



# GETTING STARTED USING LOTUS ENGINE SIMULATION







**GETTING STARTED**  
**USING**  
**LOTUS ENGINE SIMULATION**

**V**ERSION 5.05

The information in this document is furnished for informational use only, may be revised from time to time, and should not be construed as a commitment by Lotus Cars Ltd or any associated or subsidiary company. Lotus Cars Ltd assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

This document contains proprietary and copyrighted information. Lotus Cars Ltd permits licensees of Lotus Cars Ltd software products to print out or copy this document or portions thereof solely for internal use in connection with the licensed software. No part of this document may be copied for any other purpose or distributed or translated into any other language without the prior written permission of Lotus Cars Ltd.

©2001 by Lotus Cars Ltd. All rights reserved.

## Contents

### About This Guide v

- Welcome to Lotus Engine Simulation v
- What you Need to Know v

### Chapter 1. Introducing Lotus Engine Simulation 1

- Overview 1
- What is Lotus Engine Simulation? 2
- Starting Lotus Engine Simulation 3
- About the Tutorials 5
- Getting Help Online 6

### Chapter 2. Tutorial 1: 'Basic Single-Cylinder Engine Model' 9

- Overview 9
- Starting Lotus Engine Simulation 10
- Adding a Cylinder 11
- Changing the In-Cylinder Heat Transfer Model 12
- Performing Basic Functions in the Builder Environment 13
- Adding Valves 13
- Adding Ports 15
- Adding Inlet and Exit Boundaries 16
- Defining the Fuel and Fuel System 17
- Defining Steady-state Test Condition Data 18
- Setting the Names of the Results Files 19
- Saving a Model 19
- Running a Model 20
- Monitoring the Progress of a Job 21
- Displaying the .MRS Results – Text Viewer 22
- Displaying the .MRS Results – Graphical Viewer 23

### Chapter 3. Tutorial 2: 'Basic Single-Cylinder Engine Model with Intake Pipes' 27

- Overview 27
- Re-Naming a Model 28
- Adding Pipes 28
- Changing Plotting Options 30
- Examining the .MRS Results 31
- Comparing the .MRS Results with Tutorial 1 32
- Displaying the .PRS Results 34
- Animating .PRS Results 36
- Exercise 39

**Chapter 4. Tutorial 3: ‘Parametric Studies Using Basic Single-Cylinder Model with Intake Pipes’ 41**

- Overview 41
- Element Groups 42
- Creating Element Groups 42
- Adding Elements to a Group 42
- Viewing Elements within a Group 43
- Starting the Parametric Tool 44
- Defining a Scoring System 44
- Setting Up Parameters 46
- Launching a 1D Parametric Run 47
- Exercise – 1D Parametric Analysis 49
- Running a 2D Parametric Analysis 50
- Running the Optimizer 52

**Chapter 5. Tutorial 4: ‘Building a Multi-Cylinder Engine Model’ 57**

- Overview 57
- Adding the Exhaust Pipes 58
- Copying Model Segments 59
- Creating the 4-2-1 Exhaust Manifold 61
- Setting the Firing Order 64
- Running the Multi Cylinder Model 65
- Using the Pressure Loss Junction 66
- Running the Pressure Loss Junction Model 69

**Chapter 6. Tutorial 5: ‘Using the Concept Tool’ 71**

- Overview 71
- Opening the Concept Tool 72
- Entering the Minimum Data Requirements 74
- Setting Data Values Directly 74
- Editing the User Defined Rules 75
- Changing the Warning Limits 76
- Selecting Inlet and Exhaust Options 77
- Creating the Simulation Model from the Concept Tool 78
- Concept Tool Test Points 79
- Running the Concept Tool Model 79

**Chapter 7. Tutorial 6: ‘Using Sensors and Actuators’ 81**

- Overview 81
- Loading the Model 82
- The Importance of Groups 82
- Adding Sensors and Actuators to a Model 84
- Using Control Elements 86
- Running the VVT Model 88
- Running the Model with Fixed Geometry 89
- Exercise – Adding an Exhaust cam Phaser 89

## **Chapter 8. Tutorial 7: 'Using the Load Finder' 91**

- Overview 91
- Loading the Model 92
- 'Load Finder' Objectives 92
- Adding the Throttle to a New Group 93
- Setting Up the 'Load Finder' Cases 93
- Monitoring the 'Load Finder' 96
- Running the 'Load Finder' Job 98
- Reviewing the 'load Finder' Results 99
- Exercise – Virtual Links 99

## **Chapter 9. Tutorial 8: 'Multi-Cylinder Turbocharged Model' 101**

- Overview 101
- Loading the Original Model 102
- Adding the Turbocharger 102
- Modifying the Turbocharger Data 104
- Monitoring the Turbocharger Speed 106
- Running the Turbocharged model 107
- Viewing the Results 109
- Exercise 112

## **Chapter 10. Tutorial 9: 'Turbocharged Model with a Waste-gate and Charge-cooler' 113**

- Overview 113
- Loading the Original Model 114
- Adding the Waste-gate 114
- Adding Control Elements to the Waste-gate 115
- Running the Turbocharged Model with a Waste-gate 118
- Viewing the Results 119
- Waste-gate for Variable Boost Pressure 120
- Damping the Waste-gate Motion 124
- Exercise 125

## **Chapter 11. Tutorial 10: 'Modelling a Multi-Cylinder Turbocharged Diesel Engine' 127**

- Overview 127
- Description of the Model 128
- Running the Model 133
- Discussion of Results 134
- Exercise 135

## **Chapter 12. Tutorial 11: 'Modelling the Transient Response of a Turbocharged Diesel Engine' 137**

- Overview 137
- Setting Up the Part-load Condition 138
- Entering Mechanical Property Data and Attaching a Shaft 139
- Defining a Transient Event 141
- Running the Model 144
- Discussion of Results 145
- Exercise 145

**Chapter 13. Tutorial 12: ‘A Two-Stroke Engine Model’ 149**

- Overview 149
- Building the Model 150
- Creating Test Points 154
- Running the Model 156
- Exercise 156

**Chapter 14. Tutorial 13: ‘Co-simulation with Simulink’ 159**

- Overview 159
- Building the Simulink Model 160
- Linking the LES model to Simulink 165
- Registering the LES Solver 168
- Running the Simulink Model 169



## About This Guide

### Welcome to Lotus Engine Simulation

Welcome to Lotus Engine Simulation. Using Lotus Engine Simulation, you can predict the performance of an engine.

### What You Need to Know

This guide assumes the following:

- Lotus Engine Simulation is installed on your computer or network and you have permission to execute the Lotus Engine Simulation module.
- The necessary password files are installed to allow you to run the necessary modules.
- You have a basic understanding of engine performance simulation and combustion and the influence of hardware changes on engine performance.



# 1

## Introducing Lotus Engine Simulation

### Overview

This chapter introduces you to Lotus Engine Simulation and explains the benefits of the code. It also explains how you can learn more about Lotus Engine Simulation and introduces tutorials that will help you to get started.

This chapter contains the following sections:

- What is Lotus Engine Simulation?, 2
- Starting Lotus Engine Simulation, 3
- About the Tutorials, 5
- Getting Help Online, 6

## What is Lotus Engine Simulation?

LOTUS ENGINE SIMULATION is a simulation program capable of predicting the complete performance of an engine system. The program can be used to calculate:

- **the full- and part-load performance of the engine under steady-state and transient operating conditions;**
- **in-cylinder heat transfer data;**
- **instantaneous gas property variations within the engine manifolds;**
- **turbocharger and supercharger matching conditions.**

LOTUS ENGINE SIMULATION is designed to run on a desktop PC with Windows NT/98/2000. The user interface is based on the standard LOTUS ENGINEERING software 'look-and-feel' and offers the same intuitive approach as other popular Windows applications.

Using the simulation program typically follows the procedure below,

- The user constructing the simulation model enters the engine and manifold specifications. This includes data for:
  - engine bore, stroke and connecting rod dimensions;
  - compression ratio;
  - valve sizes and valve event timing data;
  - intake and exhaust port flow data;
  - intake and exhaust manifold dimensions;
  - maps defining the performance of turbines and compressors;
  - engine operating speed;
  - heat release data' characterising the combustion event;
  - air / fuel ratio and inlet air temperature and pressure.
- Having defined the engine specification data and test condition data the cycle simulation is performed. The user can monitor the progress of the simulation using the Job Status screen.
- Calculation results for cycle-averaged data, such as volumetric efficiency, BSFC, torque, and power, and intra-cycle data, such as pressure, temperature, and mass flow rate are available to the user in the form of a report quality summary sheet and through quick-to-use graph plotting systems.

LOTUS ENGINE SIMULATION has been applied extensively by world-wide clients and validated thoroughly at LOTUS over a wide range of engine types. The program is capable of simulating most existing and projected engine systems and is continually updated by LOTUS in co-operation with it's partners.

## Starting Lotus Engine Simulation

The three basic steps necessary to create and run a LOTUS ENGINE SIMULATION model are:

- Step 1 - generate a model through either the 'drag-and-drop' network builder, loading and editing data from an existing model, or using the integrated 'Concept' Tool.
- Step 2 - define the required engine operating parameters for the test.
- Step 3 - launch the simulation solver.
- Step 4 - load the simulation results as either textual or graphical displays to review the calculated values.

### Start-up Wizard

When the application is opened, the first dialog box displayed is the Startup Wizard, shown below in Fig. 1.1.

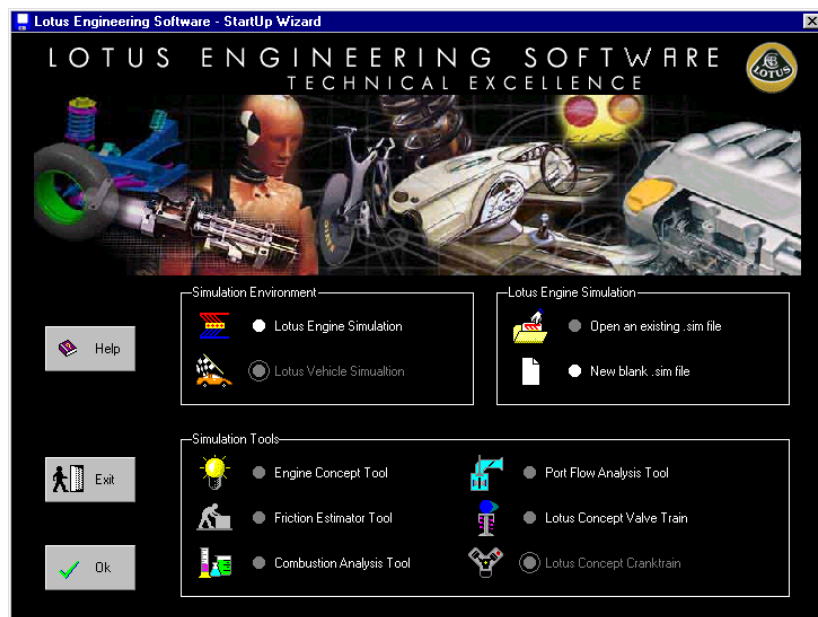


Figure 1.1. Startup wizard

This window consists of three main panels, each panel containing a number of options. The wizard provides access the Lotus Engine Simulation and Lotus Vehicle Simulation environments and a number of other Lotus Engineering **Simulation Tools**. It should be noted that only the codes for which the user is licensed can be selected. An item 'greyed out' indicates that that program or tool is not currently licensed.

