tool does not delete any data thus the original raw data is still available in OptimumT. Therefore if desired all of the data can still be graphed or just the collapsed data.

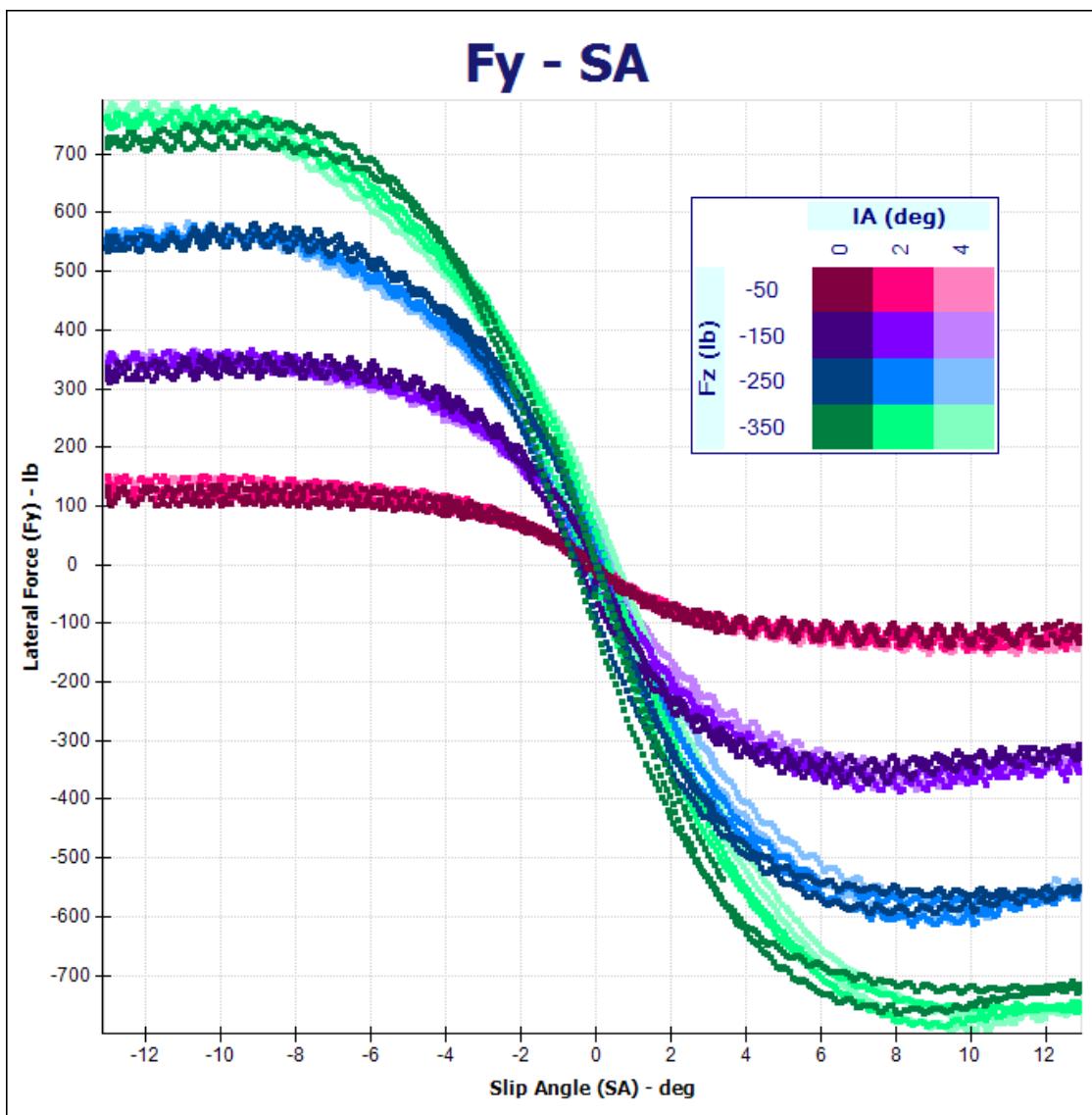


Figure 2.7: Raw Data Before Collapsing

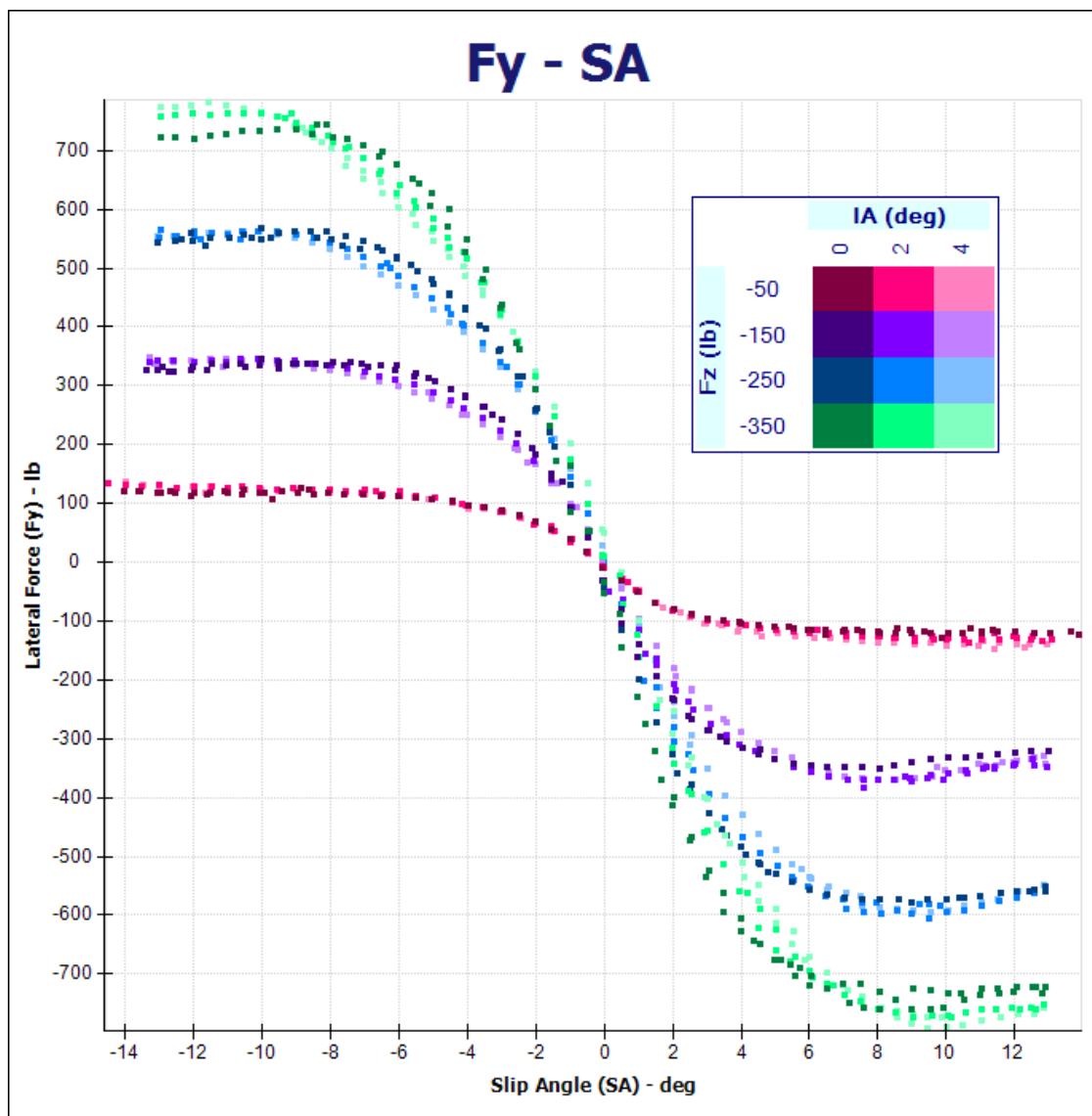


Figure 2.8: Collapsed Data

Now the procedure for collapsing raw data will be described. First select the raw data to be collapsed from the project tree. Then in the data entry area click on the *Collapse Data* button. This will open the data collapsing tool shown in Figure 2.9.

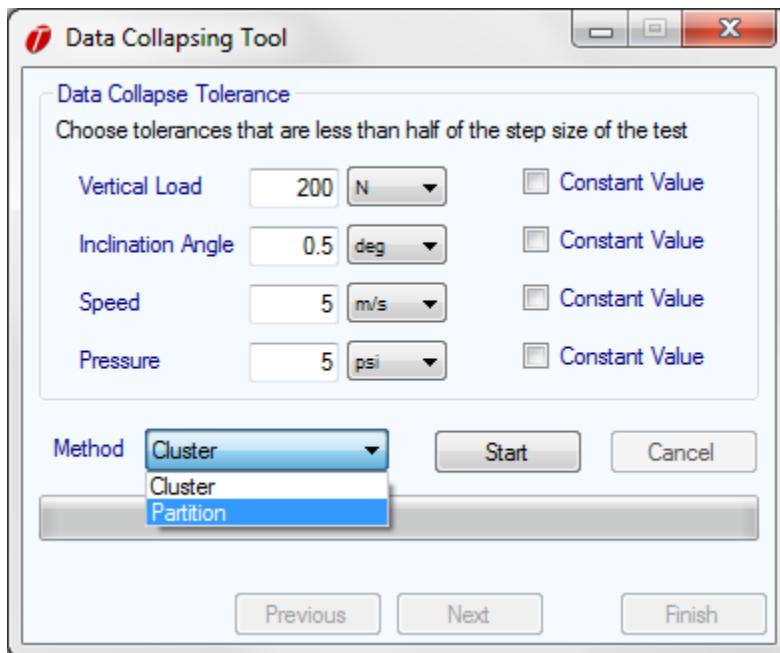


Figure 2.9: Data Collapsing Tool

First the data is sorted into different sets depending on the test conditions. In order to do this the data collapsing tolerances should be set at less than half of the step sizes used in the tire testing. For example if a tire was tested at vertical loads of 100, 200, and 300 lbs the data tolerance should be set to at least less than 50 lbs. This will separate the different test conditions as well as sort out any irregular data. Select the constant value check box for test conditions that are kept constant throughout the test. This will increase the speed of the sorting.

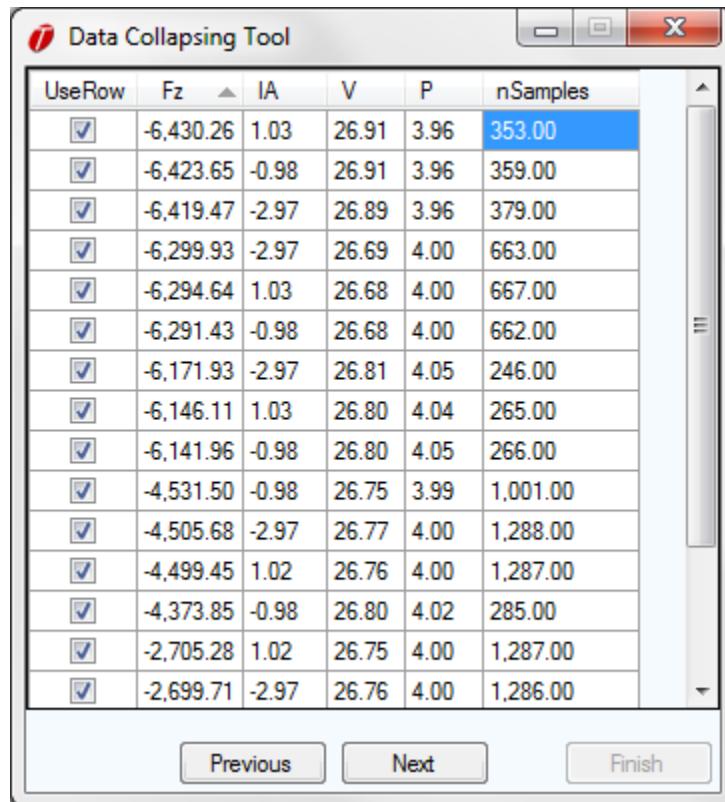
There are two sorting methods available *cluster* and *partition*. The cluster method works by creating a number of evenly spaced clusters and assigns each data point to the nearest cluster. The sizes of the initial clusters is determined by the variable tolerance. The algorithm then checks all the clusters. If the range of the cluster is too large the cluster is split if it is too small the cluster is deleted. The algorithm continues this process until all the clusters meet the size criteria. The partitioning method works by assigning each data point to its own cluster. The algorithm then checks the difference between adjacent clusters if the clusters are too close they are merged. The algorithm continues until the minimum distance between clusters has met the conditions defined by the variable tolerance.

Clicking on the *Start* button will sort the data. Once this is completed click on the *Next* button.

A list of the different sets of data will then be displayed. This is shown in Figure 2.10. The rows of data can be sorted by value by clicking on the column headers.

It should be checked that these sets represent the data that you want to work with. Also if

the data at the very beginning or end of the test was not removed with the cropping tool it can be removed here. This data will normally be at significantly different speeds or vertical loads than the rest of the data. Also if the data has a relatively low number of samples it most likely was not intended to be tested at those conditions. To remove sets of data just unselect the checkboxes next to that set. Once this is completed click on Next.



The screenshot shows a software window titled "Data Collapsing Tool". The main area is a table with the following columns: "UseRow", "Fz", "IA", "V", "P", and "nSamples". There are 15 rows of data, each with a checked checkbox in the "UseRow" column. The data is sorted by "nSamples" in descending order. The last row, which has the highest value of 1,286.00, is highlighted with a blue background. At the bottom of the window are three buttons: "Previous", "Next", and "Finish".

UseRow	Fz	IA	V	P	nSamples
<input checked="" type="checkbox"/>	-6,430.26	1.03	26.91	3.96	353.00
<input checked="" type="checkbox"/>	-6,423.65	-0.98	26.91	3.96	359.00
<input checked="" type="checkbox"/>	-6,419.47	-2.97	26.89	3.96	379.00
<input checked="" type="checkbox"/>	-6,299.93	-2.97	26.69	4.00	663.00
<input checked="" type="checkbox"/>	-6,294.64	1.03	26.68	4.00	667.00
<input checked="" type="checkbox"/>	-6,291.43	-0.98	26.68	4.00	662.00
<input checked="" type="checkbox"/>	-6,171.93	-2.97	26.81	4.05	246.00
<input checked="" type="checkbox"/>	-6,146.11	1.03	26.80	4.04	265.00
<input checked="" type="checkbox"/>	-6,141.96	-0.98	26.80	4.05	266.00
<input checked="" type="checkbox"/>	-4,531.50	-0.98	26.75	3.99	1,001.00
<input checked="" type="checkbox"/>	-4,505.68	-2.97	26.77	4.00	1,288.00
<input checked="" type="checkbox"/>	-4,499.45	1.02	26.76	4.00	1,287.00
<input checked="" type="checkbox"/>	-4,373.85	-0.98	26.80	4.02	285.00
<input checked="" type="checkbox"/>	-2,705.28	1.02	26.75	4.00	1,287.00
<input checked="" type="checkbox"/>	-2,699.71	-2.97	26.76	4.00	1,286.00

Figure 2.10: Sorted Data

Then the discretization range over which the data will be collapsed must be selected. This is shown in Figure 2.11. This determines the quantity and range of the collapsed data points that will be generated. Therefore for pure cornering data the number of steps used for the slip angle should be higher than that for the slip ratio and vice versa for the combined lateral and longitudinal data. For combined data the number of steps used should be approximately equal.

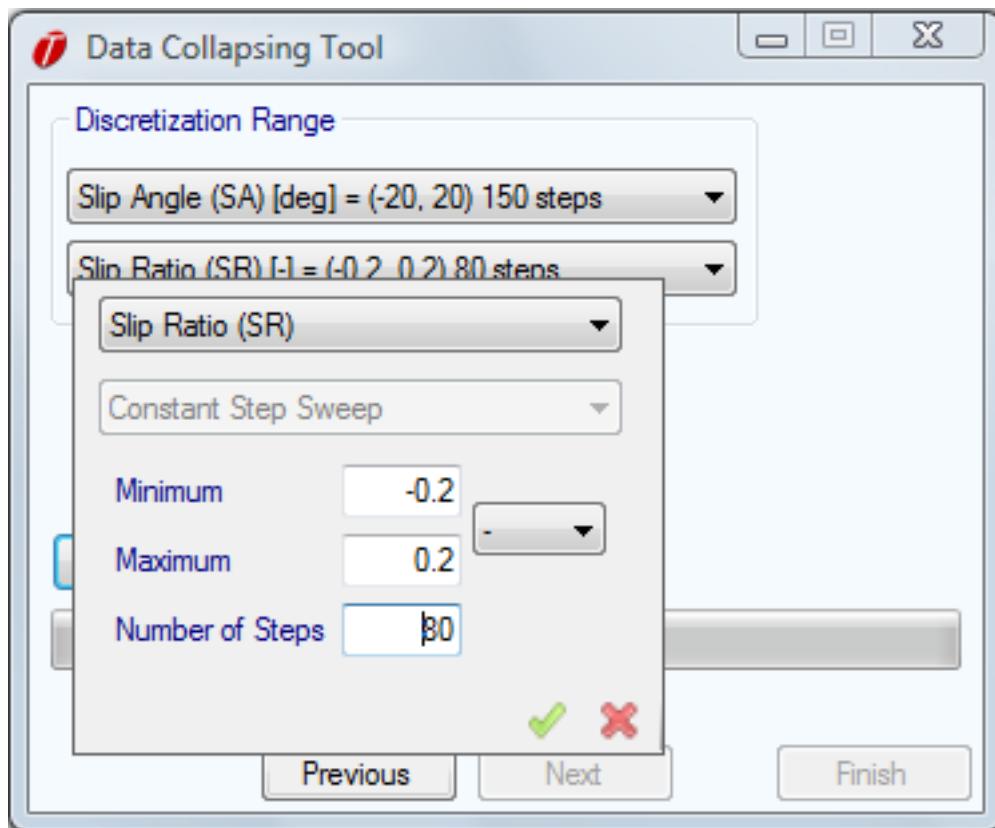


Figure 2.11: Setting Discretization Range

Clicking on the *Collapse Data* button will begin the collapsing process. This will compress each set of data points that fall within each discretization step. For each step, the subsequent data points are normalized with respect to their average vertical load. This is demonstrated in the following equations where F_y is normalized with respect to the vertical load. A similar formulation is used to normalize the other output parameters (i.e. F_x , M_z , etc...).

$$F_y = \frac{F_z}{F_{z0}} * F_{y0}(\alpha_0) \quad (2.1)$$

With the normalized slip angle

$$\alpha_0 = \frac{F_{z0}}{F_z} * \alpha \quad (2.2)$$

Clicking on the *Finish* button will close the *Data Collapsing Tool* and return to the main OptimumT window. A summary of the collapsed tire data will now appear in the data entry area when the tire data is selected in the project tree. The check-box labeled *Use Collapsed Data* will automatically be checked after the data is collapsed. This checkbox specifies whether the original or collapsed data will be used for graphing and fitting of tire models. This can be seen in Figure 2.12.

Collapse Data

Use Collapsed Data

Collapse Data

Selected Test Cases

Use Row	Vertical Load	Inclination Angle	Speed	Pressure	Data Points
<input checked="" type="checkbox"/>	-5,500.0	-3.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-4,000.0	-3.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-2,500.0	-3.00	55.9	29.01	374
<input checked="" type="checkbox"/>	-1,000.0	-3.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-5,500.0	-1.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-4,000.0	-1.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-2,500.0	-1.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-1,000.0	-1.00	55.9	29.01	375
<input checked="" type="checkbox"/>	-5,500.0	1.00	55.9	29.01	375

Vertical Load Unit: N

Inclination Angle Unit: deg

Speed Unit: mph

Pressure Unit: psi

Figure 2.12: Summary of Collapsed Data in the Data Entry Form

The summary of the collapsed data shown in Figure 2.12 allows the user to quickly see the data sets and the conditions of the test. The data can be sorted numerically by clicking on the column headers. The data can also be sorted by the tolerances by selecting the *Sort by Tolerance* checkbox. The checkboxes in the first column allow the user to exclude or include certain data sets from graphing and tire model fitting. Since the original raw data is still stored in OptimumT the same data can be collapsed multiple times.

2.4 Exporting Data

By selecting the Options-Export button at the top of the raw data entry form the tire data can be exported to an OptimumT Raw Data File or a CSV file. After the desired format is selected a dialog box will appear. Clicking on the Yes button will export the cropped and collapsed tire data while clicking on the No button will export the raw tire data.

Chapter 3

Tire Models

Currently, seven tire models are implemented in OptimumT:

- Fiala Model
- Brush Model
- Harty Model
- Pacejka Magic Formula '96 Model
- Pacejka Magic Formula 2002 Model
- Pacejka Magic Formula 2002 Model with Inflation Pressure effects
- Pacejka Magic Formula 2006 Model

The coefficients of these tire models can be manually inputted or imported from an OptimumT Native File. OptimumT can also fit these models to raw tire data. More specific information regarding the tire models is included in the section 8.

3.1 Manually Input Model

If the tire model coefficients have already been determined, they can be manually inputted into OptimumT. First select the *New Tire Model* button at the top of the project tree. Then choose the type of tire model you would like to add to the project. This tire model will be added to the project tree. Right clicking on the tire model allows the user to rename, delete, or copy the model as well as many other functions. The model coefficients can now be entered into the model input form, which appears in the data entry area when the model is clicked on. An example of this form is shown in Figure 3.1. For coefficients that have

units the units can be specified in the dropdown boxes to the right of the value. The small plus and minus buttons also to the right of the values allow the user to adjust the model. This feature is covered in detail in section 3.6.

Pacejka 2002	Coefficients	Options
Description		
<input type="checkbox"/> Override default name in legend <input type="text"/>		
Coordinate System		
<input type="button" value="SAE, Default"/>		
General		
Fz0	-1112.0	N
r0	0.254	m
V0	11.176	m/s
Pure Lateral		
pCy1	1.322	±
pDy1	-2.18	±
pDy2	0.182	±
pDy3	-19.837	±
pEy1	-0.716	±
pEy2	-0.342	±
pEy3	-0.684	±
pEy4	-6.49	±
pKy1	39.675	±
pKy2	2.015	±
pKy3	29.783	±
pHy1	0.005	±
pHy2	0.002	±

Figure 3.1: Tire Model Input Form

3.2 Import and Export Models

Before a tire model is imported into OptimumT the appropriate tire model needs to be added to the project. This is done by clicking on the *New Tire Model* button at the top of the tire project tree. Once the tire model is added clicking on it will display its coefficients in the data entry area. Since it is a new model all of the coefficients will be zero. At the top of the data entry area click on Options-Import as shown in Figure 3.2. Tire model coefficients can then be imported from an OptimumT Native file or from a TIR, or similar, file. Note that when you import a model, OptimumT will overwrite the data contained in the model input form.

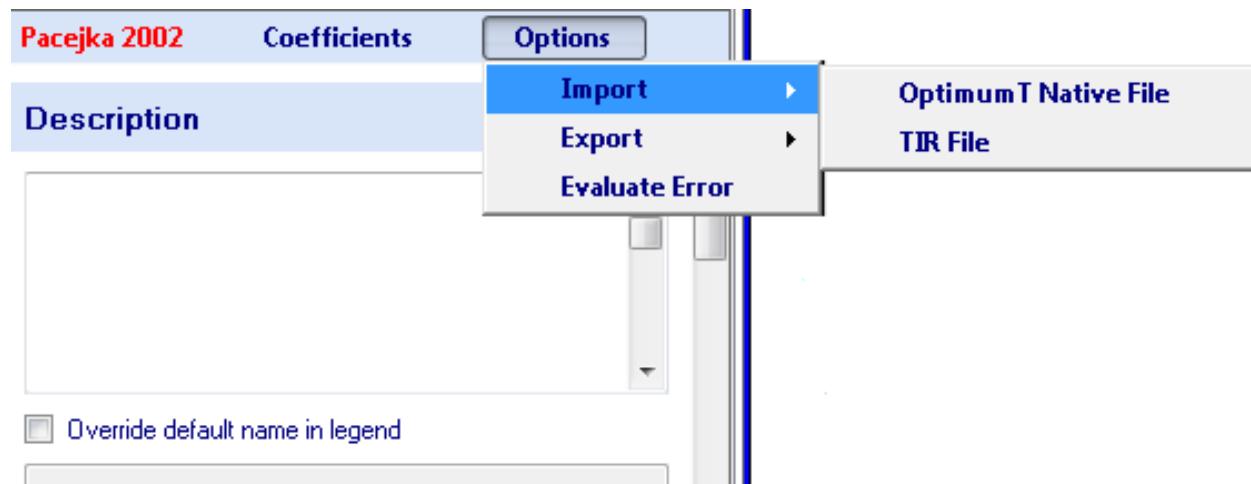


Figure 3.2: Import Tire Model

Tire models can also be exported from OptimumT by clicking on the desired tire model in the project tree. Then, at the top of the data entry area, click on *Options-Export* and select the file format you prefer. You can export to OptimumT Native files, Excel, Lookup tables and into text-based files, such as TIR files. You also have the option of copying the tire model to the clip board for use with the addin (it will be in an encoded text format).

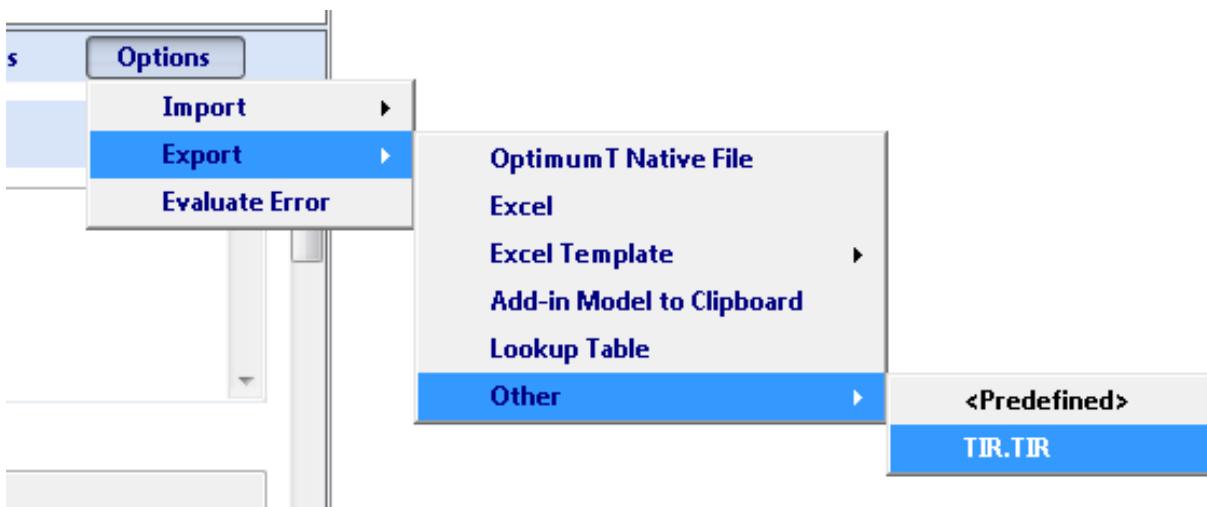


Figure 3.3: Export Tire Model

3.2.1 Export Templates

When exporting to a text-based format format, like a TIR file, OptimumT uses export templates. You can edit the export templates in the template manager. When creating a new template, it is recommended that you start with an existing template and modify it. (If you start from a Predefined template, you will need to clone it first because these are read-only). When OptimumT reads a template, it looks for parameter fields in the format `#PARAM#DEFAULT-VALUE#`. Note that there are a total of three # symbols for each parameter. PARAM denotes the name of the parameter, DEFAULT-VALUE denotes the default value for this parameter if it is not found in the tire model. If there are parameters specified in the export template that are not in the tire model, the user will be asked to specify them in a dialog when exporting a model. The default value will be used if the user does not specify a different value when exporting. The model description and coordinate system can be exported using the tags `#DESC#` and `#COORDSYS#` respectively.

When exporting models using templates be aware that the simulation program you intend to use the model with may require the model in a specific coordinate system. Tir and similar files do not contain coordinate system information so the coordinate system must be modified within OptimumT before exporting. In the case of the MF5.2 and P96 templates, which have been designed for use in RaceSim, you can not export using the SAE coordinate system. It is recommended that you export using the Iso coordinate system for these models.

OptimumT also offers the ability to export lookup tables for simulation programs that do not directly import model coefficients or equations. For more details on how to use this feature of OptimumT see section 6.5.

3.2.2 Excel Export Templates

OptimumT can also export to Microsoft Excel using export templates similar to those explained in the previous section 3.4. An Excel template is a regular Excel worksheet with cells that contain text that can be recognised by OptimumT. When OptimumT reads an Excel template it looks for parameter fields in the format #PARAM# where PARAM is the name of the coefficient to be exported, eg. pCy1 as shown in Figure 3.4. The model description and coordinate system can be exported using the tags #DESC# and #COORDSYS# respectively (Note OptimumT will only search the first worksheet of an Excel template for the parameter fields). Once the template has been created the file should be copied and pasted into the application data folder using the template manager or saved in the application data folder:

Windows Vista and Windows 7:

C:\Users\UserName\AppData\Roaming\OptimumT\ExcelTemplates

Windows XP:

C:\Documents and Settings\UserName\Application Data\Roaming\OptimumT\ExcelTemplates

	C31		f _x	#PHY1#
	A	B	C	D
1	Pacejka 2006 Model			
2				
3				
4				
5	Model Type			
6	Pacejka 2006			
7				
8				
9	General			
10	Fz0	#FNOMIN#		
11	R0	#UNLOADED_RADIUS#		
12	V0	#LONGVL#		
13				
14	Pure Lateral			
15	pCy1	#PCY1#		
16	pDy1	#PDY1#		
17	pDy2	#PDY2#		
18	pDy3	#PDY3#		
19	pEy1	#PEY1#		
20	pEy2	#PEY2#		
21	pEy3	#PEY3#		
22	pEy4	#PEY4#		

Figure 3.4: Example Excel Export Template

3.3 Fitting Models to Raw Data

Before fitting a tire model to the raw data the data should be properly cropped and collapsed. This will make the fitting more accurate and significantly quicker. Also make sure that the *Use Collapsed Data* option is selected for the tire data that will be used. This can be found at the top of the *Collapse Data* section in the tire data input area. The following sections demonstrate how to fit a tire model. This example will use pure lateral and combined longitudinal and lateral data to create a complete tire model. Depending on the data available and the goal of the model fitting the order in which the models are fit can vary. More information about the order the models should be fit is included in section 3.4.

3.3.1 Pure Lateral Model

Fitting of the cornering data to the model lateral force coefficients is generally done first. To begin select the tire data to be modeled in the project tree. Go to the last section of the data entry area, labeled *Model Fitting*. This is shown in Figure 3.5. In the drop down box select the type of tire model to be fitted. Once you have selected the type of model to be used click the *Fit Model* button. This will open the Model Fitting Tool as can be seen in Figure 3.6.

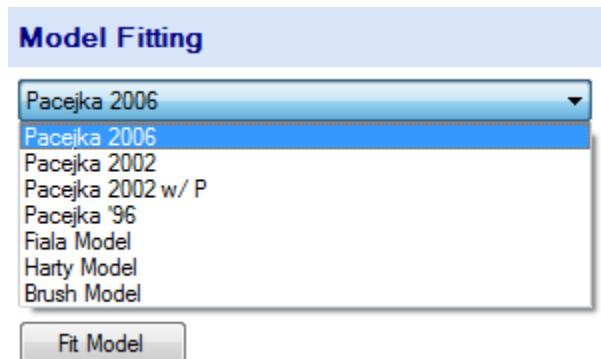


Figure 3.5: Selecting the Tire Model to be Fit

Specify Coefficients to be Calculated

In the *Model Fitting Selection* window the coefficients to be fit are selected. The coefficients that the error calculation is based on are also selected by choosing the *Fit and Calculate Error* option in the dropdown boxes. For this case you would select *Fit and Calculate Error* in the drop down box labeled *Fy Pure* as is shown in Figure 3.6. *Fit and Calculate Error* must always be selected for at least one of the sets of coefficients. At the bottom of this window the coordinate system of the model can be selected. To proceed click the *Next* button.

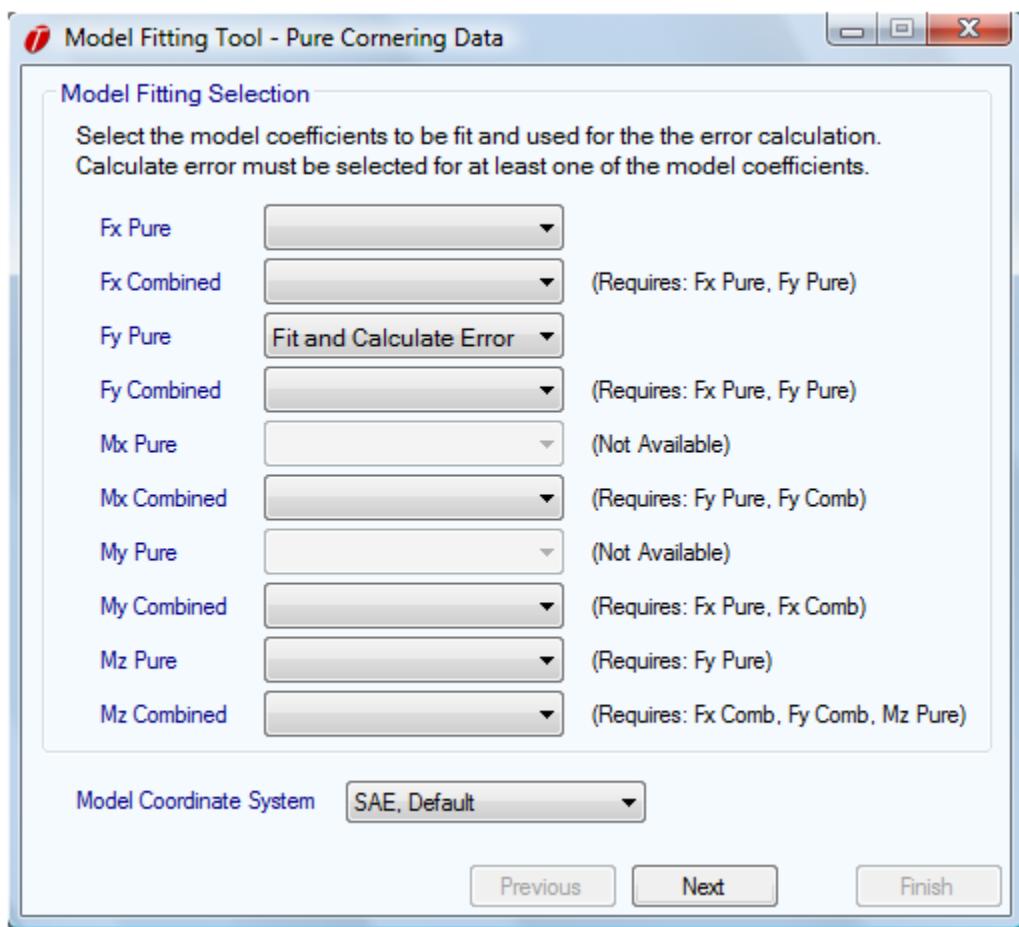


Figure 3.6: Model Fitting Selection

Model Constraints

Now the model constraints must be set. The constraints required vary depending on what type of model is being fit. For the Pacejka models the constraints include the nominal (rated) tire load, F_{z0} , the unloaded tire radius, R_0 , the reference velocity, V_0 , and the reference pressure, P_0 . Typically the nominal load is set to the largest or second largest load that the tire was tested at, the reference velocity is set to the test velocity, and the reference pressure is set to the highest inflation pressure the tire was tested at. More information regarding these parameters is included in section 8.

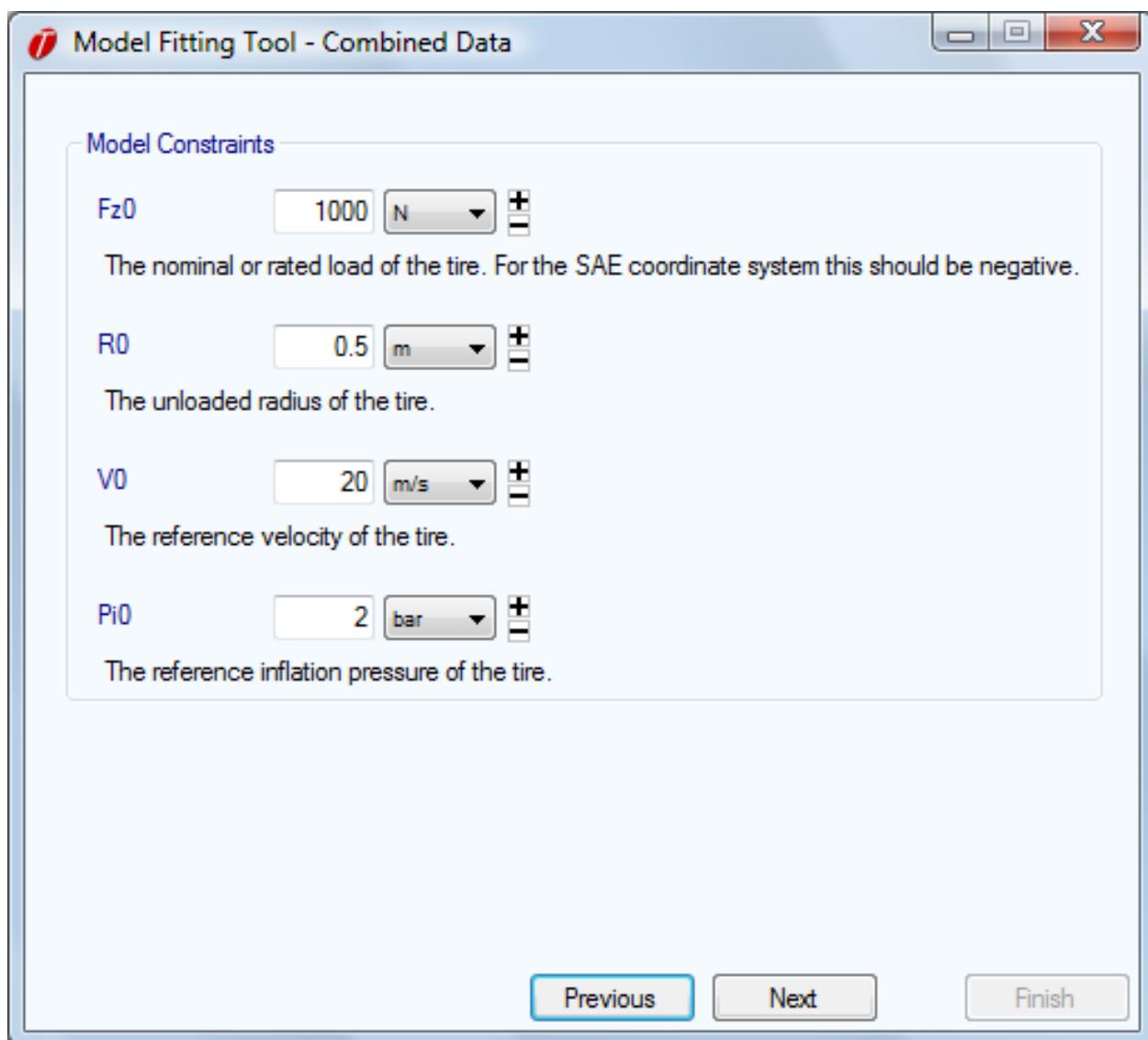


Figure 3.7: Model Constraints

Define the Coefficient Boundary

Now the *Coefficient Boundary* to be used in the fitting must be set. The coefficient boundary allows the solver to begin fitting the model in a restricted, more accurate range of values for the coefficients. However, the solver is not necessarily restricted to the range of coefficients set by the boundaries. In Figure 3.8 the Coefficient Boundary window is shown. A *Coefficient Boundary* is chosen by selecting it from the list. Click on *Next* to proceed.

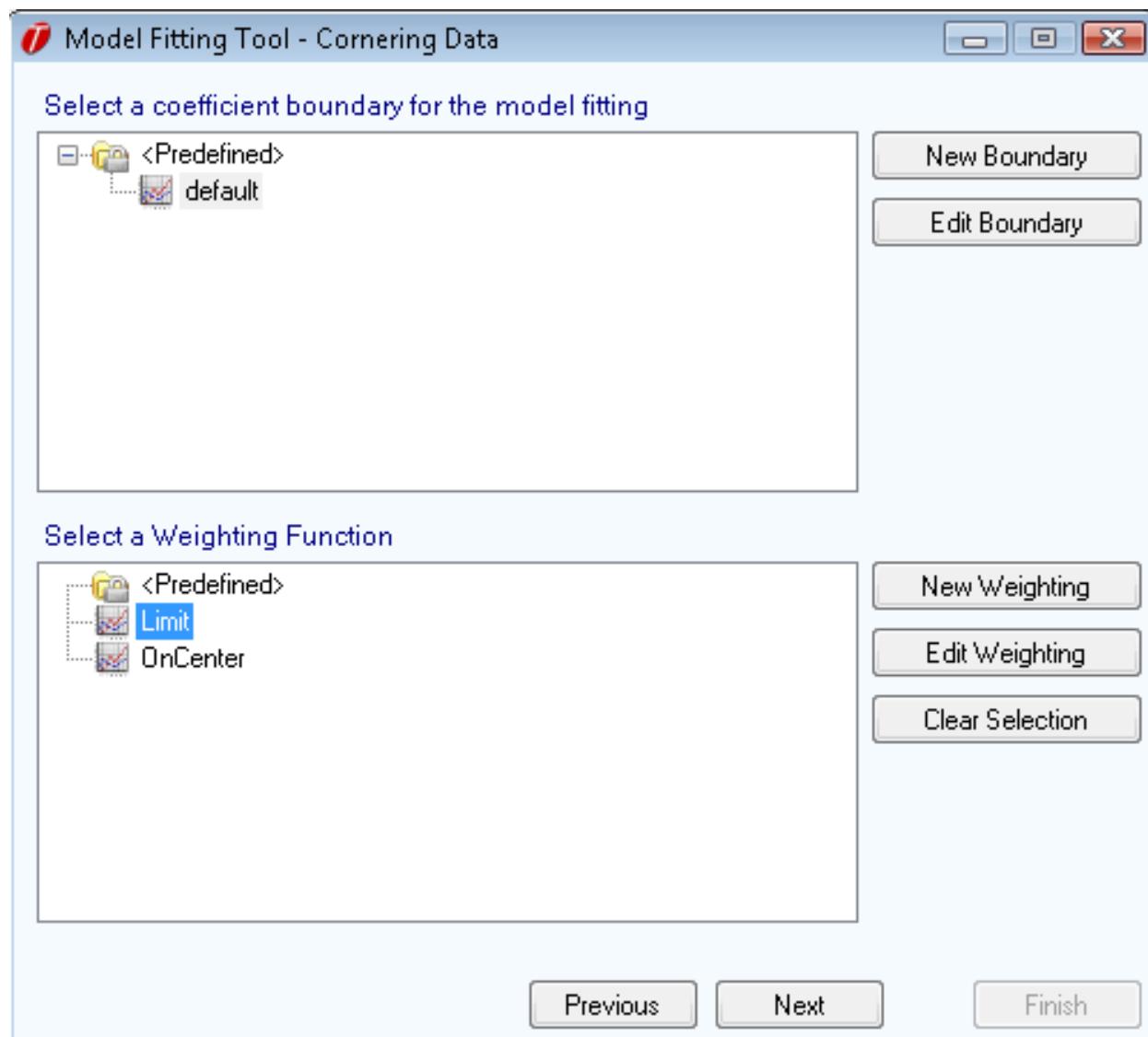


Figure 3.8: Coefficient Boundary

Coefficient boundaries can easily be created or edited by clicking on the *New Boundary* and *Edit Boundary* buttons to the right. However the coefficients in the *Predefined Folder* cannot be modified. If they are edited, the edited version will be saved outside of the *Predefined* folder and the original version will stay unchanged. When these buttons are clicked the *Coefficient Boundary Editor* window will open as shown in Figure 3.9. This window allows the boundaries for each coefficient to be changed. If the *Hard Boundary* check boxes are selected the solver will be restricted to finding a solution within this range of coefficients. Save the new or edited boundary by typing in a name at the top of the window and clicking on save. Now you can close this window and the new boundary will be available for selection in the *Coefficient Boundary* window.

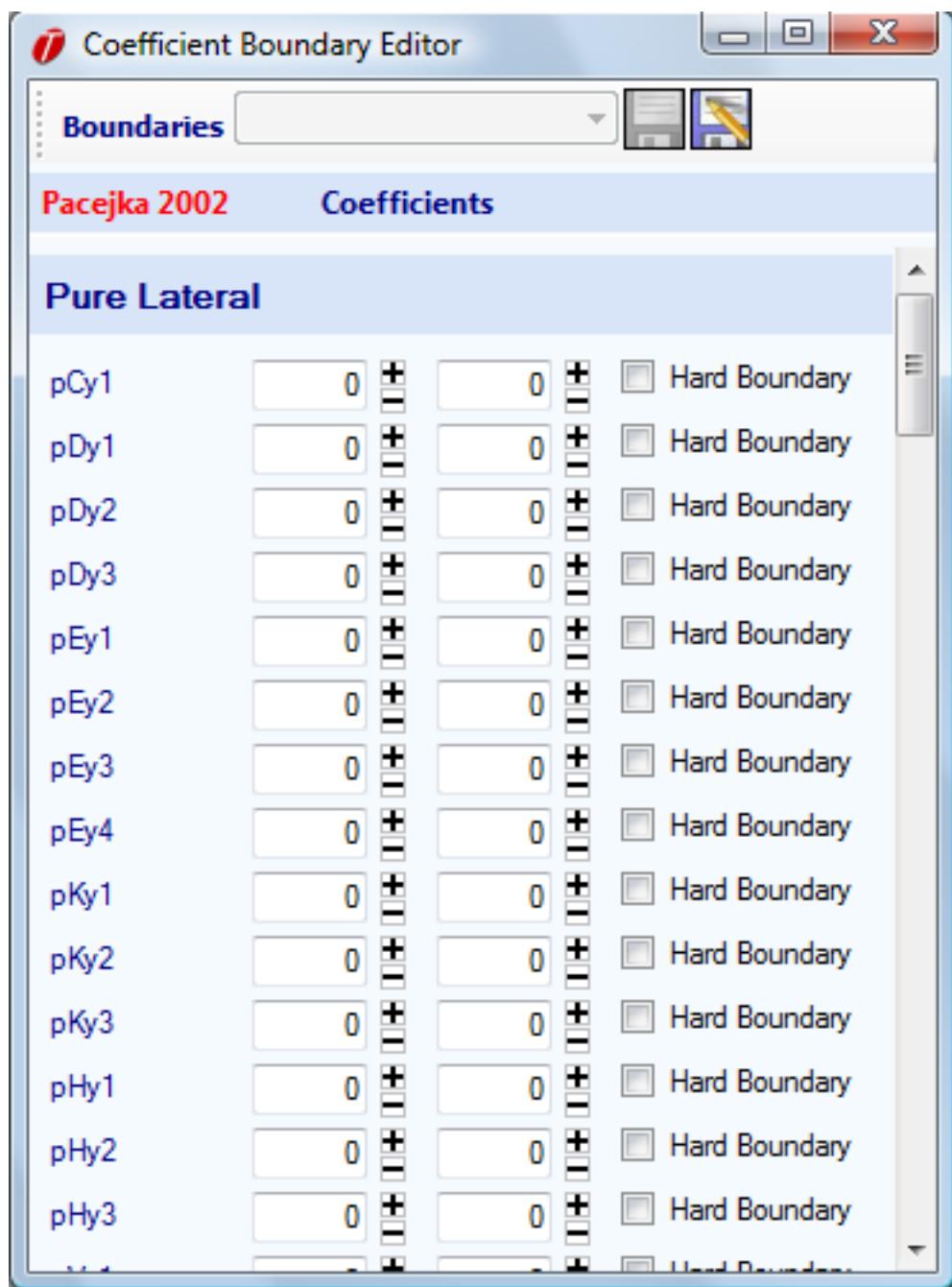


Figure 3.9: Coefficient Boundary Editor

Weighting Functions

Optionally, you can select a weighting function to apply to the fit. This will change the way the error is evaluated. For details about the error evaluation, please see Section 3.3.1. You can select an existing weighting function by selecting it from the list in the coefficient

boundary selection dialog (Figure 3.8). You can also create a new weighting function or edit an existing one in this dialog. If you have selected a weighting function and wish to remove this selection, click the *Clear Selection* button.

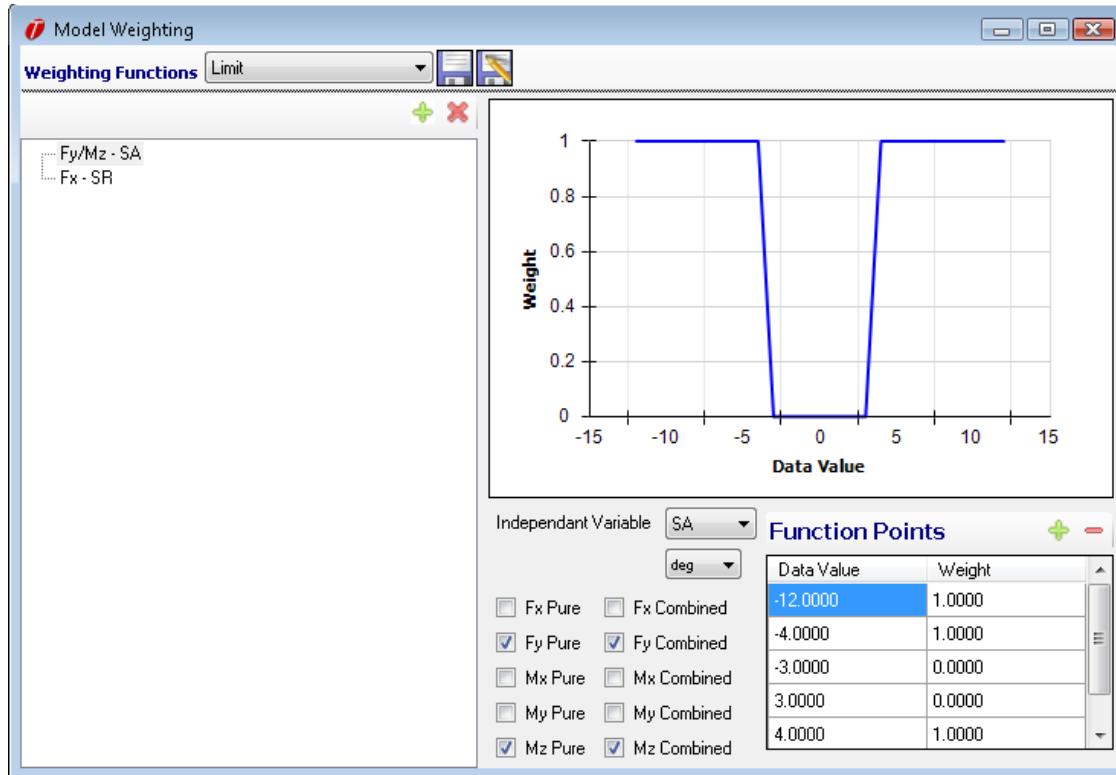


Figure 3.10: The weighting function editor

Figure 3.10 shows the weighting function editor. It is accessible by clicking either the *New Weighting* or *Edit Weighting* button. At the very top of the dialog, you can save or load a weighting function. On the left side, you can select the *Weighting Pages* which define a part of a weighting function. There is no limit to the number of weighting pages that you can add (though there is a practical limit of 60, as it is not possible to define more than 60 non-redundant weighting pages). Each weighting page has one independent variable one ore more dependent variables. If you want to make the solver treat the error of the FxPure fit with more "importance" at high slip ratios, you would define the independent variable to be slip ratio (SR) and the dependent variable to be FxPure. A given pair of independent and dependent variables cannot be defined twice within the same weighting function. You will receive an error if you try to do this.

Once the independent and dependent variables have been selected for a weighting page, the values of the weighting function can be entered. The function is defined as a table. OptimumT will do a linear interpolation between the entries in the table. The *Data Value* column indicates the values of the independent variable, and the *Weight* column defines the weight. Typically the values in the weight column will be between 0 and 1.

Solver Parameters and Error Evaluation

Next the *Population* and *Iterations* needs to be set. These parameters are used by the solver and are not related to the tire models. The *Population* is the number of initial values that the solver will distribute between the coefficient boundaries. Therefore if the coefficient boundaries are well defined the population can be less and vice versa. *Iterations* are the number of steps the solver will take when fitting the model. With a more complex model a larger number of iterations is required. The window that these parameters are set is shown in Figure 3.11.

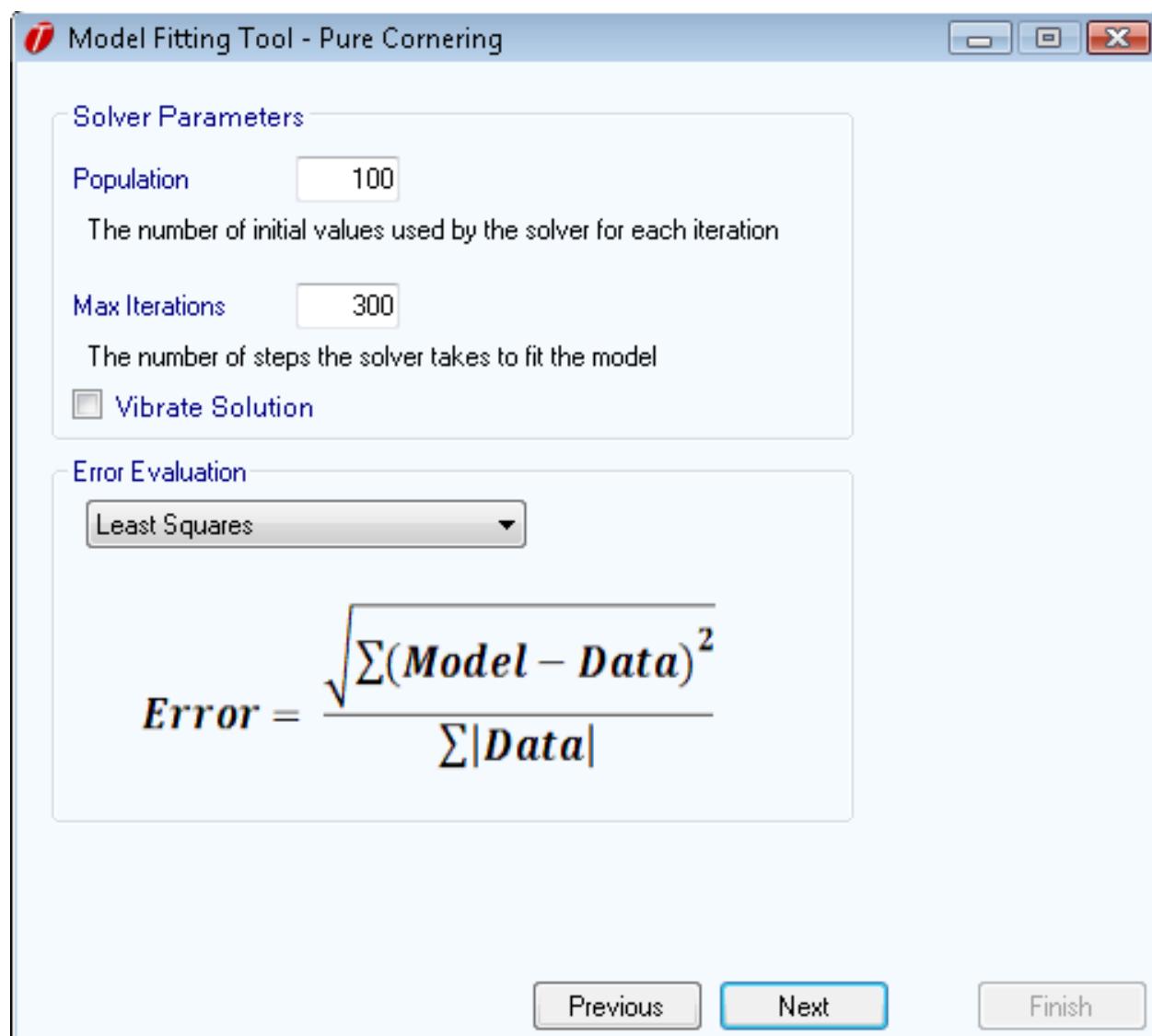


Figure 3.11: Solver Parameters and Error Evaluation

The *Vibrate Solution* option will cause a small random variation to be applied to the data

at each step. This can help the solver converge towards a global minimum slightly faster. Statistically, the variation is applied equally in both direction, so this technique does not bias the results.

The *Error Evaluation* can now be selected. This determines what type of error criteria is used when fitting the model. The four types of error criteria included in OptimumT are described below. In the equations *Model* represents the value found by the model fitter and *Data* represents the value of the actual raw data.

- Least Squares Error:

$$Error = \frac{\sqrt{\sum (Model - Data)^2}}{\sum |Data|}$$

- Normalized Least Squares Error:

$$Error = \frac{\sqrt{\sum \left(\frac{Model - Data}{Data}\right)^2}}{\#of datapoints}$$

- Total Error:

$$Error = \frac{\sum |Model - Data|}{\sum |Data|}$$

- Normalized Total Error:

$$Error = \frac{\sum \left|\frac{Model - Data}{Data}\right|}{\#of datapoints}$$

Typically the *Least Squares Error* criteria will result in the best overall model fit. However depending on the variance and testing conditions of the raw data some of the other error criteria may produce a better fit. For example when fitting an aligning torque model *Total Error* will often produce slightly better results. The *Normalized Error* criteria gives equal weight to all of the data points. Therefore it will improve the model fitting at lower loads.

In the case that a weighting function is used, the error evaluation is modified to be the following. The weightin is denoted as w_i and is evaluated for each data point separately.

- Least Squares Error:

$$Error = \frac{\sqrt{\sum w_i (Model - Data)^2}}{\sum |Data|}$$

- Normalized Least Squares Error:

$$Error = \frac{\sqrt{\sum w_i \left(\frac{Model - Data}{Data}\right)^2}}{\#of datapoints}$$

- Total Error:

$$Error = \frac{\sum w_i |Model - Data|}{\sum |Data|}$$

- Normalized Total Error:

$$Error = \frac{\sum w_i \left| \frac{Model - Data}{Data} \right|}{\# of datapoints}$$

Model Fitting

Now OptimumT is ready to fit the tire model. However, if something is incorrect or the user would like to change some of the settings the *Previous* button can be used to go back to the previous windows. Alternatively, the *Model Fitting Tool* window can be closed and the model fitting restarted.

Clicking on the *Summary* button allows the user to check to make sure all of the solver settings are correct. It will show the model being fit, the raw data to be used, the error calculation method, the coefficient boundary, and the solver parameters selected. To return to the *Model Fitting Tool* window just close the model summary. Once the model fitting is finished this information will be automatically transferred into OptimumT and will be stored with the associated tire model in the models data entry form.

Clicking on the *Start* button in the upper right corner of the window will start the solver. Figure 3.12 shows this window after a fitting has been completed. The graph shows the convergence of the solution and the textbox shows the current error as the model is fitted. When the final solution is found click on the *Finish* button to return to the main screen of OptimumT.

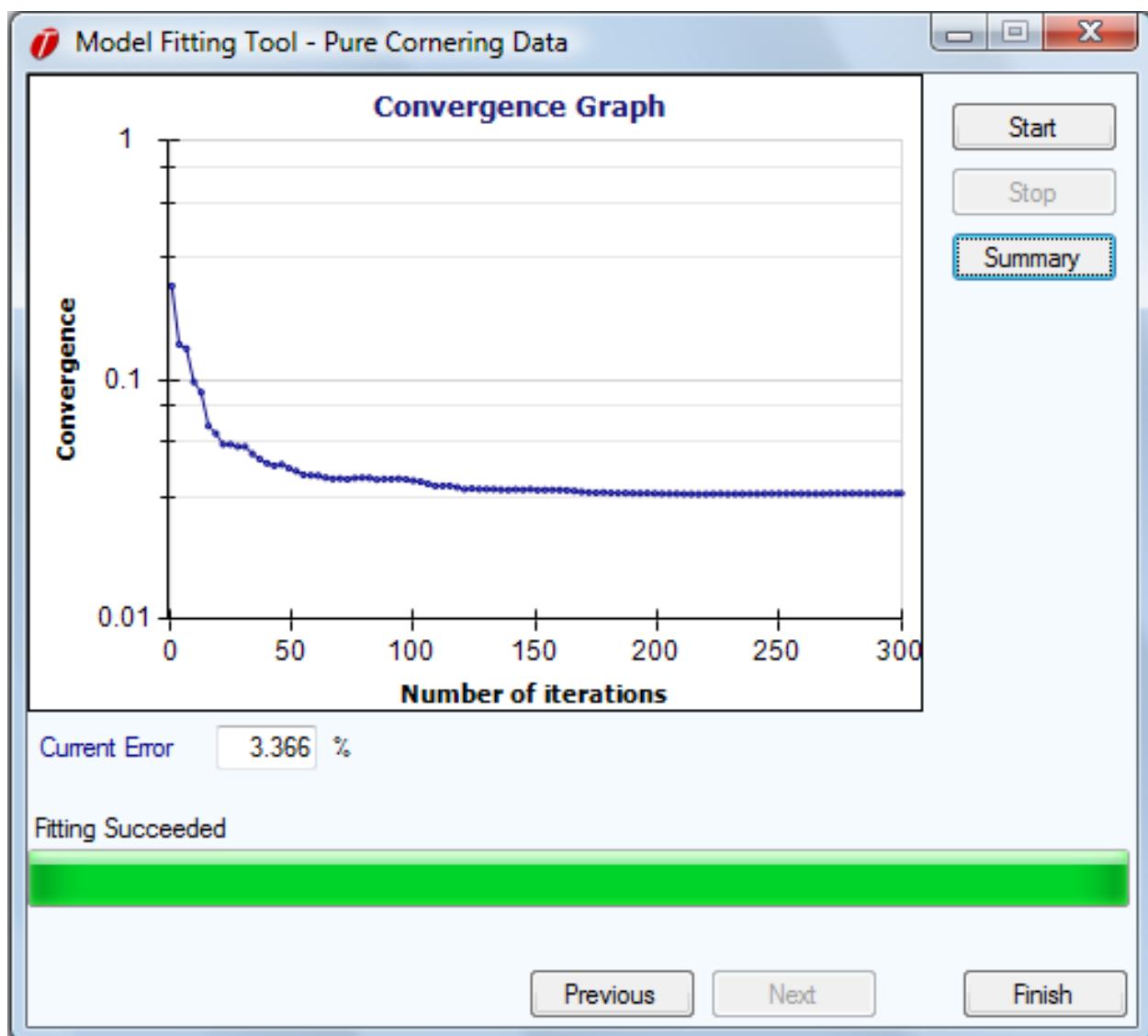


Figure 3.12: Model Fitting Convergence and Error

The newly created tire model should now appear in the project tree. This model can now be graphed by selecting the check box next to it (Graphing is covered in Chapter 6). It should be compared to the raw data to ensure accuracy. Lateral force as a function of the slip angle at the tested inclination angles should be graphed and compared to the raw data as shown in Figure 3.13. Also the lateral force as a function of the normal load, F_z , for several slip angles should be graphed as in Figure 3.14. The graphs should be checked for accuracy by comparing them with the raw data. The model should also be checked to ensure the curves are well behaved outside of the measurement area. This is especially important if the tire models are going to be used for simulation. Refer to section 3.6 for information on adjusting the tire model.

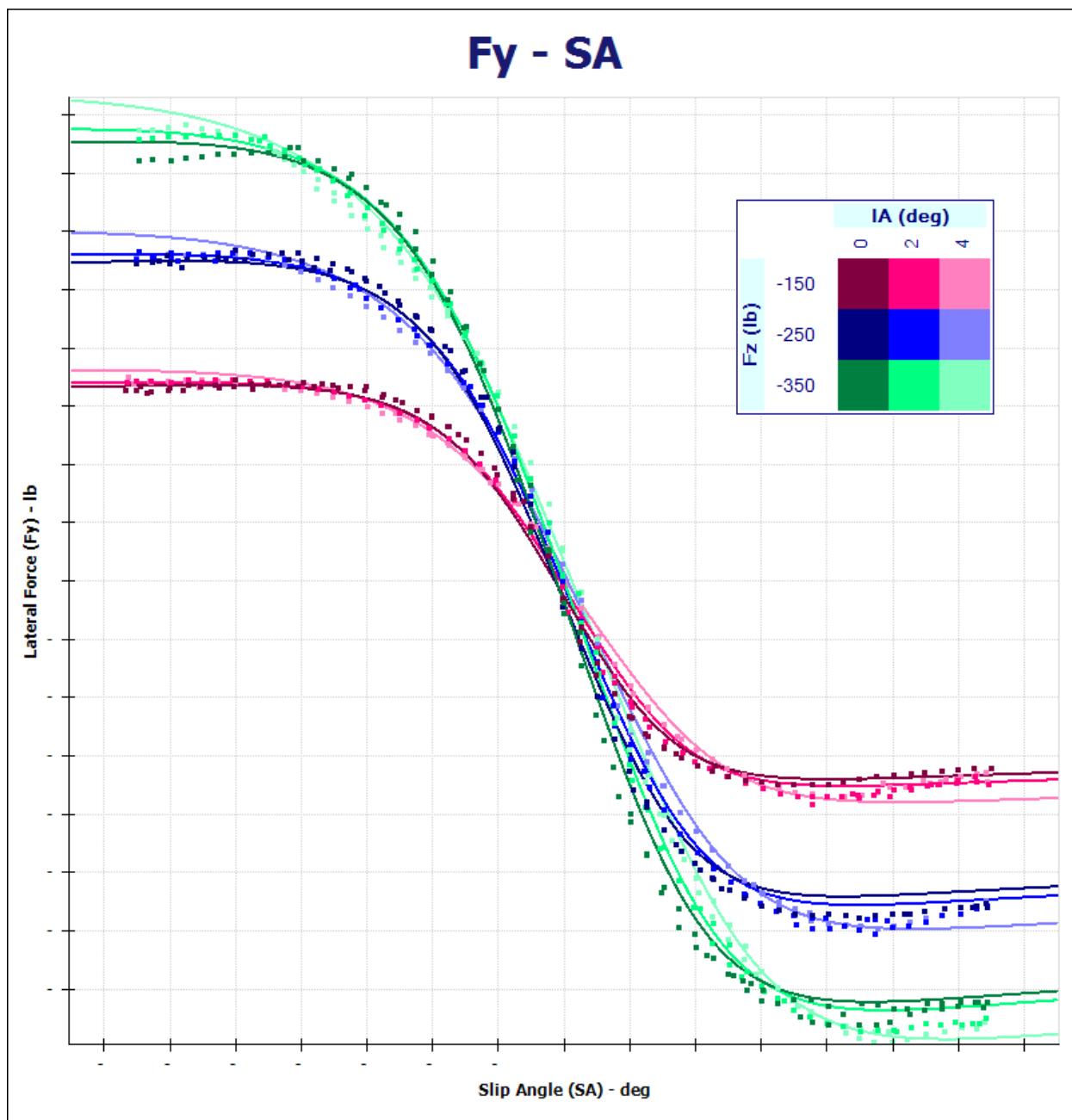


Figure 3.13: Lateral Force vs. Slip Angle at Different Inclination Angles

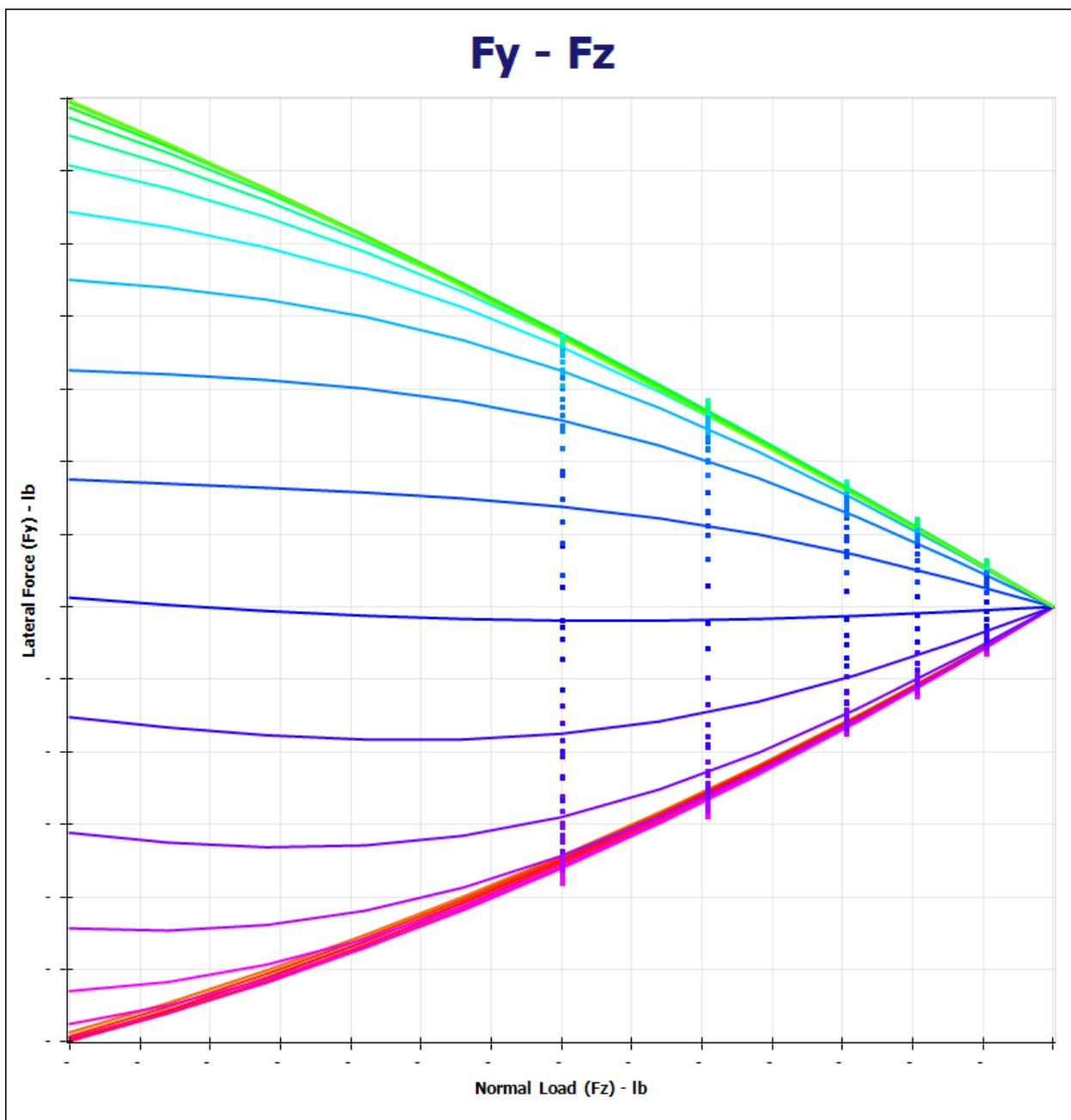


Figure 3.14: Lateral Force vs. Normal Load at Different Slip Angles

3.3.2 Aligning Torque Model

Fitting the aligning torque model is very similar to the lateral force model. However, there are a few small differences since this model will be combined with the lateral force model.

First select the tire data used to fit the lateral force model. In the *Model Fitting* section

select the same type of tire model as was used for the lateral force model and click on the *Fit Model* button. This will open a different window than before since another tire model already exists in this project. Figure 3.15 shows the *Advanced Fitting Options* window. In this window coefficients that were already determined in previously created models can be fixed for the current fitting. So for this example you would set the *Fy Pure* category to the previously created tire model. In later sections you will see that you can set multiple sets of coefficients and also include model scaling factors. Once the correct coefficients are selected click on the *Done* button.

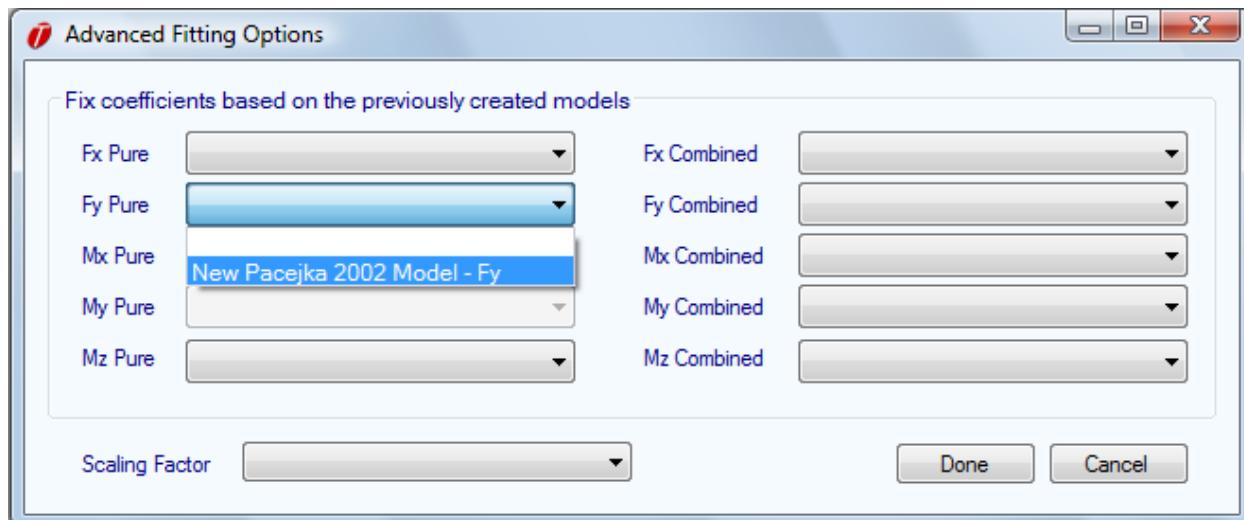


Figure 3.15: Advanced Fitting Options

Now the *Model Fitting Selection* window that was used to fit the previous models will open. For this case *Calculate and Fit Error* in the drop down box labeled *Mz Pure* would be selected. As can be seen in Figure 3.16 the dropdown box corresponding to *Fy Pure* is disabled because the coefficients for this model were fixed in the previous step. The rest of the tire model can now be created in the same way as the lateral force model. When finished a new tire model will be created that contains both the pure lateral force and the aligning torque coefficients.

The aligning torque model could also have been created at the same time as the pure lateral force model, but for demonstration purposes was done separately. This would have been achieved by selecting *Fit* or *Fit and Calculate Error* for both *Fy Pure* and *Mz Pure* when the coefficients to be fit were specified. The aligning torque as a function of the slip angle should be graphed with the tire data to check the accuracy of the model. An example of this is shown in Figure 3.17.

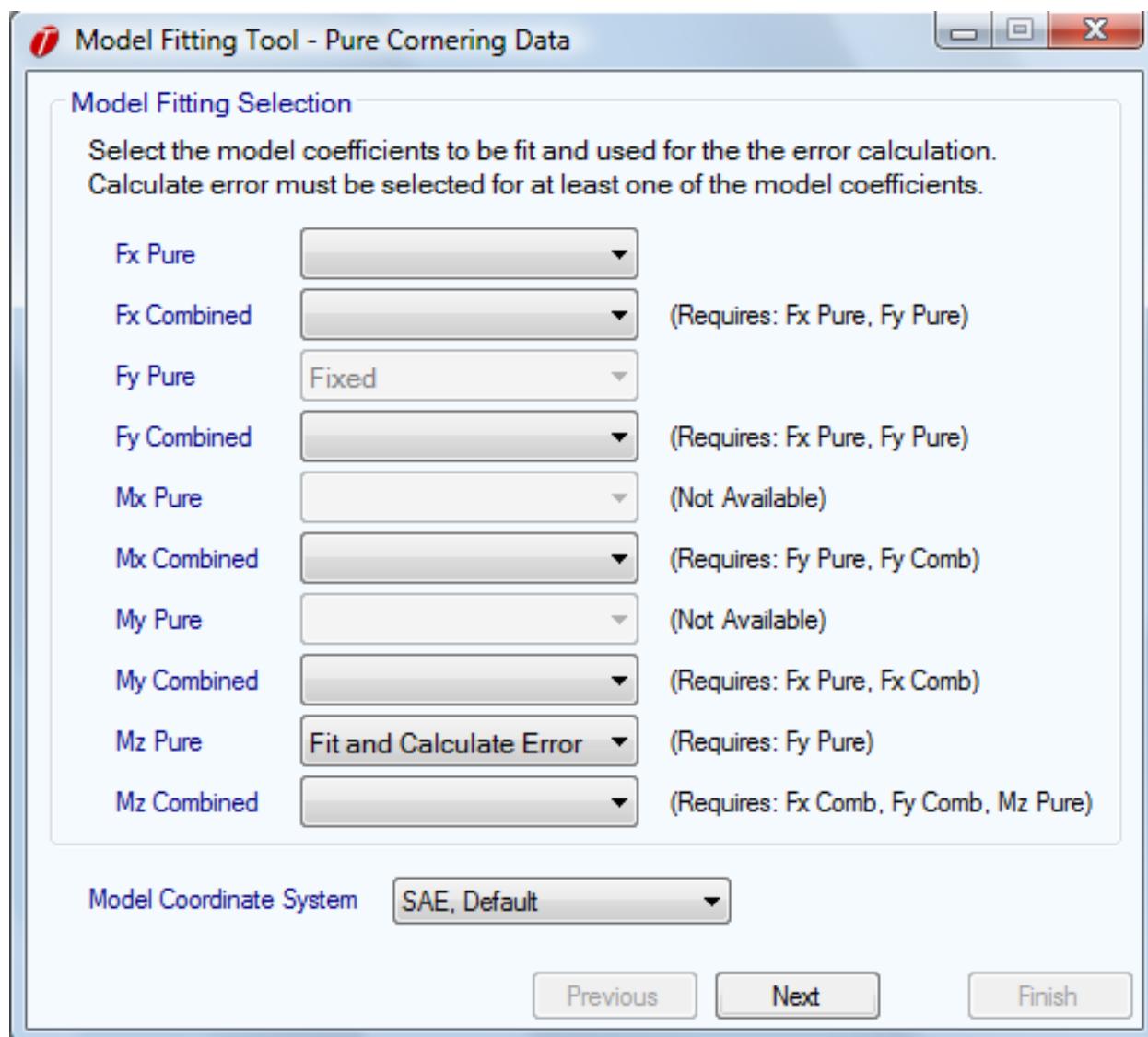


Figure 3.16: Specify Coefficients to be Fit

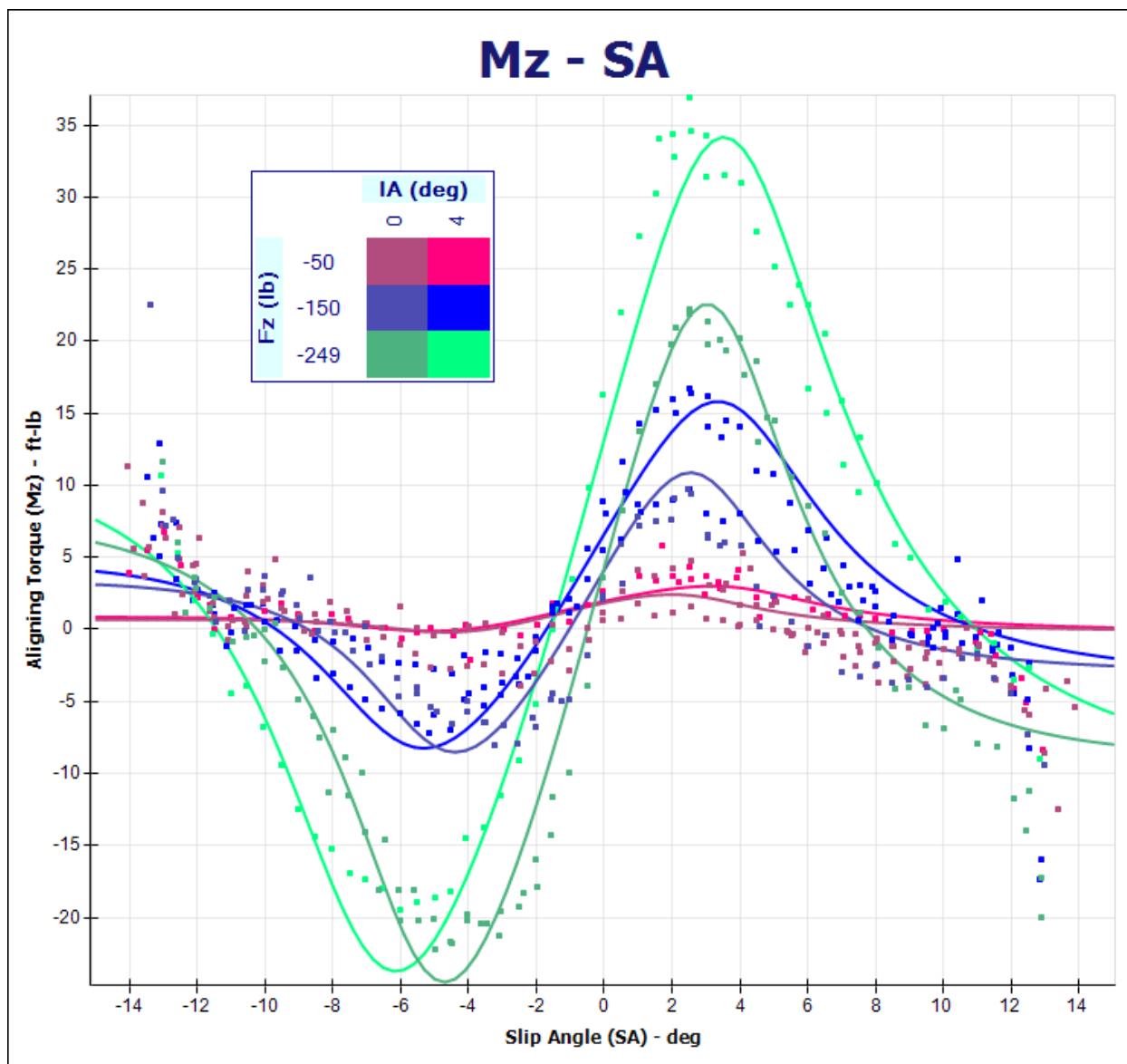


Figure 3.17: Aligning Torque vs. Slip Angle at Different Inclination Angles

3.3.3 Model Scaling Factors

In order to combine the data from a pure cornering test and a combined lateral and longitudinal test it will often be required to use a scaling factor. It will adjust the pure lateral force model created from the cornering data to match the combined lateral and longitudinal data at zero slip ratio.

Figure 3.18 shows the discrepancy between the lateral force model and the combined data for the same tire. The pure lateral force model is represented by the solid lines and the

combined data is the clusters of points at approximately 0, 3, and 6 degrees of slip angle. These points represent the force at zero slip ratio. This discrepancy is an effect of the pure cornering test being performed at a constant slip ratio and varying slip angle, while the combined test is performed at a constant slip angle with a varying slip ratio. These tests are also often performed at different speeds. Additionally, the tire will experience different heat cycling in these tests and therefore different results would be produced.

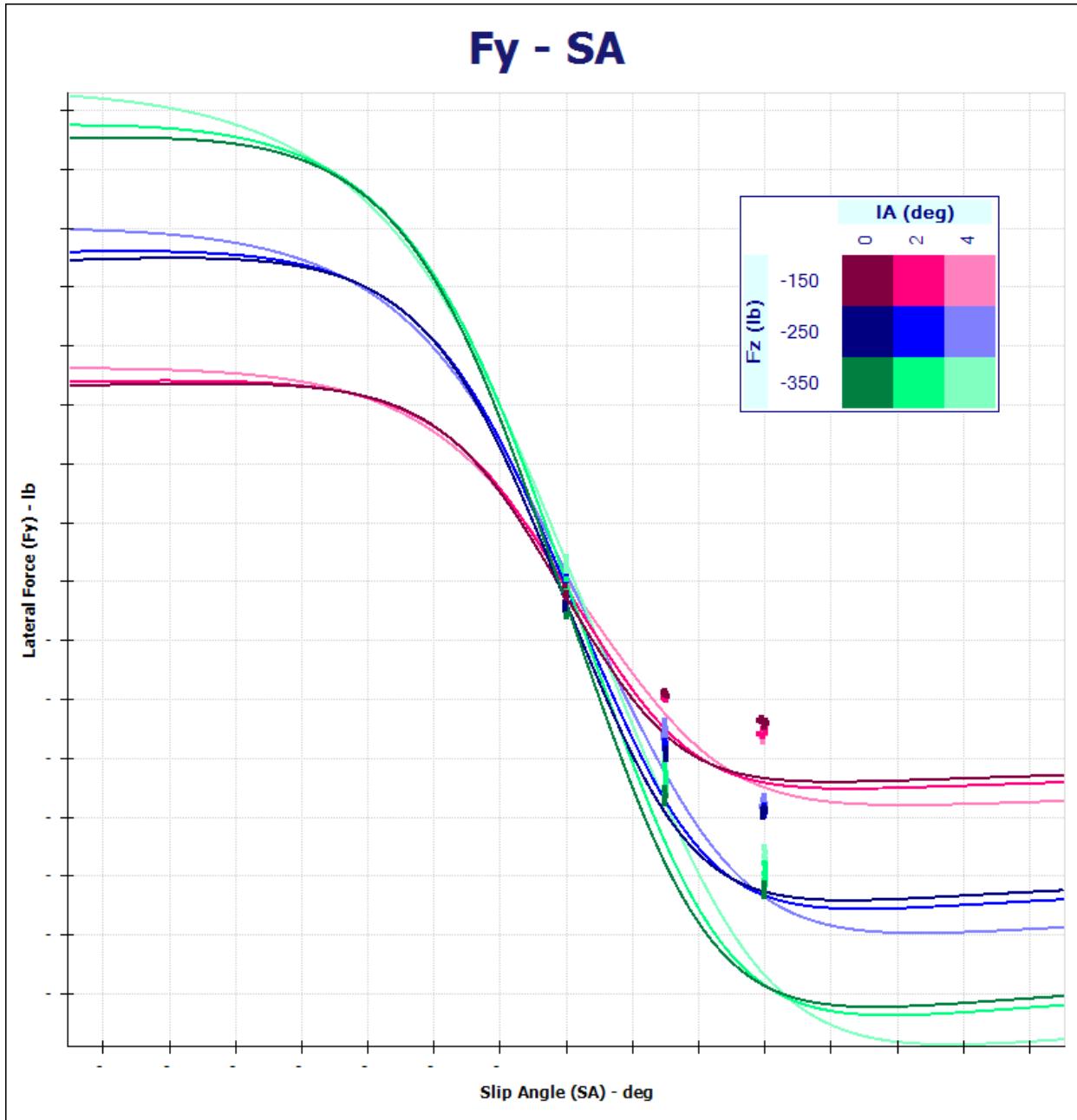


Figure 3.18: Lateral Force vs. Slip Angle for the Model and Data before Scaling

Therefore to remedy this problem a scaling factor will be created and applied to the previously created models. To create a scaling factor click on the *New Scaling Factor* button in the upper right corner of the project tree area. Figure 3.16 shows this button, circled in red. Scaling factors can only be created for the Pacejka models. Similarly to the tire data and models, the scaling factor can be deleted, copied, or renamed by right clicking on it in the project tree.

In Figure 3.19 a previously added scaling factor can be seen in the project tree. The scaling factor can be applied to a tire model by dragging and dropping it onto the tire model or by right clicking on the tire model and choosing *Add Scaling Factor*. The small red "S" that appears on the selected tire model indicates that a scaling factor has been applied to the model. The scaling factor applied to a tire model can be removed or viewed by right clicking on the tire model.