The **Simulation environment** panel allows the user to chose between opening the main application at either the Engine Simulation or the Vehicle Simulation environment level. When an option is selected (identified by the toggle being shown in white) the second panel will display file options relevant to that environment level. If you are not licensed for either Engine Simulation or Vehicle Simulation the second panel will not be visible.

For the Engine Simulation environment there are two file options available in the second panel, **(Lotus Engine Simulation)**:

The first option is '**Open an Existing .sim File**'. This option should be selected if the user has previously saved a Simulation model and wishes to load it into the interface. If this option has been chosen then the standard Windows file browser will be displayed and will use the \*.sim filter, allowing the user to select the correct data file.

The second option is '**New Blank .sim File**' and should be selected if no simulation models have yet been constructed or if the user wishes to create a new model.

The **Engine Concept Tool**, the **Friction Estimator Tool**, the **Combustion Analysis Tool**, the **Port Flow Analysis Tool**, and **Lotus Concept Valve Train** can all be used in conjunction with LOTUS ENGINE SIMULATION. These tools analyse and prepare data in a form which is compatible with the input requirements of LOTUS ENGINE SIMULATION.

## About the Tutorials

It is assumed that the user will work through the tutorials in sequential order. Therefore more guidance is given in the early tutorials and less as you proceed through the guide

**Note:** If you choose not to work through the tutorials in sequential order you may have to refer to earlier tutorials for guidance on some of the basic actions.

The following tutorials are included in the guide:

- Tutorial 1 – ‘Basic Single-Cylinder Engine Model’
- Tutorial 2 – ‘Basic Single-Cylinder Engine Model with Intake Pipes’
- Tutorial 3 – ‘Parametric Studies Using Basic Single-Cylinder Model with Intake Pipes’
- Tutorial 4 – ‘Building a Multi-Cylinder Model’
- Tutorial 5 – ‘Using the Concept Tool’
- Tutorial 6 – ‘Using Sensors and Actuators’
- Tutorial 7 – ‘Using the Load Finder’
- Tutorial 8 – ‘Multi-Cylinder Turbocharged Model’
- Tutorial 9 – ‘Turbocharged Model with a Waste-gate and Charge-cooler’
- Tutorial 10 – ‘Modelling a Multi-Cylinder Turbocharged Diesel Engine’
- Tutorial 11 – ‘Modelling the Transient Response of a Turbocharged Diesel Engine’
- Tutorial 12 – ‘A Two-Stroke Engine Model’
- Tutorial 13 – ‘Co-simulation with Simulink’

## Getting Help Online

When working in Lotus Engine Simulation, you can get help in several ways, as follows:

- Displaying Bubble Help
- Using Status Bar Messages
- Accessing the Online Documentation
- Displaying Information About Lotus Engine Simulation

### Displaying Bubble Help

The bubble help gives a brief description of a particular icon or buttons function. Rest the cursor over the desired icon to view the bubble help message. To turn bubble help 'off', from the **Help** menu select **Display Bubble Help**. Changes in the visibility of the bubble help only take effect the next time the program is run.

### Using status Bar Messages

The bubble help messages are also displayed in the first status bar pane at the lower left of the screen. They are displayed irrespective of the visibility setting of the bubble help. The other panes in the status bar are used for displaying data values at relevant times.



## Accessing the Online Documentation

The online help documentation can be accessed from the **Help** menu, select **Contents** to open the help file at the contents page, (see Figure 1.1)

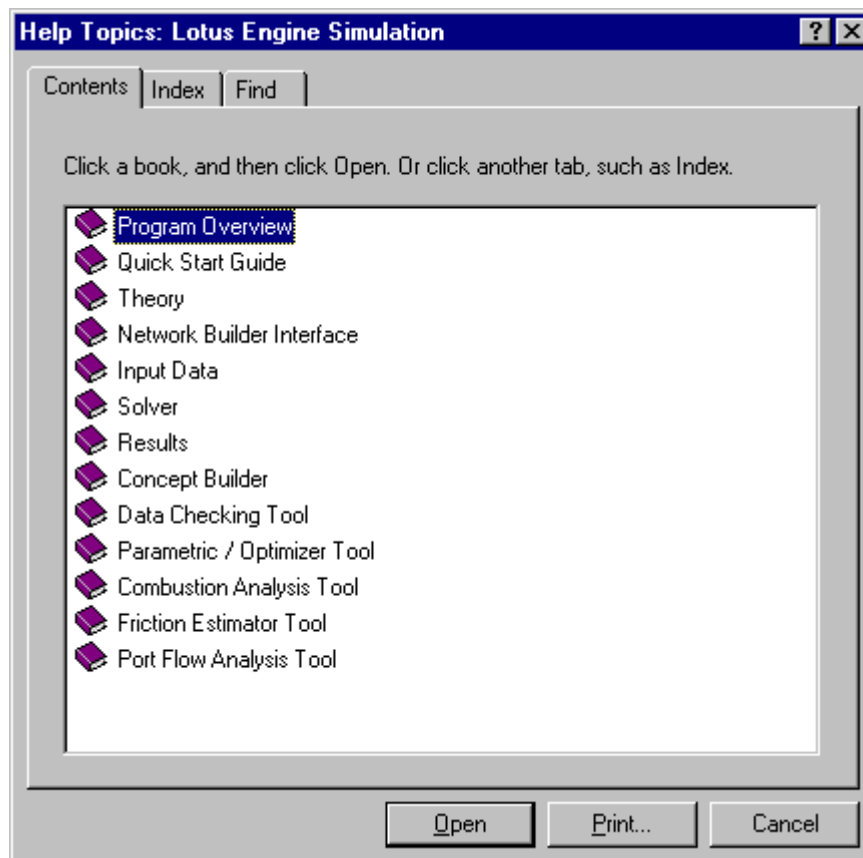


Figure 1.2. On-line help contents page

To access the most relevant page based on the current section selected, from the **Help** menu select **Help on Lotus Engine Simulation...**

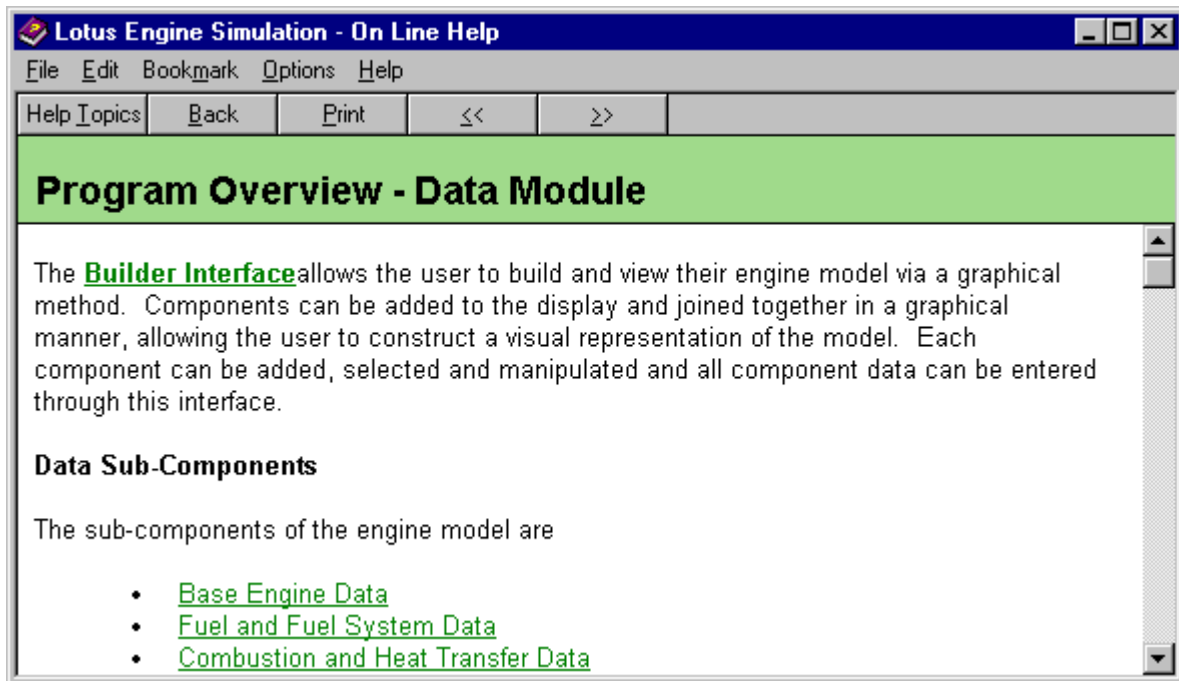


Figure 1.3. On-line help, context sensitive pages

You can also access the help file directly at a relevant page whenever you see the help icon.



## Displaying Information About Lotus Engine Simulation

To display information about Lotus Engine simulation:

From the **Help** menu, select **About Lotus Engine Simulation**.

## Overview

This tutorial shows the user how to create a single-cylinder engine model, run the model, and display the answers. The tutorial also introduces the user to some basic operations within Lotus Engine Simulation, such as checking a model for connectivity errors and describes various elements of the Lotus Engine Simulation interface.

This chapter contains the following sections:

- Starting Lotus Engine Simulation, 10
- Adding a Cylinder, 11
- Changing the In-Cylinder Heat Transfer Coefficients, 12
- Performing Basic Functions in the Builder Environment, 13
- Adding Valves, 13
- Adding Ports, 15
- Adding Inlet and Exit Boundaries, 16
- Defining the Fuel and Fuel System, 17
- Defining Steady-state Test Condition Data, 18
- Setting the Names of the Results Files, 19
- Saving a Model, 19
- Running a Model, 20
- Monitoring the Progress of a Job, 21
- Displaying the .MRS Results – Text Viewer, 22
- Displaying the .MRS Results – Graphical Viewer, 23

## Starting LOTUS ENGINE SIMULATION

From the start menu, point to **Programs**, point to **Lotus Engineering Software**, and then select **Simulation Tools**. The start-up 'splash screen' is then displayed for a few seconds before the start-up wizard appears, as shown below in Fig. 2.1.

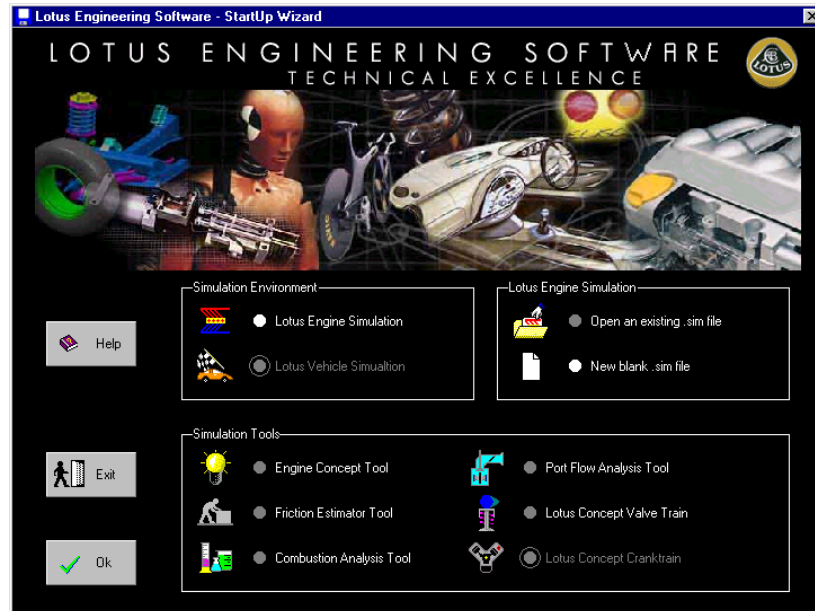


Figure 2.1. Start-up wizard

From the start-up wizard select **Lotus Engine Simulation** from the **Simulation Environment** panel and select **Open blank .sim file**. Then select **OK**.

Before proceeding further in this tutorial please ensure that the **Show Connectivity Errors** option has been selected from the **View** menu at the top of the interface window as shown in Figure 2.2.

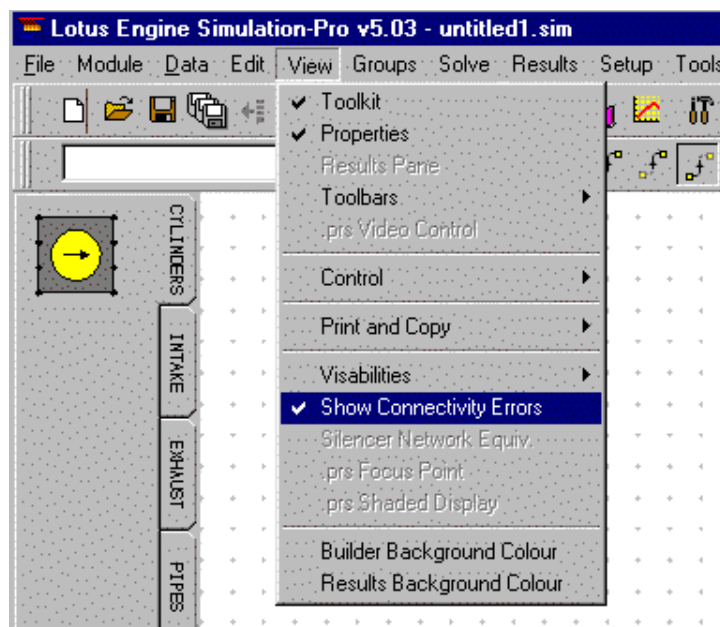


Figure 2.2. Turning on the display connectivity errors option.

This will cause elements which do not have their connectivity's sufficiently well defined to appear in red.

## Adding a Cylinder

Select the **Cylinders** tool kit tab from the left-hand side of the window (the partial screen shot shown below shows the Cylinder Tool Kit). Using the left mouse button drag and drop a cylinder on to the central region of the window known as the **Builder Area**. The cylinder background will turn red once it is placed in the builder area as shown in Fig. 2.3.

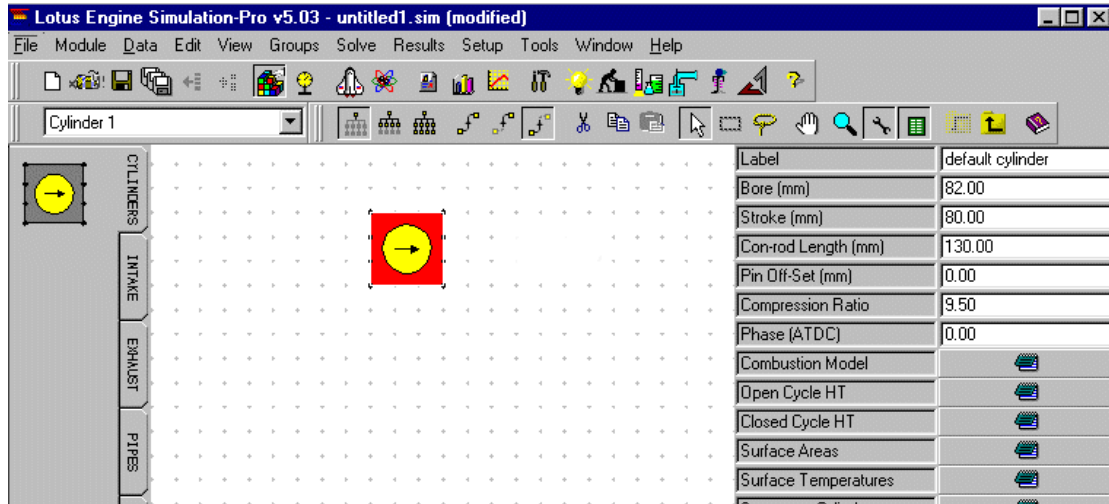


Figure 2.3. The builder area

In the cylinder **Property Sheet** on the right-hand side of the screen change the cylinder **Bore** to **87 mm** and the stroke to **84 mm**.

Also change the **Compression Ratio** to 11.0.

Now click on the **Surface Areas** option on this property sheet. In the window which is generated select the **User Defined Values for Surface Areas** option. This will change the appearance of the window to that shown below. This screen enables the user to define the increase in the surface area over the bore cross-sectional area of the cylinder head and piston. It also enables the user to define the 'bump clearance' of the engine (i.e. the exposed cylinder liner length above the crown when the piston is at top-dead-centre). The last number defines the number of segments in to which the cylinder liner can be sub-divided in order to discretize liner heat transfer data. Set the **Head/Bore (ratio)** to 1.1, the **Piston/Bore (ratio)** to 1.05 and the **Exp. Liner** (length above piston crown at tdc) to 1 mm. All these parameters are used in the heat transfer calculations.

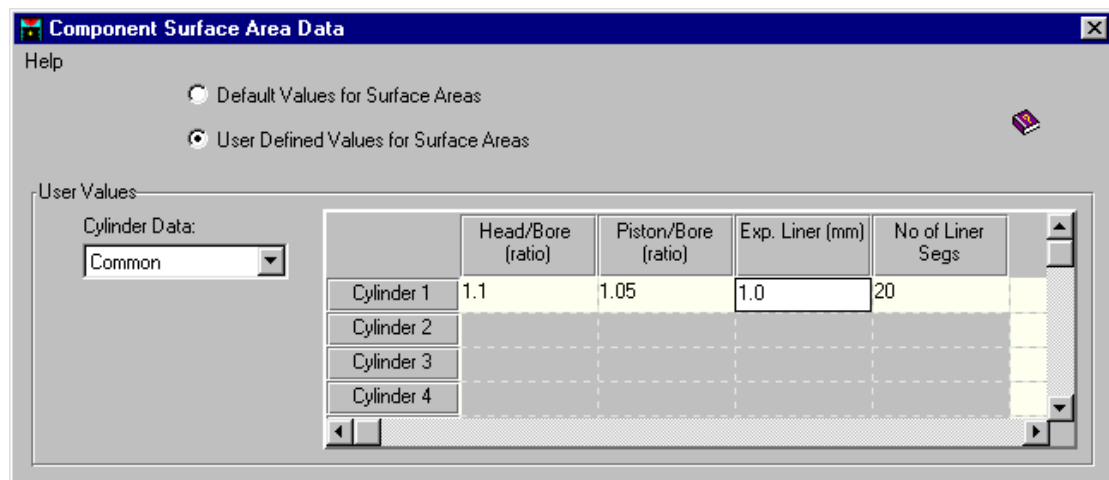


Figure 2.4. Editing component surface areas.

Select **Common** from the **Cylinder Data** box on the left-hand side of the window – this allocates the properties just defined to any other cylinders which exist at the time of editing the data, or are

subsequently created. Close the window by clicking on the cross in the top right-hand corner of the window.

### Changing the In-Cylinder Heat Transfer Model

LOTUS ENGINE SIMULATION provides a choice of three in-cylinder heat transfer models:

1. the Annand model;
2. the Woschni model;
3. the Eichelberg model.

These models characterise the convective and radiative heat transfer coefficients in the cylinder – the Annand model is selected as the default model. Different coefficients are used in the open and closed periods of the cylinder due to the different gas compositions and temperatures, and the different fluid motion regimes which prevail during these periods. Click on the **Closed Period** option in the cylinder property list. This will produce the window shown in Fig. 2.5. Click in the **User** bullet box and change the Annand **A** coefficient to 0.20, ensure the **B** coefficient is set to 0.8 and **C** to  $4.29\text{e-}9$ .

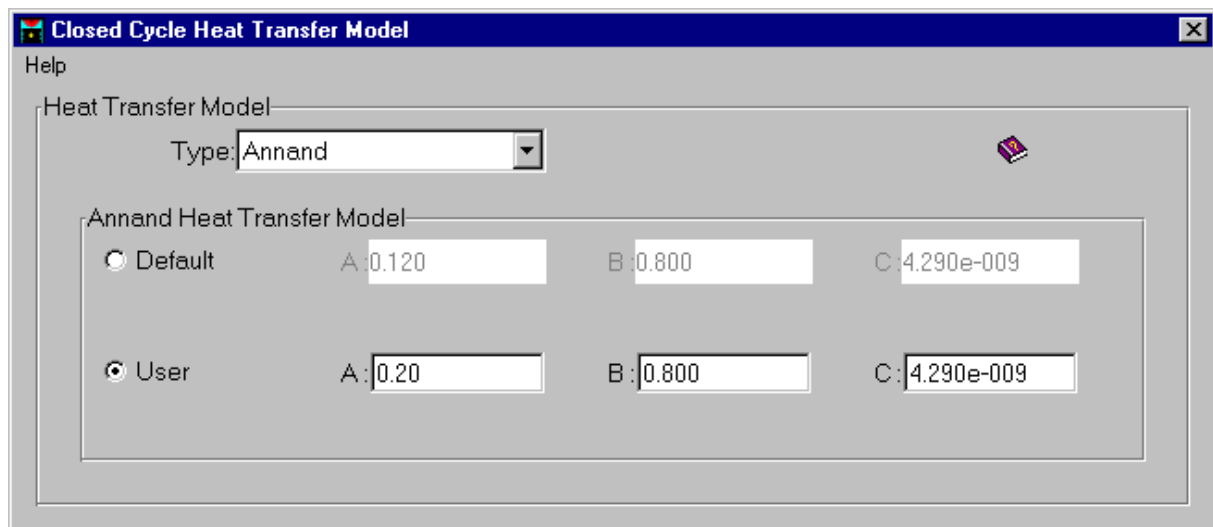


Figure. 2.5. Defining the closed cycle heat transfer model.