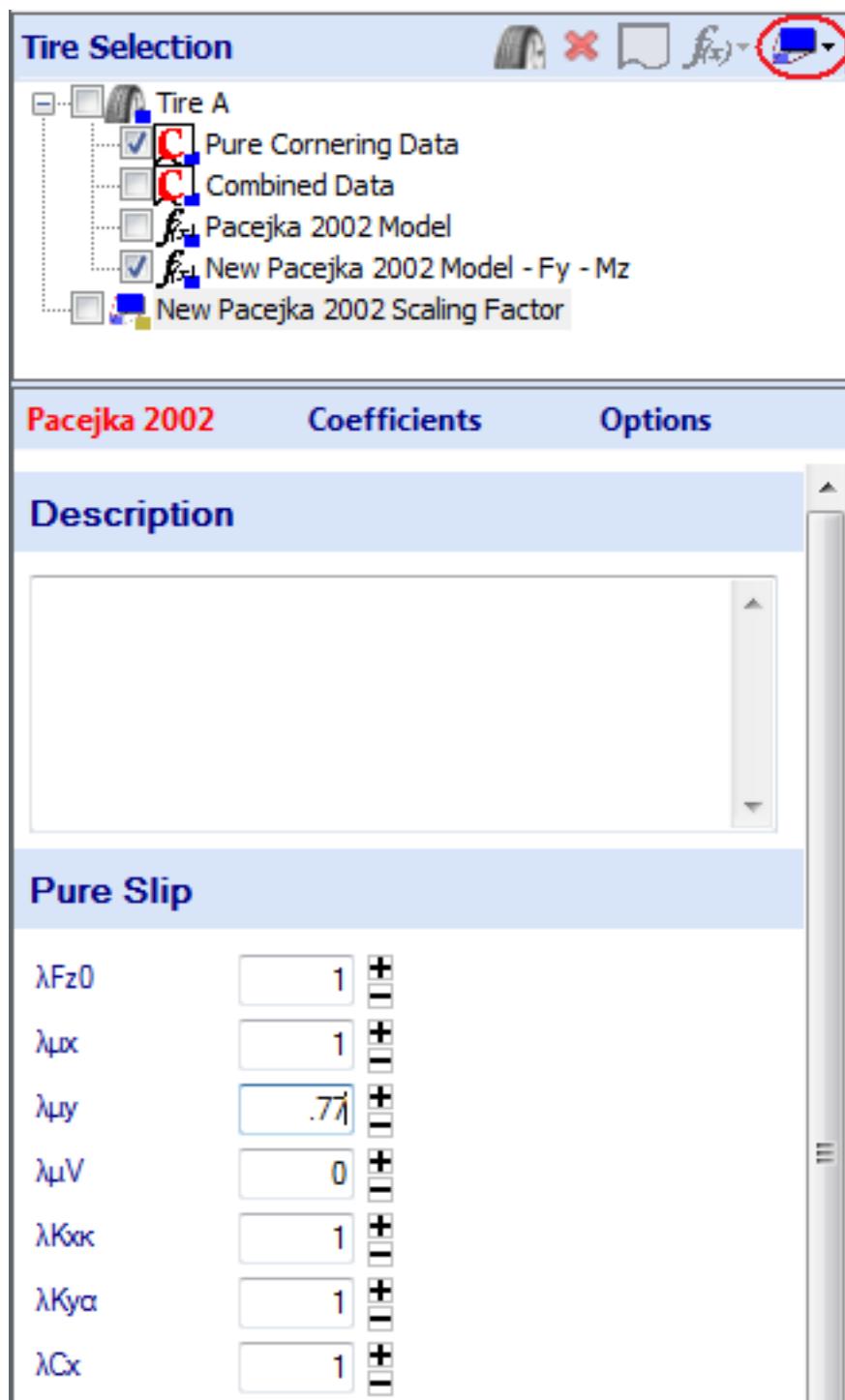


Figure 3.19: New Scaling Factor

When a scaling factor is selected in the project tree its coefficients will appear in the data entry area as can also be seen in Figure 3.19. A description of these coefficients appears in Table 8.9. The coefficients can be modified manually or by double clicking on the "+" or "-".

This will increase or decrease the value of the scaling factor by 10%. If the scaling factor is applied to a graphed model holding down on these buttons will show a preview of the change (For more information on adjusting models refer to section 3.6). Decreasing the peak lateral friction coefficient, $\lambda\mu_y$, will decrease the lateral force of the model. Figure 3.20 shows the lateral force of the model and data shown in Figure 3.18.

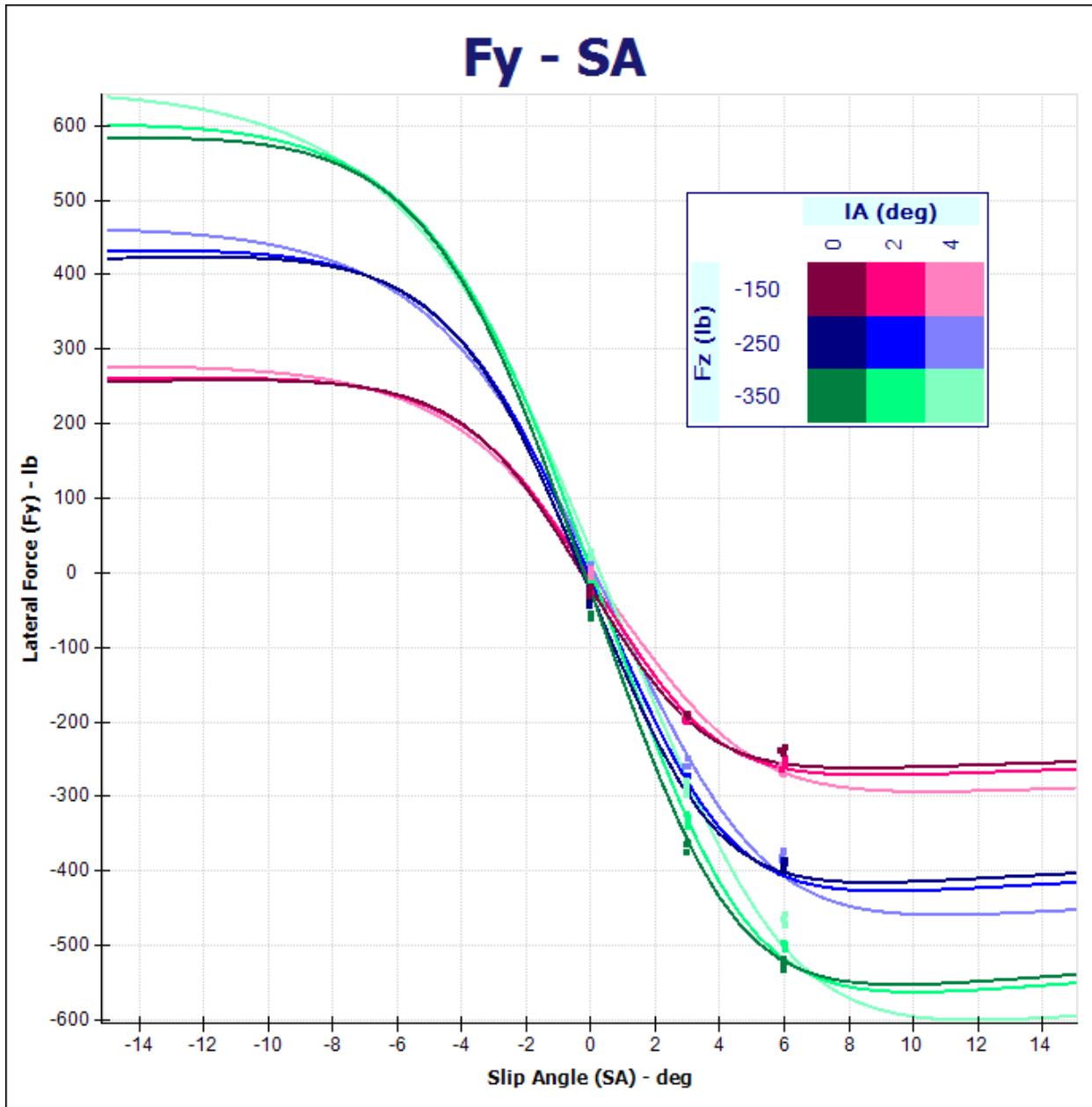


Figure 3.20: Lateral Force vs. Slip Angle for the Model and Data after Scaling

Scaling factors are also commonly used to adjust tire models to more accurately represent the actual vehicle performance. This is necessary because the conditions and surfaces the tires

are tested on are typically different than those they are to be used on. For more information regarding the scaling coefficients and their effect on the models please refer to the section 8.9.

3.3.4 Pure and Combined Longitudinal Model

Now that the previously created model is properly scaled a pure longitudinal and combined longitudinal model can be created. This process will be similar to the previous models but both models will be created simultaneously. They will be created at the same time because the combined lateral and longitudinal tire data is collected at various slip angles. Thus the tire data is representative of pure and combined longitudinal force.

First, the combined data to be used should be imported and properly collapsed (These operations are covered in section 2). Then select the combined lateral and longitudinal tire data in the project tree. In the *Model Fitting* section select the appropriate tire model and click on the *Fit Model* button. This will open the *Advanced Fitting Options Window*. Set the dropdown boxes corresponding to *Fy Pure* and *Mz Pure* to the appropriate tire model to fix these coefficients. Click *Done* once this is completed.

The *Model Fitting Selection* window will now open. For this case you would select *Fit* in the *Fx Pure* drop down box and *Fit and Calculate Error* in the *Fx Combined* drop down box as shown in Figure 3.21. You can also calculate the combined error for both models by selecting *Fit and Calculate Error* for both models. *Fit and Calculate Error* must be selected for at least one of the models. The rest of the tire model can now be created in the same way as the lateral force model. When finished a new tire model will be created that contains both the pure and combined longitudinal force coefficients.

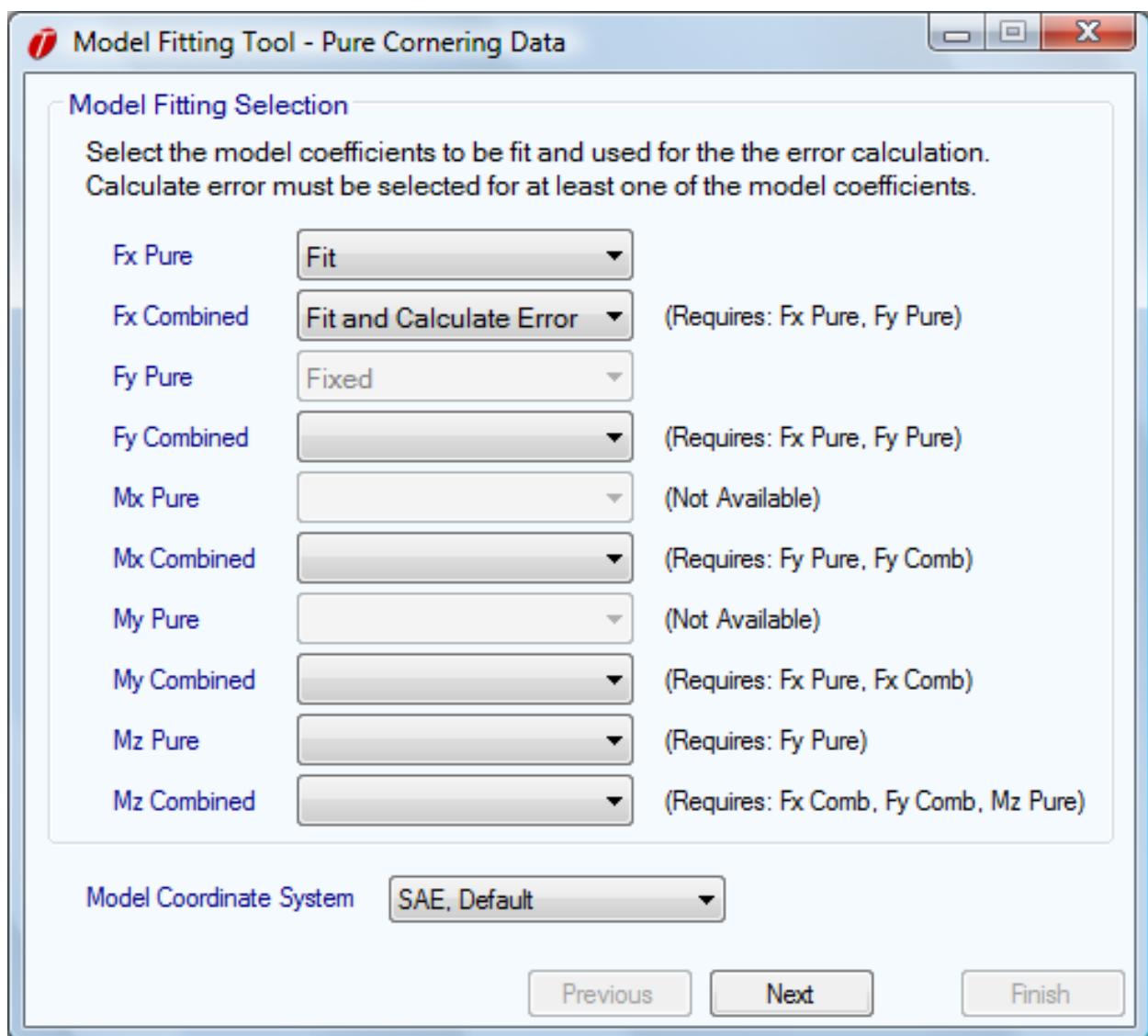


Figure 3.21: Fitting Multiple Models Simultaneously

This model should be checked by graphing the longitudinal force as a function of the slip ratio and as a function of the normal load, F_z . Examples of these graphs are shown in Figure 3.22 and Figure 3.23. You should check that the model curves correlate well to the tire data and are well behaved outside of the measurement area especially if the tire model is going to be used for simulation. If the models need to be adjusted please refer to section 3.6.

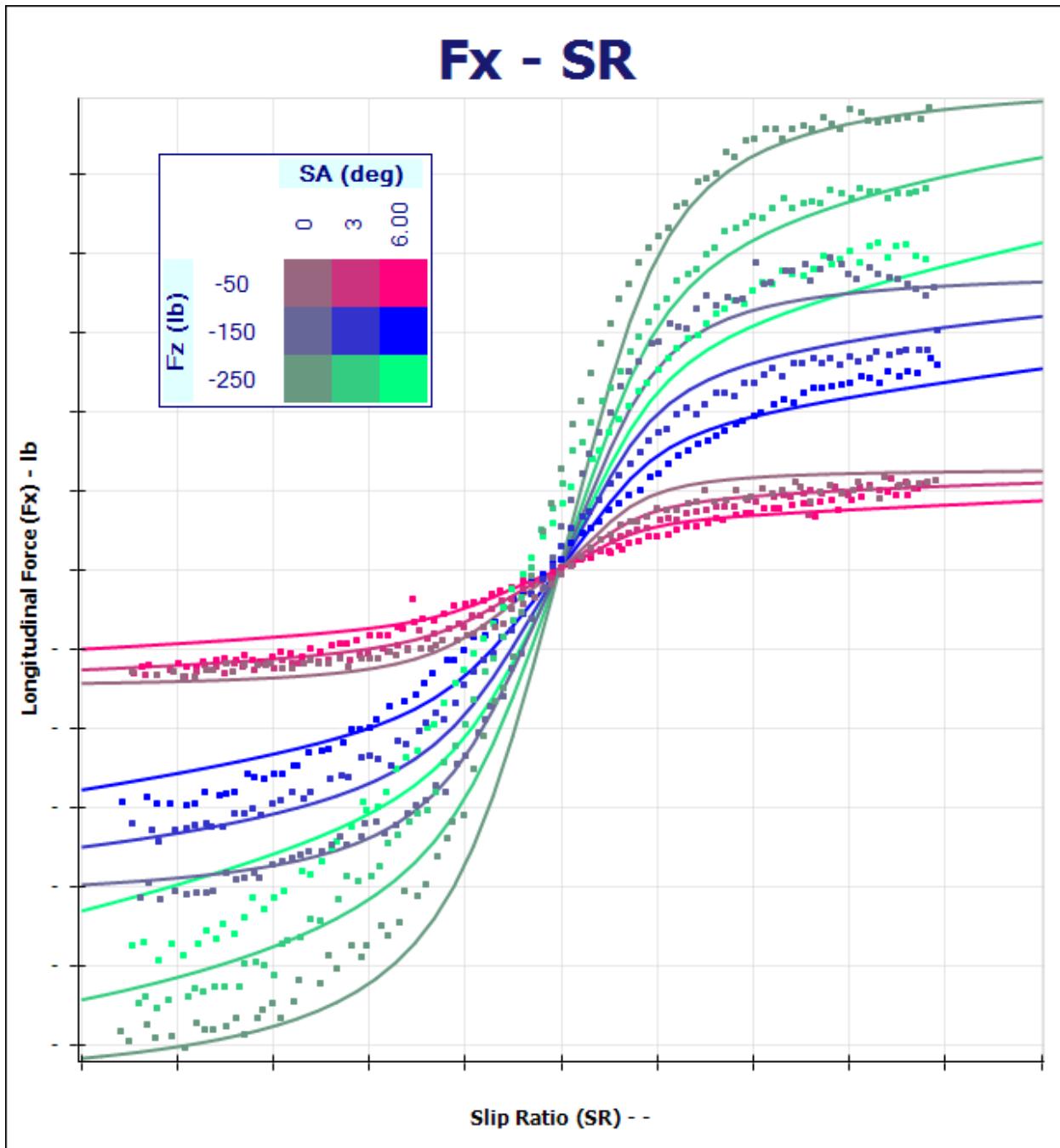


Figure 3.22: Longitudinal Force vs. Slip Angle at Different Slip Angles

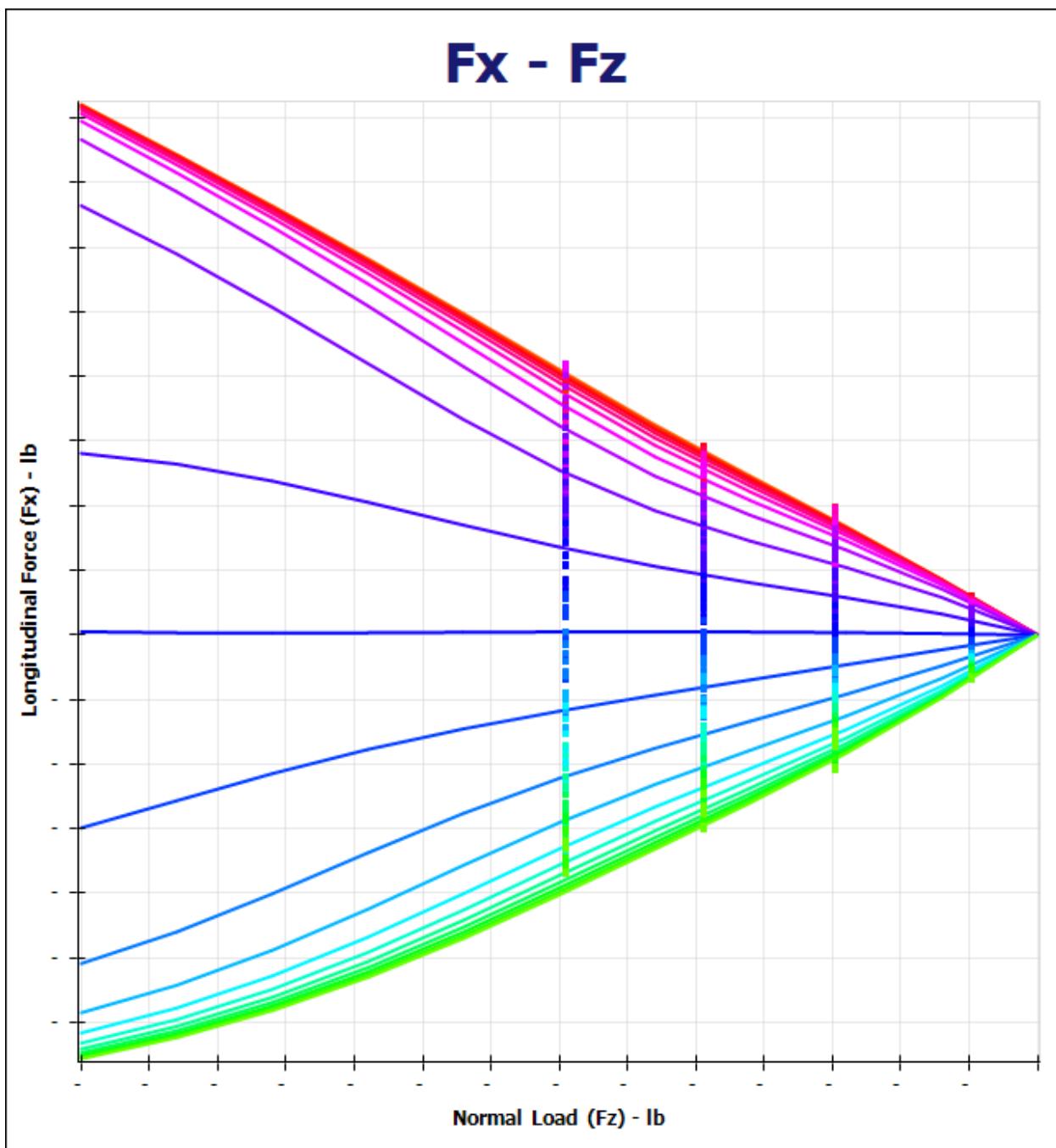


Figure 3.23: Longitudinal Force vs. Normal Load at Different Slip Ratios

3.3.5 Combined Lateral Model

The combined lateral model is also fit from the combined lateral and longitudinal data. The procedure is very similar to the other models except that the scaling factor should be applied

when the model is fit. This is the case because the model will be based on the unscaled Fy Pure coefficients. This is done by selecting the appropriate scaling factor in the *Advanced Fitting Options* as is shown in Figure 3.24. As can also be seen in the figure all of the previously determined coefficients are set to the appropriate tire model.

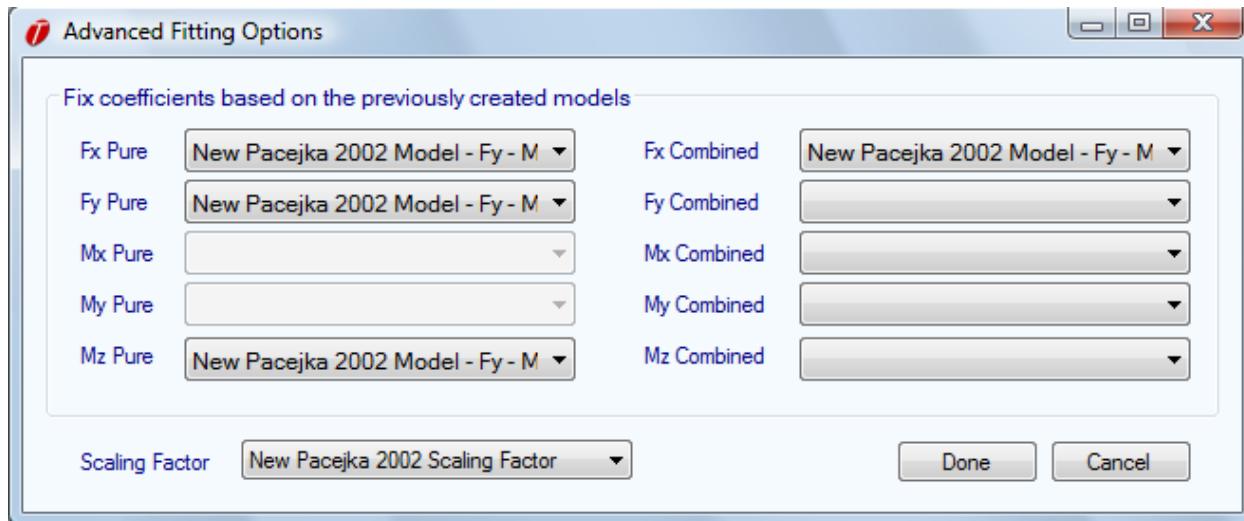


Figure 3.24: Fitting with a Scaling Factor

A friction ellipse can now be plotted to ensure the accuracy of the combined lateral and longitudinal models. A graph of a friction ellipse is shown in Figure 3.25. If these models are to be used for simulation it is very important to check that the model curves are well behaved outside of the measurement area.

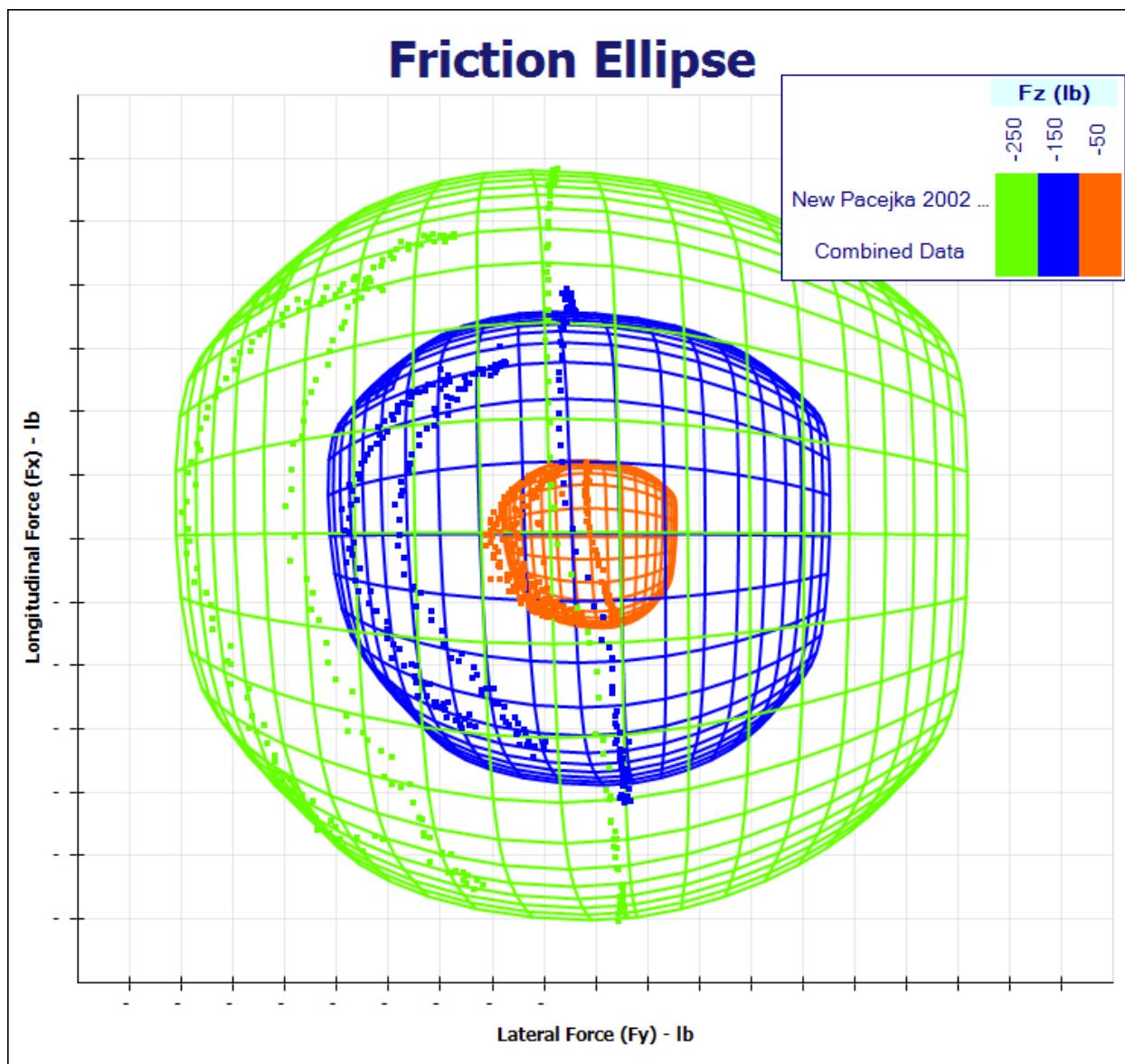


Figure 3.25: Friction Ellipse

3.3.6 Additional Models

Additional tire properties can also be fitted to the Pacejka models. These include combined models of aligning torque, rolling resistance, and overturning moment. Table 3.1 summarizes all of the models that can be fit to Pacejka coefficients in OptimumT. It also shows what models are required before creating a new model. The required models can either be fixed prior to or fit concurrently with the new model. These models are fit in the same way as the previous example.

Table 3.1 displays the separate sets of coefficients that are available in the Pacejka models. Only combined, pure, rolling resistance and overturning models can be fit with the Pacejka models. The overturning moment and rolling resistance models are not available in the Pacejka '96 model.

Model		Requires
Fx Pure	Longitudinal Force	
Fx Comb	Combined Longitudinal Force	Fx Pure, Fy Pure
Fy Pure	Lateral Force	
Fy Comb	Combined Lateral Force	Fx Pure, Fy Pure
Mx Pure	Overturning Moment	Not Available
Mx Comb	Combined Overturning Moment	Fx Pure, Fx Comb (not available in '96)
My Pure	Rolling Resistance	Not Available
My Comb	Combined Rolling Resistance	Fy Pure, Fy Comb (not available in '96)
Mz Pure	Aligning Torque	Fy Pure
Mz Comb	Combined Aligning Torque	Fx Comb, Fy Comb, Mz Pure

Table 3.1: Pacejka Models (Mx Comb and My Comb are not available in the Pacejka '96 model)

3.4 Model Fitting Order

The order in which the models are fit can vary depending on the tire data available and the goals of the project. However, since some of the models are dependent on each other there are some restrictions on the order. Table 3.1, above, shows what models are required before other models can be fit. This is also displayed in the *Model Fitting Selection* window in OptimumT. These requirements, which will vary depending on the type of model being fit, can be seen to the right of the dropdown boxes in Figure 3.26.

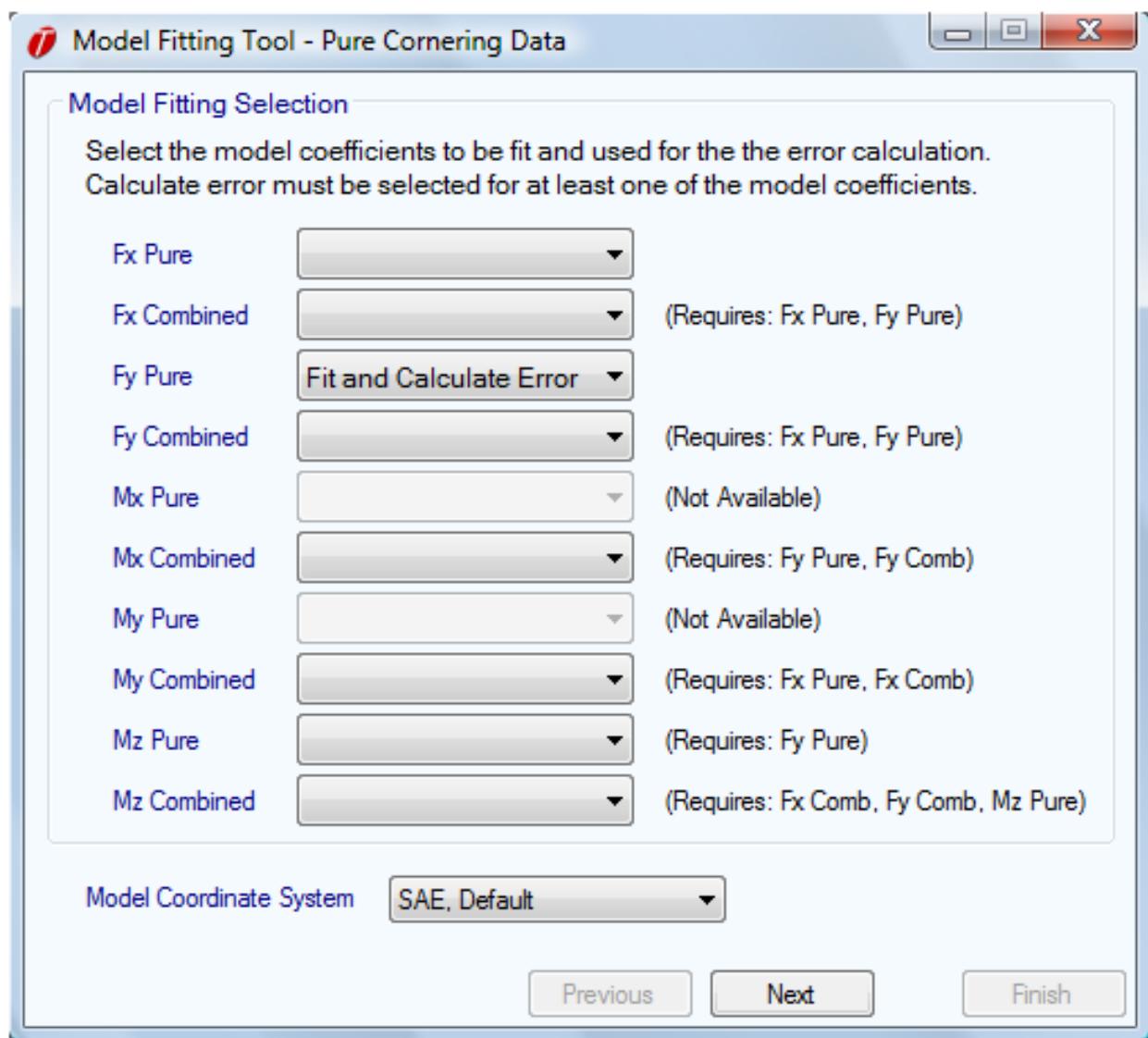


Figure 3.26: Model Fitting Requirements

The two most common model fitting sequences will be described in Table 3.2 and Table 3.3. The first sequence uses pure lateral and combined data like the model fitting example in the previous sections. The second sequence uses pure lateral, pure longitudinal and combined data. Under *Coefficients to be Fixed* the "Fix" indicates that these coefficients should be selected in the Advanced Fitting Options window. Under *Coefficients to be Fit* the "FE" indicates that these coefficients should be set to *Fit and Calculate Error* and the "Fit" indicates that these coefficients should be set to *Fit* in the *Model Fitting Selection* window.

Model Fitting with Pure Lateral and Combined Data													
Step	Data Used	Coefficients to be Fixed						Coefficients to be Fit					
		Pure			Combined			Pure			Combined		
		Fx	Fy	Mz	Fx	Fy	Mz	Fx	Fy	Mz	Fx	Fy	Mz
1	Pure Lat.								Fe				
2	Pure Lat.		Fix							Fe			
5	Combined.		Fix	Fix				Fit			Fe		
4	Combined.	Fix	Fix	Fix	Fix						Fe		
5	Combined.	Fix	Fix	Fix	Fix	Fix						Fe	

Table 3.2: Model Fitting Order with Pure Lateral and Combined Data

Model Fitting with Pure Lateral, Pure Longitudinal and Combined Data													
Step	Data Used	Coefficients to be Fixed						Coefficients to be Fit					
		Pure			Combined			Pure			Combined		
		Fx	Fy	Mz	Fx	Fy	Mz	Fx	Fy	Mz	Fx	Fy	Mz
1	Pure Lat.								Fe				
2	Pure Lat.		Fix							Fe			
5	Combined.		Fix	Fix				Fit			Fe		
4	Combined.	Fix	Fix	Fix									
5	Combined.	Fix	Fix	Fix	Fix								
6	Combined.	Fix	Fix	Fix	Fix	Fix							

Table 3.3: Model Fitting Order with Pure Lateral, Pure Longitudinal and Combined Data

3.5 Model Coefficient Form

After a model is fit the coefficients corresponding to it can be accessed by clicking on the model in the tire project tree. This will display the model coefficient form in the data entry area as shown in Figure 3.27. This form includes the model coefficients, a description box, and the option to change the name displayed in the legend for the model. In this form the model coefficients can be modified by the user. This will be covered in detail in the next section.

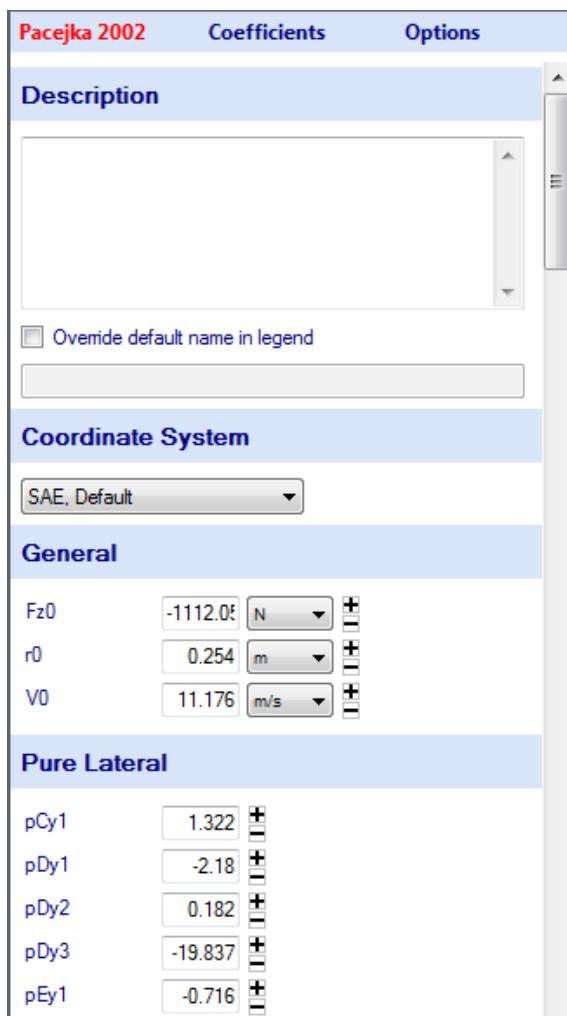


Figure 3.27: Model Coefficient Form

After a model is fit information regarding the model fitting will be directly imported into the description box as can be seen in Figure The information included in the description box is the final error of the fitting, the model that was fit, the data file used, the coefficients fit, the error evaluation method, the coefficient boundary, and the solver parameters.

3.6 Adjusting Models

Once the model is created and graphed against the raw data you might want to adjust it to improve its accuracy in a certain region or improve its behavior beyond the measurement area. This can be done easily in OptimumT.

Selecting a tire model in the project tree will show the model coefficients in the data entry area. You can modify the coefficients by 10% by double clicking on the "+" or "-" button

next to the coefficient¹. If the model is shown on a graph (see chapter 6 for information about graphing), holding down the "+" or "-" button will show a preview of the model with the coefficient modified by 10%. An example of this is shown in Figure 3.28. In this figure the "+" button of the $pDy1$ coefficient is being held down. As can be seen in Figure 3.28 this change has a large effect on the graph. Changing other coefficients will have a much different effect on the curve both in terms of its magnitude and shape. The coefficients can also be adjusted manually by changing the values in the boxes. When applied to a tire model the coefficients of a scaling factor can be adjusted in the same way.

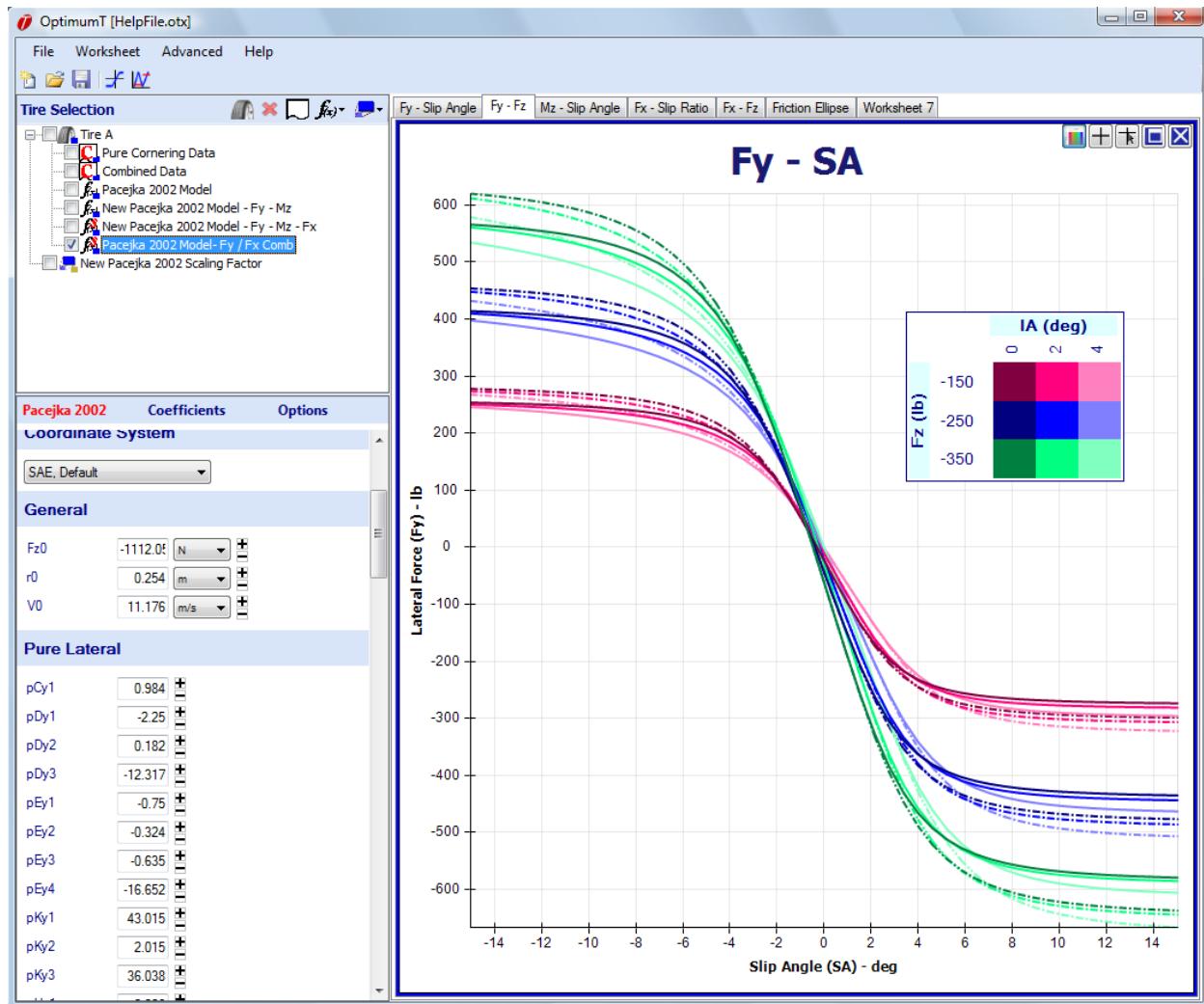


Figure 3.28: Model Adjustment Preview Feature

¹When clicking the "+" and "-" button, the coefficient is either multiplied or divided 1.10

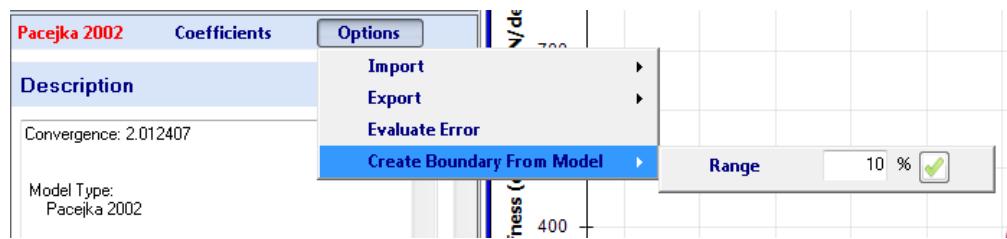


Figure 3.29: Creating a boundary from an existing model.

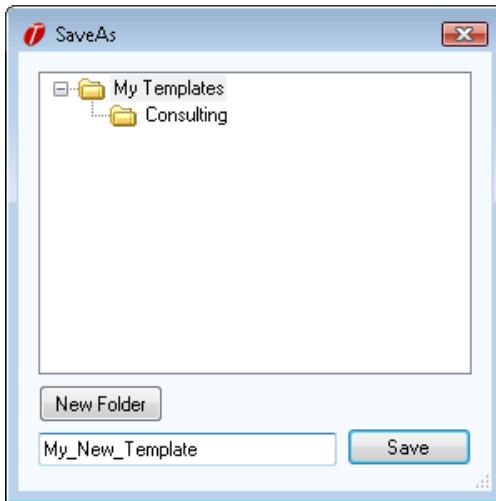


Figure 3.30: To save the boundary created from a model, specify the name to save it as in this dialog.