Use the **Help File** to discover the definitions of A, B and C coefficients in the Annand model. Click on the Help File icon shown below and click on the Theory link which is highlighted in green. Scroll down this window until you reach the section which defines the Annand equation.



Close the Help File window when you have read the section describing the Annand equation.

Close the Closed Cycle Heat Transfer Model window.

## Performing Basic Functions in the Builder Environment

Now that an element is present in the builder environment it is appropriate to explore some of the basic viewing functions which can be performed. The viewable area can be automatically scaled, so that all of the elements present in the builder environment are visible, by pressing the **Control-A** keys. Alternatively, pressing the right-mouse button in the builder environment when no element is in focus produces a menu from which the **Auto-scale** option can be selected. Performing this **auto-scaling** task should show the cylinder which was added and the green Fuel Element. The Fuel Element enables the user to define the fuel type and fuel system specification as described below.

Clicking on the magnify glass icon on the top tool bar enables the element to be zoomed in or out by moving the magnifying glass up or down respectively.



Particular areas of the model can be zoomed in on so that the selected region fills the viewable area by clicking on the right mouse button with no element in focus and selecting the **Zoom** option. This produces a 'cross-hair' which can be placed with the left-mouse button on the required point and dragged to encapsulate the region which is to be zoomed. The hand icon enables the model to be moved around within the viewable area.

## Adding Valves

Click on the **Intake Components** tab on the Tool Kit menu. Select the element at the top of this list – this is a poppet valve. Attach the valve to one of the connectors on the cylinder element as shown in Fig. 2.6.

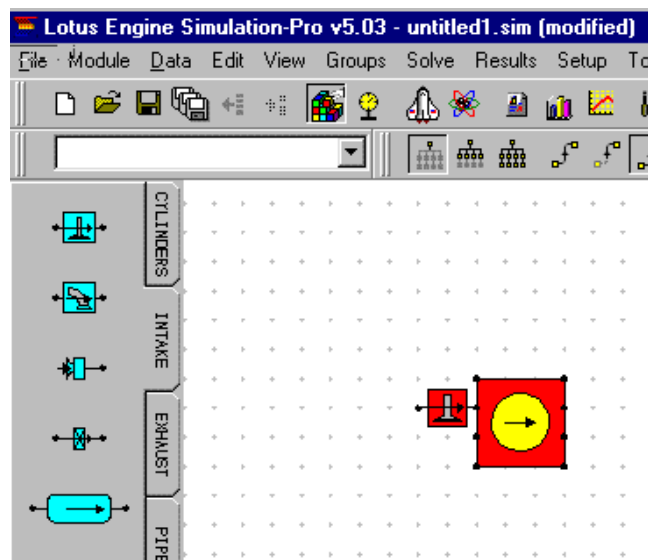


Figure. 2.6. Adding an intake valve.

The timing of this valve is to be changed. We shall make use of the **Help File** in order to discover the convention for setting the valve timings.

Using the information contained in the Help File set the intake valve opening (**Valve Open (deg)**) and closing (**Valve Close (deg)**) points to 15 degrees before top-dead-centre and sixty degrees after bottom-dead-centre, respectively. Having set these values, click on the **Valve Timing Display** button to give a visual check on the data – the window should appear as below. The timings can also be entered from the Valve Timing Display window.

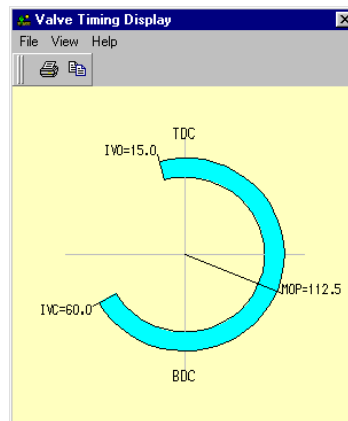


Figure 2.7. Intake valve timing.

Click on the **Exhaust Components** tab on the Tool Kit menu. Select the poppet valve element at the top of this list. Attach the valve to one of the connectors on the opposite side of the cylinder element to the intake poppet valve. You will notice that the background colour of the **Cylinder** element has changed to grey, indicating that the cylinder is now a fully connected element within the model.

Modify the exhaust valve timing to 40 degrees before bottom-dead-centre and 20 degrees after top-dead-centre for the opening and closing points, respectively. Clicking on the **Valve Timing Display** button now produces a diagram such as that shown in Fig. 2.8.

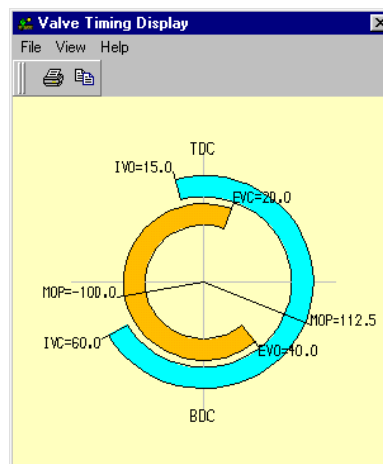


Figure 2.8. Intake and exhaust valve timing.

Note that it is conventional to define the Maximum Opening Point (MOP) as positive and negative for conventional inlet and exhaust valve timing, respectively, for a four-stroke engine. This is the point of maximum valve lift.

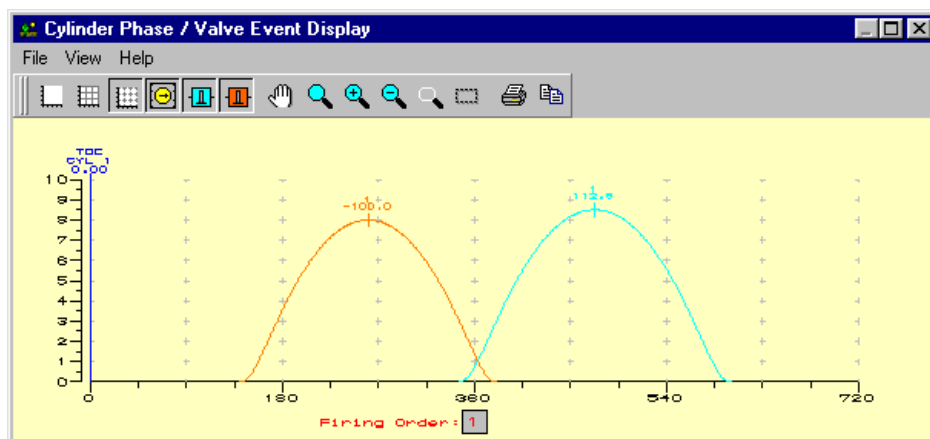


Figure 2.9. Intake and exhaust valve lift profiles.



A further visual check that the valve timing data is as intended is to click on the **Valve Event Display** button which should give the window shown below:

In this window the exhaust valve lift profile is shown in orange and that for the inlet valve is shown in blue.

## Adding Ports

Now select the exhaust **Port icon** from the Exhaust Tool Kit menu and connect it to the exhaust valve. Repeat the process for an inlet port. The use of separate valve and port combinations enables the user to try different cam and port combinations. The model should now appear as indicated below:

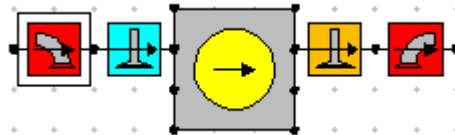


Figure 2.10. Connecting ports to poppet valves.

The click on the intake port icon and change the **Valve Throat Diameter** to 28. This number represents the diameter of each individual valve (port) throat in the region of the valve seat. Note that the number of valves specified (on the intake side of the cylinder) is 2. This means that it is not necessary to model the individual port duct leading to each valve. With basic four-valve engines this does not introduce any significant modelling inaccuracy if the individual ducts after the bifurcation are of equal length. The simulation will treat the port as a single duct with an equivalent diameter equal to the total of the two individual ducts. Note that the length of the port cannot be modelled using the port data – the port flow data simply constitutes a map of data which defines the size of the valve orifice at a given lift.

Default Port Flow		
File		
No of Values:10		
	L/D Ratio	Flow Coeff.
1	0	0
2	0.05	0.14
3	0.1	0.27
4	0.15	0.395
5	0.2	0.51
6	0.25	0.605
7	0.3	0.67
8	0.35	0.71
9	0.4	0.72
10	1	0.72
11		
12		

Figure 2.11. Port flow data – default good port.

Select the **Port Data** button from the property sheets. This reveals a table of port flow coefficients vs. valve lift / port throat diameter data ( $C_f$  vs. L/D – see Fig. 2.11) which can be viewed by clicking on the graph icon in the top right of the window.

The View / Autoscale command should be invoked if the graph cannot be seen or is not correctly scaled. Figure 2.12 shows how the data should appear. The data for this table is obtained via steady flow tests on the engine to be simulated or from an engine with similar port geometric details.

The port icons can only be attached to poppet valves.

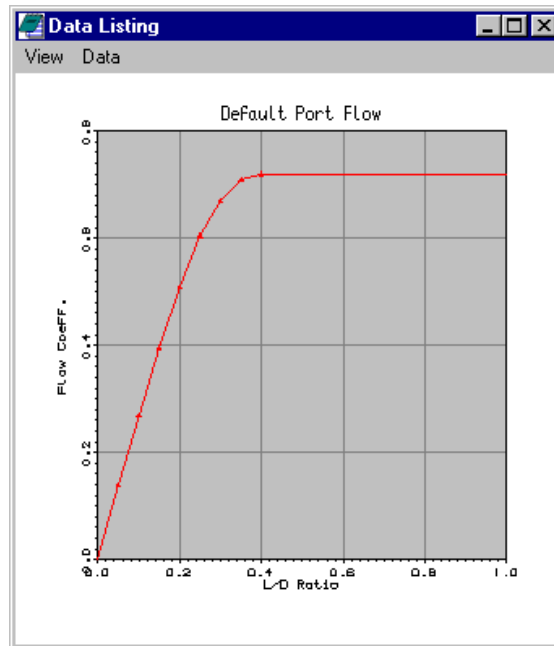


Figure 2.12. Variation of flow coefficient with valve lift / diameter ratio.

After closing the plot window shown in Figure 2.12 it is worth taking note of the menus which define the upstream and downstream elements which can be connected to the element in focus. Clicking on the inlet valve reveals the elements shown in the bottom right-hand corner of the interface screen, shown in Figure 2.13. This indicates that the upstream elements which can be connected to an inlet poppet valve are an inlet port and a virtual pipe (see later), whilst the downstream elements which can be connected to an inlet poppet valve are a cylinder and a virtual pipe.