3.7 Creating Coefficient Boundaries From an Existing Model

You can create a boundary from an existing model. When doing this, you specify the half-width of the boundary. In the model coefficient form, click on *Options* then *Create Boundary From Model* to arrive at the range specification (shown in Figure 3.29). Enter the desired range and click the check-mark button. The minimum for each coefficient is created by reducing the coefficient value by the user specified percentage, and the maximum is created by increasing the value by the same amount. For example, if one of the coefficients of the model has a value of 1.00 and you specify a boundary range of 10%, the minimum will be 0.90 and the maximum will be 1.10.

Once you have clicked the check-mark button to create the boundary, you will be presented with the dialog shown in Figure 3.30. Enter the name to save the boundary as and choose a folder to put the boundary in.

Chapter 4

Graphing

The OptimumT plotting tool is a powerful feature that allows users to create nearly any type of tire graph that they please. Two different types of graphs can be created in OptimumT. One is a *standard graph* which allows plotting of any quantity against any other quantity in either two- or three-dimensions. The other is a *report graph* which shows measured quantities versus sample number. It is only capable of plotting raw data.

4.1 Standard Graphs

Standard graphs can display raw data and tire models. The data or model to be graphed can be specified by selecting the checkbox next to the desired items in the project tree. A new standard graph can be added to a worksheet in multiple ways. Right clicking on the worksheet or worksheet tabs and selecting *Add Graph*, selecting *Add Graph To Worksheet* from the *Worksheet* menu on the main toolbar or by clicking on the *Add Graph* button in the toolbar next to the Save button. Up to four graphs can be placed on a single worksheet. When a graph is added to the worksheet the graph setup form will appear in the data entry area. To show the graph setup form when the data entry area is showing another form, click on the graph. You must specify six graph inputs and two or three graph outputs to create a graph. These will be further discussed in the following sections.

4.1.1 Types of Standard Graphs

There are three types of standard graphs. Single direction line graphs, crossed line graphs and surface graphs. You can select the type of graph by choosing the appropriate option in the list of radio buttons near the bottom of the data entry form.

Single Direction Line Graphs

An example of a single direction line graph is shown in Figure 4.1. When plotting models using this type of graph, lines are generated based on the first of the graph inputs. For example, if the first graph input (labeled as *Indep. Variable*) is slip angle, lines will be generated by varying the slip angle while keeping all the other graph inputs constant.

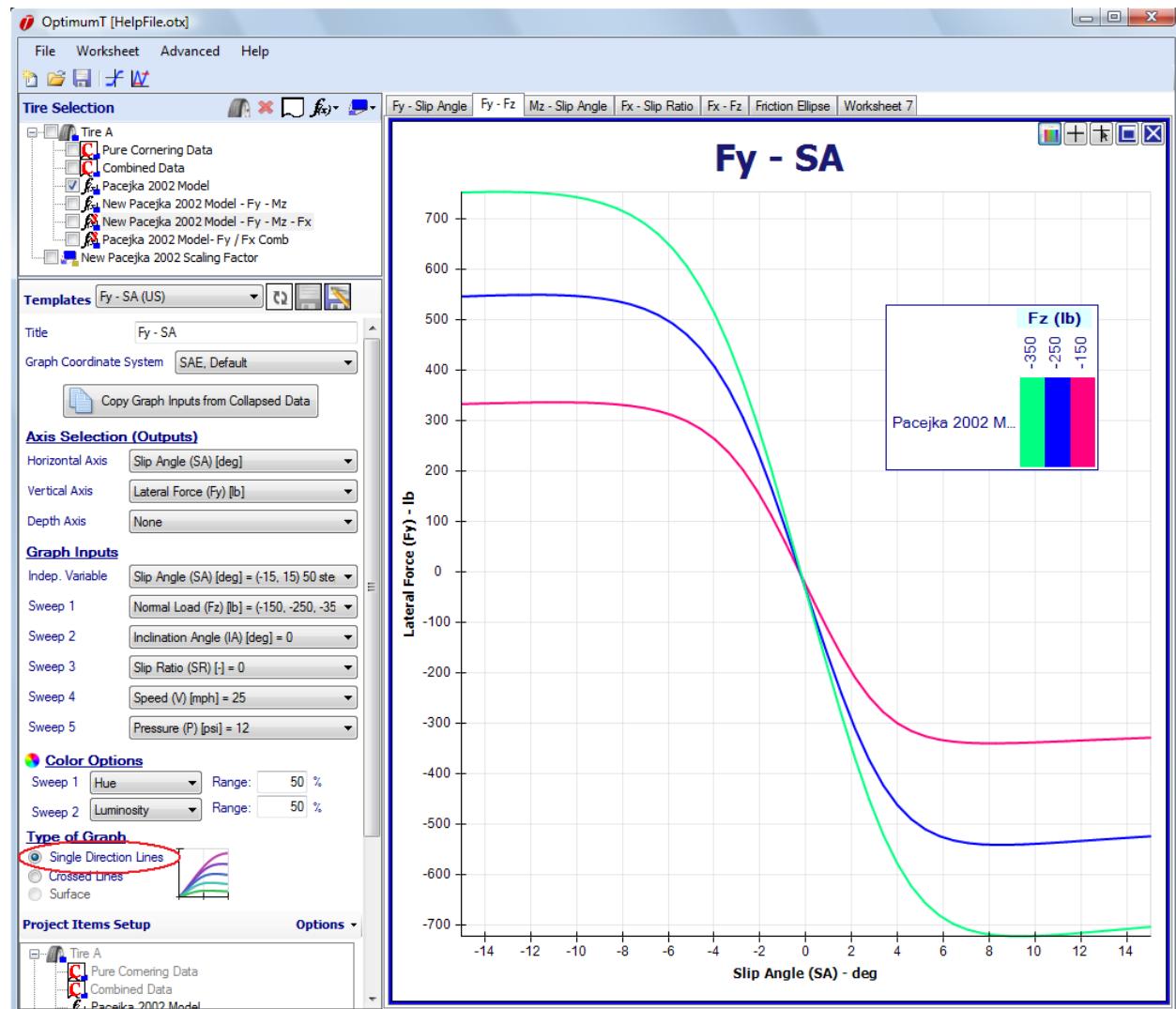


Figure 4.1: Single Direction Line Graph

Crossed Line Graphs

An example of a crossed line graph is shown in Figure 4.2. As can be seen in the figure there are two independent variables for this type of graph. Therefore, two sets of lines will be generated for this graph. The first set of lines is generated by varying the first graph

input while holding all others constant. The second set of lines is generated by varying the second graph input while holding all the others constant. For example, if the first graph input (labeled *First Indep.*) is set as slip angle and the second input (labeled *Second Indep.*) is set as slip ratio, then one set of lines will be generated with constant slip ratio and varying slip angle and another set of lines will be generated with constant slip angle and varying slip ratio. This type of graph is useful when creating a friction ellipse where you want lines of constant slip angle and lines of constant slip ratio.

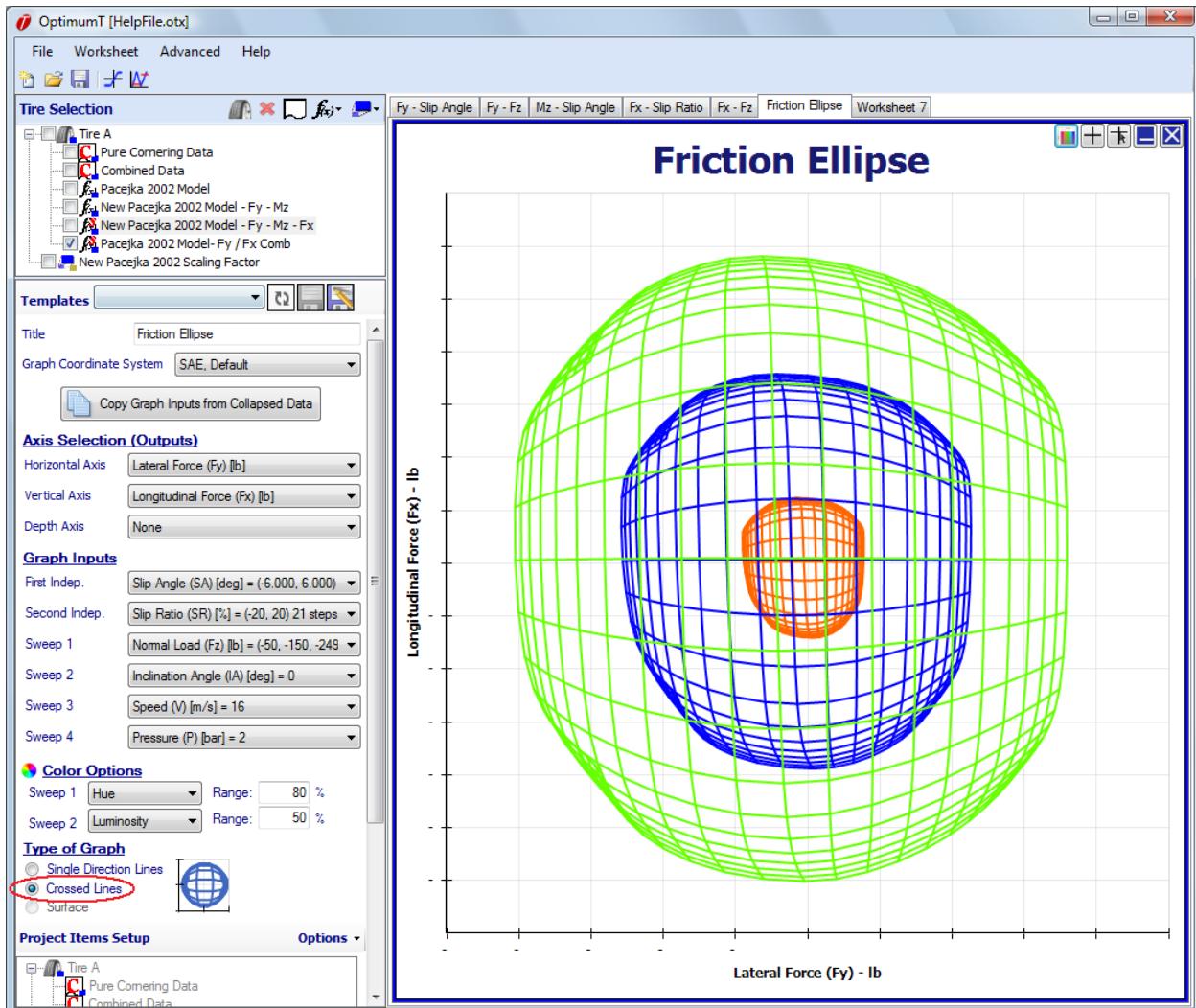


Figure 4.2: Crossed Line Graph

Surface Graphs

An example of a surface graph is shown in Figure 4.3. Surface graphs are only available when a third graph output is selected in the Depth Axis drop down box. Surfaces are generated

in the same way that crossed line plots are generated, but the area between the lines is filled to generate the surface.

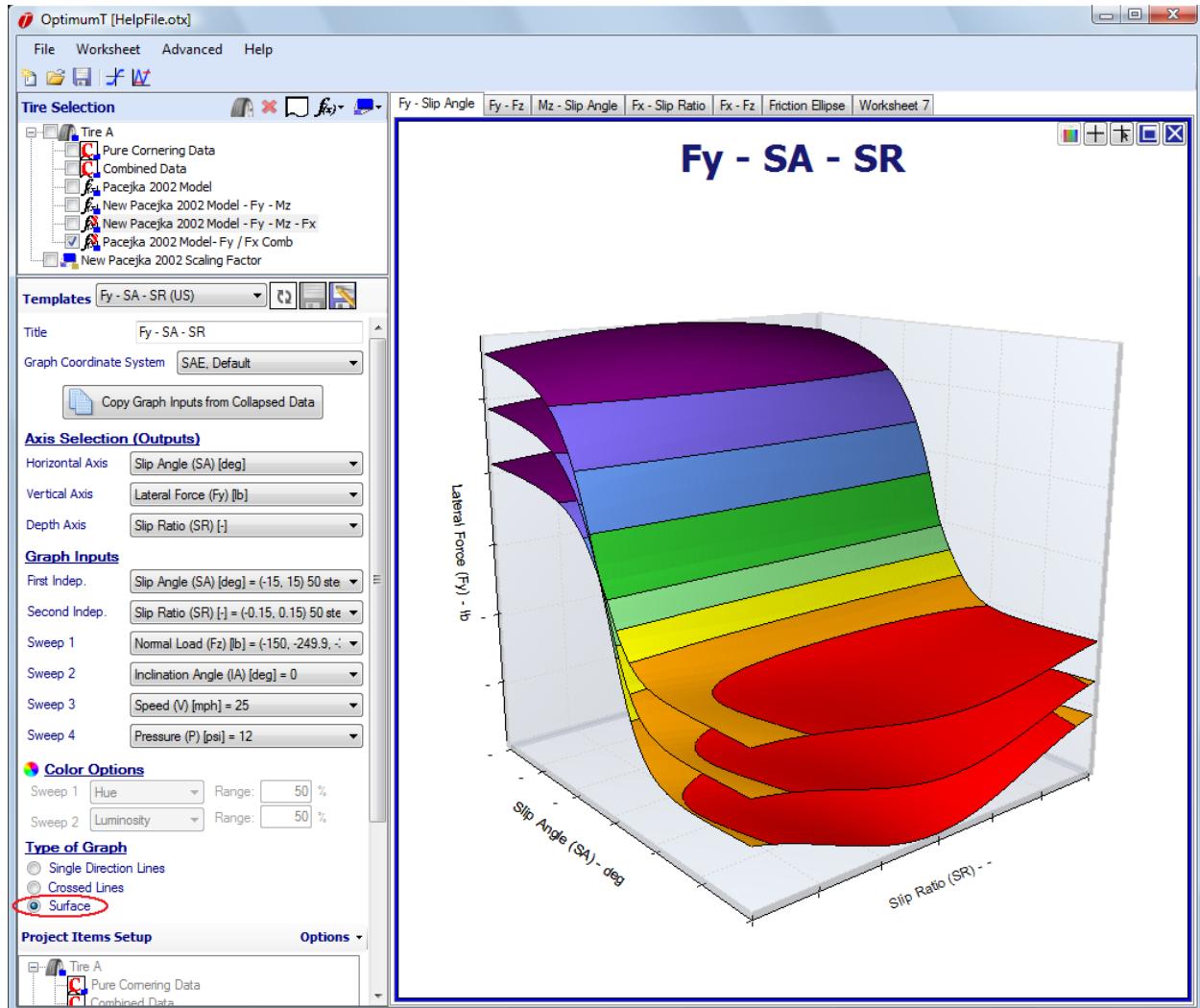


Figure 4.3: Surface Graph

4.1.2 Setting up a Standard Graph

For OptimumT to generate a graph all six possible steady state tire conditions must be specified. These graph inputs are:

- Slip angle, α
- Slip ratio, κ
- Normal load, F_z

- Inclination angle, γ
- Velocity, V
- Inflation pressure, P

In addition to the six graph inputs, the two graph outputs, the *Horizontal Axis* and *Vertical Axis* must be set to create a graph. If you are making a 3D graph a third output, the *Depth Axis*, must also be specified. Essentially, the OptimumT plotting tool is evaluating a tire model or searching through the tire data for the three axes as follows:

$$X = f_1(\alpha, \kappa, F_z, \gamma, V, P)$$

$$Y = f_2(\alpha, \kappa, F_z, \gamma, V, P)$$

$$Z = f_3(\alpha, \kappa, F_z, \gamma, V, P)$$

4.1.3 Graph Outputs

The graph outputs specify what quantity and data range that each graph axis should represent. The graph output dialog is shown in Figure 4.4. In this dialog the graph output to be used for each axis can be chosen. The unit displayed and the axis scaling can also be chosen. If the Auto Scale check box is selected, the scaling will adjust according to the data shown on the graph. The Auto Scale is unchecked then minimum and maximum value for the axis can be entered. Selecting Hide Axis Values will remove the numeric values from the graph axis.

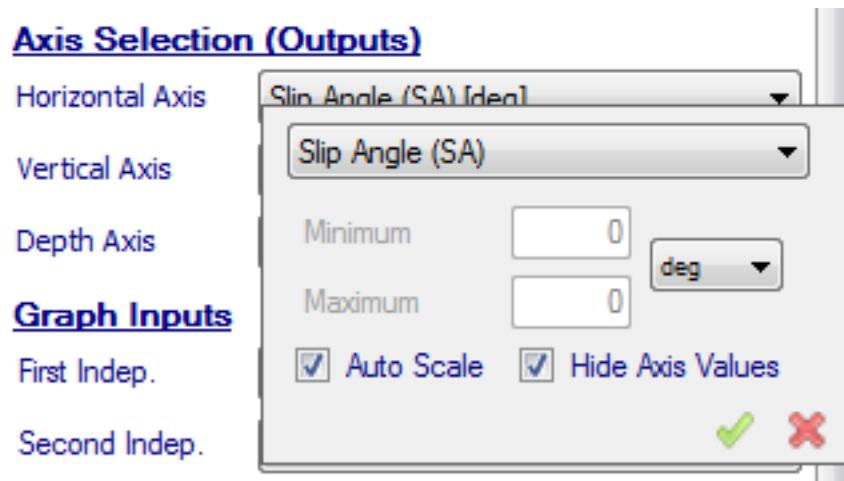


Figure 4.4: Graph Output Dialog

Currently OptimumT includes over 30 outputs that can be graphed. These outputs as well as their associated unit types are displayed in Table 4.1. The unit types are displayed because

OptimumT allows the user to select between many different units. For more information about the available units in OptimumT please refer to section 8.3.

Graph Outputs	
Basic	Unit Type
Inclination Angle (IA)	angle
Slip Angle (SA)	angle
Slip Ratio (SR)	ratio
Speed (V)	velocity
Pressure (P)	pressure
Loaded Radius (RL)	length
Force / Moment	Unit Type
Longitudinal Force (Fx)	force
Lateral Force (Fy)	force
Normal Load (Fz)	force
Overturning Moment (Mx)	moment
Rolling Resistance (My)	moment
Aligning Torque (Mz)	moment
Derivatives	Unit Type
Cornering Stiffness	force / angle
Inst. Cornering Stiffness	force / ratio
Slip Stiffness	force / angle
Inst. Slip Stiffness	force / ratio
Camber Stiffness	force / angle
Inst. Camber Stiffness	force / angle
Lateral Load Sensitivity	ratio
Longitudinal Load Sensitivity	ratio
Aligning Moment Load Sensitivity	length
Overturning Moment Load Sensitivity	length
Rolling Resistance Load Sensitivity	length
Normalized	Unit Type
Normalized Longitudinal Force	ratio
Normalized Lateral Force	ratio
Normalized Inst. Cornering Stiffness	1 / angle
Normalized Inst. Slip Stiffness	ratio
Normalized Inst. Camber Stiffness	1 / angle
Cornering Stiffness Coefficient	1 / angle
Slip Stiffness Coefficient	ratio
Camber Stiffness Coefficient	1 / angle

Coefficient of Friction	Unit Type
Lateral Coefficient of Friction	ratio
Longitudinal Coefficient of Friction	ratio
Moment Arm	Unit Type
Pneumatic Trail	length
Pneumatic Scrub Radius	length
Peak	Unit Type
Slip Angle at Peak Fy (negative)	angle
Slip Angle at Peak Fy (positive)	angle
Slip Ratio at Peak Fx (negative)	ratio
Slip Ratio at Peak Fx (positive)	ratio
Peak Lateral Force (negative)	force
Peak Lateral Force (positive)	force
Peak Longitudinal Force (negative)	force
Peak Longitudinal Force (positive)	force
Offset	Unit Type
Fx Offset (Fx @ SR = 0)	force
Fy Offset (Fy @ SA = 0)	force

Table 4.1: Graph Outputs

4.1.4 Graph Inputs

The graph inputs can be *fixed values*, *constant sweeps* or *custom sweeps*. The graph inputs are specified by clicking on the box next to one of the six graph inputs. A dialog box similar to that shown in Figure 4.5 will appear. The first item to be chosen is the variable for the graph input. This is chosen in the first dropdown box of the dialog. As can be seen in Figure 4.5 "Slip Angle" is currently selected as the first independent. As previously mentioned, all six inputs must be present.

For a *Single Line* graph the first graph input, labeled *Indep. Variable*, will be the quantity used to generate the graph lines. For a *Crossed Line* or *Surface graph* the first two graph inputs, labeled *First Indep.* and *Second Indep.*, will be the quantities used to generate the graph lines. The rest of the graph inputs, labeled *Sweep*, allow the plotting of multiple lines

depending on the other tire conditions. For the first two of these, the option of changing the color of the graphed lines according to these inputs is available (see section 4.1.8 for more information).

The second dropdown box in this dialog allows the user to specify the type of sweep to be graphed. For each input in the graph input dialog three different types of sweeps can be chosen. The three sweep types available are: *Fixed Value*, *Constant Step Sweep*, and *Custom Sweep*. For the independent variables either the *Constant Step Sweep* or *Custom Sweep* should be used. For the sweep variables any of the sweep types can be used.

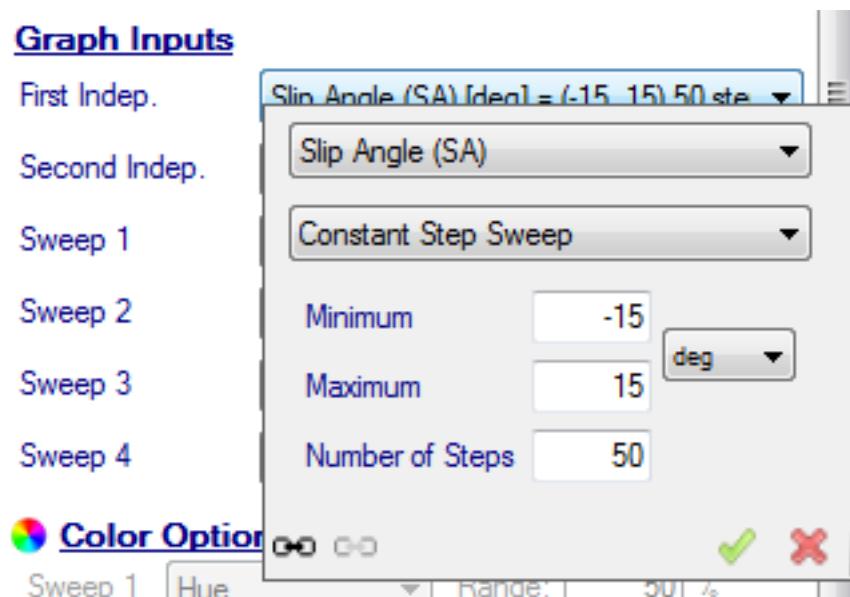


Figure 4.5: Graph Input Dialog

Fixed Value Graph Inputs

As the name suggests, fixed value graph inputs hold a particular variable constant when generating the graph. In the example shown in Figure 4.6, *Fixed* is selected in the second drop down box. Therefore the inclination angle is being held at a constant 2 degrees. In the fixed value dialog, there is also a box for data tolerance. This value tells OptimumT what tolerance to use when searching raw data. Since there will always be some noise in the raw data, OptimumT needs to know how far from the nominal value it should consider to be "close enough" to include in the plot.

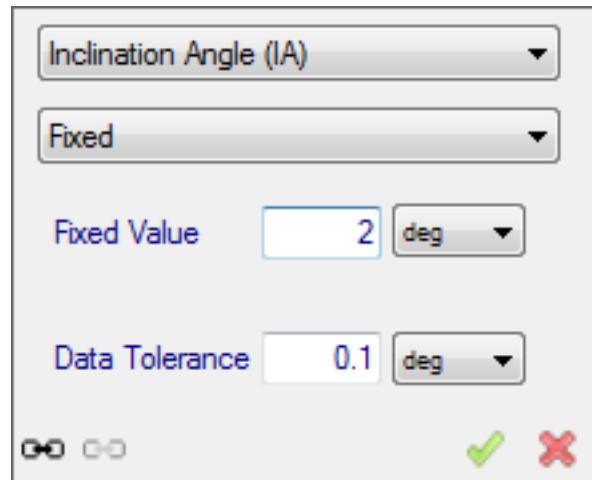


Figure 4.6: Fixed Value Dialog

Constant Step Sweep Graph Inputs

Constant step sweep graph inputs generate a set of evenly spaced points between a minimum and a maximum value. The constant sweep dialog is shown in Figure 4.7. A minimum and a maximum value for the sweep as well as the number of steps needs to be entered into the dialog. Constant sweep graph inputs do not have data tolerances as fixed graph inputs do. All data that falls within the specified range is used.



Figure 4.7: Constant Step Sweep Dialog

Custom Sweep Graph Inputs

A custom sweep graph input allows you to specify a number of arbitrary values. Figure 4.8 shows the custom sweep dialog. You can specify any number of values by typing them into the table. Additional rows can be added by clicking on the green "+" and rows can be removed by clicking on the red "-" above the input list. Like the fixed value graph input, custom sweep graph inputs have data tolerances.

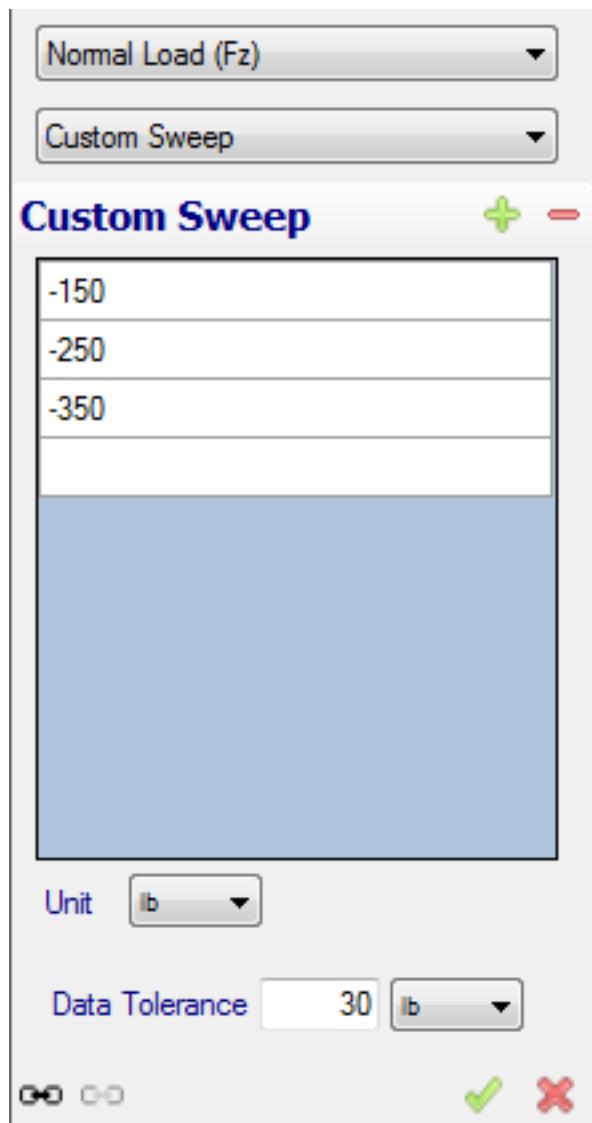


Figure 4.8: Custom Sweep Dialog

4.1.5 Plotting Raw Data

Initially when raw data is imported and graphed in OptimumT all of the data will be plotted regardless of the graph input parameters. This *Plot all Data* feature allows the user to quickly view data to ensure that it is the correct data and that it was imported successfully. If the user would like to graph data at certain conditions specified by the graph inputs, this feature needs to be unselected.

When the *Plot All Data* option is unselected, OptimumT will search for raw data points that correspond to all six of the graph inputs. Points matching the six conditions will be plotted on the graph.

There are three different ways to unselect the *Plot All Data*. The first one is by right clicking on the raw data in the project tree and selecting Remove "Plot All Data" In All Graphs. This is shown in Figure 4.9. This will disable the feature for the selected data set in all of the graphs.

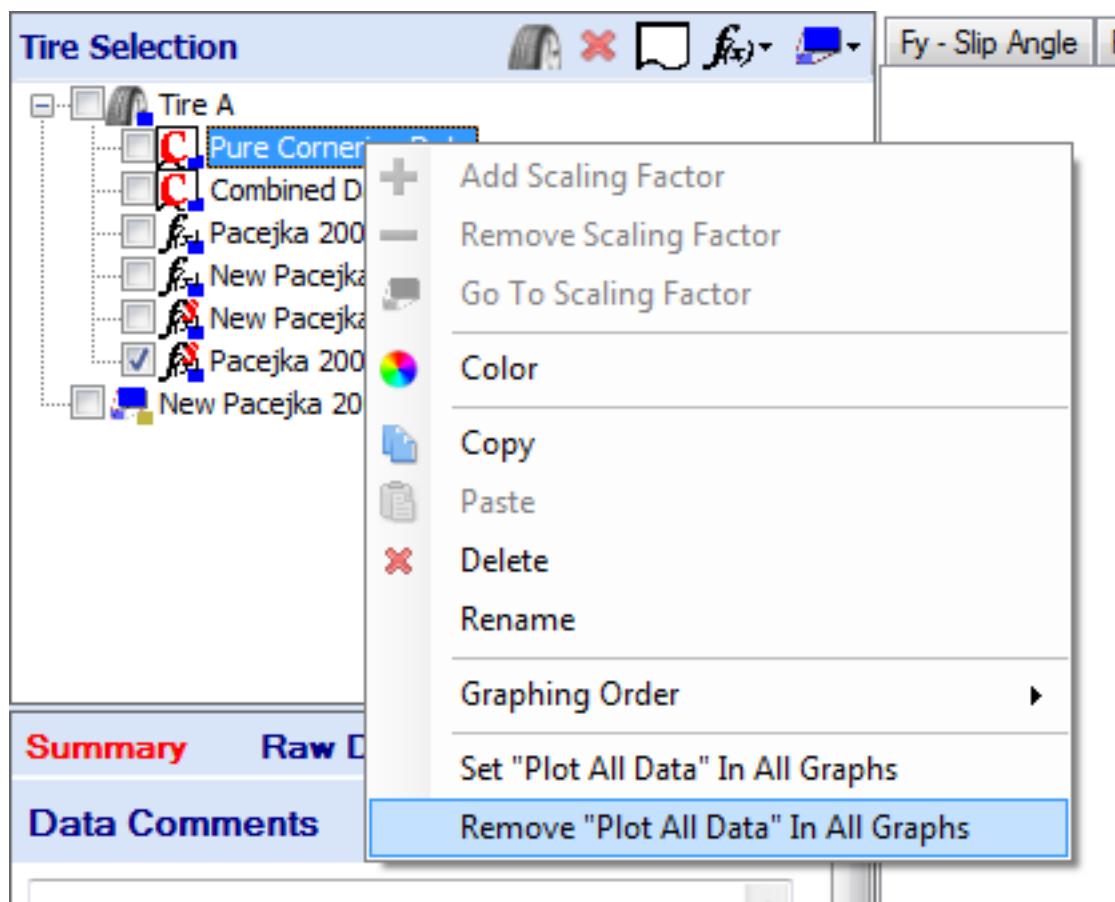


Figure 4.9: Plot All Data Option in Project Tree

The other two procedures to unselect this option are located in the *Project Items Setup* at

the bottom of the graph setup form as shown in Figure 4.10. Since this is located in the graph setup form it will only affect the graph that it corresponds to. As can be seen "<all data>" will appear to the right of the name of the data if *Plot All Data* is enabled. By right clicking on the data and clicking on the *Plot All Data* selects or unselects this feature. The *Options* button in the upper right of the figure allows the user to set all or no items to *Plot All Data*. Again this will change all of the items for the current graph only.

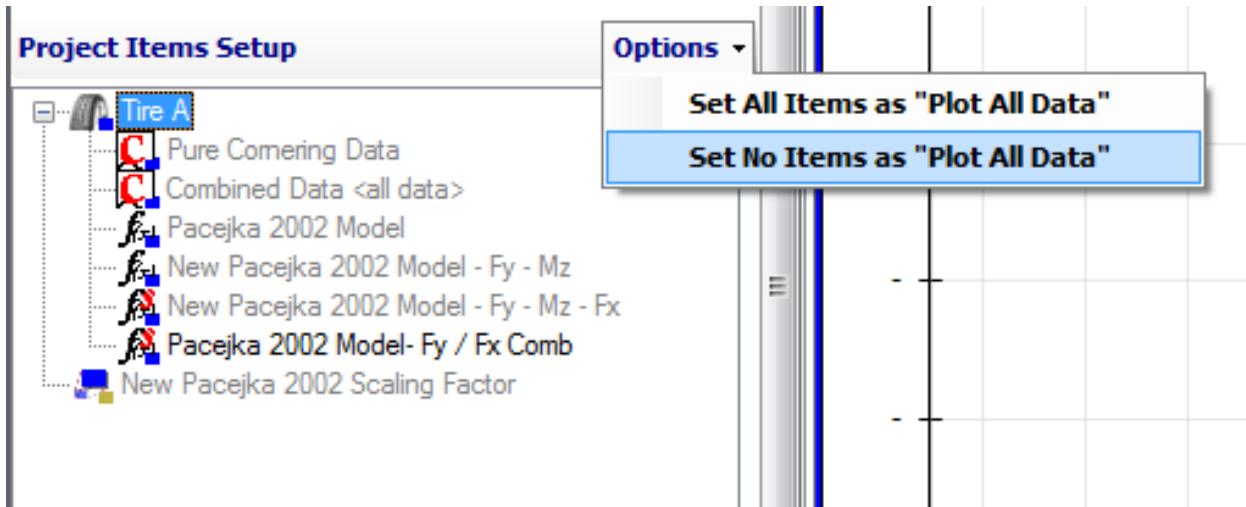


Figure 4.10: Project Item Setup

4.1.6 Copy Graph Inputs from Collapsed Data

The *Copy Inputs from Collapsed Data* button located on the graph setup form is shown in Figure 4.11. This feature allows the user to quickly set the graph inputs. It will copy the values and tolerances set in the data collapsing into the graph setup inputs. Therefore to use this feature the current project must contain collapsed data.

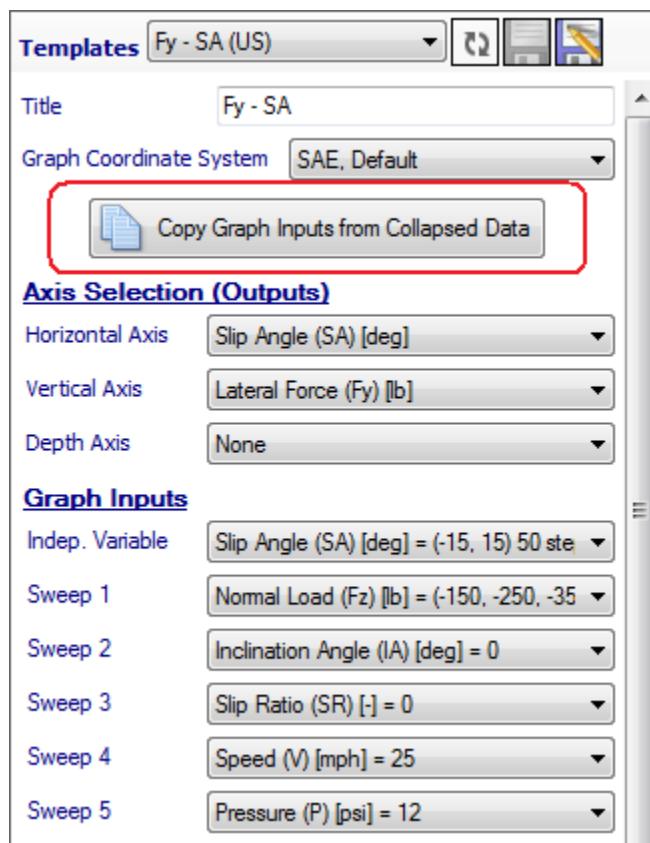


Figure 4.11: Copy Inputs from Collapsed Data Button

When this button is clicked a window similar to the one shown in Figure 4.12 will appear. This window displays all of the collapsed data in the project. Selecting one of the data sets and pressing the *Copy* button will import the collapsed data values and tolerances into the graph setup form, therefore instantly plotting the data at all of its tested conditions.

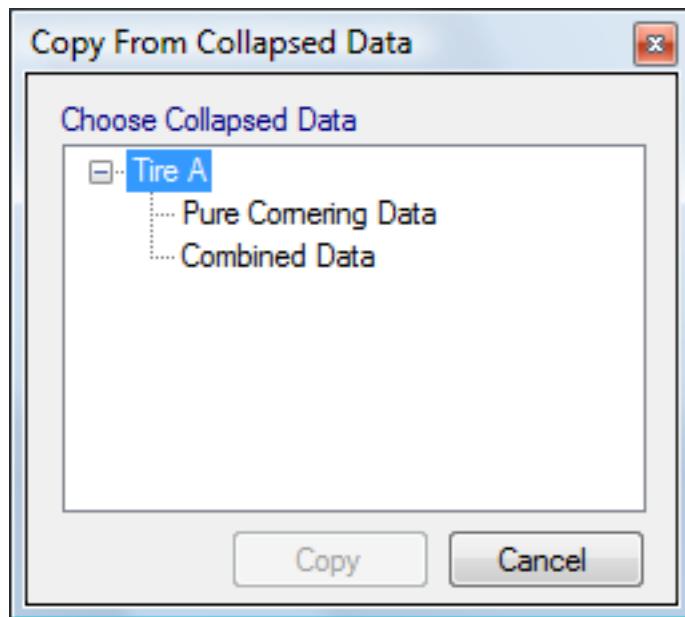


Figure 4.12: Copy Inputs from Collapsed Data Window

4.1.7 Linking Graphs

The data input in different graphs can be linked together. Thus when the linked input in one of the graphs is changed it will be applied to the other graphs that are linked to it. The inputs are linked by clicking on the *Link Sweep* button. This button as well as the *Unlink Sweep* button is circled in Figure 4.13. After clicking on this button a message box that says *Please select a graph* will appear. Clicking on any of the other graphs will link them to the original graph. The linked input of the original graph will now change to reflect the input of the selected graph. If you would like to link another graph to these two, click the *Link Sweep* button on that graph and then choose one of the previously linked graphs. Multiple graphs and inputs can be linked together in the same fashion.

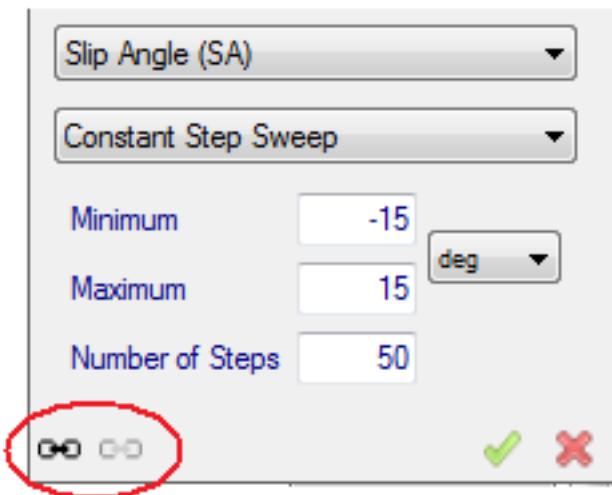


Figure 4.13: Linking Graphs

4.1.8 Coloring Graphs

The color of the graphs is based on the color assigned to the project item. This can be changed by right-clicking on the item in the project tree and selecting *color*. Then a window will open where the user can choose the base color to be used. This will change the color of the item in all of the graphs. The color of all of the items in one tire can also be changed. This is done by right clicking on the tire itself and selecting *color*. The color in the bottom right corner of the item icon displays the current base color of that item.

The base color of an item can also be changed in each graph separately. This can be done in the *Project Items Setup* at the bottom of the graph setup form. By right clicking on an item in this list the user is given the option to *Set Custom Color or Return to Default Color*. Selecting *Set Custom Color* will allow the user to change the base color of this item in only the current graph. Selecting *Return to Default Color* will change the color to the color of the same item in the project tree.

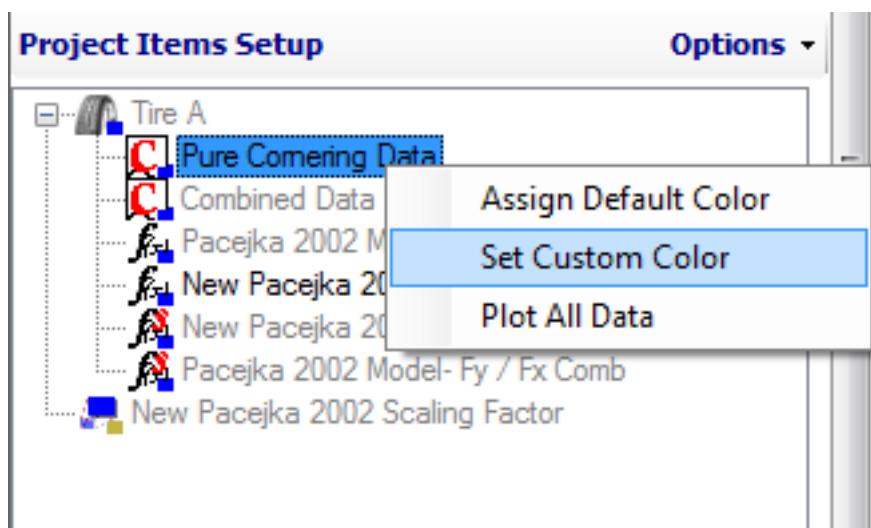


Figure 4.14: Plot All Data Option

If you choose, you can also alter the color of the graph based on two of the graph inputs. This will modify the base color depending on the first two input sweeps of the graph. The color of the data sweeps can be altered in the bottom of the graph input form as seen in Figure 4.15. The color is altered according to the color quantity selected. These include *Constant Color*, *Hue*, *Luminosity*, or *Saturation*.

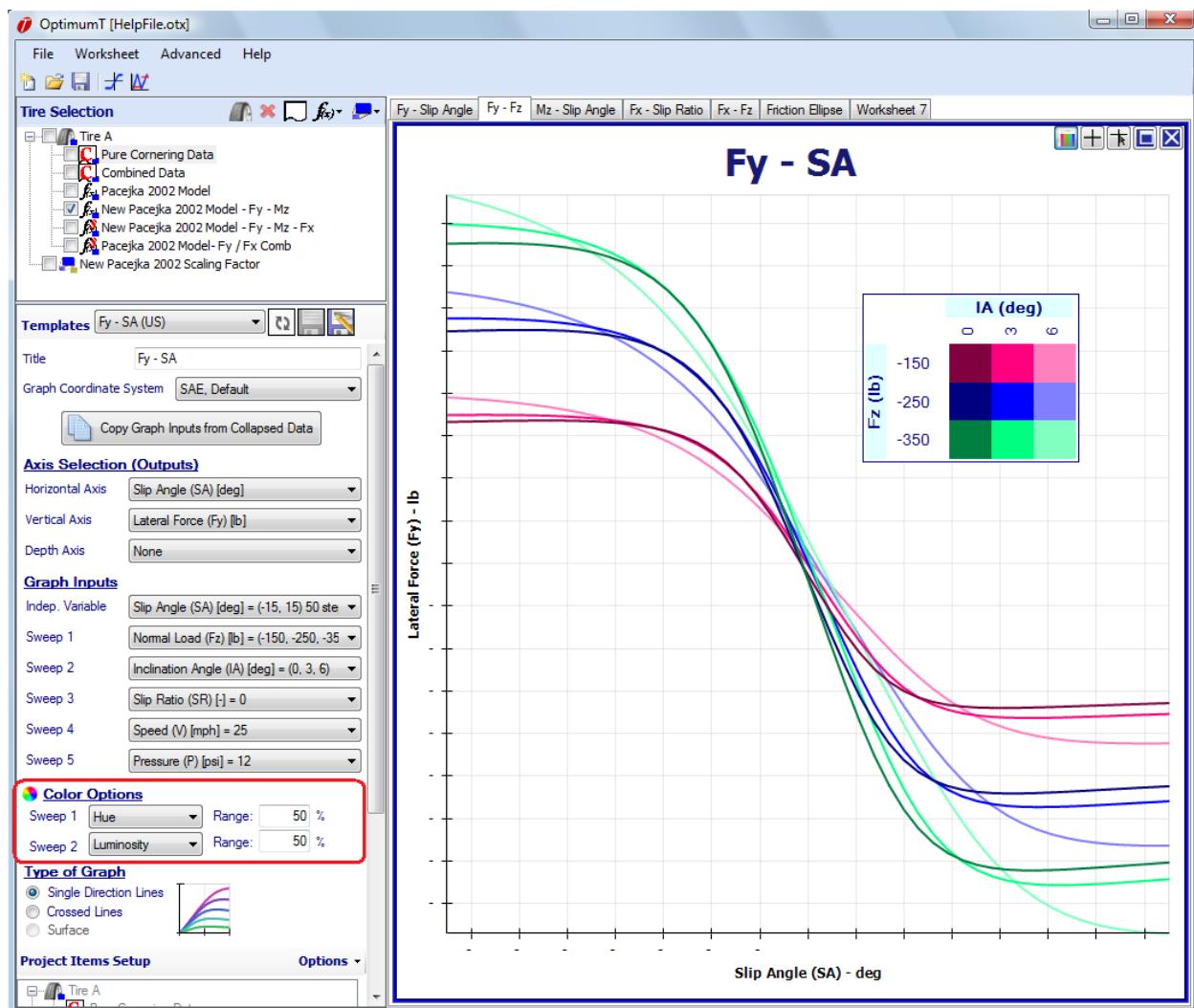


Figure 4.15: Color Setup

Hue will change the actual color of the lines according to the input sweeps as shown in Figure 4.16. Luminosity will affect the visibility of the lines as shown in Figure 4.17 and Saturation will affect the brightness of the lines as shown in Figure 4.18. In Figure 4.15 hue is varied with the vertical load and the luminosity is varied with the inclination angle. The range for these quantities in percentage can be entered in the textboxes to the right. A higher range will create a larger difference in color between the lines. Custom coloring is not currently available for the surface graphs.