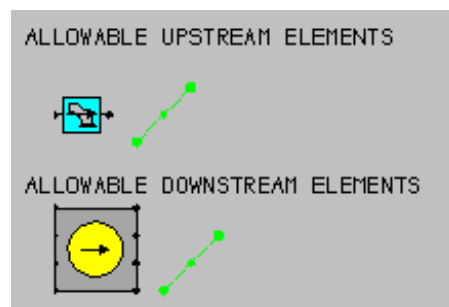


Figure 2.13. Connectivity permissions.

## Adding Inlet and Exit Boundaries

This extremely simple model can be completed by adding inlet and exit boundary conditions to the intake and exhaust ports. To do this select the intake tools tab and drag-and-drop an Inlet Element on to the builder area – attach this element to the intake port. Click on to the exhaust tool tab and attach an Exit Element to the exhaust port. The model should now appear as shown in Figure 2.14.

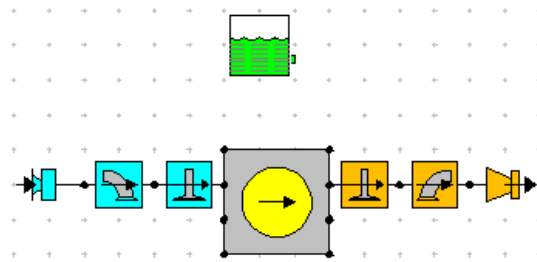


Figure 2.14. Complete model.

The model is now physically complete and it only remains to define the fuel and fuel system data operating conditions at which the engine simulation model is to run. The inlet and exit boundary conditions are defined when the Test Condition Data is set up (see below).

## Defining the Fuel and Fuel System

The fuel type and properties are defined by clicking on the Fuel Element shown in Figure 2.14. The property sheet associated with the fuel element is shown in Figure 2.15.

Label	
Fuel System	Port Injection
Fuel Type	Gasoline
Calorific Value (kJ/kg)	43000.0
Density (kg/litre)	0.7500
H/C Ratio Fuel (molar)	1.800
O/C Ratio Fuel (molar)	0.000
Molecular Mass (kg/k.mol)	114.230
Maldistribution Factor	1.000

Figure 2.15. Property sheet for fuel element.

Various default fuel types and fuel systems are available. The Help File lists the options and describes the property data. In this example we will use the default Gasoline as the **Fuel Type** and select Port Injection as the **Fuel System**. The port-injection option implies that the air and fuel enter the cylinder pre-mixed in the ratio specified in the Test Conditions Data described below.

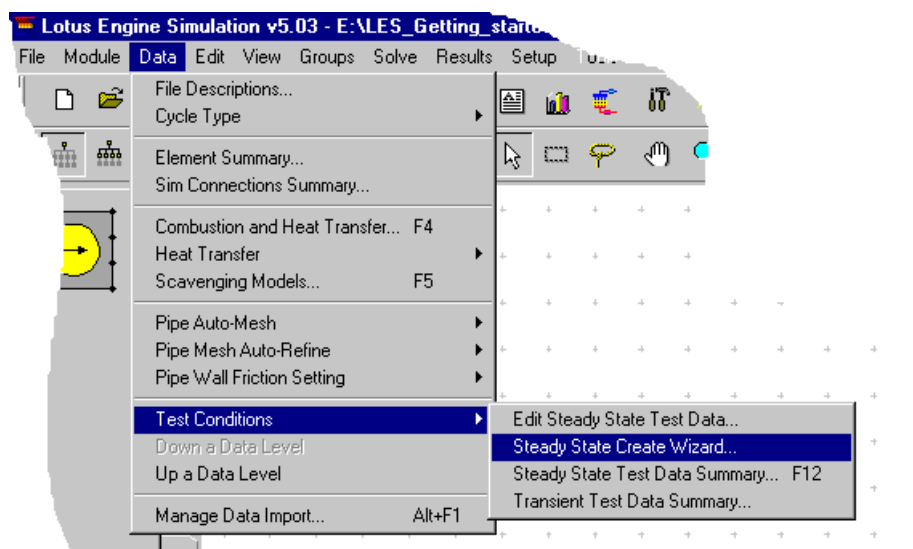


Figure 2.16. Accessing the Test Conditions Data Wizard.

## Defining Steady State Test Condition Data

The engine operating points that the simulation model is required to mimic can be set up most efficiently using the **Test Conditions Wizard**. This wizard is accessed via the **Data / Test Conditions / Create Wizard** option from the drop-down menu at the top of the screen.

Clicking on the Create Wizard option generates the input data Wizard shown in Figure 2.17.

**Test Data Wizard**

Select Definition Type

☒ By No. of Tests    ☐ By Speed Increment

No. of Tests: 7

Min. Speed (rpm): 1000.0000

Max. Speed (rpm): 7000.0000

Speed Increment (rpm): 1000.0000

Ambient Air Pressure (bar abs.): 1.0000

Ambient Air Temperature (C): 20.0000

Inlet Pressure (bar abs.): 1.0000

Inlet Temperature (C): 20.0000

Exit Pressure (bar abs.): 1.0000

Equivalence Ratio: 1.100

Specific Humidity (kg/kg): 0.0130

Options

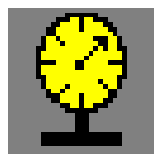
☐ Interpolate Existing

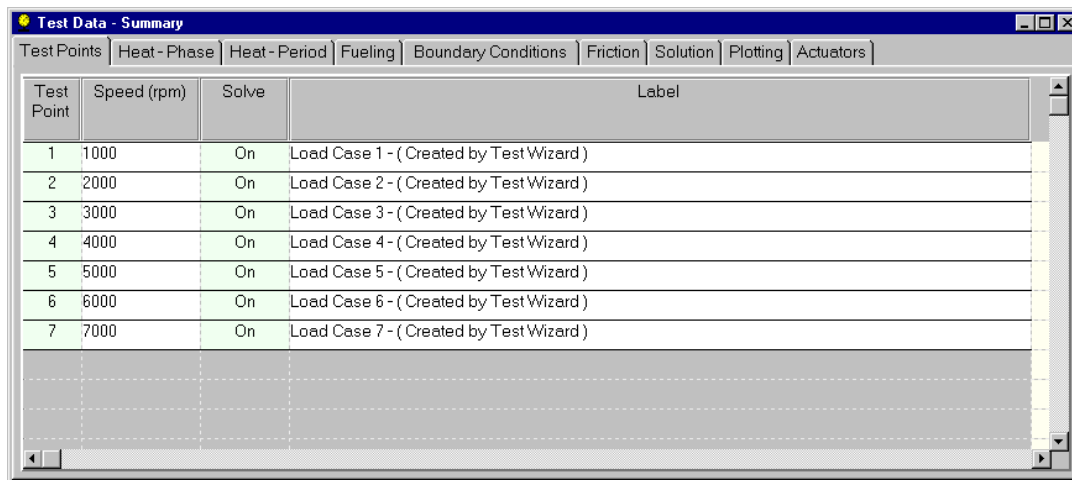
Apply    Cancel

Figure 2.17. Create Wizard for Test Condition Data

Set the desired engine speed range by changing the **No. of Tests** to 7 and the **Max. Speed (rpm)** to 7000. Also set the **Exit Pressure** (exit boundary pressure) to 1 (bar abs.) and the **Equivalence Ratio** to 1.1. Click **Apply**.

The test conditions which have been generated can be viewed or edited from the window produced by clicking on the Test Data – Summary icon shown below.





Test Point	Speed (rpm)	Solve	Label
1	1000	On	Load Case 1 - ( Created by Test Wizard )
2	2000	On	Load Case 2 - ( Created by Test Wizard )
3	3000	On	Load Case 3 - ( Created by Test Wizard )
4	4000	On	Load Case 4 - ( Created by Test Wizard )
5	5000	On	Load Case 5 - ( Created by Test Wizard )
6	6000	On	Load Case 6 - ( Created by Test Wizard )
7	7000	On	Load Case 7 - ( Created by Test Wizard )

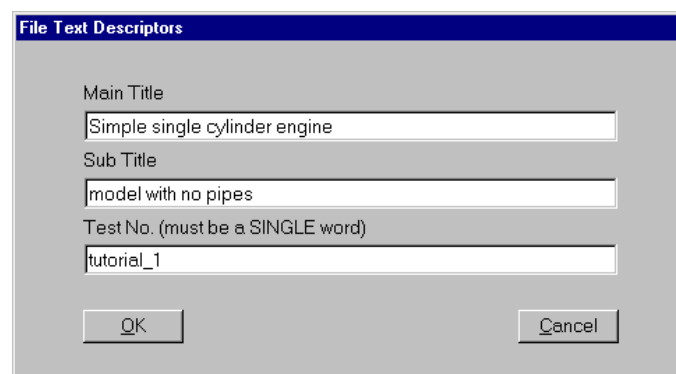
Figure 2.18. Test Data Summary window.

The top tabs can be used to move through the different data sections that have been created. The wizard has generated and defined some data which is not controlled by the selections made in the wizard. This additional data includes 'heat release' parameters which, for the gasoline engine we have generated, consists of defining the point of 50% mass fraction burned and the 10%-90% mass fraction burned period. A default relationship for calculating the engine friction has been selected. This is necessary in order for LOTUS ENGINE SIMULATION to predict 'brake' output parameters (torque, power, specific fuel consumption, BMEP). Note that the default models (in this case the Barnes-Moss model) estimate only the mechanical friction of the engine – the simulation code predicts the pumping work.

Close this window after checking that the test conditions have been generated as intended.

## Setting the Names of the Results Files

In order to save the simulation results to unique file names click on the **File Descriptions** section of the **Data** menu list shown in Figure 2.16. This generates the window shown in Figure 2.19. Enter the text shown.



File Text Descriptors	
Main Title	Simple single cylinder engine
Sub Title	model with no pipes
Test No. (must be a SINGLE word)	tutorial_1
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Figure 2.19. File Descriptions window.

The **Main Title** and **Sub-Title** names appear at the top of the results text file. The **Test No.** string which is entered is used to name the results files produced by the code. Click **OK** to close this window.

## Saving a Model

This is probably an appropriate juncture at which to save the model. Click on the Save Data to File icon and save the file to the directory of your choice as **tutorial\_1.sim** (note that the .sim extension will be added automatically on saving).

## Running a Model

The model can be run by clicking on the LOTUS ENGINE SIMULATION solver control icon shown below



This produces the window shown in Figure 2.20. Enter a text string in the **Label** box. This will allocate the text string entered to the job name displayed in the Job Status window (accessed from the tab shown in the **Lotus Engine Simulation Solver** window). Select **Currently Held Data** – this will run the model which has just been built in the interface. Select **Use testno** – this fills the two text boxes below the widget with the text string entered in the Test No. text box shown in Figure 2.19.

Click on the launch icon at the bottom of the window to submit the job.