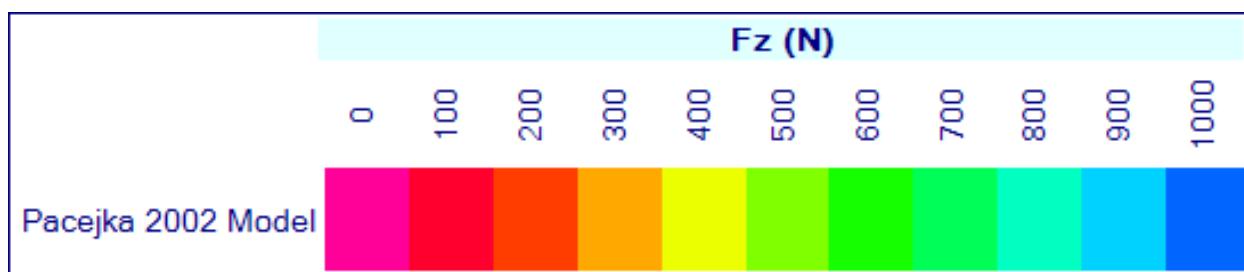


Figure 4.16: Hue

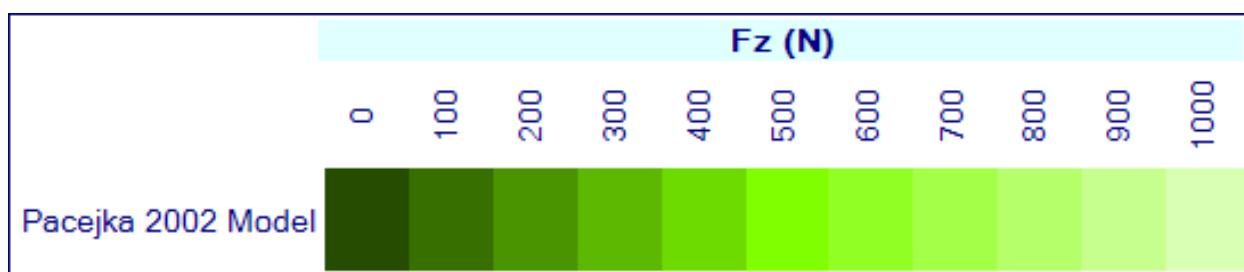


Figure 4.17: Luminosity



Figure 4.18: Saturation

4.1.9 Graph Templates

Graph templates allow the user to quickly display commonly used graphs. They can be accessed at the top of the graph setup form as shown in Figure 4.19. Graph templates in the predefined folder are supplied with OptimumT. These templates cannot directly be modified but can be changed and saved as new templates. They also can be copied into the user folder in the template manager.

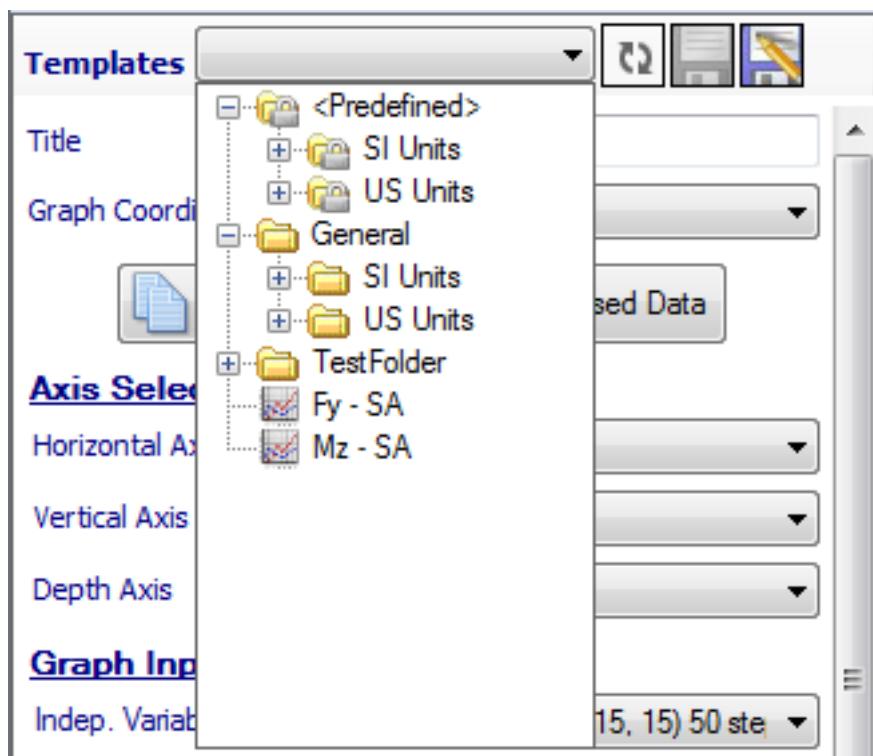


Figure 4.19: Graph Templates

To load a graph template, select it from the list and click the Load Graph Template button (which will be spinning after you choose a template). To save a new template, click on the save-as button next to the graph template list. Clicking the save button when a graph template is selected will overwrite the selected graph template.

4.2 Report Graphs

Report graphs display data versus sample number (which is proportional to time). Only raw tire data can be shown on report graphs. An example of a report graph is shown in Figure 4.20. This is the same type of graph that is used in the data cropping tool. When this graph is selected a report graph setup form will appear in the data entry area. At the top of the form a title can be added to the graph and the coordinate system that the data will be displayed in can be changed. The quantities to be graphed (i.e slip angle) can be selected from the checkboxes in this form. Multiple quantities can be graphed at the same time. At the bottom of the form the user can set the axis parameters including the minimum and maximum range (or Auto Scale) and the display units.

By clicking on the graph, the data cursor (vertical black line) can be moved. The data values to the right of the graph quantities in the setup form represent the values of the data at the

data cursor location. The graph can be zoomed in and out with the middle mouse button. While holding down the middle mouse button, draw a box from the top left to the bottom right corner of the area you want to zoom in on. Drawing a box in the opposite direction will un-zoom the graph. Right clicking on the mouse gives the user options to copy or print the graph as well add additional graphs or worksheets to the project.

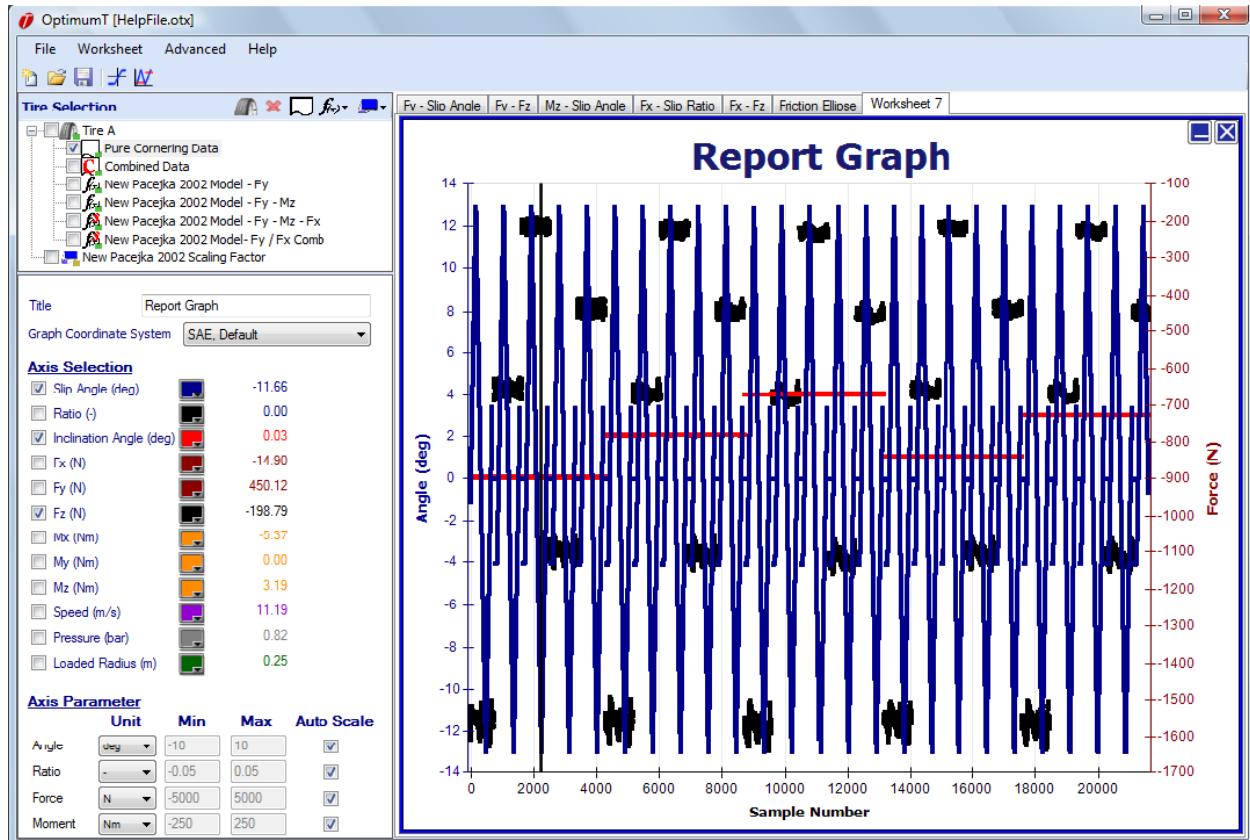


Figure 4.20: Report Graph

4.3 Graph Area Options

In the upper right corner of each standard graph there are buttons that allow the user to access additional graphing options. These buttons are shown in Figure 4.21. The two buttons on the far right are similar to typical windows buttons. The one on the far right will permanently close the graph and the other one will minimize or maximize the graph depending on its current state. The other three buttons are custom options in OptimumT. These will first be discussed in detail in this section and then other graph options that can be accessed by right clicking on the graph will be covered.

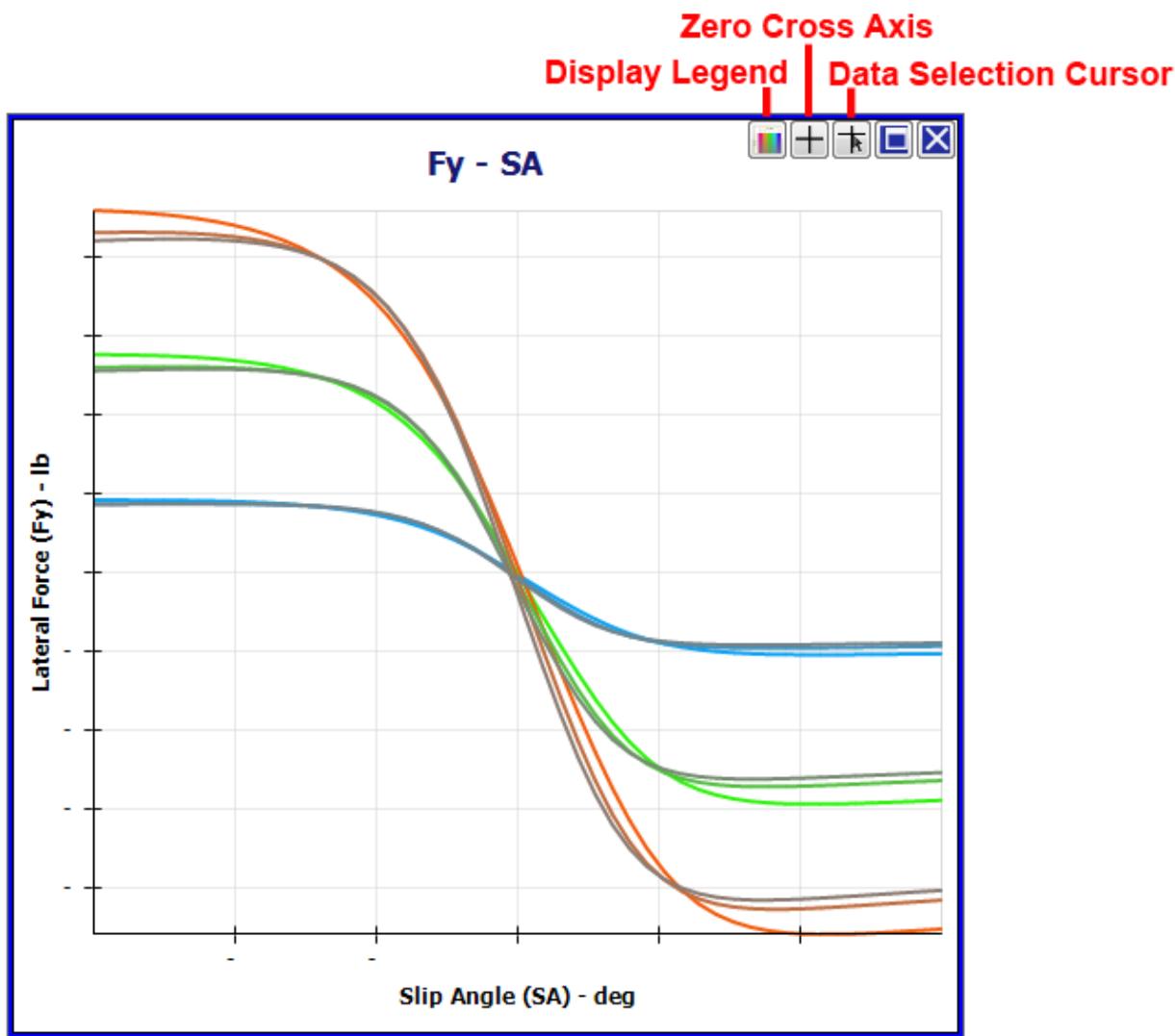


Figure 4.21: Graph Area Options

4.3.1 Display Legend

The *Display Legend* button on the far left allows the user to show the graph legend. Clicking on this button will display the legend as shown below in Figure 4.22. The legend can be moved by holding down the mouse button on it and dragging it to the desired location. It can also be resized by holding the mouse button down on the corners of the legend and dragging it to the desired size.

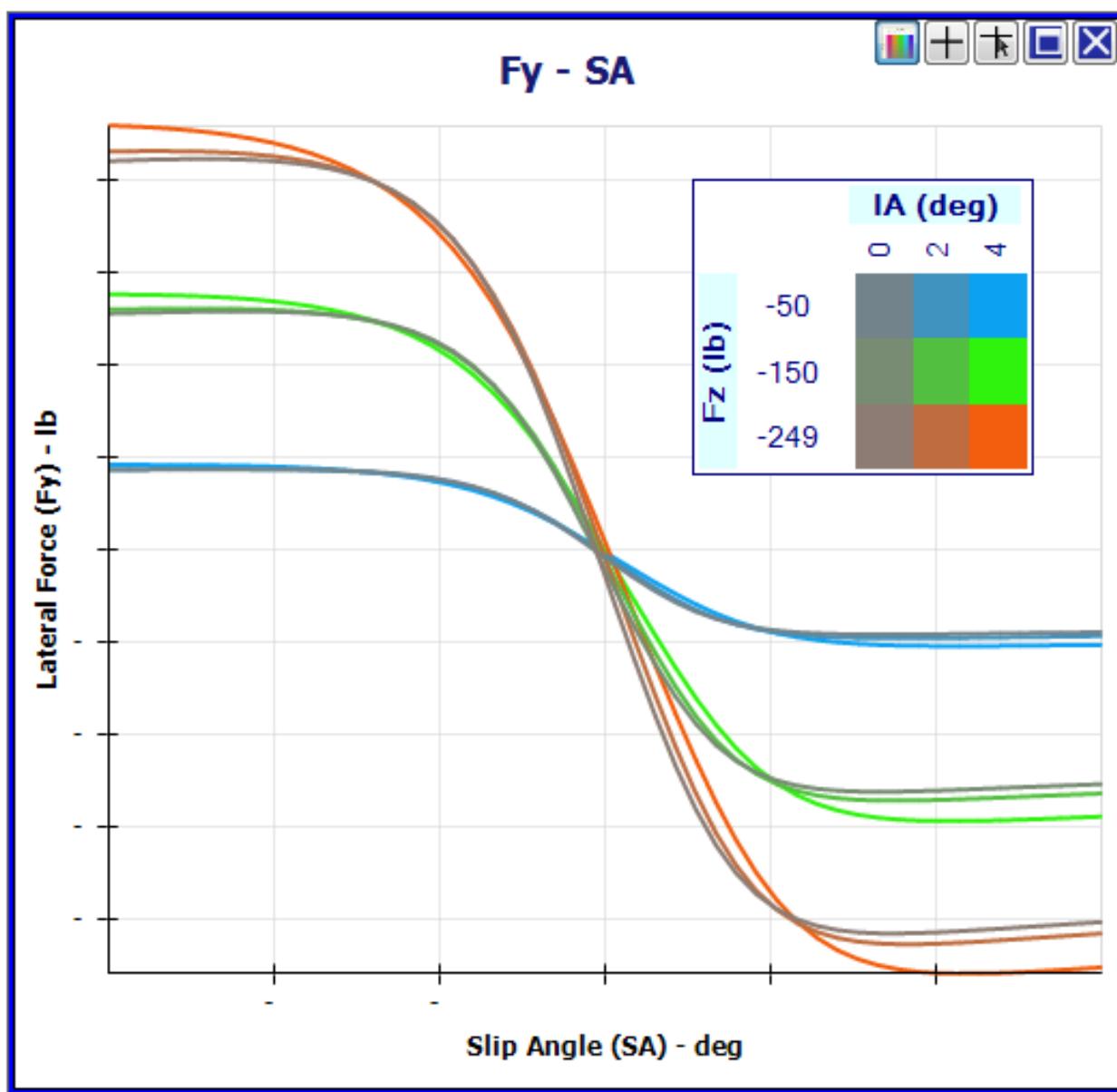


Figure 4.22: Graph Legend for two Data Sweeps

In Figure 4.22 the legend displays the values and colors corresponding to the inputs in Sweep 1 and Sweep 2 in the graph setup form. Since both inputs were set as sweeps both are displayed in the legend.

If neither or only one of the first two graph inputs include multiple values, the legend will display the names of the items in the project tree that are being graphed. This is shown below in Figure 4.23.

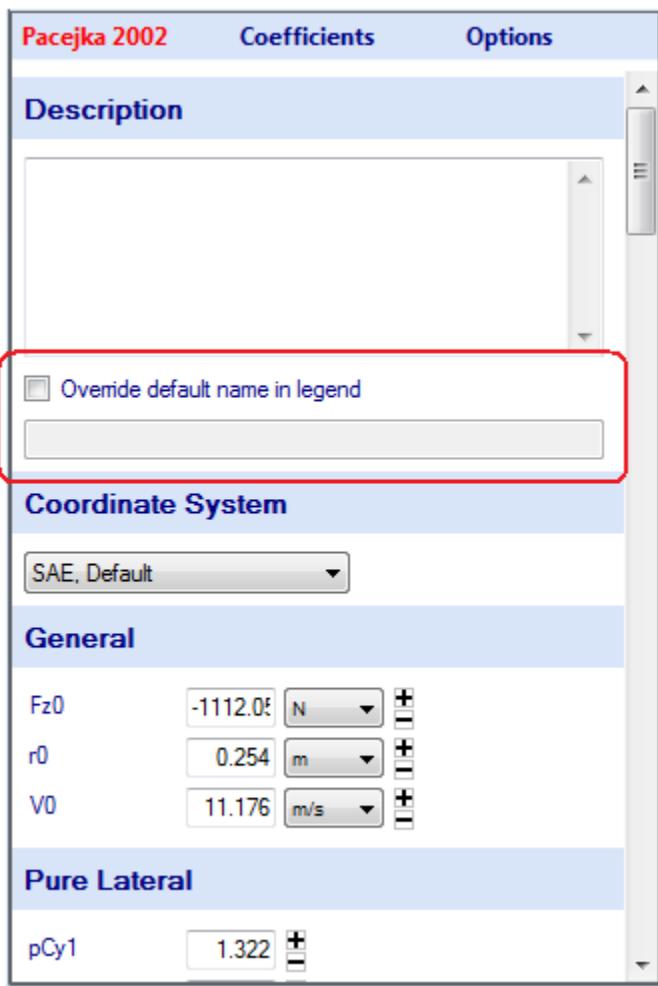


Figure 4.23: Graph Legend for one Data Sweep

The names displayed in the legend can also be modified in the data entry area corresponding to the selected Item. Therefore to change the legend name without renaming the item, click on the item in the project tree. Then the form shown in Figure 4.23 will appear in the data entry area. Selecting or deselecting the checkbox labeled *Override the default name in legend* will change the name of the item shown in the legend. When it is checked whatever text is entered into the textbox below the checkbox will appear in the legend instead of the name of the item.

4.3.2 Axis Zero Cross

The *Axis Zero Cross* button is the second button from the left. By default the graph axis will be displayed at the borders of the graph. When this button is selected the axis will be displayed on the gridlines corresponding to the zero value of the graph outputs. This is

shown in Figure 4.24.

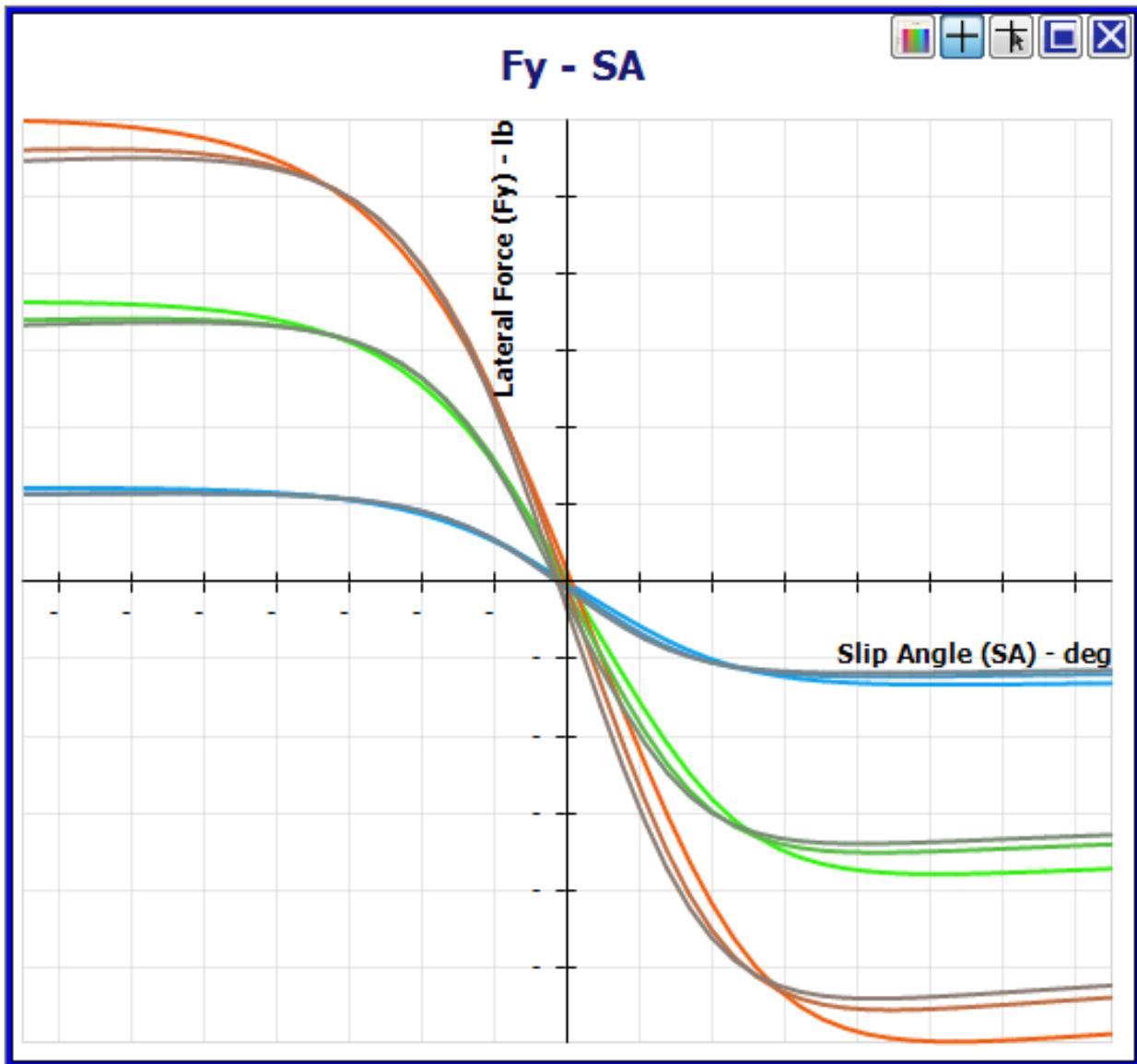


Figure 4.24: Zero Cross Axis

4.3.3 Data Selection Cursor

The *Data Selection Cursor* allows the user to easily and quickly get information about a specific data point or line. An example of this is shown in Figure 4.25. After clicking on the *Data Selection Cursor* button, click on either a data line or point in the graph to see the test conditions and values corresponding to that specific location.

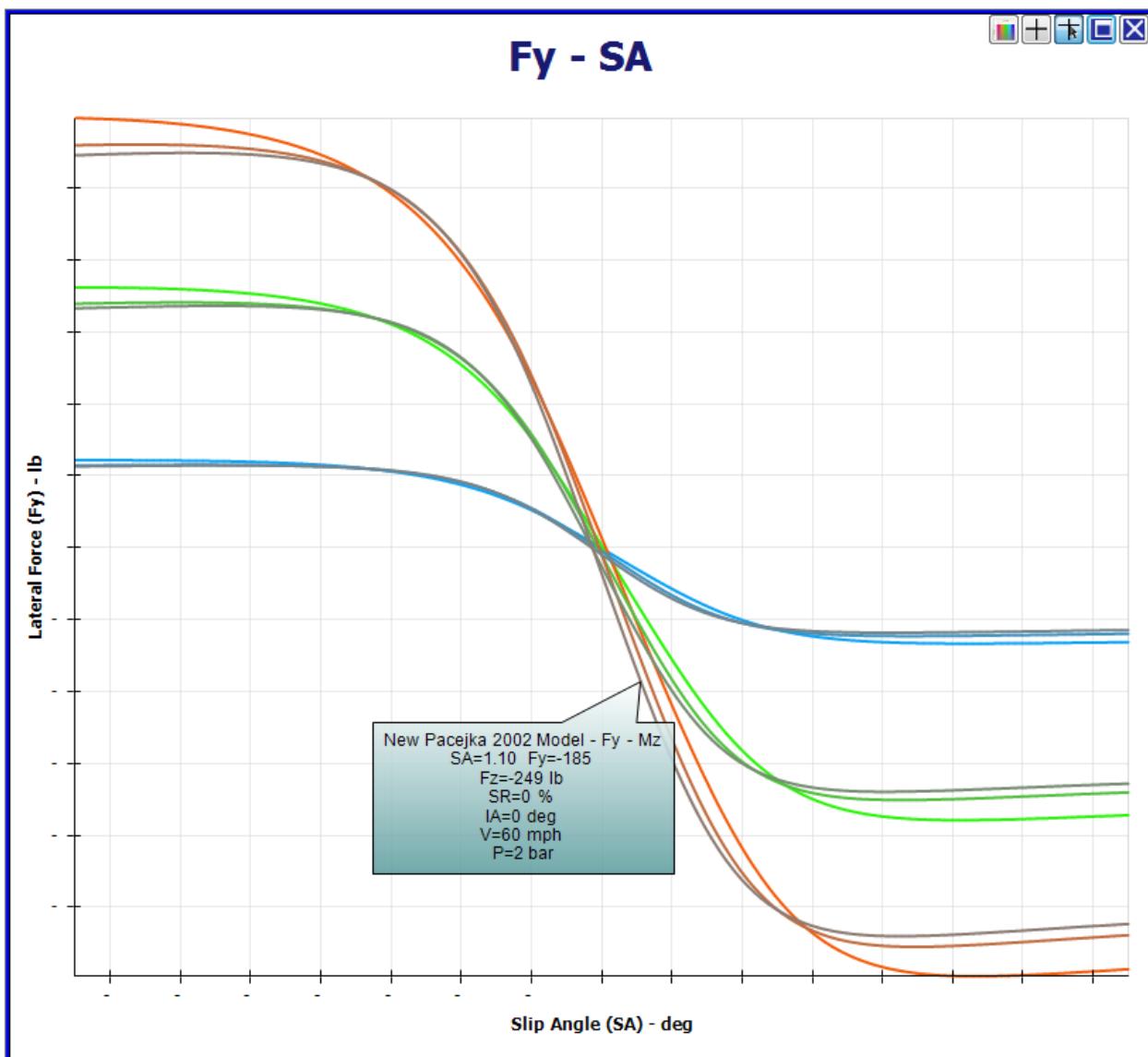


Figure 4.25: Data Cursor Selection

4.3.4 Copy and Print Graphs

Graphs can be easily copied or printed from OptimumT. These options are available through right clicking on either a standard or report graph as shown in Figure 4.26. If *Copy graph to Clipboard* is selected than an image of the current graph will be copied to the windows clipboard. The user can then paste this image into other programs. If *Print Graph* is selected the standard printing options window will open allowing the user to select the printer and set the printing preferences. When printed the graph will be automatically resized and centered to fit on a standard 8.5" x 11" piece of paper.

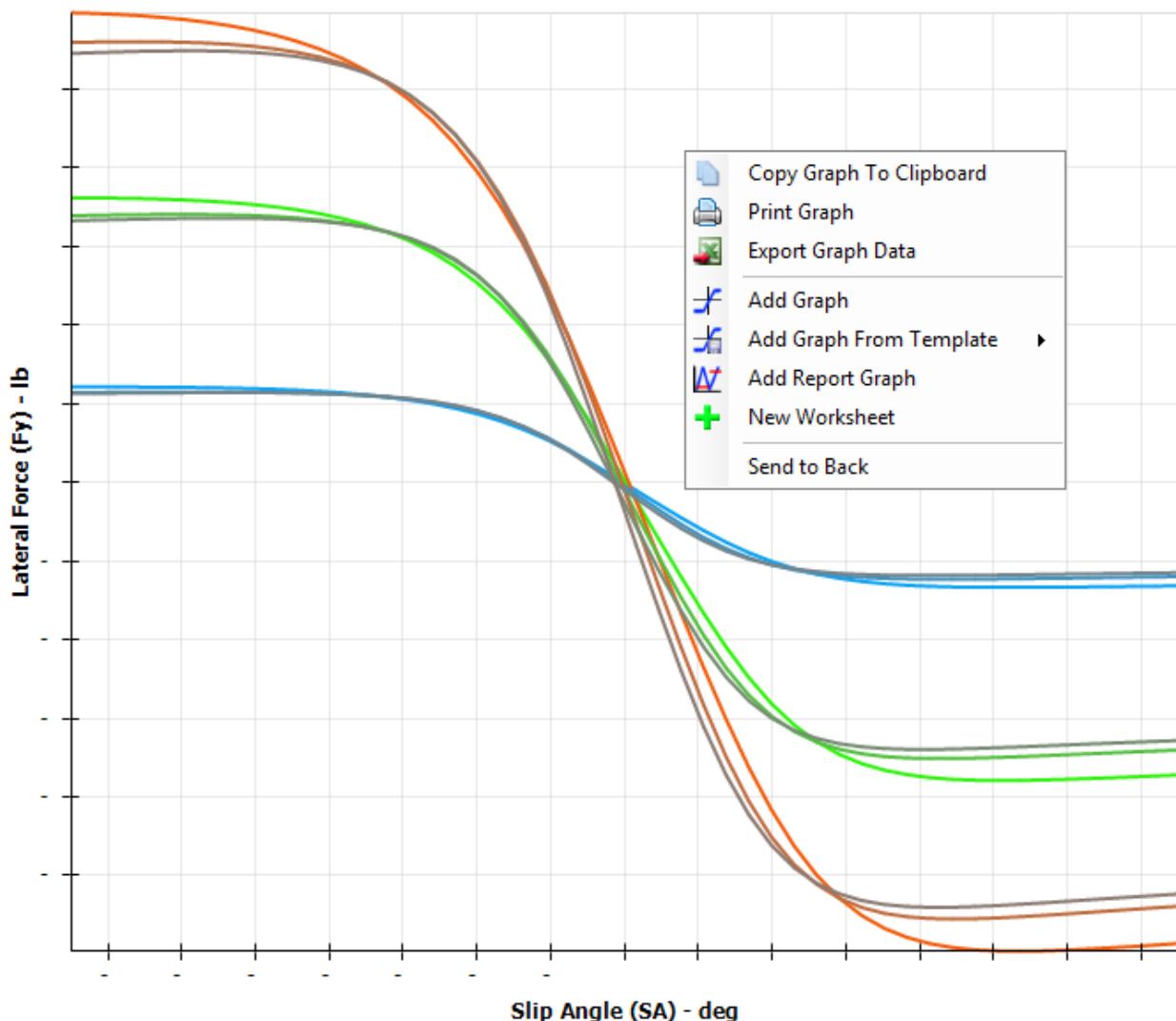


Figure 4.26: Right Click Options

4.3.5 Export Graph Data

Another feature shown in Figure 4.26 is the Export Graph Data. When this is selected the data that is currently plotted will be exported and saved to an Excel file. The different data sweeps will be separated by different headers indicating the conditions that they represent. An example of this is shown in Figure 4.27.

	A	B	C	D	E	F	G	H	I
1	Pacejka 2002 Model- Fy / Fx Comb								
2									
3	9/29/2009								
4									
5	Model Type								
6	Pacejka 2002								
7									
8	Description								
9									
10	Coordinate System								
11	SAE, Default								
12									
13	General								
14	Fz0	-1112.06							
15	r0	0.254							
16	V0	11.176							
17									
18	Pure Lateral								
19	pCy1	0.983539							
20	pDy1	-2.25							
21	pDy2	0.182111							
22	pDy3	-12.3175							
23	pEy1	-0.75							
24	pEy2	-0.324							
25	pEy3	-0.635							
26	pEy4	-16.652							
27	pKy1	43.015							
28	pKy2	2.014907							
29	pKy3	36.03797							

Figure 4.27: Excel Export Format

Chapter 5

Custom Models

5.1 Introduction to Custom Models

Custom models are part of the advanced features available in OptimumT. This functionality gives users the flexibility to include user developed custom models within the OptimumT environment. With the custom models feature users can write their own models in house, therefore keeping proprietary models confidential and providing the ability to tailor models to the exact specifications required. Custom models need only to be coded once, they can then be used repeatedly and as efficiently as any of the hard coded models available within OptimumT. The custom model package is an advanced feature which requires an additional activation license available by contacting engineering@optimumg.com

5.2 Creating Custom Models

Custom models are based on a template written in C# this template is provided by OptimumG with the purchase of the custom models package. To create a new model the user must first define a .dll (dynamically linked library) file which can then be loaded into OptimumT. This file references the template provided by OptimumG so it must follow a certain syntax that makes it compatible with OptimumT. This syntax is outlined in Sections 5.2.4 and 5.2.5. The user .dll contains a list of coefficients and equations that define the user's custom model.

An example project is included with OptimumT to provide an example that users can copy and edit when developing custom models. The project is installed in My Documents on the users machine and can be opened with Visual C#.

5.2.1 Software Requirements

The custom model template is written in Visual C# a CLI (Common Language Infrastructure) language. In order to create a custom model the user must first download a version of Visual C# by Microsoft. This software is available as a free *Express Edition* for the general user or *Visual Studio Professional Edition* for professional software developers. For the purposes of creating OptimumT custom models the *Visual C# Express Edition* is sufficient. The download is available at the following site www.microsoft.com/express/Downloads

OptimumT requires two files a *Coefficient File* and a *Calculation File* to define a model. The following sections outline how to create these files.

5.2.2 Creating the Custom Model Project

The first step in creating custom model is to create a new C# project with the coefficient and calculation classes. To do this open Visual Studio or Visual C#. Create a new model from the file menu, select *C# Class Library* and enter the name of your model as shown.

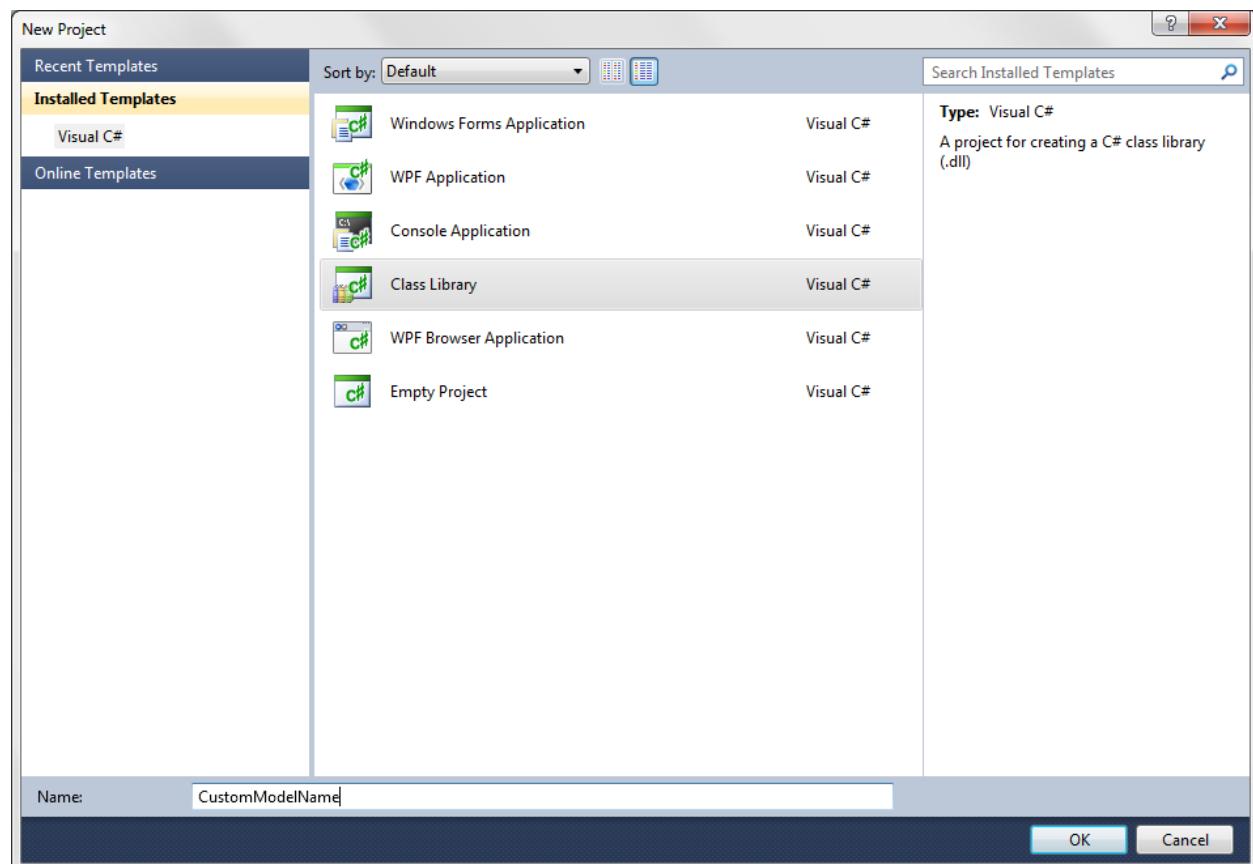


Figure 5.1: Creating a New Project

Alternatively you can open the example project provided with OptimumT, modify it and save it under a different name.

Next right click on the project just created and select add->new item to add a class to the project.

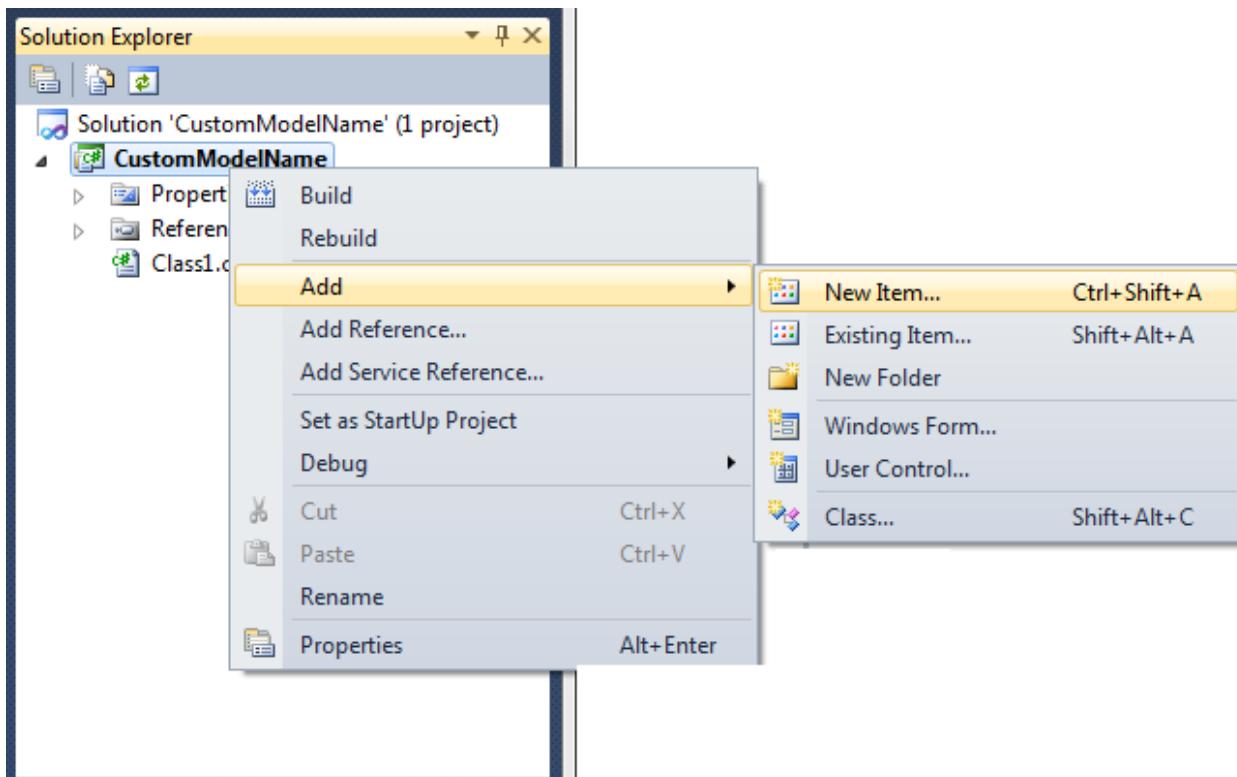


Figure 5.2: Creating a New Class

Add two classes, the coefficient class and the calculation class. Name these classes *CustomTireModelCoef* and *CustomTireModelCalculation*. Save the project and then continue with the following steps.