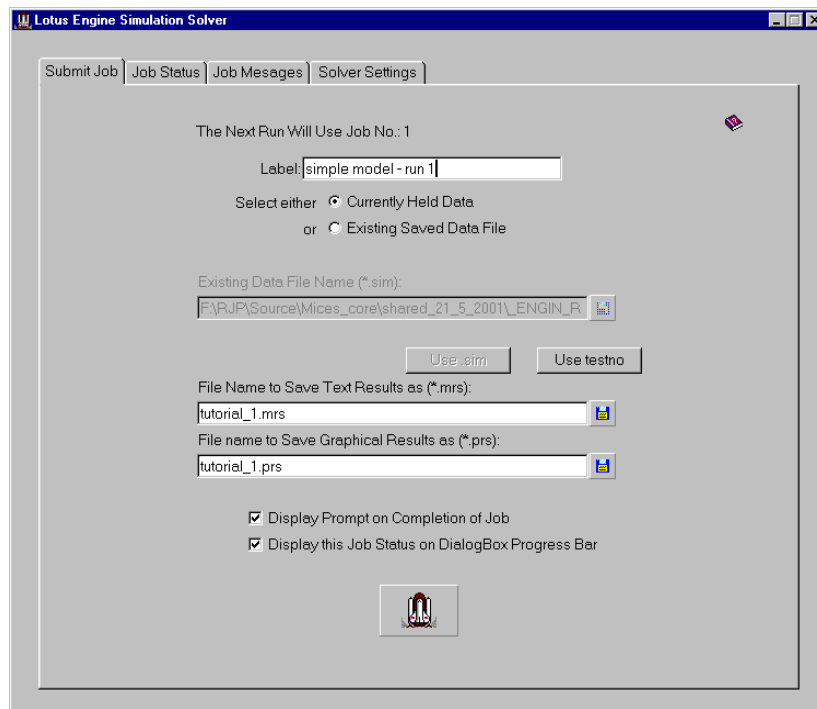


Figure 2.20. The Lotus Engine Simulation Solver window

Submitting the job will open the *solver runtime window* shown in Figure 2.21. Minimise this window by clicking on the minimise tab. **Do not close the window as this will stop the job from running.**

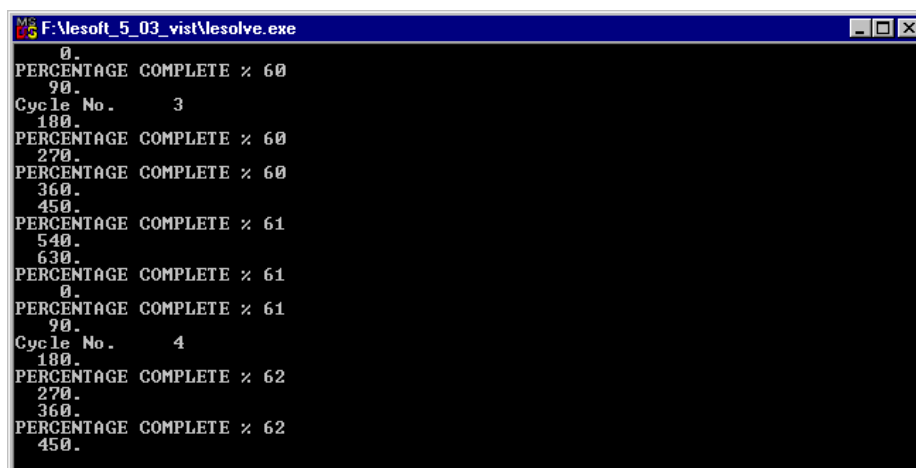


Figure 2.21. Solver runtime window produced on submission of job.

## Monitoring the Progress of a Job

Click on the **Job Status** tab on the LOTUS ENGINE SIMULATION *solver runtime* window. This will produce a window as shown in Figure 2.22.

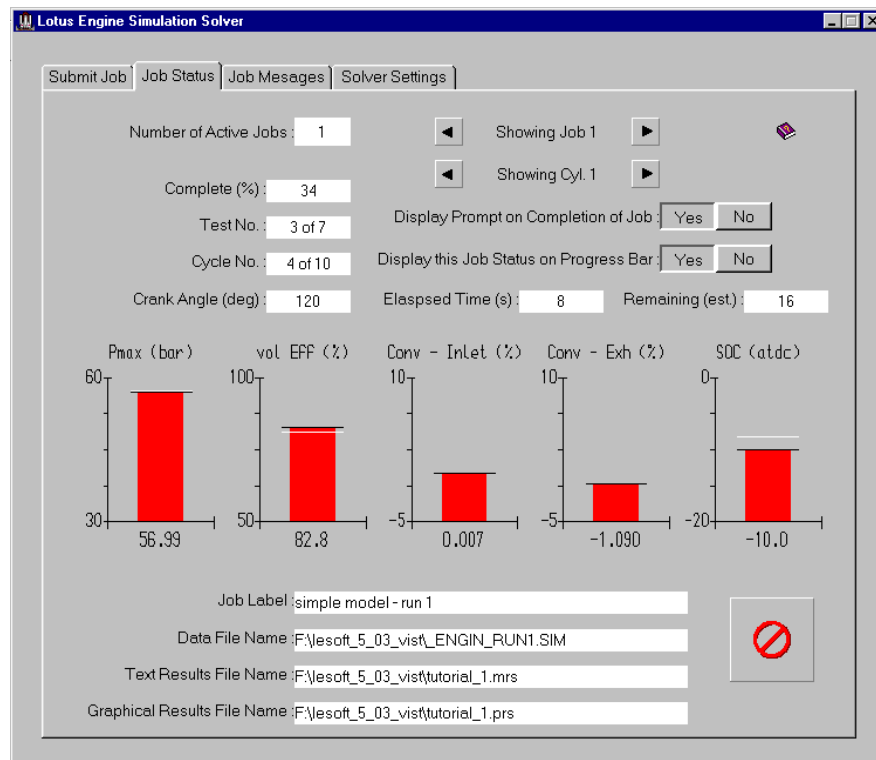


Figure 2.22. Job Status window.

The Job Status window provides a means of monitoring the progress of a simulation job. The top text box indicates the **Number of Active Jobs** being run by LOTUS ENGINE SIMULATION. The arrow tabs to the right of this box enable the user to display information regarding the different jobs being run (in this case only one job has been submitted).

Various measures of the progress of the job are displayed, including the **Test No.**, **Cycle No.**, and the **Crank Angle** position within the cycle. The **Elapsed Time** and an estimate of the **Remaining** time are also displayed.

The five red bars which appear after the first cycle simulated are cycle-averaged parameters from which the user can obtain an impression of the viability of the run. Eg. The maximum cylinder pressure (**Pmax**) and volumetric efficiency (**Vol Eff**) values are updated every engine cycle and indicate whether the run is progressing in a satisfactory manner.

At the end of the run the window shown in Figure 2.23 will appear on the screen and the Job Status window will display the 'Job Complete or Not Current' message. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. Tick the boxes to **Load Text Results (.mrs)** and **Load Graphics Results (.mrs)**. This places the .MRS results file, which contains all the relevant cycle-averaged data into the dedicated text reader and graphical facilities. The .PRS results will be discussed in a later tutorial – leave this box un-ticked. Click **Load**. Now close the **Lotus Engineering Simulation Solver** window shown in Figure 2.22.

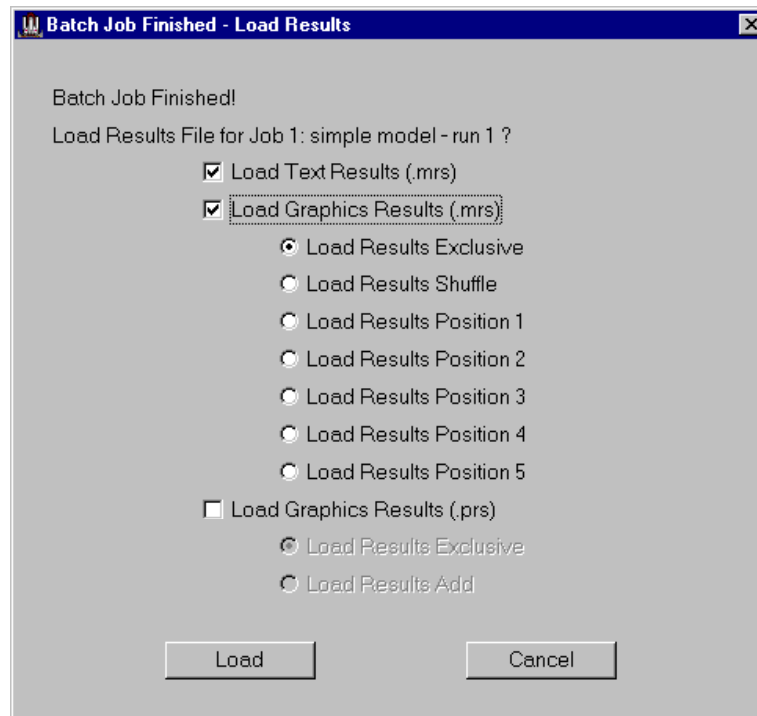


Figure 2.23. Batch Job Finished window.

### Displaying the .MRS Results – Text Viewer

The .MRS file contains all the cycle-averaged results from the engine simulation, such as brake torque, brake power, volumetric efficiency, BMEP, BSFC, and heat rejection data. This file can be displayed in a dedicated text file viewer by clicking on the icon shown below on the top tool bar.



The data at the top of the .MRS file summarises the input data – this information can be used by the user in order to identify the results and to ensure that the input data has been submitted as intended. At the end of the input data summary section the results for each Test Point, or operating condition, are given. For each test point the operating conditions and boundary conditions are printed first. Detailed information regarding mass flow rates through the valves of each cylinder (only a single-cylinder is present in this case) is then provided, including the volumetric efficiency of each cylinder<sup>1</sup>. The trapped conditions within each cylinder are then given, followed by a 'Performance Section' where the IMEP, FMEP, BMEP, and power values are given for each cylinder. Fuel consumption and efficiency values for each cylinder are then provided, before a Performance Summary table is written to the file. In addition, bulk heat transfer data for each cylinder is provided. The Miscellaneous Data and Results section contains component resolved heat transfer data for the cylinders and the heat transfer distribution down pipes (there are no pipes in the model considered in this tutorial). Combustion data is also written to the file in this section. This is followed by a listing of the mass flow convergence results (see a later tutorial) and a summary table of the total engine performance parameters. The remaining portion of the file consists of a copy of the input data file, which may, if desired, be re-loaded. A sample of the .MRS text file is shown in Figure 2.24.

<sup>1</sup> Volumetric efficiency is defined as the ratio of the mass of air and fuel trapped in the cylinder when the intake valve closes to the mass of air and fuel which could be trapped in the cylinder at a reference temperature and pressure. The reference condition is usually taken as the ambient conditions for naturally aspirated engines. Intake plenum conditions are sometimes used for pressure charged engines.