5.2.3 Referencing the Custom Model Template

Once the project has been created the next step is to add the reference for the custom model template. Without this reference the custom model classes defined by the user cannot inherit from the template. To add a reference to a project first right click on the C# project and select add reference from the dropdown menu as shown in Figure 5.3.

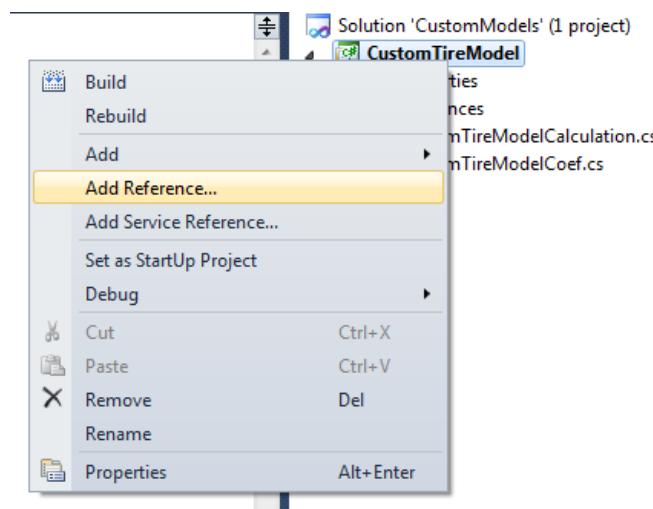


Figure 5.3: Creating a Reference

Select the browse tab in the add reference dialog and browse to the OptimumT installation folder. Select the CustomTireModelTemplate.dll and select OK as shown in Figure 5.4. The reference to the template.dll should now appear in the project references folder in the Visual Studio Solution Explorer.

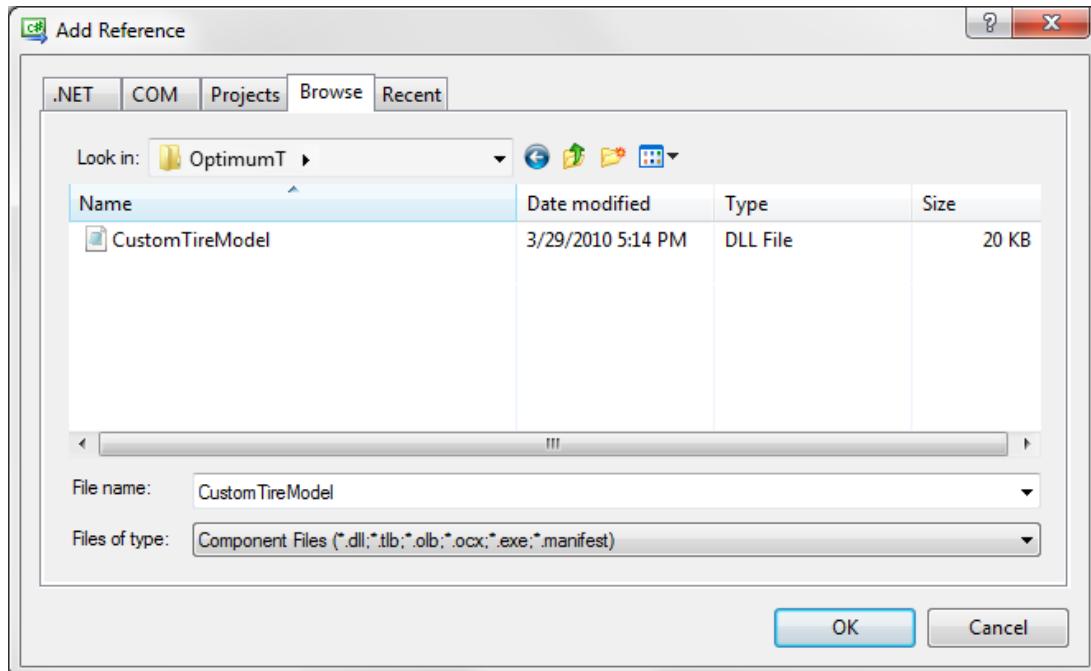
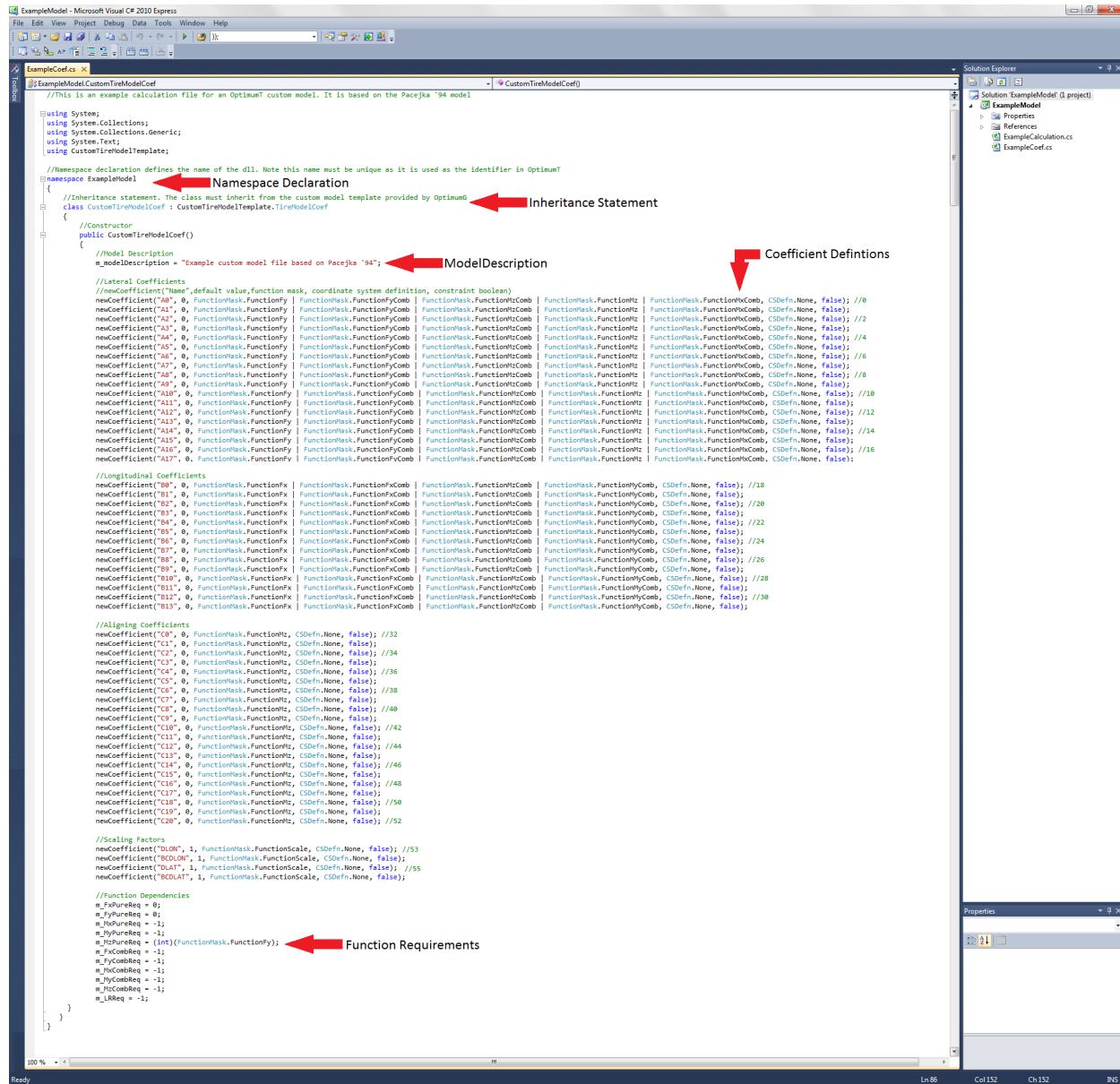


Figure 5.4: Selecting the Reference .dll

5.2.4 Creating a Coefficient File

An example coefficient file has been included with the OptimumT custom package. It is shown in Figure 5.5. It has a number of features.



```

ExampleModel - Microsoft Visual Studio 2010 Express
File Edit View Project Debug Data Tools Window Help
ExampleCoef.cs - CustomTireModelCoef.cs
//This is an example calculation file for an OptimumT custom model. It is based on the Pacejka '94 model
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using CustomTireModelTemplate;

<_namespace> namespace ExampleModel
{
    //Inheritance statement. The class must inherit from the custom model template provided by OptimumT
    class CustomTireModelCoef : CustomTireModelTemplate.TireModelCoef
    {
        //Constructor
        public CustomTireModelCoef()
        {
            //Model Description
            m_modelDescription = "Example custom model file based on Pacejka '94";
        }

        //Lateral Coefficients
        //newCoefficient("Name", default value, function mask, coordinate system definition, constraint boolean)
        newCoefficient("A0", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //0
        newCoefficient("A1", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //1
        newCoefficient("A2", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //2
        newCoefficient("A3", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //3
        newCoefficient("A4", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //4
        newCoefficient("A5", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //5
        newCoefficient("A6", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //6
        newCoefficient("A7", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //7
        newCoefficient("A8", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //8
        newCoefficient("A9", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //9
        newCoefficient("A10", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //10
        newCoefficient("A11", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //11
        newCoefficient("A12", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //12
        newCoefficient("A13", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //13
        newCoefficient("A14", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //14
        newCoefficient("A15", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //15
        newCoefficient("A16", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //16
        newCoefficient("A17", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionMzComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //17

        //Longitudinal Coefficients
        newCoefficient("B0", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //18
        newCoefficient("B1", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //19
        newCoefficient("B2", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //20
        newCoefficient("B3", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //21
        newCoefficient("B4", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //22
        newCoefficient("B5", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //23
        newCoefficient("B6", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //24
        newCoefficient("B7", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //25
        newCoefficient("B8", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //26
        newCoefficient("B9", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //27
        newCoefficient("B10", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //28
        newCoefficient("B11", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //29
        newCoefficient("B12", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //30
        newCoefficient("B13", 0, FunctionMask.FunctionFx | FunctionMask.FunctionVfComb | FunctionMask.FunctionVfComb, CSDefn.None, false); //31

        //Aligning Coefficients
        newCoefficient("C0", 0, FunctionMask.FunctionMz, CSDefn.None, false); //32
        newCoefficient("C1", 0, FunctionMask.FunctionMz, CSDefn.None, false); //33
        newCoefficient("C2", 0, FunctionMask.FunctionMz, CSDefn.None, false); //34
        newCoefficient("C3", 0, FunctionMask.FunctionMz, CSDefn.None, false); //35
        newCoefficient("C4", 0, FunctionMask.FunctionMz, CSDefn.None, false); //36
        newCoefficient("C5", 0, FunctionMask.FunctionMz, CSDefn.None, false); //37
        newCoefficient("C6", 0, FunctionMask.FunctionMz, CSDefn.None, false); //38
        newCoefficient("C7", 0, FunctionMask.FunctionMz, CSDefn.None, false); //39
        newCoefficient("C8", 0, FunctionMask.FunctionMz, CSDefn.None, false); //40
        newCoefficient("C9", 0, FunctionMask.FunctionMz, CSDefn.None, false); //41
        newCoefficient("C10", 0, FunctionMask.FunctionMz, CSDefn.None, false); //42
        newCoefficient("C11", 0, FunctionMask.FunctionMz, CSDefn.None, false); //43
        newCoefficient("C12", 0, FunctionMask.FunctionMz, CSDefn.None, false); //44
        newCoefficient("C13", 0, FunctionMask.FunctionMz, CSDefn.None, false); //45
        newCoefficient("C14", 0, FunctionMask.FunctionMz, CSDefn.None, false); //46
        newCoefficient("C15", 0, FunctionMask.FunctionMz, CSDefn.None, false); //47
        newCoefficient("C16", 0, FunctionMask.FunctionMz, CSDefn.None, false); //48
        newCoefficient("C17", 0, FunctionMask.FunctionMz, CSDefn.None, false); //49
        newCoefficient("C18", 0, FunctionMask.FunctionMz, CSDefn.None, false); //50
        newCoefficient("C19", 0, FunctionMask.FunctionMz, CSDefn.None, false); //51
        newCoefficient("C20", 0, FunctionMask.FunctionMz, CSDefn.None, false); //52

        //Scaling Factors
        newCoefficient("BL0W", 1, FunctionMask.FunctionScale, CSDefn.None, false); //53
        newCoefficient("BL0LN", 1, FunctionMask.FunctionScale, CSDefn.None, false); //54
        newCoefficient("BLAT", 1, FunctionMask.FunctionScale, CSDefn.None, false); //55
        newCoefficient("BLATL", 1, FunctionMask.FunctionScale, CSDefn.None, false); //56

        //Function Dependencies
        m_fxPureReq = 0;
        m_pyPureReq = 0;
        m_pxReq = -1;
        m_pyReq = -1;
        m_mxReq = -1;
        m_mxCombReq = -1;
        m_myReq = -1;
        m_myCombReq = -1;
        m_zReq = -1;
        m_zCombReq = -1;
    }
}

//Function Requirements
m_fxPureReq = (Int)(FunctionMask.FunctionFx);

```

Figure 5.5: Visual C# Coefficient File

The first section is the "*Namespace Declaration*" this is the name of the custom .dll file and will be used as an identifier within OptimumT.

The second statement is the "*Inheritance Statement*". This is required to ensure all the

template functions are available to the custom .dll. Without this statement the .dll will produce an error when loaded into OptimumT. The coefficient file must inherit from the coefficient template CustomTireModelTemplate.TireModelCoef.

The third statement is the "*Model Description*" which will appear in the description box on the model coefficient form sheet.

The fourth section contains the "*Coefficient Definitions*" themselves. It is a list of the coefficients to be added to the array list. These statements are in the format shown at the start of this section.

The fifth section is the "*Function Requirements*". These statements allow the user to change the function dependencies in the model. For example the aligning moment function often requires the lateral force to be calculated first. Therefore the function requirement for CalculateMz will be FunctionMask.FunctionFy. The function requirements statements are in the following syntax.

```
m_MzPureReq = (int)(FunctionMask.FunctionFy);
```

setting a function requirement equal to -1 will make that function unavailable. Setting the requirement equal to 0 indicates that the function does not have any dependencies.

The coefficient file is simply an arraylist of coefficients. Each coefficient is defined in the following format:

```
newCoefficient(("Name", Default Value, Function Mask, Coordinate System Definition,  
Constraint Boolean)
```

- **Name:** The name of the variable that will appear beside each variable in OptimumT. Must be entered as a string enclosed in quotation marks ""
- **Default Value:** The default value of the coefficient that will appear when creating empty models, boundary files and constraints. This value is a single floating point number.
- **Function Mask:** Tells OptimumT which outputs are related to each coefficient so the solver knows which coefficients to fit when fitting a specific model. OptimumT has a list of function masks to choose from:

```
FunctionMask.FunctionFx
FunctionMask.FunctionFy
FunctionMask.FunctionFz
FunctionMask.FunctionMx
FunctionMask.FunctionMy
FunctionMask.FunctionMz
FunctionMask.FunctionRl
FunctionMask.FunctionScale
```

Each coefficient must include one of these definition or a combination of the above separated by the or " | "operator.

For example a coefficient "*U*" is used in the equations for calculating the lateral force, longitudinal force and aligning torque its coefficient mask is therefore:

FunctionMask.FunctionFx | FunctionMask.FunctionFy | FunctionMask.FunctionMz

The function mask FunctionScale is used to specify that the coefficient is used as a scaling factor(see Section 3.3.2). This means the coefficient will not be fit in the fitting process and will remain the default value specified. The default value should be 1.0 in most cases for the scaling factors to have no effect on the model.

- **Coordinate System Definition:** The coordinate properties of the coefficient that determine which coordinate system conversion will cause the coefficient to change sign. Each coordinate system definition is in the form CSDfn.X which means, that if the sign of the input X changes in a coordinate system conversion the coefficients with the corresponding coordinate system definition will also change sign. OptimumT provides the following coordinate system definitions:

```

CSDefn.V
CSDefn.SA
CSDefn.SR
CSDefn.IA
CSDefn.F _x
CSDefn.F _y
CSDefn.F _z
CSDefn.M _x
CSDefn.M _y
CSDefn.M _z
CSDefn.None

```

Note: Leaving the coordinate system definition as *CSDefn.None* means the coefficient will not be affected by any coordinate system conversions within OptimumT so the coefficients will always remain in the initially selected system.

OptimumT supports four coordinate systems these are the SAE, Adapted SAE, ISO and Adapted ISO coordinate systems. For definitions of these systems see Section 8.2. These coordinate systems will apply the following coordinate system changes when converting from the default SAE coordinate system:

SAE: Default

Adapted SAE: CSDefn.SA | CSDefn.F_z

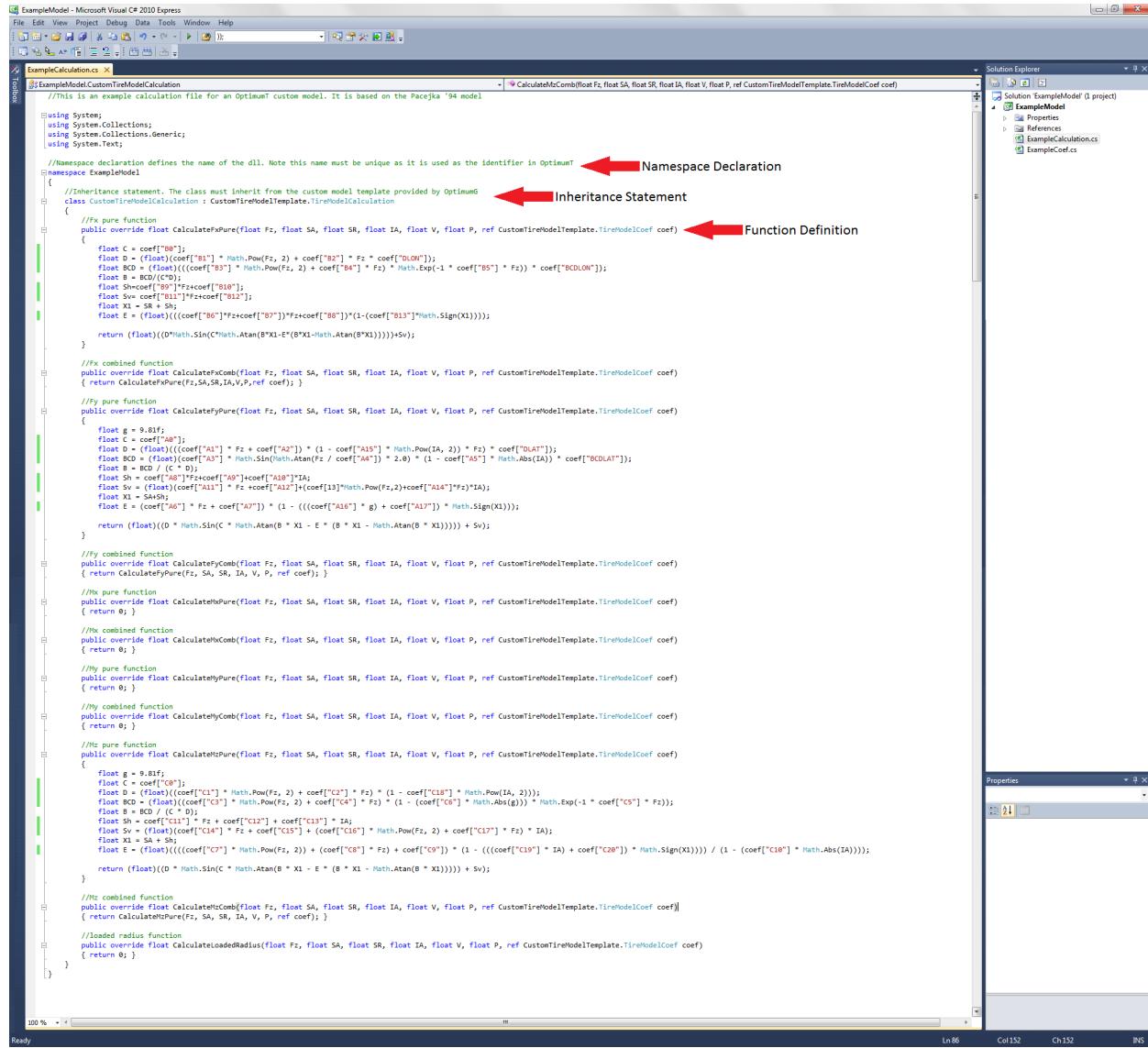
ISO: CSDefn.SA | CSDefn.F_y | CSDefn.F_z | CSDefn.M_y | CSDefn.M_z

Adapted ISo: CSDefn.IA | CSDefn.F_y | CSDefn.F_z | CSDefn.M_y | CSDefn.M_z

- **Constraint Boolean:** Defines whether a coefficient will be a fixed constraint or not. True if coefficient of fixed or false if it is a coefficient is to be fit.

5.2.5 Creating a Calculation File

The calculation file contains the equations that define the tire model. An example of how to create a calculation file is included with the OptimumT custom model package. The example is shown in Figure 5.6.



```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;

//This is an example calculation file for an OptimumT custom model. It is based on the Pacejka '94 model
namespace ExampleModel
{
    //Inheritance statement. The class must inherit from the custom model template provided by OptimumT
    class CustomTireModelCalculation : CustomTireModelTemplate.TireModelCalculation
    {
        //Fx pure function
        public override float CalculateFxPure(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            float C = coef["A0"];
            float D = (float)((coef["B1"] * Math.Pow(Fz, 2) + coef["B2"] * Fz + coef["D10"] * Math.Pow(Fz, 2) * coef["B3"] * Math.Sin(Math.Atan(Fz / coef["A4"])) + 2.0) * (1 - coef["B5"] * Fz) * Math.Exp(-1 * coef["B6"] * Fz)) * coef["BCD10"];
            float BCD = (float)((coef["B3"] * Math.Pow(Fz, 2) + coef["B4"] * Fz) * Math.Abs(IA));
            float S = (float)(C - D);
            float ShrcF = (float)(coef["B1"] * Fz + coef["B2"]);
            float Sv = (float)(coef["B13"] * Fz + coef["B12"]);
            float E = (float)((coef["B6"] * Fz + coef["B7"]) * Fz + coef["B8"]) * (1 - (coef["B13"] * Math.Sign(X1)));
            return (float)((D * Math.Sin(C * Math.Atan(B * X1 - E * (B * X1 - Math.Atan(B * X1)))) + Sv));
        }

        //Fx combined function
        public override float CalculateFxComb(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return CalculateFxPure(Fz, SA, SR, IA, V, P, ref coef);
        }

        //Fy pure function
        public override float CalculateFyPure(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            float g = 0.81f;
            float C = coef["A0"];
            float D = (float)((coef["A1"] * Fz + coef["A2"]) * (1 - coef["A15"] * Math.Pow(IA, 2)) * Fz * coef["D10AT"]);
            float BCD = (float)((coef["A3"] * Math.Sin(Math.Atan(Fz / coef["A4"])) + 2.0) * (1 - coef["A5"] * Math.Abs(IA)) * coef["BCD10AT"]);
            float S = (float)(C - D);
            float Sh = (float)(coef["A8"] * Fz + coef["A9"] * coef["A10"] * IA);
            float Sv = (float)(coef["A11"] * Fz + coef["A12"] * (coef[13] * Math.Pow(Fz, 2) * coef["A14"] * Fz) * IA);
            float X1 = (float)(coef["A6"] * Fz + coef["A7"]) * (1 - (((coef["A15"] * g) + coef["A17"]) * Math.Sign(X1)));
            return (float)((D * Math.Sin(C * Math.Atan(B * X1 - E * (B * X1 - Math.Atan(B * X1)))) + Sv));
        }

        //Fy combined function
        public override float CalculateFyComb(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return CalculateFyPure(Fz, SA, SR, IA, V, P, ref coef);
        }

        //Mx pure function
        public override float CalculateMxPure(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return 0;
        }

        //Mx combined function
        public override float CalculateMxComb(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return 0;
        }

        //My pure function
        public override float CalculateMyPure(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return 0;
        }

        //My combined function
        public override float CalculateMyComb(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return 0;
        }

        //Nx pure function
        public override float CalculateNxPure(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            float g = 0.81f;
            float C = coef["A0"];
            float D = (float)((coef["C1"] * Math.Pow(Fz, 2) + coef["C2"] * Fz) * (1 - coef["C18"] * Math.Pow(IA, 2)));
            float BCD = (float)((coef["C3"] * Math.Pow(Fz, 2) + coef["C4"] * Fz) * (1 - (coef["C6"] * Math.Abs(g)) * Math.Exp(-1 * coef["C5"] * Fz)));
            float B = BCD / ((C * D));
            float Sh = (float)(coef["C11"] * Fz + coef["C12"] * coef["C13"] * IA);
            float Sv = (float)(coef["C14"] * Fz + coef["C15"] * (coef["C16"] * Math.Pow(Fz, 2) + coef["C17"] * Fz) * IA);
            float X1 = (float)(SA + SR);
            float E = (float)((coef["C7"] * Math.Pow(Fz, 2)) + (coef["C8"] * Fz) + coef["C9"]) * (1 - (((coef["C19"] * IA) + coef["C20"]) * Math.Sign(X1))) / (1 - (coef["C10"] * Math.Abs(IA)));
            return (float)((D * Math.Sin(C * Math.Atan(B * X1 - E * (B * X1 - Math.Atan(B * X1)))) + Sv));
        }

        //Mz combined function
        public override float CalculateMzComb(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return CalculateNxPure(Fz, SA, SR, IA, V, P, ref coef);
        }

        //loaded radius function
        public override float CalculateLoadedRadius(float Fz, float SA, float SR, float IA, float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
        {
            return 0;
        }
    }
}

```

Figure 5.6: Visual C# Calculation File

The file contains several statements. The first is the "Namespace Declaration" this is the name of the .dll and will be used by OptimumT as an identifier for the custom model. It must be the same as the "Namespace Declaration" in the coefficient file.

The second statement is the "Inheritance Statement". This is required to ensure all the

template functions are available to the custom .dll. Without this statement the .dll will produce an error when loaded into OptimumT. The coefficient file must inherit from the calculation template CustomTireModelTemplate.TireModelCalculation.

The third section contains the function definitions themselves. The primary functions must be declared in the following format to override the functions in the template file.

```
public override float CalculateXXXX(float Fz, float SA, float SR, float IA,
float V, float P, ref CustomTireModelTemplate.TireModelCoef coef)
```

The custom calculation template contains the following overridable functions that can be redefined by the user.

```
CalculateFxPure(Fz, SA, SR, IA, V, P, ref coef)
CalculateFxComb(Fz, SA, SR, IA, V, P, ref coef)
CalculateFyPure(Fz, SA, SR, IA, V, P, ref coef)
CalculateFyComb(Fz, SA, SR, IA, V, P, ref coef)
CalculateMxPure(Fz, SA, SR, IA, V, P, ref coef)
CalculateMxComb(Fz, SA, SR, IA, V, P, ref coef)
CalculateMyPure(Fz, SA, SR, IA, V, P, ref coef)
CalculateMyComb(Fz, SA, SR, IA, V, P, ref coef)
CalculateMzPure(Fz, SA, SR, IA, V, P, ref coef)
CalculateMzComb(Fz, SA, SR, IA, V, P, ref coef)
CalculateLoadedRadius(Fz, SA, SR, IA, V, P, ref coef)
```

These functions define the primary outputs of OptimumT. The user may include any number of additional functions that may be called from within these primary functions. It should be noted that the quality of the code in the calculation file will affect the speed of the fitting and graphing processes. Each function must follow the signature of the functions defined in the template file. Each function returns a single floating point value for the input values Fz, SA, SR, IA, V, P and the coefficient array.

When writing the functions themselves the user can refer to the coefficients by their name as previously defined in the coefficient file. eg. `coef["A0 "]`.

5.2.6 Common Programming Errors

The following are common errors made when programming in C#. Check the following basic syntax before compiling your custom .dll.

- In C# each command should be placed on a new line which ends with the ";" character.
- Functions, classes, loops and conditional statements are defined by braces. Each "{" must have a matching "}".

- Each function must have the correct signature. Use intellisense to make sure that each function passes all the required parameters.

For more information on programming in C# see www.msdn.com

5.3 Importing Custom Models in OptimumT

Once the user has created or been given a custom model .dll file it can be loaded into OptimumT. To load the custom model select the *Advanced* menu item in OptimumT then select the *Custom Tire Model* item from the dropdown menu as shown in Figure 5.7.

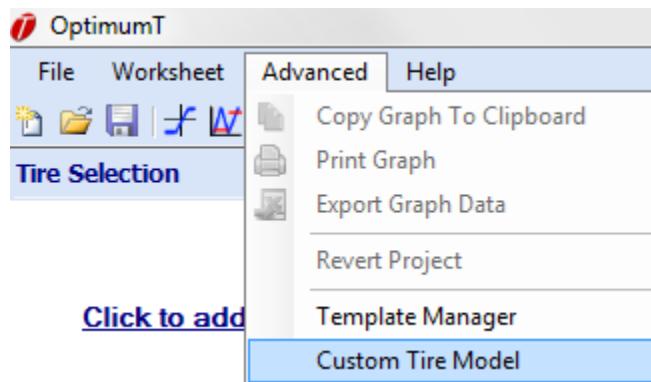


Figure 5.7: Advanced Features Menu

Selecting the *Custom Tire Model* item will launch the *Custom Tire Model Manager* shown in Figure 5.8.

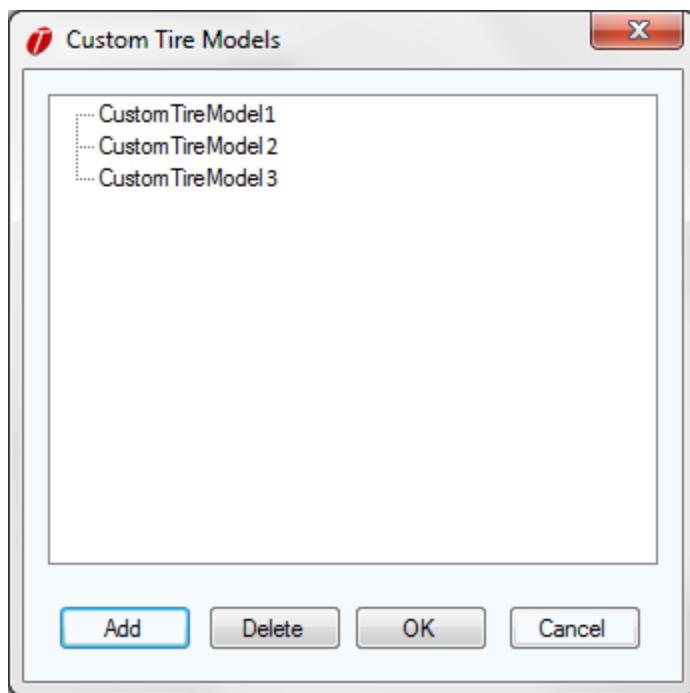


Figure 5.8: Custom Tire Model Manager

The *Custom Tire Model Manager* shows the list of custom models currently available in OptimumT. The user can add or delete models from the list using the add and delete buttons. Note that OptimumT uses the name of the .dll as an identifier so two custom models cannot be added with the same name. See sections 5.2.4 and 5.2.5 for how to name custom models.

Clicking on the *Add* button opens a standard windows dialog. Navigate to the location of the custom .dll that is to be added and click *OK* to add the model to the list.

5.4 The Custom Model Fitting Process

To fit a custom model the user can follow the same process used for standard model fitting. The *Custom Model* option should appear at the bottom of the list of models available to fit in the model fitting dropdown menu as shown in Figure 5.9.

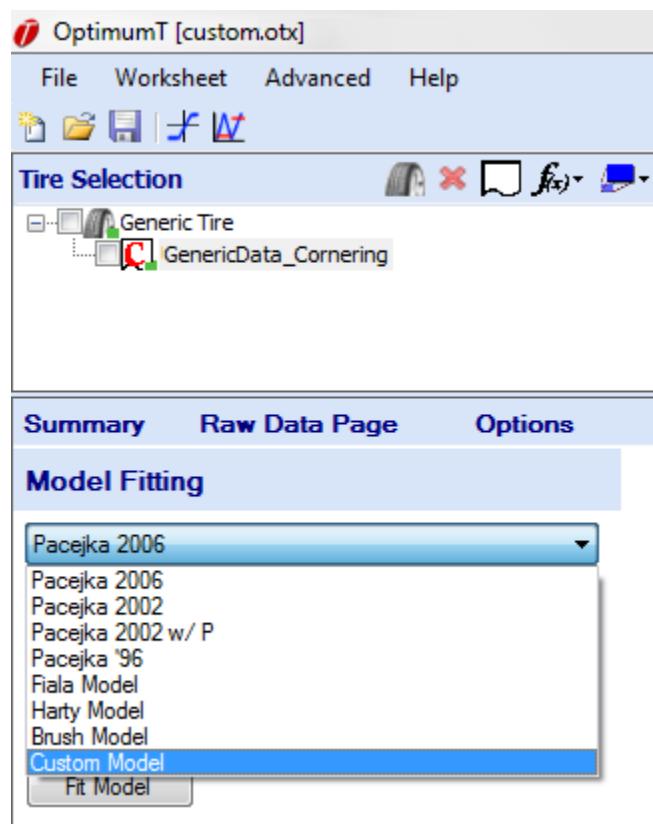


Figure 5.9: Custom Model Fitting

Selecting the *Custom Model* item from the dropdown will bring up the *Custom Model Selection Dialog* shown in Figure 5.10. Select the custom model to be fit by either double clicking on it, or selecting it and clicking the *OK* button. This will launch the fitting process.

The main differences in the custom fitting process are the constraints and boundary files. Which coefficients appear in the *Constraints Wizard* will depend on the *Constraint Boolean* specified in the coefficient file created by the user (see Section 5.2.4). Also the boundaries wizard will not contain predefined templates so a new boundary must be created for each new custom model.

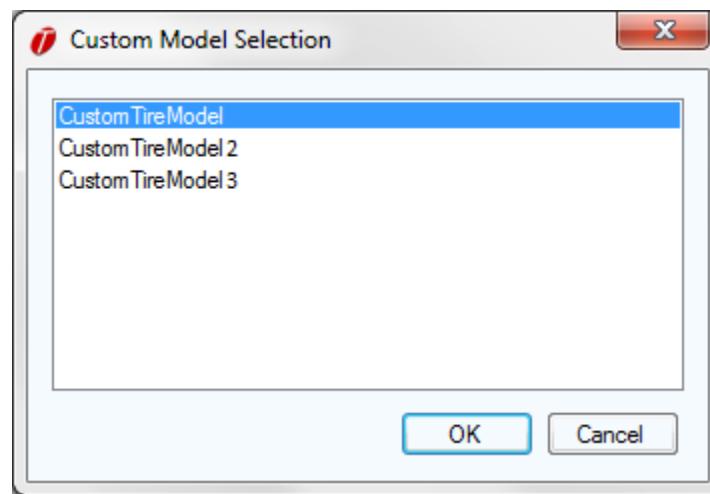


Figure 5.10: Custom Tire Model Selection

5.5 Custom Model Import and Export

The import and export functions work for the custom model in the same way as the standard tire models in OptimumT (see Section 3.2).

However to export models to .tir or Excel format the user must first create a template for each model. The process for creating and modifying export templates is illustrated in Section 3.2.1.

Chapter 6

Additional Features

6.1 OptimumT Add-in

The OptimumT add-in allows users to access all of the output quantities in OptimumT from Excel, Matlab or any other software package that supports COM components. Simulations and calculations that require a tire model can be done easily using this feature. Look-up tables of the tire characteristics can also be created quickly for use in other applications.

In the add-in, the model coefficients are specified to the OptimumT functions with long character strings. These strings can be directly exported from OptimumT. To do this first click on the desired tire model in the project tree. This will expose the model coefficients in the data entry area. Then click on the *Options* button as shown in Figure 6.1. Then select *Export* and *Add-in Model to Clipboard*. This will copy the encoded string to the computer clipboard and allow it to be pasted into Excel or any other software.

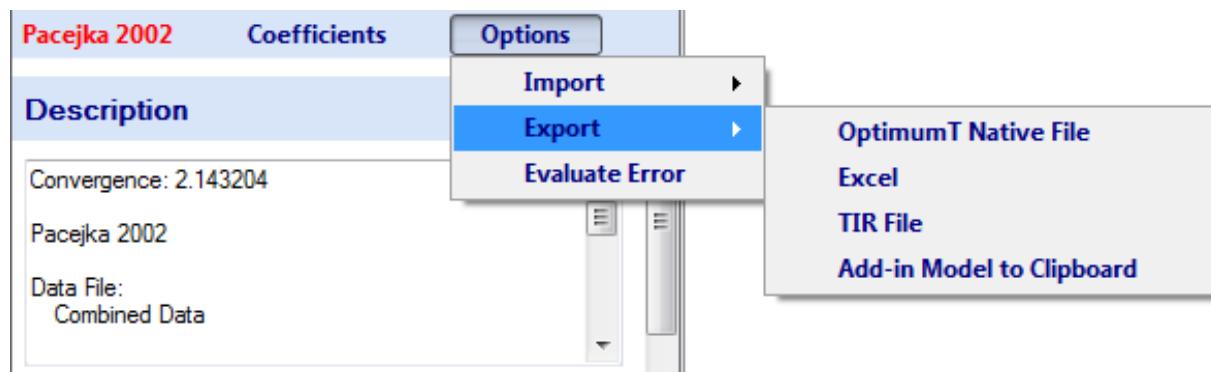


Figure 6.1: Export the Encoded Model Coefficients from OptimumT

The inputs and outputs of the add-in are expected to be in the coordinate system defined for the model. The syntax used for the inputs of the functions is demonstrated in the following

equation.

```
Output = Function(Fz, SA, SR, IA, V, P, ModelCoefficients)
```

The coordinate system can be queried using the command:

```
GetModelInfo(ModelCoefficients)
```

The units used are also restricted. The units used are displayed in Table 6.1. The values calculated are outputted in the same units as those in the table.

Add-in Units	
Inputs	Unit
Normal Load (Fz)	Newton
Inclination Angle (IA)	degree
Slip Angle (SA)	degree
Slip Ratio (SR)	ratio
Speed (V)	meter/sec
Pressure (P)	bar

Table 6.1: Units used in the OptimumT Add-in

6.1.1 Using the Add-in with Excel

The installation and use of the add-in in Excel 2007 will be demonstrated. Once OptimumT is installed the Add-in can be accessed easily from Excel. First the user should click the *Start* button in the top left corner of Excel. Then click the *Excel Options* button as shown in Figure 6.2.

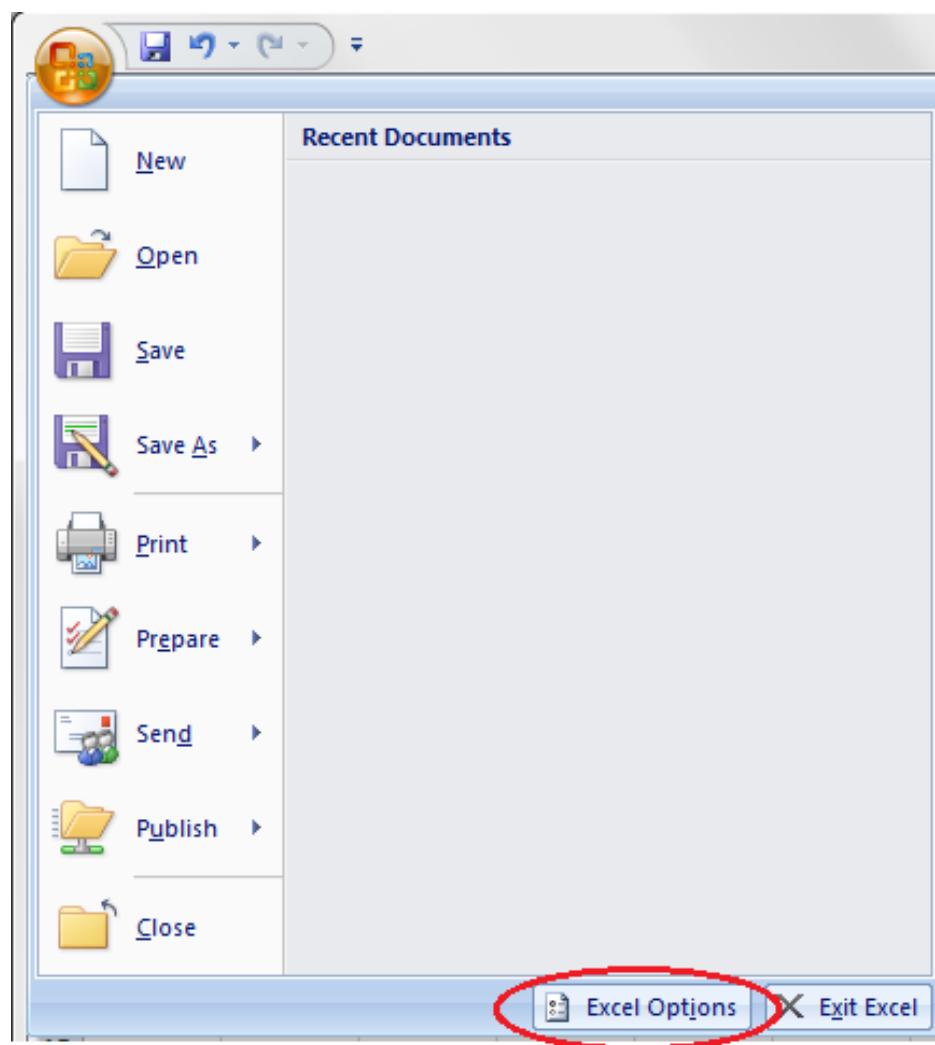


Figure 6.2: Excel Options

Once the Excel options are open click on *Add-Ins* on the left side of the window. In the dropdown box labeled *Manage:* near the bottom of the window select *Excel Add-ins*. This is shown in Figure 6.3 Then click on the *Go* button.