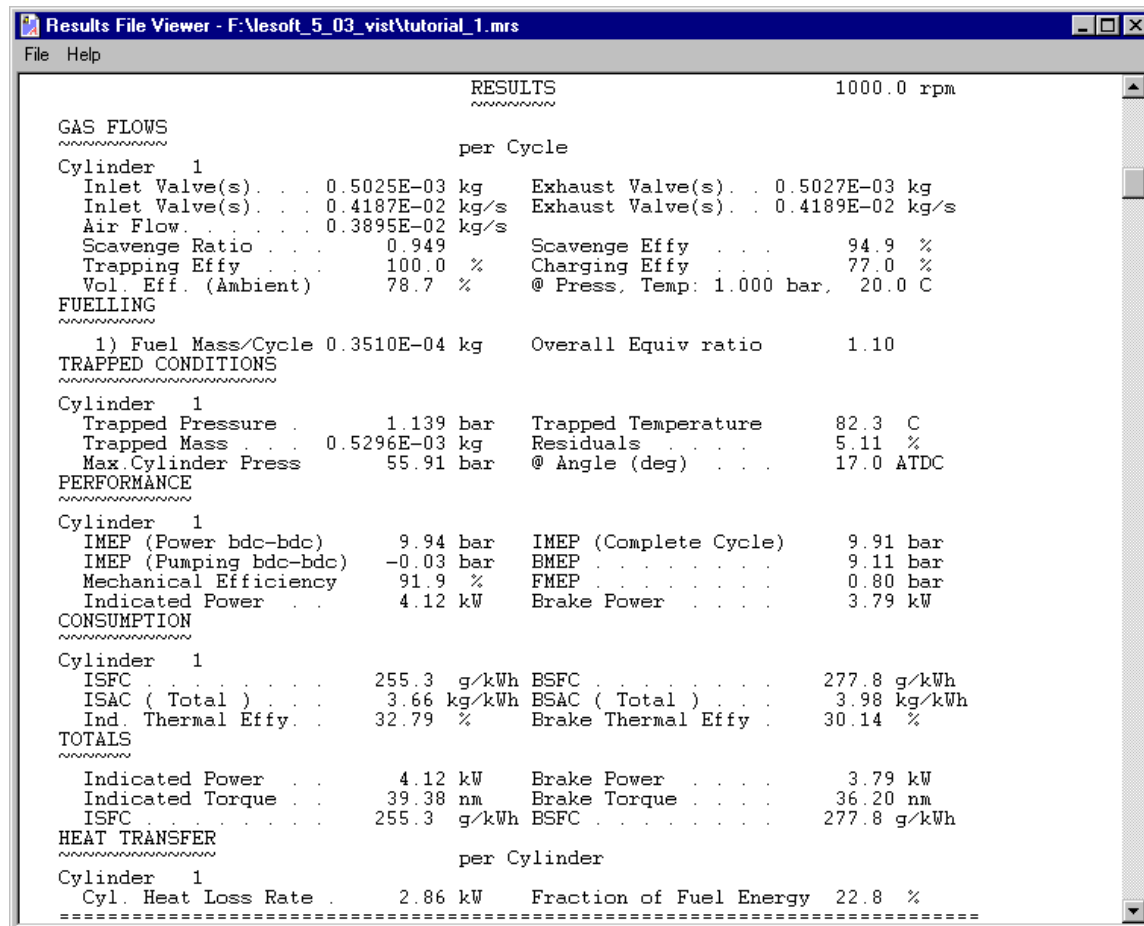


Figure 2.24. Screen shot of the .MRS file.

Take some time to familiarise yourself with the information contained in this file.

Note that the .MRS file is in ASCII format and can be read into any text editor such as WordPad.

## Displaying the .MRS Results – Graphical Viewer

A dedicated graphical viewer is provided in LOTUS ENGINE SIMULATION in order display the .MRS results. The .MRS file can be viewed by clicking on the icon shown below.



This produces the window shown in Figure 2.25. Select the **Autoscale** option from the **View** menu within this window.

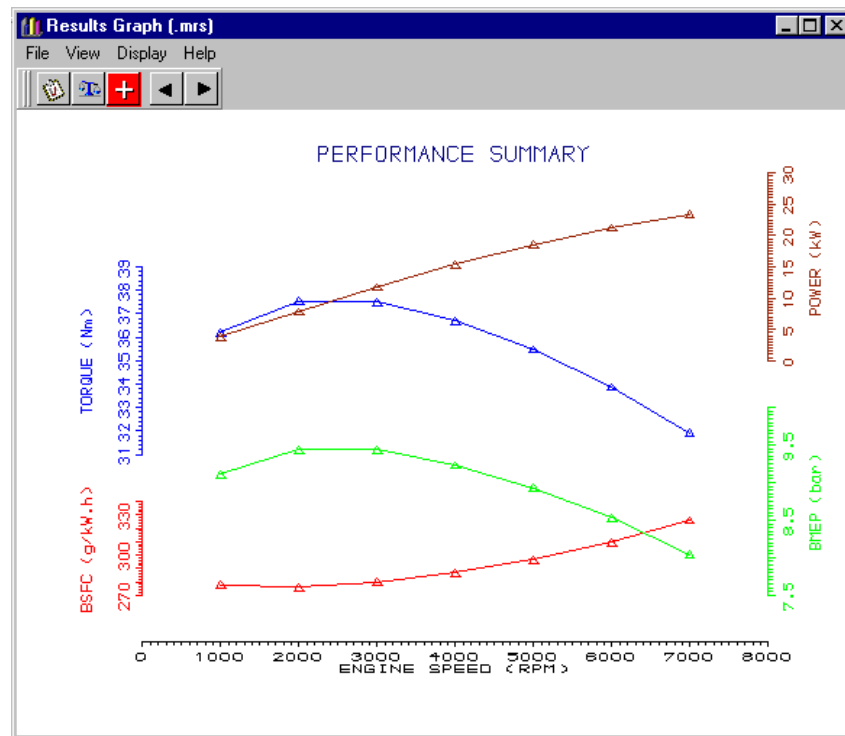


Figure 2.25. .MRS Results Graphical viewer.

The first plot which can be viewed (see Figure 2.25) shows the BSFC (brake specific fuel consumption), BMEP (brake mean effective pressure), Torque, and Power produced by the engine. It can be seen that there is a gently increasing power output from the engine as the speed rises. The torque curve peaks at about 2000 rev/min. This performance characteristic is typical of an engine with no pipes.

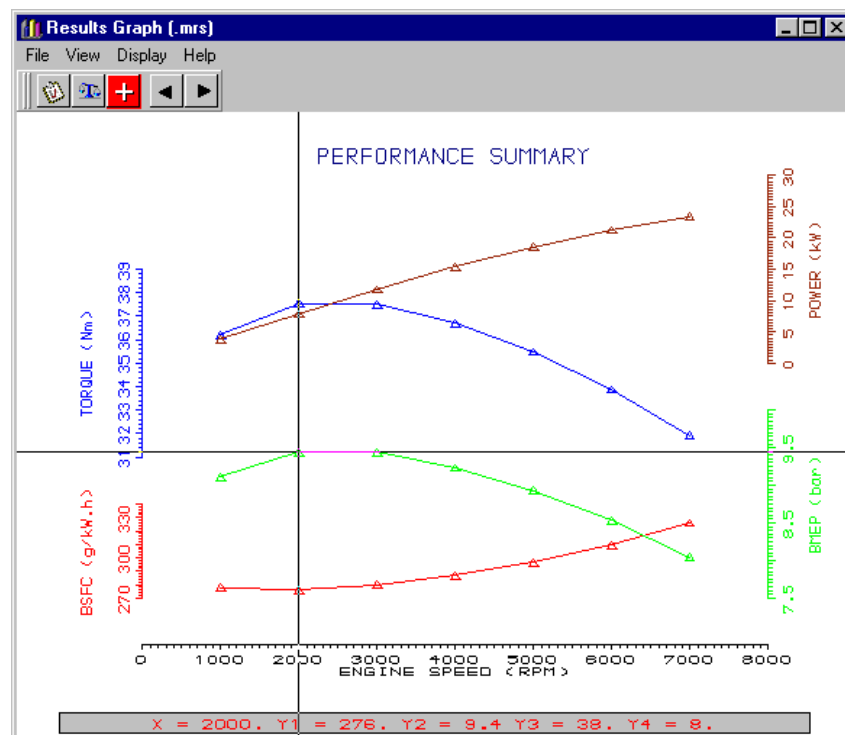


Figure 2.26. Listing a point on the .MRS graph.

Now click on the List Point option from the View menu on the .MRS plotting window. This produces a 'cross-hair' which can be placed over any point on the window to interrogate a precise value from the graphs as shown in Figure 22.26. Clicking the left-mouse button causes the values to appear. Clicking the right-mouse button releases the cross-hair. The values appear at the bottom of the window. In the case shown in Figure 22.26 the X-axis is the engine speed (the point selected is 2000 rev/min) and the Y-axis numbers relate to the four graphs shown in order of ascending height on the page: Y1 = BSFC; Y2 = BMEP; Y3 = Torque; Y4 = Power.

Confirm that the BMEP value at 2000 rev/min is 9.4 bar.

Remember to click the right-mouse button to release the cross-hair before moving to the next 'page' in the .MRS graphical viewer screen. The arrow buttons in the top left-hand corner of the screen can be used to move through the different data screens which can be displayed.

Click on the **right-arrow** to display the second screen. Click on **Autoscale** from the **View** menu. This screen shows the variation of inlet boundary pressure, volumetric efficiency, inlet boundary temperature, and plenum pressure with engine speed. In this model there is no plenum. Clicking on the right-arrow reveals the next screen which shows the same data for the exhaust system. The next screen shows the variation of combustion related parameters with engine speed, including the angles of 10%, 50%, and 90% mass fraction burned, and the maximum cylinder pressure.

This concludes the first tutorial.



# 3

## Tutorial 2. 'Basic Single-Cylinder Engine Model With Intake Pipes'

### Overview

This tutorial shows the user how to add pipes to the basic single-cylinder model developed in Tutorial 1. The .MRS results are compared with those from Tutorial 1 in order to highlight the dramatic effect which the intake pipe has on the engine performance.

Displaying the .PRS results is described and the user is taken through the procedure for generating animations using these results.

The chapter contains the following sections:

- Re-Naming a Model, 28
- Adding Pipes, 28
- Changing Plotting Options, 30
- Examining the .MRS Results, 31
- Comparing the .MRS Results with the Tutorial 1, 32
- Displaying the .PRS results, 34
- Animating .PRS Results, 36
- Exercise, 39

## Re-Naming a Model

With the model **tutorial\_1.sim** loaded into LOTUS ENGINE SIMULATION click on the **Save As** option from the **File** menu and create a new model called **tutorial\_2**.

## Adding Pipes

We wish to insert pipes between the port icon and the inlet boundary icon. In order to be able to detach a single element from other elements the **Move Element Singularly** icon on the left-hand side of the group of icons shown below should be depressed.



Click on the **Zoom** icon and zoom out a little. Click on the inlet element and move it to the left of the builder environment as shown in Figure 3.1.