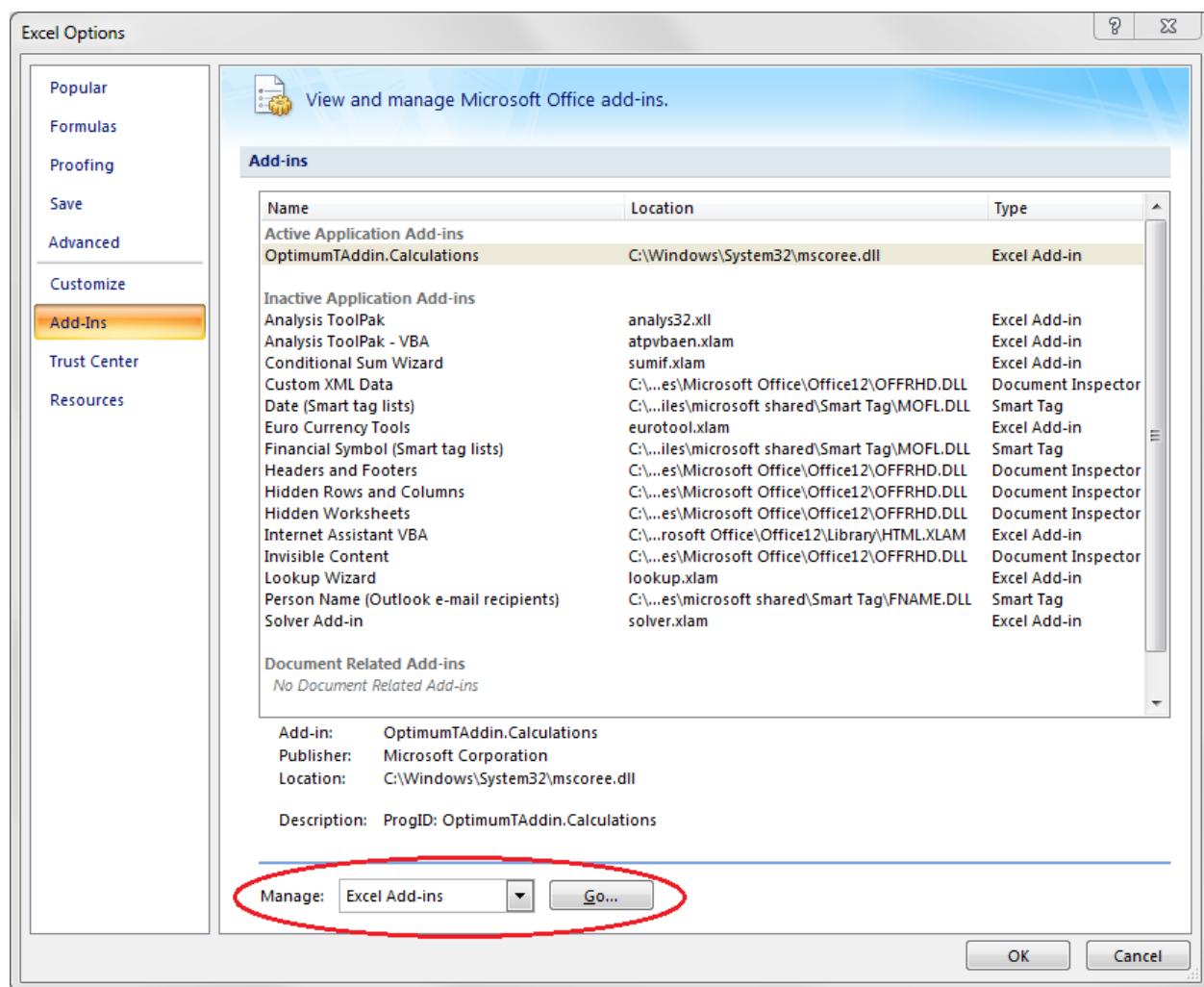


Figure 6.3: Excel Add-Ins

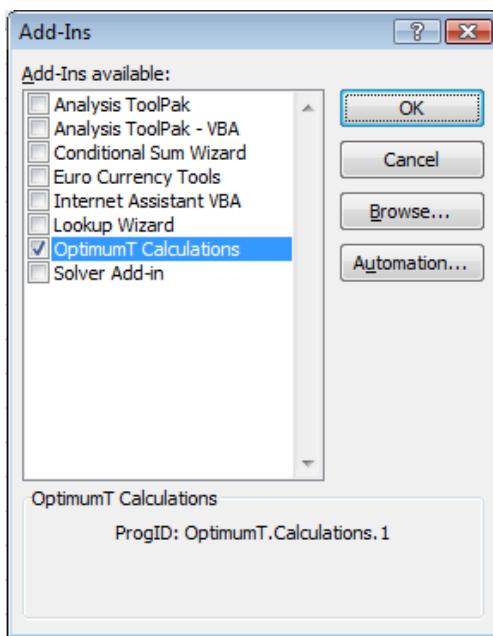


Figure 6.4: Implementing the OptimumT Add-In

The *Add-ins* selection window will now appear. In Figure 6.4 *OptimumT.Calculations* is shown in the list of add-ins. This add-in will have to be added to list by clicking on the button labeled *Automation....*. This will open the *Automation Server* window as shown in Figure 6.5. Selecting *OptimumT.Calculations* and pressing the *OK* button will load the COM add-in into Excel. Now the add-in can be used in the same way as functions built into Excel.

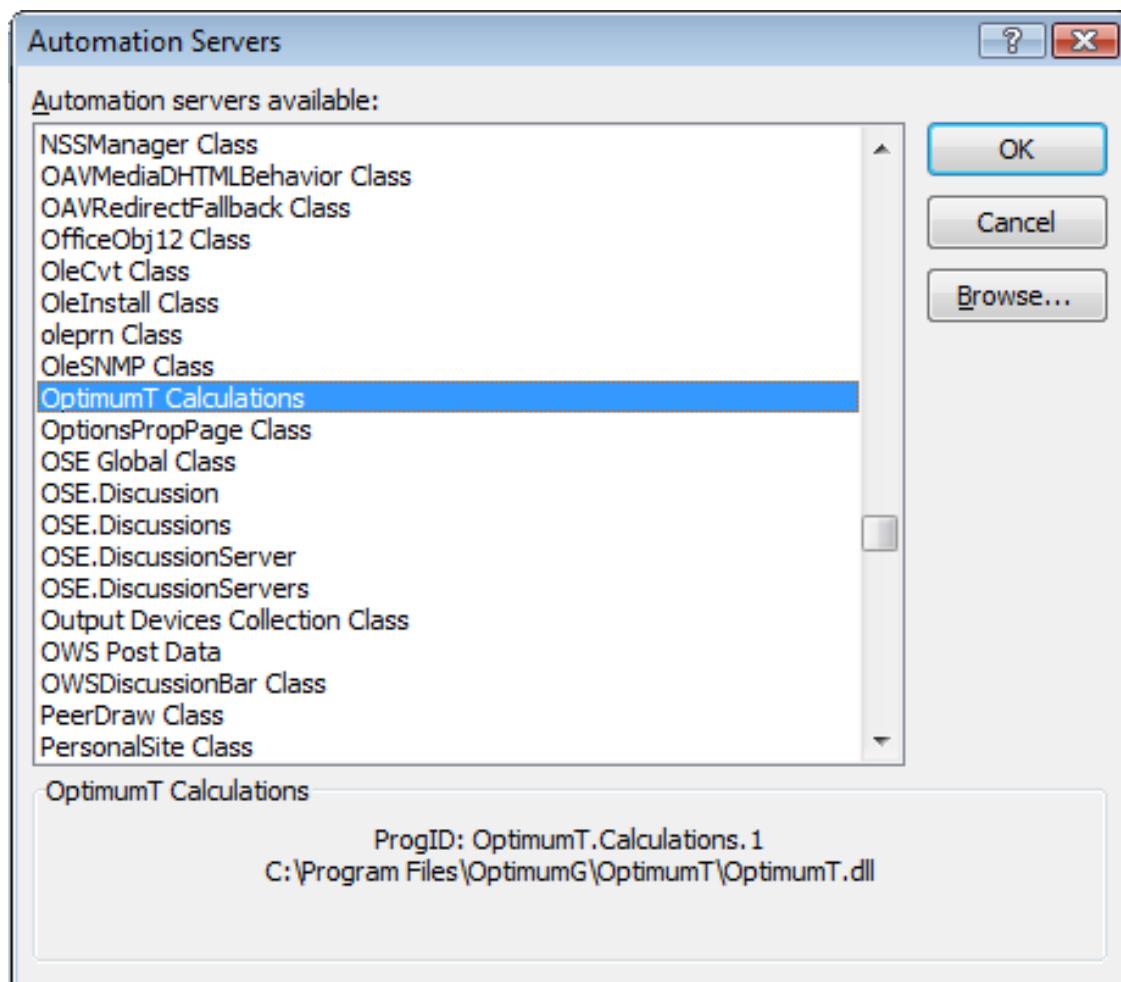


Figure 6.5: Selecting the COM Add-in in Excel

Now the Add-in can be used in Excel by clicking on the *Insert Function* button in Excel. This will open the window shown below in Figure 6.6. The available functions will be displayed by selecting *OptimumTExcelAddin* in the categories dropdown box.

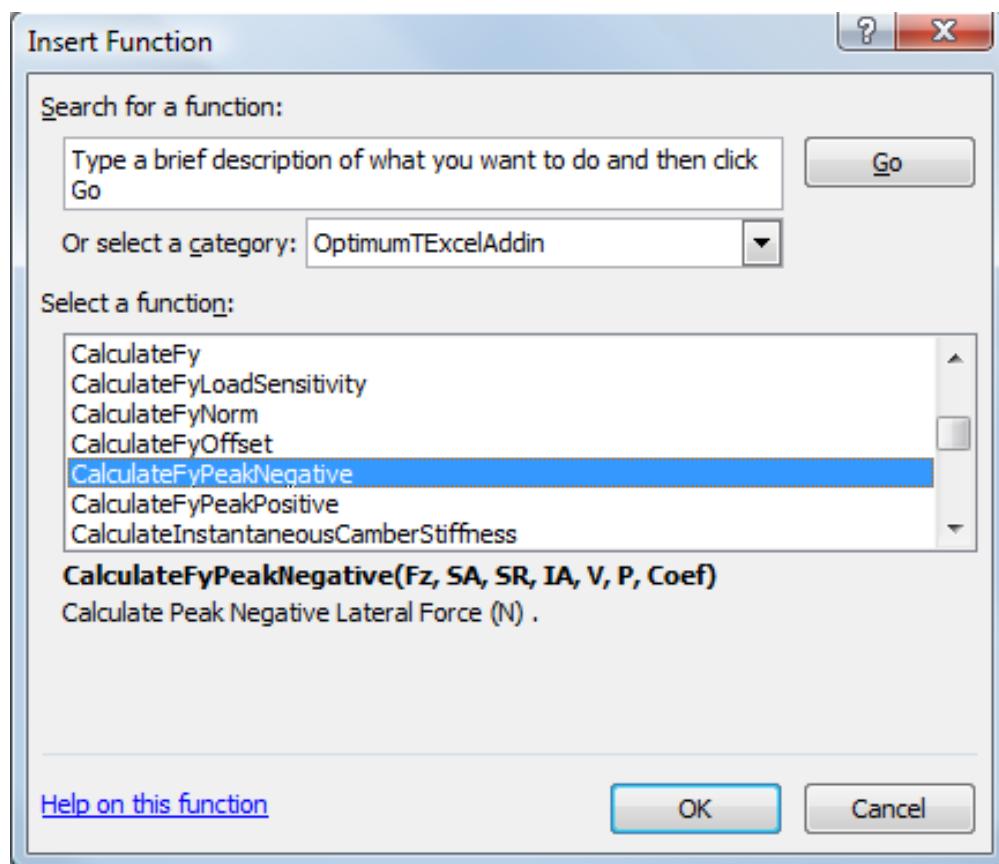


Figure 6.6: Inserting OptimumT Functions into Excel

Then the parameters of the function can be inputted as shown in Figure 6.7. As can be seen in this figure the input parameters are the six tire model inputs (normal load, slip angle, slip ratio, inclination angle, velocity and inflation pressure) and the tire model coefficients. The tire model coefficients are inputted as an encoded string as shown in Figure 6.8.

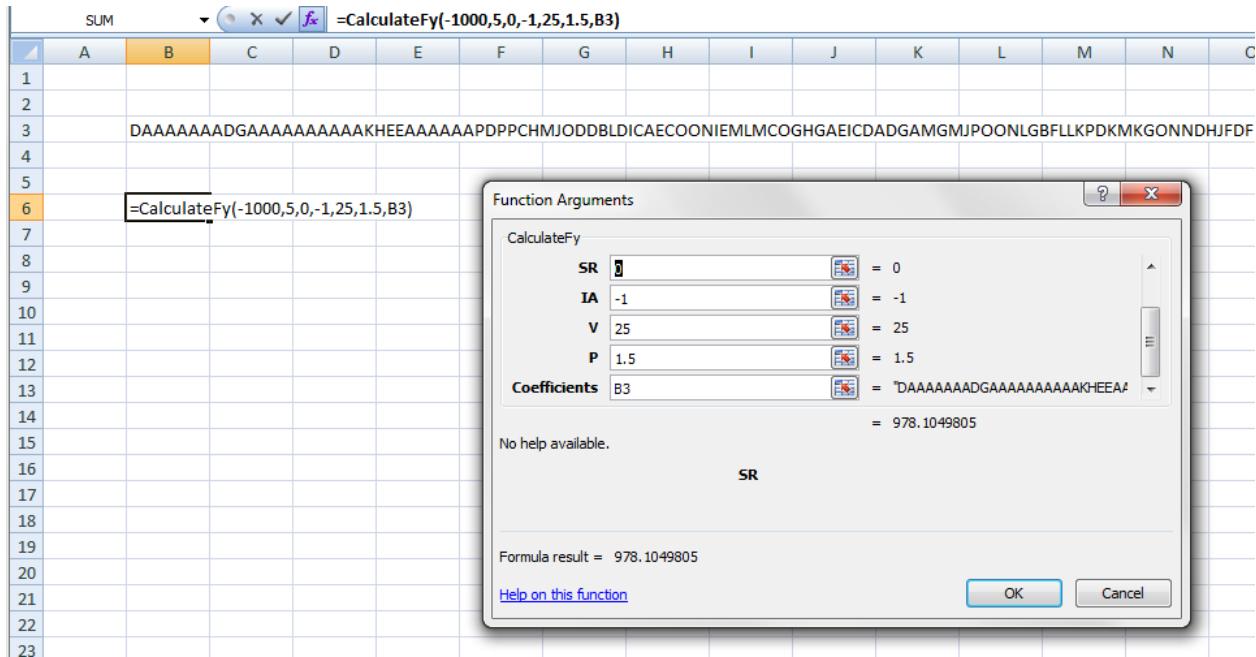


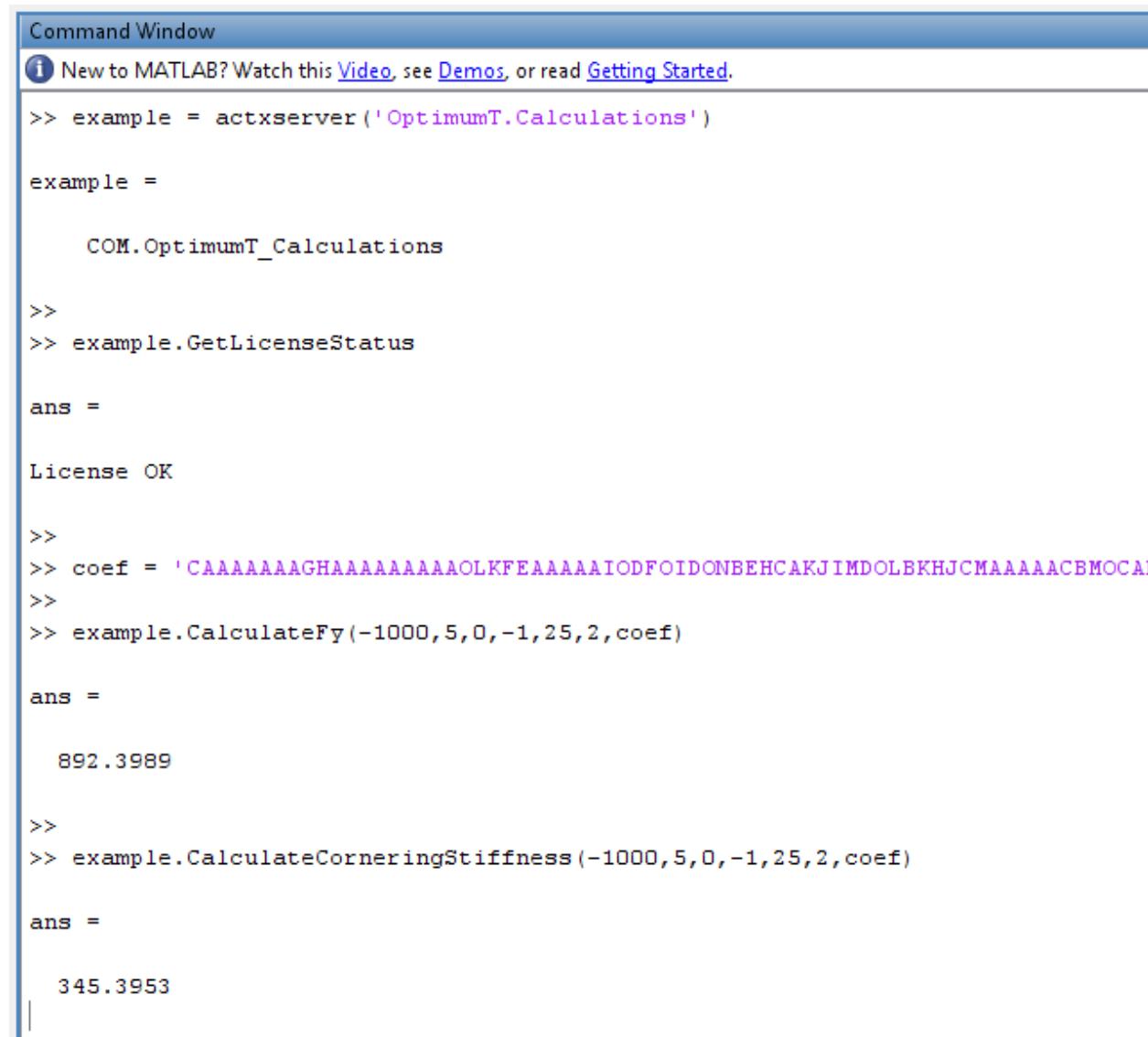
Figure 6.7: OptimumT Add-in Input Parameters

Figure 6.8: Encoded String that represents the Model Coefficients

6.1.2 Matlab COM Add-in

The use of the COM add-in in Matlab is demonstrated in this section. Every time that Matlab is opened the add-in needs to be loaded. The add-in is loaded in Matlab using the *actxserver* function as shown below in Figure 6.9. Then all of the add-in functions can be accessed through the *handle.method* syntax, were *handle* is the variable that the add-in was loaded as and *method* is the OptimumT function that is to be used. In the figure below

example is the handle and *GetLicense*, *CalculateFy* and *CalculateCorneringStiffness* are all functions. To get the full list of the available functions just type *handle.Methods* in the Matlab command window.



```

Command Window
1 New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> example = actxserver('OptimumT.Calculations')

example =
    COM.OptimumT_Calculations

>>
>> example.GetLicenseStatus

ans =
    License OK

>>
>> coef = 'CAAAAAAAGHAAAAAAAOLKFEEAAAATODFOIDONBEHCAKJIMDOLBKHJCMAAAACBMOCAI'
>>
>> example.CalculateFy(-1000,5,0,-1,25,2,coef)

ans =
    892.3989

>>
>> example.CalculateCorneringStiffness(-1000,5,0,-1,25,2,coef)

ans =
    345.3953
|
```

Figure 6.9: Using the OptimumT Add-in in Matlab

An example of the add-in being used in a m-file is shown in Figure 6.10. In this example the handle variable(*h* in this case) is checked to see if it exists. If it does not exist then it loads the add-in. If it does already exist the add-in is not reloaded. This Matlab m-file is also included with OptimumT. It is located in the *Documents* folder of the user who installed OptimumT in a folder called *Matlab COM Addin*.

i This file uses Cell Mode. For information, see the [rapid code iteration](#) video, the [publishing](#) video, or [help](#).

```

1  %% OptimumT Addin
2  % This code demonstrates the OptimumT COM Addin being used in Matlab
3
4  % Check if the Addin has been loaded and if it has not load the Addin
5  A = exist('h', 'var');
6
7  if A == 0
8    h = actxserver('OptimumT.Calculations');
9  end
10
11 % Check License
12 h.GetLicenseStatus
13
14 % Model Coefficient String
15 coef='CAAAAAAAGHAAAAAAAAAAOLKFEEAAAATODFOIDONBEHCAKJIMDOLBKJJCMAAAACBMOCAHGDOGLHEBOKPDAEENMJ';
16
17 % Calculate Lateral Force (Fy)
18 SA = -15:15;
19
20 for i = 1:31
21   Fy(i) = h.CalculateFy(-1000,SA(i),0,0,25,2,coef);
22 end
23
24 % Plot Fy - SA
25 plot(SA,Fy)
26

```

Figure 6.10: Using the OptimumT Add-in in a Matlab m-file

6.1.3 Using the COM Addin in Programs

The exact method for using the OptimumT add-in with your own programs will depend on your programming environment. Please refer to the documentation that supplied with your programming environment for details on using COM components.

In most environments, you will need to add a reference to the OptimumT type library. This is called **OptimumT 1.0 Type Library** or **OptimumTLib**.

There is an example C# project provided with OptimumT. It is located in the *Documents* folder of the user who installed OptimumT in a folder called *OptimumTAddinExample*. The main function for this program is:

```

static void Main(string[] args)
{
  Console.WriteLine("Welcome to the OptimumT Add-in example program\n\n");
  string modelCoef = "CAAAAAAAGHAAAAAAAPELJFMGGGGKODAAAAKBEAJGPMLPDA" +
    "JJIEFPLNLHAAAOEGJIGBAMFAOJCLPLOLKEMCNLBODPFNNLKNOLOEODEF" +
    "BFFFBEJBLCJLPDHONLLMDKF0AJHKDPGDCEELLNIHNCOMLODBJIJLLCMD" +

```

```

"NJANDAKBPFDPPLGJNNGIPLIMNKOJPDICBKMHPEJDNMMLGAODMDAMNELK" +
"NMNDFJJCPJLDLNEJGILLNMHPLIBMDMNBAOBEOFBKBLPDLPFNCKILEHLCC" +
"JKDNIBOCJIDBPNDMILDPDGNMABEDPBOJOAMMFEBJCJDPOIFECOKDDMPDCO" +
"MDCHNCOIOLKKJGMEBFCJCBCBEMIMFMFLDFAGGKHPMGNFBGMDMDANLHM" +
"DKPIHEKKLABLCAEOKNGEIIAEJHMKKBAEBGKBAELOHEJBDCMHFOFEJPL" +
"KFAGCFMLAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" +
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" +
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" +
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" +
"AAAAAAAAAAAAAAAAAAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPD" +
"AAAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPD" +
"AAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPDAAAAAIPD";

```

```

OptimumTLib.Calculations calc = new OptimumTLib.Calculations();

// Display the type of model and the coordinate system
Console.WriteLine("The model is:");
Console.WriteLine(calc.GetModelInfo(modelCoef));

// set up variables to plug into the tire model
float Fz = -3000.0f;           // the vertical load [N]
float IA = 1.0f;               // the inclination angle [deg]
float SR = 0.0f;               // the slip ratio [fraction]
float V = 10;                  // the speed [m/s]
float P = 2;                   // the inflation pressure [bar]

// declare a variable to store the calculated lateral force in
float Fy;

Console.Write("\n\n    SA      Fy\n");
// loop through a series of slip angles and display the resulting force
for (float SA = -10.0f; SA <= 10.0f; SA+=2.0f)
{
    Fy = calc.CalculateFy(Fz, SA, SR, IA, V, P, modelCoef);
    Console.WriteLine(string.Format("{0,5} {1,10}", SA, Fy));
}

Console.Write("\nPress any key to exit...");
Console.ReadKey();
}

```

6.2 Template Manager

The template manager allows the user to easily organize or delete templates in OptimumT. The template manager can be accessed by clicking on *Advanced* in the main toolbar and then clicking on *Template Manager*. A window similar to that shown in Figure 6.11 will open. In this window the user can manage graph templates, CSV import templates, coefficient boundaries and model export templates by clicking on the associated tabs. As can be seen in the figure the coefficient boundaries are organized by the different models. The models can be chosen from the dropdown box above the list of templates top of the window.

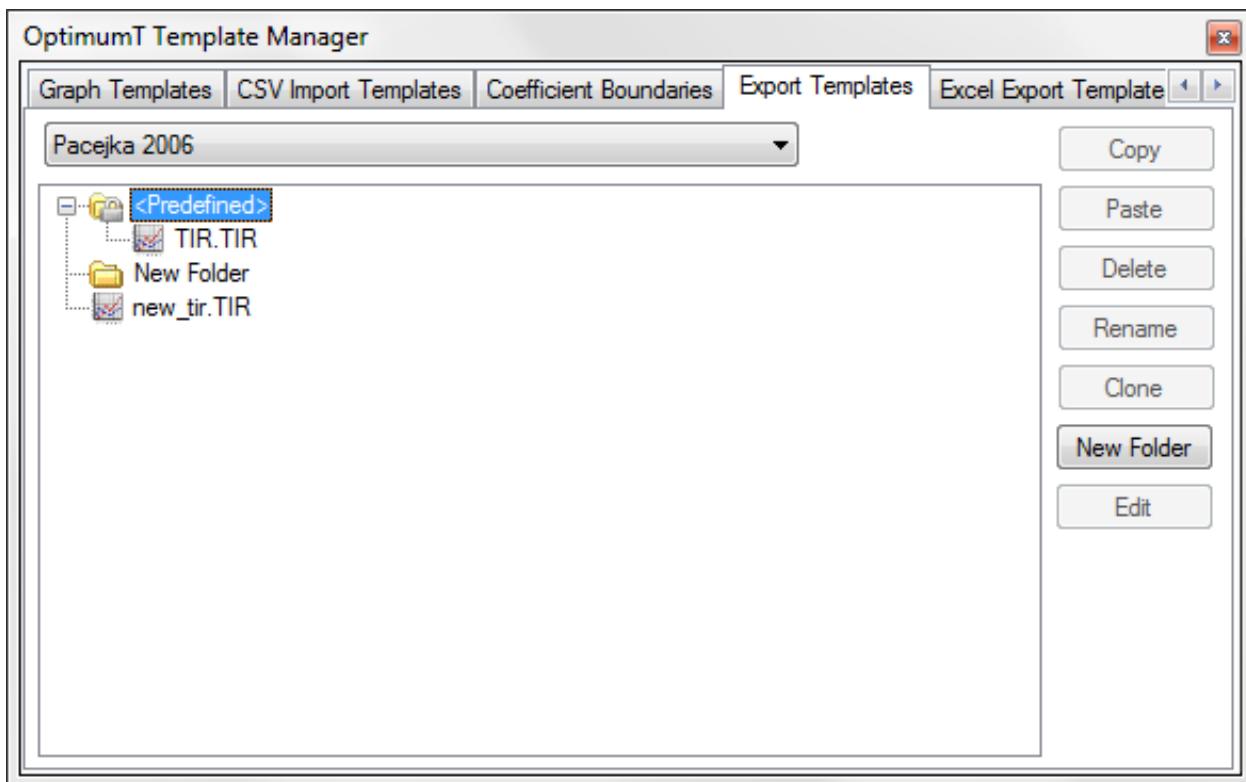


Figure 6.11: Template Manager

The templates can be deleted, renamed, or cloned using the buttons on the right side of the *Template Manager*. New folders can also be created so that the templates can be easily organized. The templates that are contained in the *Predefined* folder are included in OptimumT and cannot be edited or deleted. However they can be cloned. When they are cloned the copied version of the template will be moved outside of the *Predefined* folder, so that it can be modified. The paste button allows the user to paste files from the clipboard into the appropriate folder which a useful feature when transferring large numbers of templates.

6.3 Project Backups

OptimumT saves a backup of the project before each save operation. This allows you to recover from a corrupted project or to recover if you accidentally made an unwanted change to the project and saved it. By default the last five revisions of the project are stored.

To go to a previous version of the project, click on the Advanced menu then on *Revert Project*. You will then see the *Revert Project* dialog (shown in Figure 6.12). Select the backup that you wish to revert to and click on *Revert*. The date and time shown in this dialog is the time at which that version was initially created.

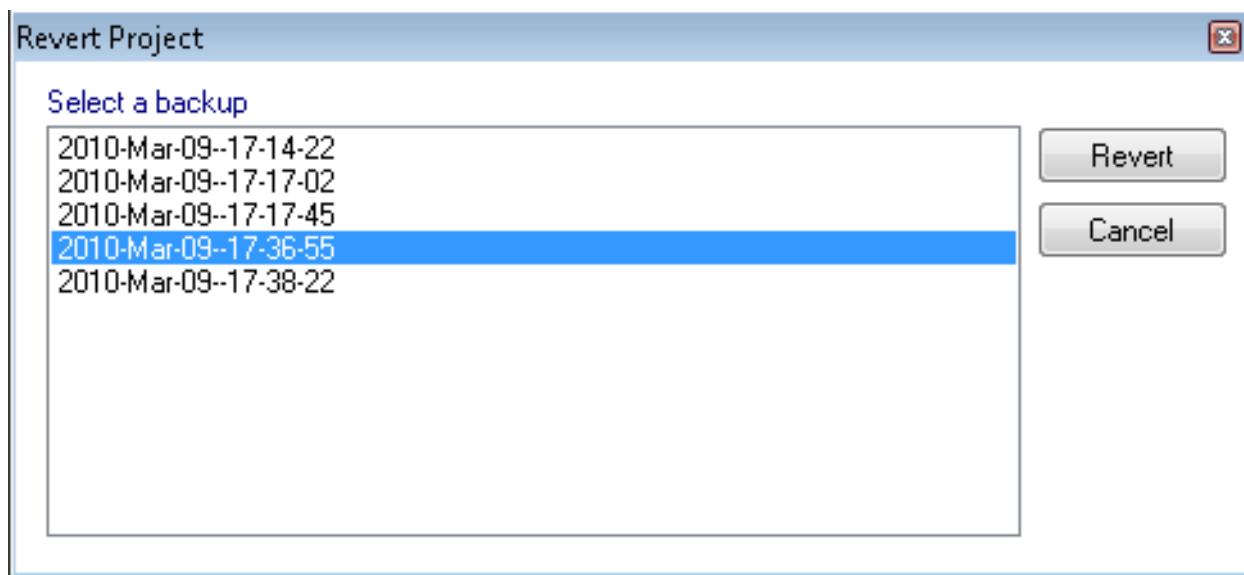


Figure 6.12: Revert Project dialog

When you revert a project, OptimumT automatically creates a backup of the project immediately before reverting it. This allows you to "undo" a revert operation.

If a project fails to load, the *Revert Project* dialog will automatically be shown. This allows you to recover your project from an earlier backup.

6.4 Error Evaluation

The *Error Evaluation* tool is shown in Figure 6.13. This tool allows quick comparison of the error of the tire models against different sets of raw data. It also allows comparison between the four different types of error calculation. To access this tool first select the tire model for the error to be evaluated in the project tree. At the top of the data entry form select *Options* and then *Evaluate Error*.

Once the *Error Evaluation* is open the data to be compared should first be selected in the list on the left. Then the models to base the error calculation should be selected. Clicking on the *Evaluate* button at the bottom of the window will calculate and display the error each of the different error calculation methods.

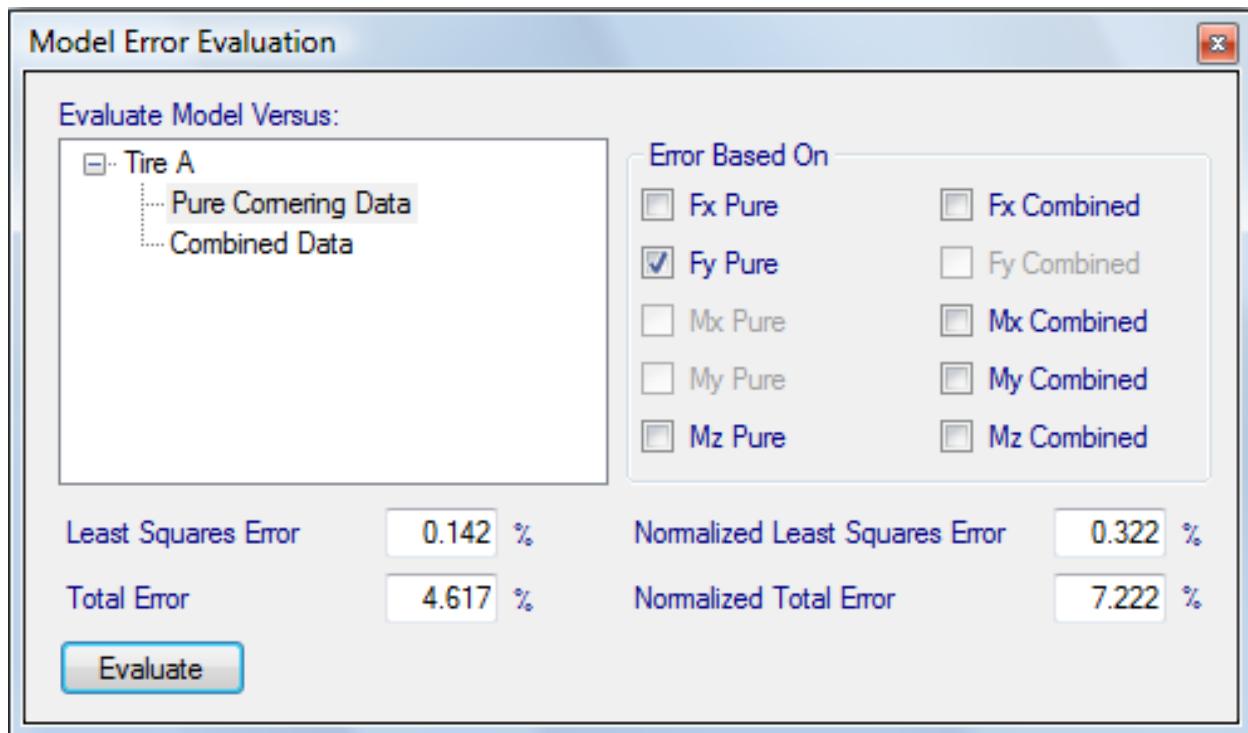


Figure 6.13: Error Evaluation

6.5 Lookup Table Export

OptimumT offers the possibility to export lookup tables. These tables are compatible with simulation products such as CarSim or other software that uses a lookup table for modelling tyre performance rather than directly importing model coefficients and equations. Lookup tables may be exported by selecting a model to export from the tree then clicking *Options->Export->Lookup table*. This will bring up the lookup table export form shown below.

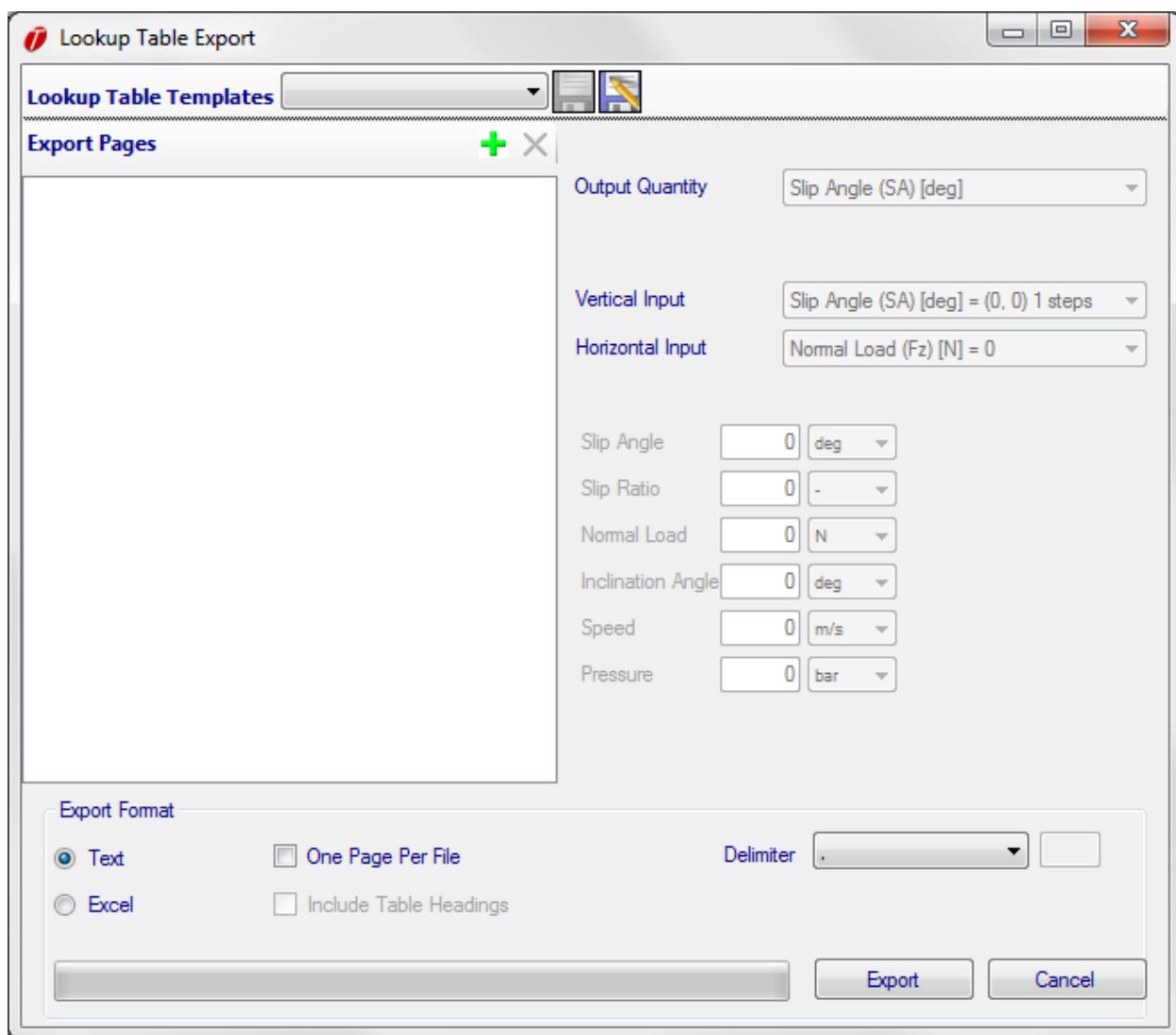


Figure 6.14: Lookup Table Export Form

The form is split into three parts. The first is the *Page Tree*. Pages can be added to the export file by clicking the green plus sign and deleted by clicking the red cross. Pages may also be renamed in the tree, these names will provide the basis for the file names when exporting text files, or the sheet names when exporting to Excel.

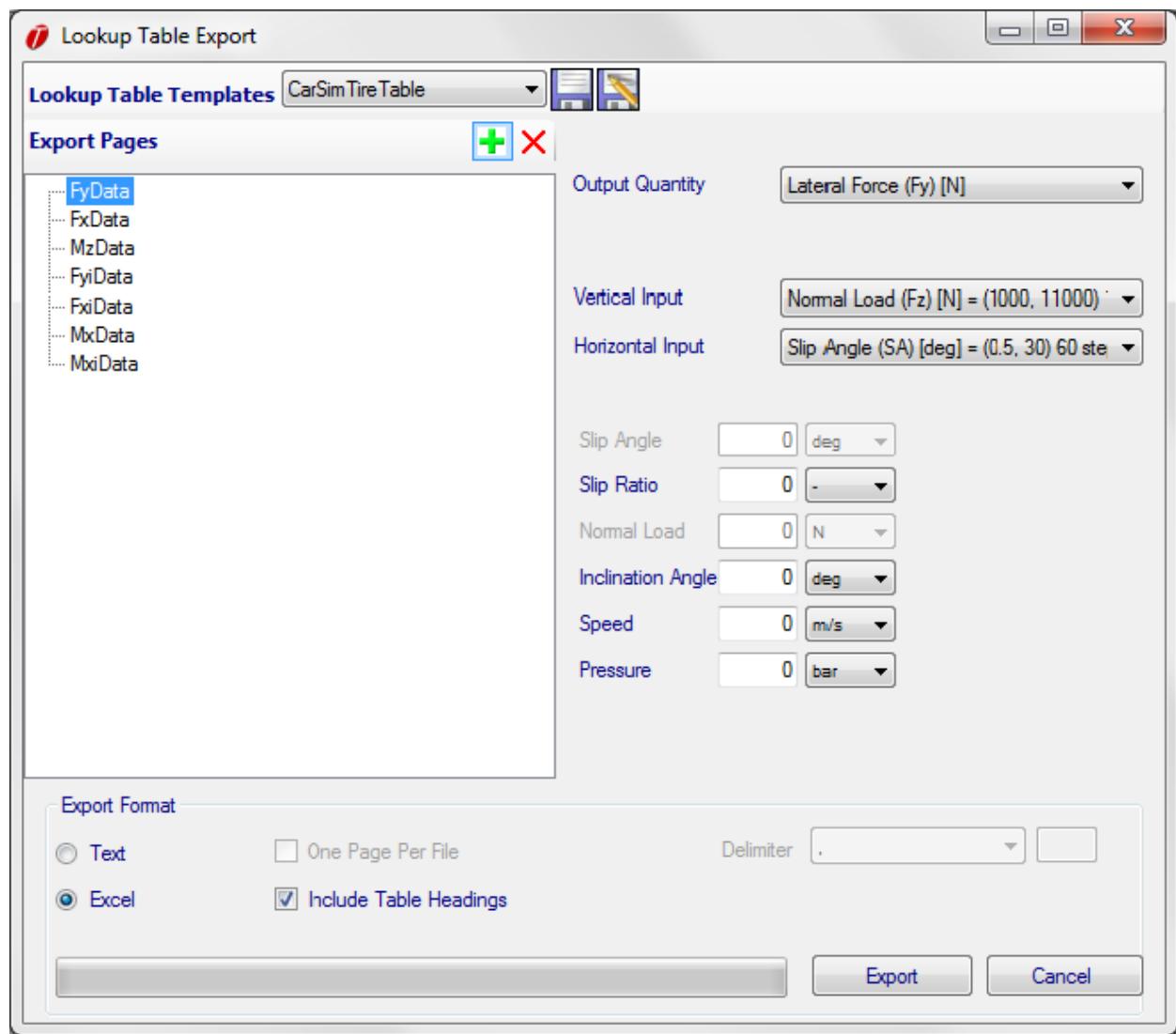


Figure 6.15: Lookup Table Export Settings

The second part of the form is the *Output Settings*. Each page in the page tree has an individual output settings sheet which specifies the parameters of the look up table. The output dropdown specifies which OptimumT output will be used to populate the table. The *Vertical* and *Horizontal* dropdowns allow you to specify the type, range and step of the vertical and horizontal axes. The remaining boxes specify the values of the outputs to be held constant in the table.

The third part of the table is the *Format section*. This specifies the global format for the export. Files may be exported as an MS Excel workbook or as text file. Text files may export multiple pages in one file or export each page to an individual file by selecting the *one page per file* radio button. When exporting files to excel a labels may be added to the tables to indicate the names of the table parameters. This is done by selecting the *Include*

Table Headings check box. The text file delimiter may be adjusted using the delimiter drop down or a custom delimiter may be used by typing a delimiter in the *custom delimiter text box*.

A useful feature of the lookup table export is the ability to save the settings in a template for repeated exports. The *lookup table template bar* is located at the top of the form. Templates may be applied using the dropdown menu or new templates saved using the save icons.

Chapter 7

Tips and Tricks

7.1 Plot All Data

By default when raw data is inputted into OptimumT the *Plot All Data* option is enabled. This option will cause all of the data to appear in graphs regardless of the graph input parameters. This allows the user to quickly view the data and check that it is correct before continuing. However the color of the graphed data will not vary with the input parameters. Therefore to look at only certain parts of the data or to have the data colored by the graphing parameters this option needs to be disabled.

There are three different ways to disable the *Plot All Data* option. The first one is by right clicking on the raw data in the project tree and selecting *Remove "Plot All Data" In All Graphs*. This is shown in Figure 7.1. This will disable the feature for the selected data set in all of the graphs.

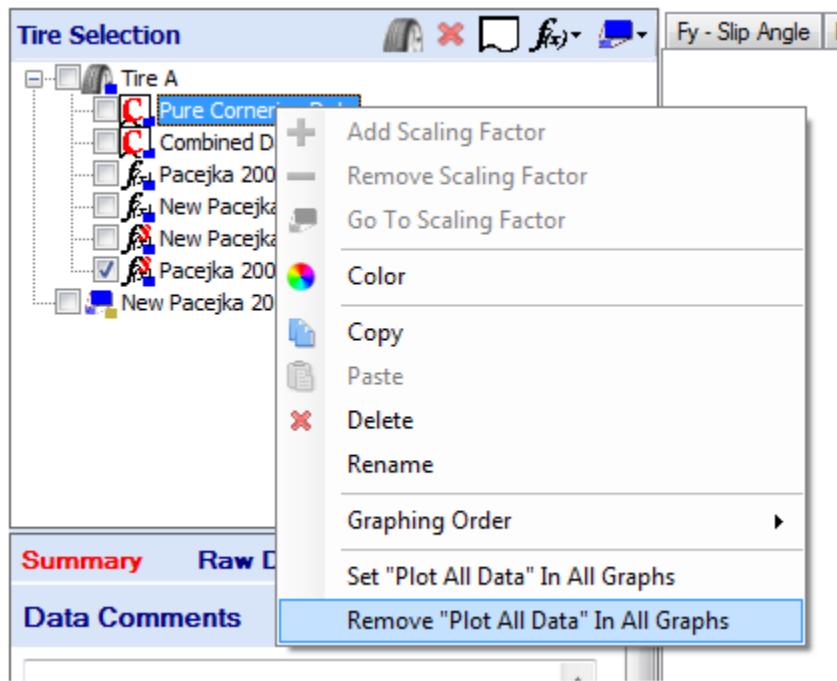


Figure 7.1: Plot All Data Option in the Project Tree

The other two procedures to disable this option are located in the Project Items Setup at the bottom of the graph setup form as shown in Figure 7.2. Since this is located in the graph setup form it will only affect the graph that it corresponds to. As can be seen "*<all data>*" will appear to the right of the name of the data if *Plot All Data* is enabled. By right clicking on the data and clicking on the *Plot All Data* selects or unselects this feature. The Options button in the upper right of the figure allows the user to set all or no items to *Plot All Data*. Again this will change all of the items for the current graph only.

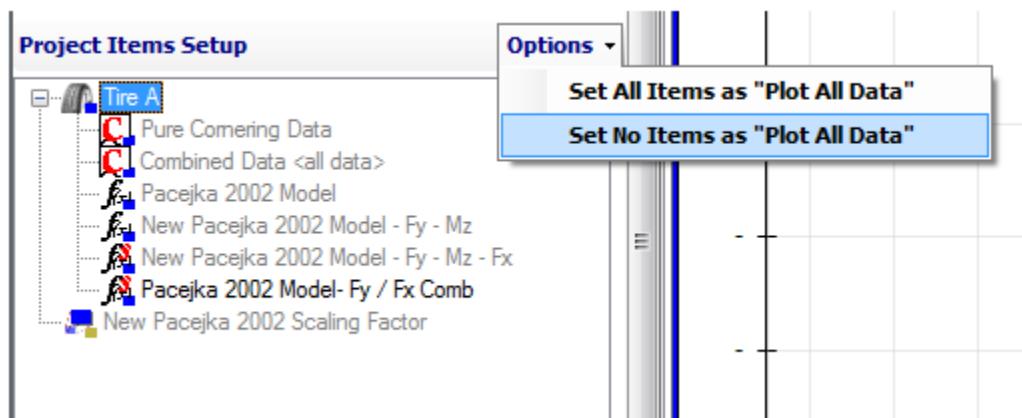


Figure 7.2: Plot All Data Option in the Project Item Setup

7.2 Large Tolerance Graph Inputs

There are cases when you want to make graphs without regard to a certain tire condition. For example, if you want to plot data for several different inflation pressures, without typing in all of the specific pressures into the display. In cases like this, you can simply set the graph input as some arbitrary fixed value and set a very large tolerance. Taking the inflation pressure example, if you had tire data taken at 1.50, 1.75, 2.00 and 2.25 bar, you could set the graph input for inflation pressure to a fixed value of 2.00 bar and set the tolerance to 2.00 bar. All of the inflation pressures will then be displayed.

7.3 Override Default Name in Legend

Often the name of an item (a tire model or raw data) will be too long to conveniently view in the legend. Therefore you can easily display a custom name in the legend without having to change the name of the item. This is done by clicking on the item in the project tree that a custom name is to be given to. In the data entry area the input form of the selected item will appear as shown in Figure 7.3. The custom legend name will be used if the checkbox labeled *Override default name in legend* is checked. The custom name can be inputted into the textbox below the checkbox below. The resulting graph with the custom legend name can be seen in Figure 7.4.

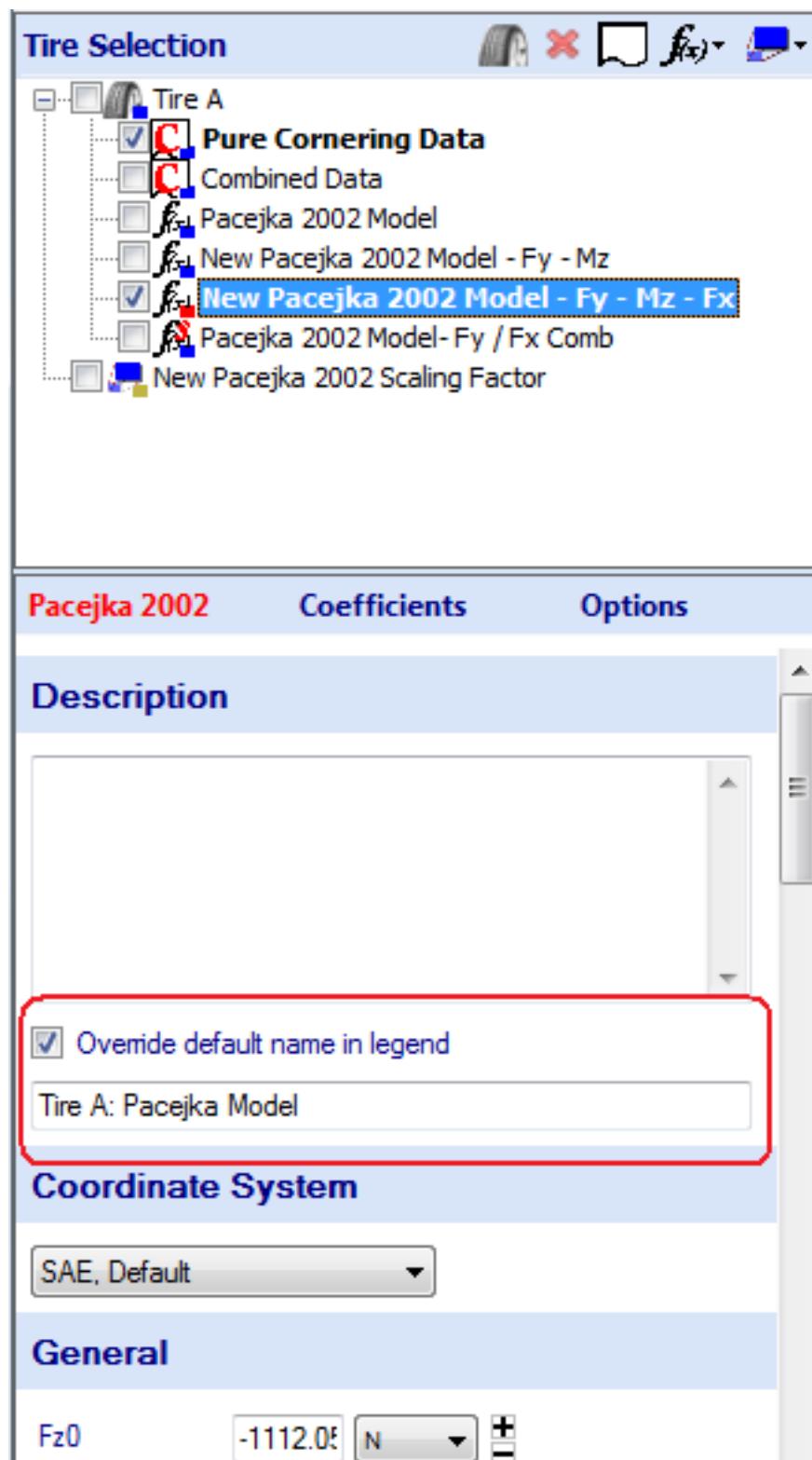


Figure 7.3: Override Default Name in Legend

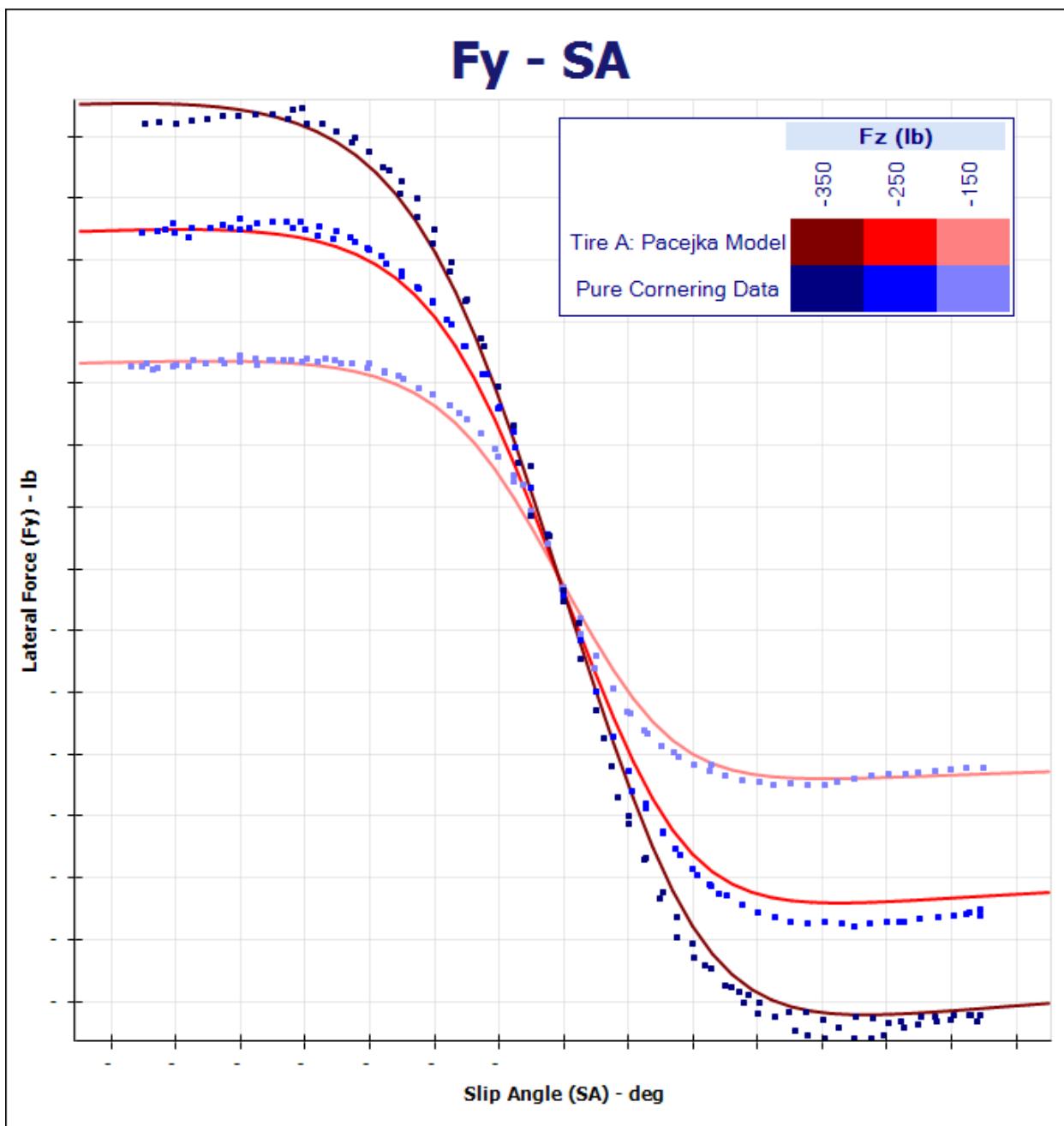


Figure 7.4: Custom Tire Model Name

7.4 Changing Units

When a unit is changed in OptimumT its corresponding value is automatically converted to the new unit. If you wish to change the unit of a quantity without performing a unit conversion, hold down the *shift* key while selecting the new unit. For example, if you type

10 into an input box with the unit m selected, but you wish to enter 10mm, simply changing the unit to *mm* will perform a unit conversion, resulting in 10000mm. If you hold down the *shift* key while selecting the new unit, the desired result of 10mm will result.

7.5 Preview Model Coefficient Change

The values of the model coefficients can be changed by double clicking on the "+" or "-" button next to the coefficient. If the model is shown on a graph, holding down the "+" or "-" button will show a preview of the model with the coefficient modified by 10%. An example of this is shown in Figure 7.5. In this figure the "+" button of the *pDy1* coefficient is being held down. This change can be made permanent by double clicking on this button.

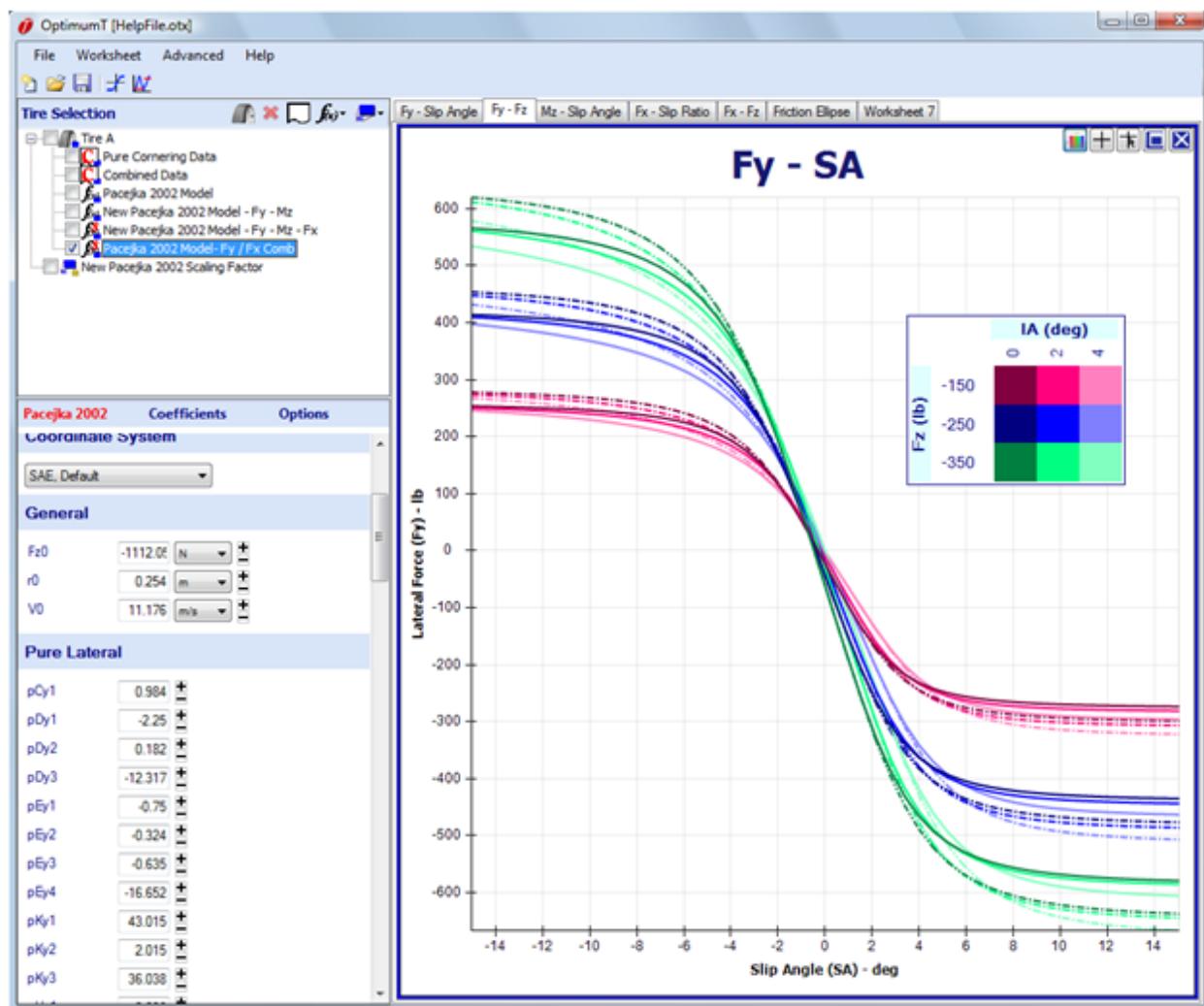


Figure 7.5: Preview Model Coefficient Change

7.6 Hide Axis Values

The user can choose whether or not to display axis values on the graphs. This can be very important to ensure confidentiality of the data. This option is available in the Axis Selection drop down dialog boxes as shown in Figure 7.6. If the *Hide Axis Values* box is checked the numeric values on the specified graph axis will not be displayed.

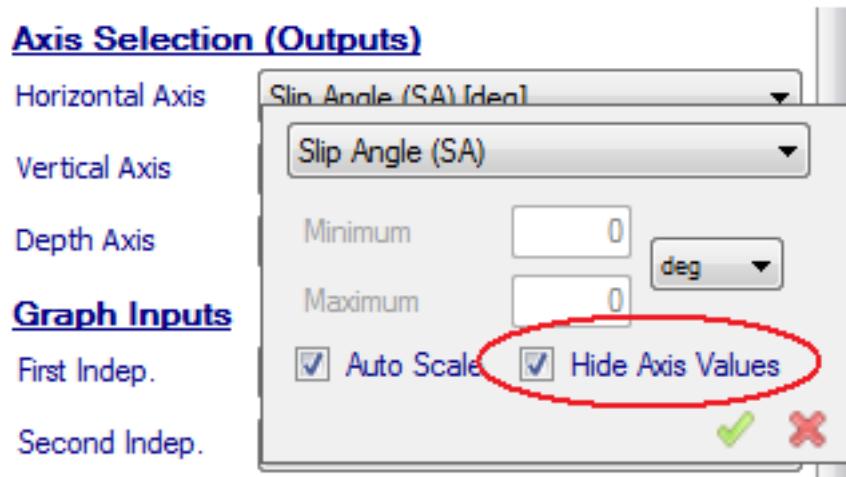


Figure 7.6: Hide Axis Values

7.7 Importing Multiple Data Files

If more than one data file needs to be imported and merged, then they can be simply be selected together when choosing files to import. Hold down the shift or ctrl key to select multiple files. Note that all the files selected must be in exactly the same format (columns in the same order and the same units used). This feature can be useful if multiple files are produced in a single test. For example, if cornering data for different inclination angles are contained in different files, it is useful to multi-select when importing so that these files are merged within OptimumT when they are imported.

Chapter 8

References

8.1 Example Files

The following examples are installed with OptimumT.

- OptimumT Add-in Example
- Matlab Examples
- OptimumT Demo Project
- Custom Model Example Project

These files are installed with OptimumT in the users documents folder under *OptimumT Samples*

8.2 Coordinate Systems

OptimumT allows the user to select between four different coordinate systems:

- Society of Automotive Engineers (SAE) J670e
- Adapted SAE
- International Organization for Standardization (ISO)
- Adapted ISO

Figure 8.1 shows the orientation of these coordinate systems and Figure 8.2 shows graphs of typical tire parameters in the different coordinate systems.

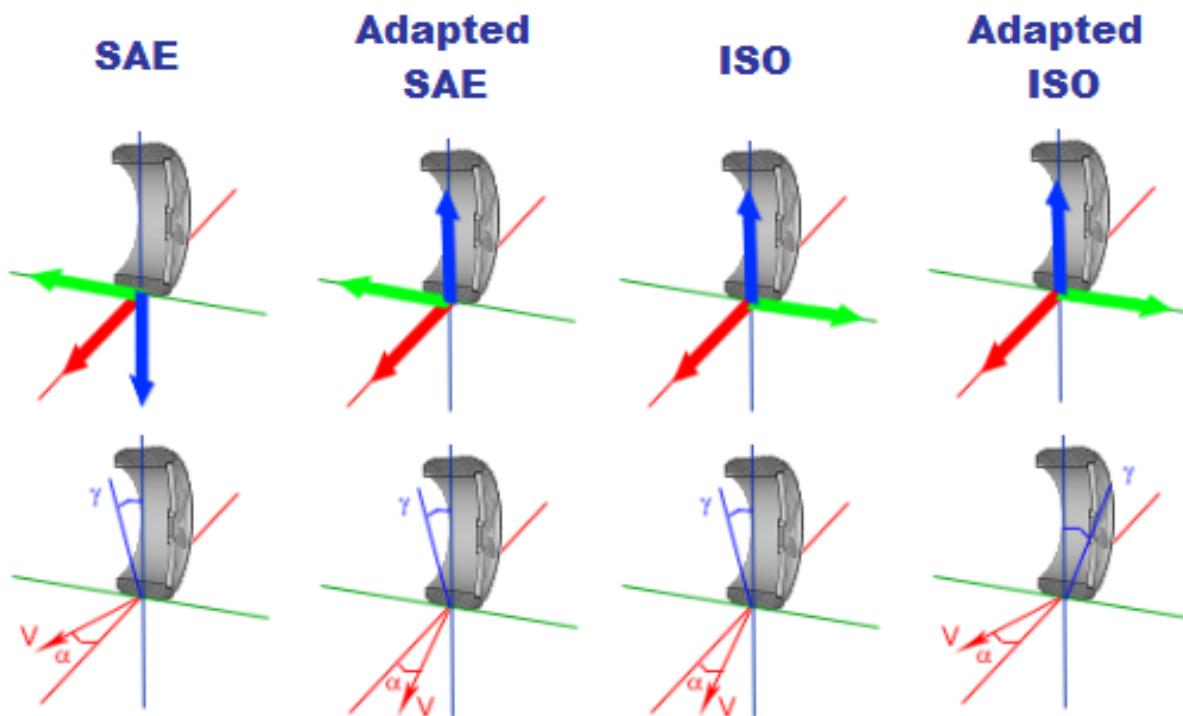


Figure 8.1: Coordinate Systems (viewed from front)

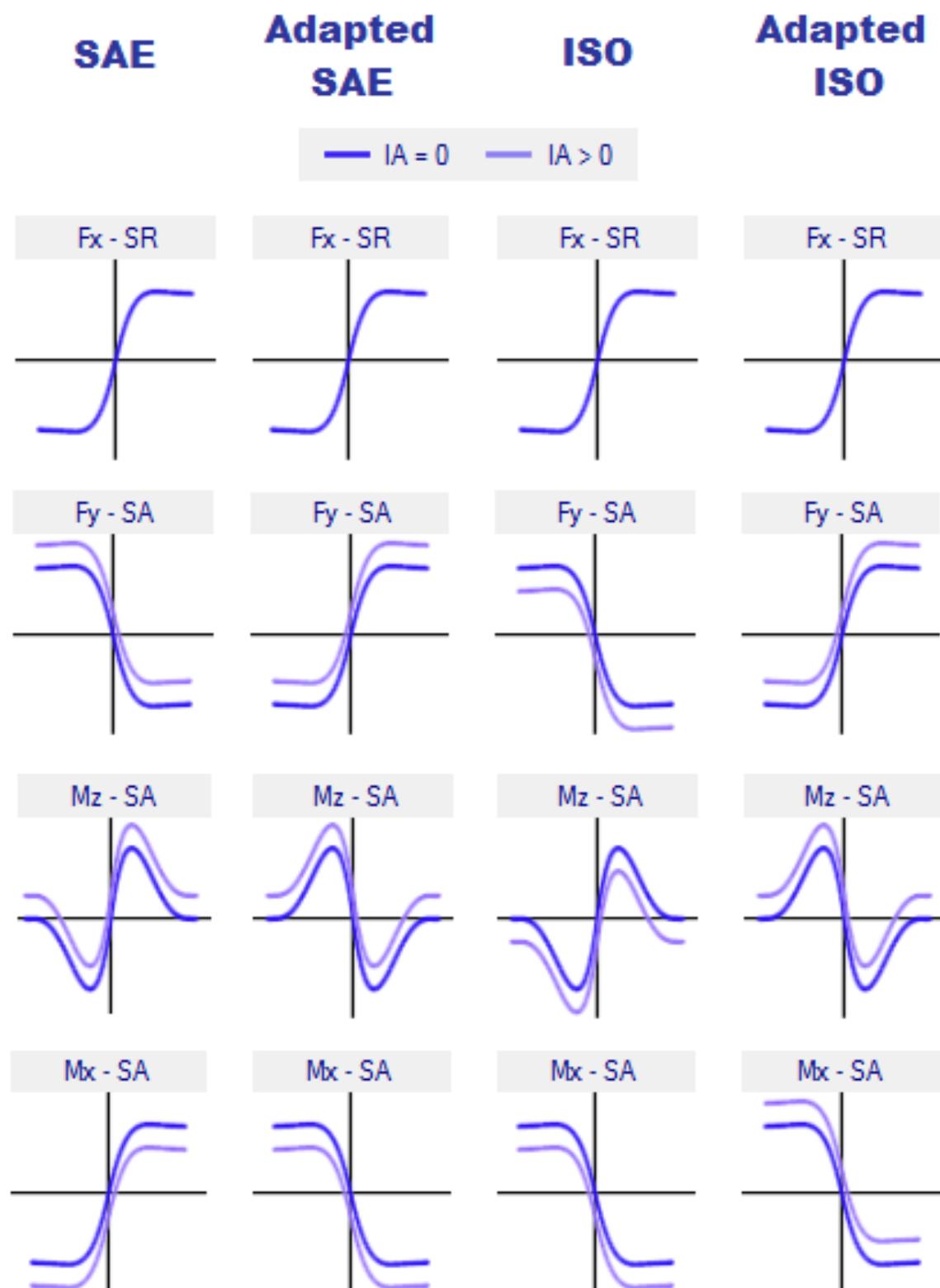


Figure 8.2: Common Tire Parameters in different Coordinate Systems

8.3 Units

OptimumT allows the user to select the units to be displayed. A summary of the available units are included in Table 8.3.

Units in OptimumT				
Unit Type	Angle	Force	Force / Angle	Force/Ratio
Units	degree	Newton	Newton/Degree	Newton
	radian	kilonewton	Newton / radian	Newton / percent
		kilogram-force	kilonewton / degree	kilonewton
		pound	kilonewton / radian	kilonewton / percent
			kilogram-force / degree	kilogram-force
			kilogram-force / radian	kilogram-force / percent
			pound / degree	pound
			pound / radian	pound / percent

Table 8.1: Units in OptimumT

Units in OptimumT				
Unit Type	length	Moment	Pressure	Ratio
Units	meter	Newton meter	bar	Unit-less
	centimeter	Newton millimeter	Pascal	percent
	millimeter	kilonewton meter	kilopascal	
	foot	kilogram-force meter	pound / square inch	
	mile	foot pound		
		inch pound		

Table 8.2: Units in OptimumT

Units in OptimumT			
Unit Type	Stiffness	Time	Velocity
Units	Newton / meter	second	meter / second
	Newton / millimeter	hour	kilometer / hour
	kilonewton / meter		feet / second
	kilonewton / millimeter		mile / hour
	kilogram-force / meter		
	kilogram-force / millimeter		
	pound / foot		
	pound / inch		

Table 8.3: Units in OptimumT

8.4 Fiala Model

The Fiala tire model is based on the physical characteristics of the tire. Table 8.4 summarizes these characteristics. This model does not include combined longitudinal or lateral force, the effect of inclination angle, the lateral force offset at zero slip (from tire conicity or ply steer), or tire load sensitivity. More information about the Fiala model can be found in "The Multibody Systems Approach to Vehicle Dynamics", 2004, by Mike Blundell and Damian Harty.

Fiala Model Parameters		Unit Type
R1	Tire tread width divided by two	length
Cs	Longitudinal tire slip stiffness	force / ratio
Calpha	Lateral tire slip stiffness	force / angle
Cr	Rolling resistance moment coefficient	length
U0	Tire static friction coefficient	ratio
U1	Tire sliding friction coefficient	ratio

Table 8.4: Fiala Model Parameters

8.5 Harty Model

The Harty tire model aims to provide a compromise between the complex Pacejka models and the limited Fiala model. Features of the Harty model include the ability to model camber thrust and the load dependency of cornering stiffness.

The model does not include the calculation of the overturning moment. The model also treats driving and braking forces as symmetric. Note the Harty model is only compatible with the SAE coordinate systems due to the method used to model the camber thrust. The model may still be graphed in other coordinate systems after fitting. For more information on the Harty model see "Intermediate tyre model for vehicle handling simulation" by M V Blundell and D Harty.

Longitudinal Model

$$D = D_x + (|F_z| - R_L) \frac{dD}{dF_z}$$

$$F_x = (1 - e^{(-A_x|S/S_c|)}) \mu D F_z \text{sign}(1, \alpha)$$

$$if(S < S_c)$$

$$F_x = (1 - e^{-A_x}) \mu D F_z sign(1, \alpha)$$

Lateral Model

$$B = B_y + (|F_z| - R_L) \frac{dB}{dF_z}$$

$$if(\alpha < \alpha_c)$$

$$F_{y\alpha} = (1 - e^{(-A_y|\alpha/\alpha_c|)}) \mu B F_z sign(1, \alpha)$$

$$if(\alpha > \alpha_c)$$

$$F_{y\alpha} = (1 - e^{-A_y}) \mu B F_z sign(1, \alpha)$$

$$F_{y\gamma} = -F_z \tan(\gamma)$$

$$F_y = -F_{y\alpha} + F_{y\gamma}$$

Aligning Torque Model

$$C = C_{tz} + (|F_z| - R_L) \frac{dC}{dF_z}$$

$$L_{CP} = 2(R_1^2 - (\frac{R_1 + F_{zk}}{K_z})^2)^{0.5}$$

$$x_{pt} = \frac{L_{CP}}{4}$$

$$if(\alpha < \alpha_c)$$

$$M_z = -F_{y\alpha} C x_{pt} (1 - |\frac{\alpha}{\alpha_c}|)$$

$$if(\alpha > \alpha_c)$$

$$M_z = 0$$

Rolling Resistance Model

$$if(V > 0)$$

$$M_y = -C_r F_z$$

$$if(V < 0)$$

$$M_y = C_r F_z$$

The Harty model requires 14 parameters to model the tire.

Harty Model Parameters		Unit Type
F_{z0}	Reference tire load	force
$R1$	Tire loaded radius	length
α_c	Critical slip angle	angle
A_y	Curvature factor for lateral force	
B_y	Scale factor for lateral force at reference tire load	
dB/F_z	Dimunition of lateral force scale factor with load	
S_c	Critical slip ratio	ratio
A_x	Curvature factor for longitudinal force	
C_{tz}	Scale factor for aligning moment at reference tire load	
dC/F_z	Dimunition of aligning moment scale factor with load	
D_x	Scale factor for longitudinal force at reference tire load	
dD/F_z	Dimunition of longitudinal force scale factor with load	
K_z	Tire vertical stiffness	force / length
C_r	Rolling resistance coefficient	length

Table 8.5: Harty Model Parameters

8.6 Brush Model

Brush tire models can be very simple or very complex. The model included in OptimumT is a very simple example of the brush model. The brush model is a physically based model that represents the tire as a row of elastic bristles that can deflect in the direction of the road. The deformation of these elements to applied forces represents the combined elasticity of the tire belt, carcass, and tread.

The model included in OptimumT does not include the effects of inclination angle, the lateral force offset at zero slip (from tire conicity or ply steer), tire load sensitivity, or the fall off of force after the optimum slip has been reached. However, it does include the effects of combined lateral and longitudinal slip.

Brush Model Parameters		Unit Type
Fz0	Nominal vertical load	force
mu	Coefficient of friction	ratio
cpy	Lateral tread element stiffness	force / length
cpx	Longitudinal tread element stiffness	force / length
a0	Contact patch length at Fz0 divided by two	ratio

Table 8.6: Brush Model Parameters

8.7 Pacejka Models

The Pacejka "Magic Formula" tire models are empirical relations that model the steady-state forces and moments produced by the tire as a function of the tire conditions (i.e. slip angle, slip ratio, inclination angle, etc . . .). These models include combined longitudinal and lateral force effects, inclination angle effects, lateral and longitudinal force offset, and tire load sensitivity. The primary form for the Pacejka models is given in the equations below. The table following these equations describes the various parameters.

$$y = D \sin [C \arctan \{Bx - E(Bx - \arctan Bx)\}]$$

With

$$Y(X) = y(x) + S_V$$

$$x = X + S_H$$

General Pacejka Parameters	
Input/Output	Description
Y	Output Lateral force, longitudinal force, or aligning torque
X	Input Slip ratio or tangent of slip angle
Parameters	
B	Stiffness Factor Slope at the origin
C	Shape Factor Shape of the resulting curve
D	Peak Value Peak Value with C>=1
E	Curvature Factor Curvature and horizontal position of the peak
H	Horizontal Shift
V	Vertical Shift

Table 8.7: Pacejka Model

A description of the Pacejka models implemented in OptimumT is given in this section. In the following section the coefficients used in these models are summarized.

8.7.1 Pacejka '96

This model is given in the 1996 paper "The Tire as a Vehicle Component" by Hans B. Pacejka. This model includes the combined lateral and longitudinal tire response as well as lateral camber response and load sensitivity. This model does not include the rolling resistance or overturning moment of the tire. This model includes 78 coefficients.

8.7.2 Pacejka 2002

This model is given in Pacejka's book "Tire and Vehicle Dynamics" published in 2002. It is similar to the '96 model but has additional coefficients in the combined lateral and longitudinal models. It also includes models for the rolling resistance and overturning moment. This model includes 89 coefficients.

8.7.3 Pacejka 2002 with Inflation Pressure Effects

This model is described in the paper "Extending the Magic Formula and SWIFT Tyre Models for Inflation Pressure Changes" by Dr. Ir. A.J.C. Schmeitz, Dr. Ir. I.J.M. Besselink, Ir. J. de Hoogh, and Dr. H. Nijmeijer. This model incorporates the effect of inflation pressure into the Pacejka 2002 model. Ten additional coefficients, including the reference pressure P_{i0} , are added to the model. These coefficients appear in the pure lateral, longitudinal, and aligning torque models. This model includes 99 coefficients.

8.7.4 Magic Formula 5.2

This model is a close development of the the Pacejka 2002 model. This model differs from Pacejka 2002 in the way that it models the effect of camber. The main advantage of the MF5.2 model is that models the effect of camber on the longitudinal coefficient of friction. This model includes 90 coefficients.

8.7.5 Pacejka 2006

This model is given in the second edition of Pacejka's book "Tire and Vehicle Dynamics" published in 2006. This model is based off of the 2002 model but includes significant modifications to the pure lateral and aligning torque models. An additional coefficient is also added to both the combined lateral and longitudinal models. This model includes 97 coefficients.

8.8 Pacejka Coefficients

Table 8.8 displays all of the Pacejka coefficients used in OptimumT. These coefficients are grouped into nine different categories depending on what tire characteristics they describe: General, Pure Lateral, Pure Longitudinal, Aligning Torque, Combined Lateral, Combined Longitudinal, Combined Aligning Torque, Overturning Moment and Rolling Resistance. On the left the name and a brief description of the coefficient are given. On the right the "x" in the boxes indicates whether or not each coefficient is included in each of the specific Pacejka models.

Pacejka Coefficents		Models			
	General	96	02	02Pi	06
Fz0	Nominal load	x	x	x	x
r0	Tire unloaded radius	x	x	x	x
V0	Reference velocity		x	x	x
Pi0	Reference pressure			x	
Pure Lateral		96	02	02Pi	06
pCy1	Shape factor	x	x	x	x
pDy1	Lateral coefficient of friction at Fz0	x	x	x	x
pDy2	Variation of friction with load	x	x	x	x
pDy3	Variation of friction with camber squared	x	x	x	x
pEy1	Lateral curvature at Fz0	x	x	x	x
pEy2	Variation of curvature with load	x	x	x	x
pEy3	Zero order camber dependency of curvature	x	x	x	x
pEy4	Variation of curvature with camber	x	x	x	x
pEy5	Camber curvature			x	
pKy1	Maximum cornering stiffness	x	x	x	x
pKy2	Load at which maximum stiffness occurs	x	x	x	x
pKy3	Variation of stiffness with camber	x	x	x	x
pKy4	Variation of stiffness with camber squared			x	
pKy5	Lateral stiffness dependency with camber			x	
pKy6	Camber stiffness factor			x	
pKy7	Load dependency of camber stiffness factor			x	
pHy1	Horizontal shift at Fz0	x	x	x	x
pHy2	Variation of horizontal shift with load	x	x	x	x
pHy3	Variation of horizontal shift with camber	x	x	x	
pVy1	Vertical shift at Fz0	x	x	x	x
pVy2	Variation of vertical shift with load	x	x	x	x
pVy3	Variation of vertical shift with camber	x	x	x	x
pVy4	Variation of vertical shift with camber and load	x	x	x	x
pPy1	Variation of cornering stiffness with inflation pressure			x	
pPy2	Variation of cornering stiffness with inflation and load			x	

pPy3	Variation of friction with inflation pressure	x
pPy4	Variation of friction with inflation pressure squared	x
Pure Longitudinal		96 02 02Pi 06
pCx1	Shape factor	x x x x
pDx1	Longitudinal coefficient of friction at Fz0	x x x x
pDx2	Variation of friction with load	x x x x
pDx3	Variation of friction with camber	
pEx1	Longitudinal curvature at Fz0	x x x x
pEx2	Variation of curvature with load	x x x x
pEx3	Variation of curvature with load squared	x x x x
pEx4	Factor in curvature while driving	x x x x
pKx1	Longitudinal slip stiffness at Fz0	x x x x
pKx2	Variation of slip stiffness with load	x x x x
pKx3	Exponent in slip stiffness with load	x x x x
pHx1	Horizontal shift at Fz0	x x x x
pHx2	Variation of horizontal shift with load	x x x x
pVx1	Vertical shift at Fz0	x x x x
pVx2	Variation of vertical shift with load	x x x x
pPx1	Variation of slip stiffness with inflation pressure	x
pPx2	Variation of slip stiffness with inflation pressure squared	x
pPx3	Variation of friction with inflation pressure	x
pPx4	Variation of friction with inflation pressure squared	x
Aligning Torque		96 02 02Pi 06
qBz1	Pneumatic trail slope factor at Fz0	x x x x
qBz2	Variation of trail slope with load	x x x x
qBz3	Variation of trail slope with load squared	x x x x
qBz4	Variation of trail slope with camber	x
qBz5	Variation of trail slope with absolute camber	x x x x
qBz6	Variation of trail slope with camber squared	x x x x
qBz9	Slope factor of residual torque	x x x x
qBz10	Slope factor of residual torque	x x x x
qCz1	Shape factor for pneumatic trail	x x x x
qDz1	Peak pneumatic trail	x x x x
qDz2	Variation of peak trail with load	x x x x
qDz3	Variation of peak trail with camber	x x x x
qDz4	Variation of peak trail with camber squared	x x x x
qDz6	Peak residual torque	x x x x
qDz7	Variation of peak torque with load	x x x x
qDz8	Variation of peak torque with camber	x x x x
qDz9	Variation of peak torque with camber and load	x x x x
qDz10	Variation of peak torque with camber squared	x

qDz11	Variation of peak torque with camber squared and load	x
qEz1	Pneumatic trail curvature at Fz0	x x x x
qEz2	Variation of curvature with load	x x x x
qEz3	Variation of curvature with load squared	x x x x
qEz4	Variation of curvature with sign of slip angle	x x x x
qEz5	Variation of curvature with camber and sign of slip angle	x x x x
qHz1	Pneumatic trail horizontal shift at Fz0	x x x x
qHz2	Variation of horizontal shift with load	x x x x
qHz3	Variation of horizontal shift with camber	x x x x
qHz4	Variation of horizontal shift with camber and load	x x x x
qPz1	Variation of peak with inflation pressure	x
Combined Lateral		96 02 02Pi 06
rBy1	Slope factor for combined slip lateral force reduction	x x x x
rBy2	Variation of lateral force slope reduction with slip angle	x x x x
rBy3	Shift factor for slip angle in lateral force slope reduction	x x x x
rBy4	Variation of lateral force combined stiffness with camber	x
rCy1	Shape factor for combined slip lateral force reduction	x x x
rEy1	Curvature factor of combined lateral force	x x x
rEy2	Curvature factor of combined lateral force with load	x x x x
rHy1	Horizontal shift factor for lateral force reduction	x x x
rHy2	Horizontal shift factor for lateral force reduction with load	x x x x
rVy1	Vertical shift at Fz0 for lateral force reduction	x x x x
rVy2	Variation of vertical shift factor with load	x x x x
rVy3	Variation of vertical shift factor with camber	x x x x
rVy4	Variation of vertical shift factor with slip angle	x x x x
rVy5	Variation of vertical shift factor with slip ratio	x x x x
rVy6	Variation of vertical shift factor with the arctan of slip ratio	x x x x
Combined Longitudinal		96 02 02Pi 06
rBx1	Slope factor for combined slip longitudinal force reduction	x x x x
rBx2	Variation of longitudinal force slope reduction with slip ratio	x x x x
rBx3	Variation of longitudinal force combined stiffness with camber	x
rCx1	Shape factor for combined slip longitudinal force reduction	x x x x
rEx1	Curvature factor of combined longitudinal force	x x x
rEx2	Curvature factor of combined longitudinal force with load	x x x
rHx1	Shift factor for combined slip longitudinal force reduction	x x x x

Combined Aligning Torque		96	02	02Pi	06
sSz1	Effect of longitudinal force on aligning torque	x	x	x	x
sSz2	Variation of aligning torque with lateral force	x	x	x	x
sSz3	Variation of aligning torque with camber	x	x	x	x
sSz4	Variation of aligning torque with camber and load	x	x	x	x
Overturning Moment		96	02	02Pi	06
qSx1	Vertical force induced overturning moment	x	x	x	
qSx2	Camber induced overturning moment	x	x	x	
qSx3	Lateral force induced overturning moment	x	x	x	
Rolling Moment		96	02	02Pi	06
qSy1	Rolling resistance torque coefficient	x	x	x	
qSy2	Variation of rolling resistance torque with load	x	x	x	

Table 8.8: Pacejka Coefficents

8.9 Pacejka Scaling Factors

Table 8.9 below gives a summary of the Pacejka scaling coefficients. The default value for all of these coefficients is 1 except for $\lambda\mu V$ which is 0.

Scaling Factors		Models			
Pure Slip		96	02	02Pi	06
λF_{z0}	Nominal load factor	x	x	x	x
$\lambda\mu_x$	Peak longitudinal coefficient of friction	x	x	x	x
$\lambda\mu_y$	Peak lateral coefficient of friction	x	x	x	x
$\lambda\mu V$	Friction decay with slip speed	x	x	x	x
$\lambda K_{x\kappa}$	Longitudinal slip stiffness	x	x	x	x
$\lambda K_{y\alpha}$	Cornering stiffness	x	x	x	x
λC_x	Longitudinal shape factor	x	x	x	x
λC_y	Lateral shape factor	x	x	x	x
λE_x	Longitudinal curvature factor	x	x	x	x
λE_y	Lateral curvature factor	x	x	x	x
λH_x	Longitudinal horizontal shift	x	x	x	x
λH_y	Lateral horizontal shift	x	x	x	x
λV_x	Longitudinal vertical shift	x	x	x	x
λV_y	Lateral vertical shift	x	x	x	x
$\lambda K_{y\gamma}$	Camber force stiffness	x	x	x	x

$\lambda K z\gamma$	Camber torque stiffness	x	x	x	x
λt	Pneumatic trail (effects aligning torque stiffness)	x	x	x	x
λMr	Residual torque	x	x	x	x
Combined Slip			96	02	02Pi 06
$\lambda x\alpha$	Slip angle influence on longitudinal force	x	x	x	x
$\lambda y\kappa$	Slip ratio influence on lateral force	x	x	x	x
$\lambda Vy\kappa$	Slip ratio induced lateral force from ply steer	x	x	x	x
λs	Aligning torque moment arm of longitudinal force	x	x	x	x
Other			96	02	02Pi 06
$\lambda Cz\alpha$	Radial tire stiffness	x	x	x	
$\lambda Mx\kappa$	Overturning couple stiffness	x	x	x	
$\lambda My\kappa$	Rolling resistance moment	x	x	x	

Table 8.9: Pacejka Scaling Coefficients

Additional Information

For additional information regarding tire data and tire model theory, OptimumG recommends the following publications.

- Pacejka, H.B. "The Tyre as a Vehicle Component", Proceedings of XXVI FISITA Congress, ed. M. Apetaur, Prague, 1996.
- Pacejka, H.B. Tire and Vehicle Dynamics (First Edition), Butterworth-Heinemann, Oxford, 2002.
- Pacejka, H.B. Tire and Vehicle Dynamics (Second Edition), SAE, WarrenDale, PA, 2006.
- Blundell, M.V., and Harty, D. The Multibody Systems Approach to Vehicle Dynamics, SAE, WarrenDale, PA, 2004.
- Blundell, M.V., and Harty, D. "Intermediate Tyre Model for Vehicle Handling Simulation", IMechE, Vol 221 Part K: J. Multi-body Dynamics, 2007.
- Gäfvert, M., and Svendenius, J. "Construction of Semi-Empirical Tire Models for Combined Slip", Technical Report ISRN LUTFD2/TFRT- - 7606- -SE, Department of Automatic Control, Lund Institute of Technology, Sweden, 2003.
- Schmeitz, A.J.C., Besselink, I.J.M., Hoogh, J de., and Nijmeijer. H. "Extending the Magic Formula and SWIFT Tyre Models for Inflation Pressure Changes". TU Eindhoven, Eindhoven, TNO Automotive, Helmond, The Netherlands, 2005.