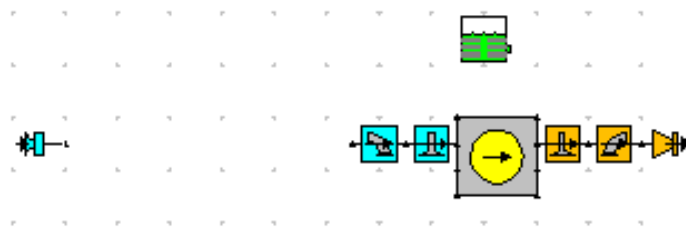


Figure 3.1. Detaching the inlet element.

Now click on the Pipes tab on the Tool-Kit. Click on a (black) pipe from the top of the menu and drop, by clicking the left-mouse button, the left-hand end about  $\frac{1}{4}$  of the way between the port icon and the inlet element. Then attach the right-hand end of the pipe to the port icon by clicking again the left-mouse button. Pressing Control-A will bring the model into a fuller view now. The model should appear as shown in Figure 3.2. Note that the green pipes on the pipe tool kit tab are virtual pipes which only imply commonality of connection points which they link together.

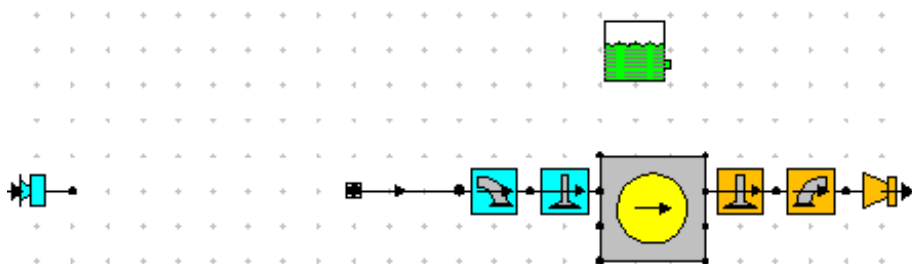


Figure 3.2. Model with pipe attached to port icon.

The intention is to modify the properties of this pipe so that it represents the physical extent of the port. Click on the pipe and change the **Total Length** in the pipe property sheet to 100 mm. Note that the **No. of Meshes** has been set to 7 by the automatic meshing option which is turned on by default. Now change the **Start Diameter** and **End Diameter** to 39 mm. Change the **Wall Thickness** to 4 mm and the **Cooling Type** to water cooled. You will notice that the **Temperature** and **Ext. HTC** values have been changed to 99.85 deg. C and 5000 W m<sup>-2</sup> K<sup>-1</sup>. The water cooled pipe option assumes that the pipe is entirely surrounded by cooling water and these values define the temperature and convective heat transfer coefficient for the coolant. These values can also be defined by the user if they differ from the defaults.

Note that the **Wall Material** is set to aluminium – this fixes the Thermal Conductivity of the pipe wall which is used in the heat transfer calculation in the pipes.

A further feature to note on the pipe property sheet is that the **Wall Factor Type** is set as Default Surface Roughness – this implies that a default surface roughness value has been assigned to the pipe wall which is then used by the code to infer a pipe wall friction factor from a curve fit to the Moody diagram (which portrays pipe wall friction factor against Reynolds number as a function of pipe wall surface roughness). Click on the cactus icon to display a schematic of the pipe which has been created (see Figure 3.3).

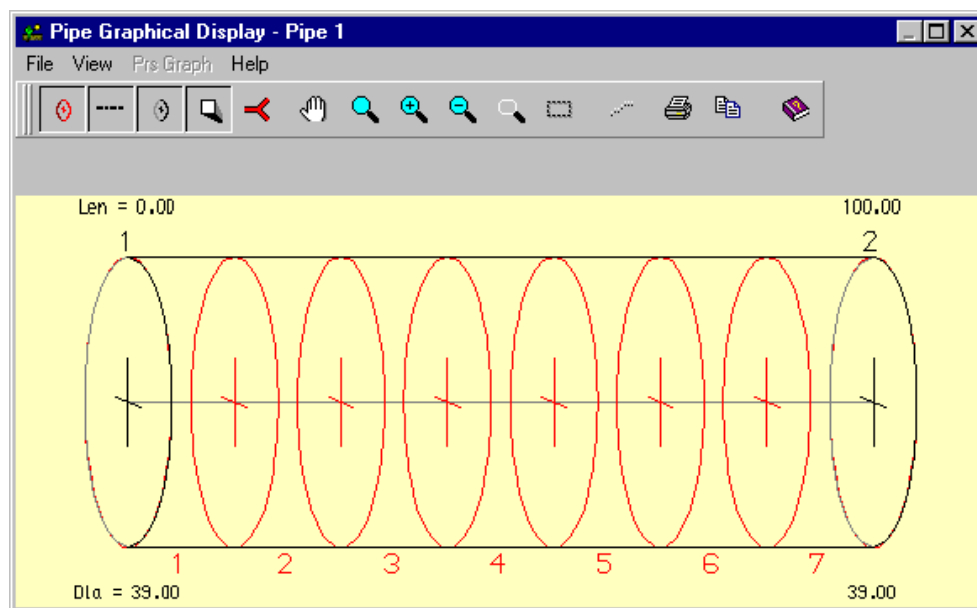


Figure 3.3. Schematic diagram of pipe.

The red circles represent pipe mesh-points. The black circles indicate points at which the pipe diameter values have been defined.

Close this window.

We now wish to add a pipe to represent the intake runner of the engine which extends from the cylinder head outwards. This pipe will be surrounded by ambient air.

Repeat the procedure described above for inserting a pipe on to the builder interface.

Click on the pipe and change the **Total Length** in the pipe property sheet to 350 mm. Now change the **Start Diameter** and **End Diameter** to 39 mm. We will accept the default wall thickness of the pipe and the cooling type.

The model should now appear as shown in Figure 3.4.

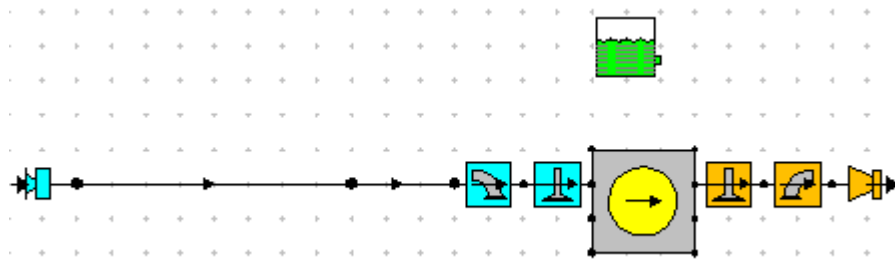


Figure 3.4. Single-cylinder model with port and inlet runner pipe.

### Changing Plotting Options.

Before running the model we will select an output data option which will allow us to look at the instantaneous variation of gas properties in the inlet system.

Select the **Data / Test Conditions / Edit Steady State Test Data Summary** option from the drop down menu at the top of the screen. Click on the **Plotting** tab in the screen shown in Figure 3.5

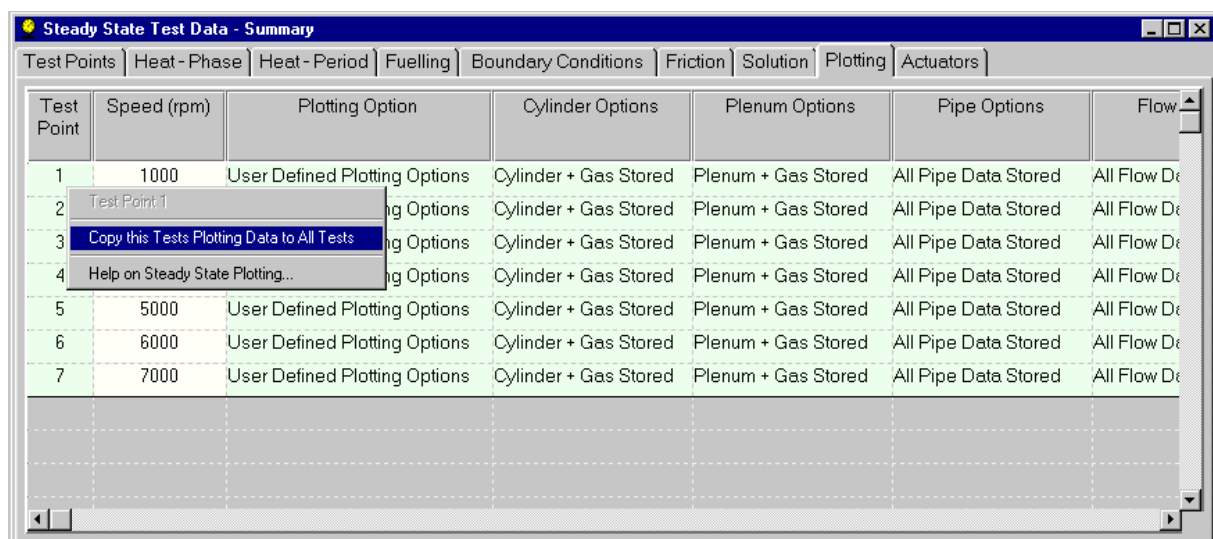


Figure 3.5. Plotting options window in Test Conditions Data window.

Select **User Defined Plotting Options** from the *Plotting Option* selection box and choose **All pipe data stored** from the *Pipe Options* selection box. Then click on the **Copy this Tests Plotting Data to All Tests** button at the top right-hand side of the plotting window. This will affect the amount of data which is written to the **.PRS** plotting files which contain all the data which may vary on an intra-cycle basis.

Close the Test Conditions Data window

Select the **File Descriptions** Option from the **Data** menu and change the text in the **Test No.** box to tutorial\_2. Click **OK**.

**Now Save the model.**

Run the model by following the same procedure described in Chapter 2. Click on the **Use Test No.** option in the Lotus Engine Simulation Solver window in order to define the file names for the **.MRS** and **.PRS** files and submit the job.

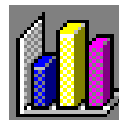


Minimise the solver runtime window produced and click on the **Job Status** tab on the Lotus Engine Simulation Solver window in order to monitor the progress of the simulation.

At the end of the run the window shown in Figure 2.23 (Chapter 2) will appear on the screen and the Job Status window will display the 'Job Complete or Not Current' message. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. Tick the boxes to **Load Text Results (.mrs)**, **Load Graphics Results (.mrs)**, and **Load Graphics Results (.prs)**. This places the **.MRS** results file, which contains all the relevant cycle-averaged data, and the **.PRS** file, which contains all the intra-cycle data, into the dedicated text reader and graphical facilities. Click **Load**. Now close the **Lotus Engineering Simulation Solver** window shown in Figure 2.22.

## Examining the Results

Click on the .MRS file viewer icon (shown below) and **Autoscale** the plots (**View / Autoscale**):



This will produce the results shown in Figure 3.6.

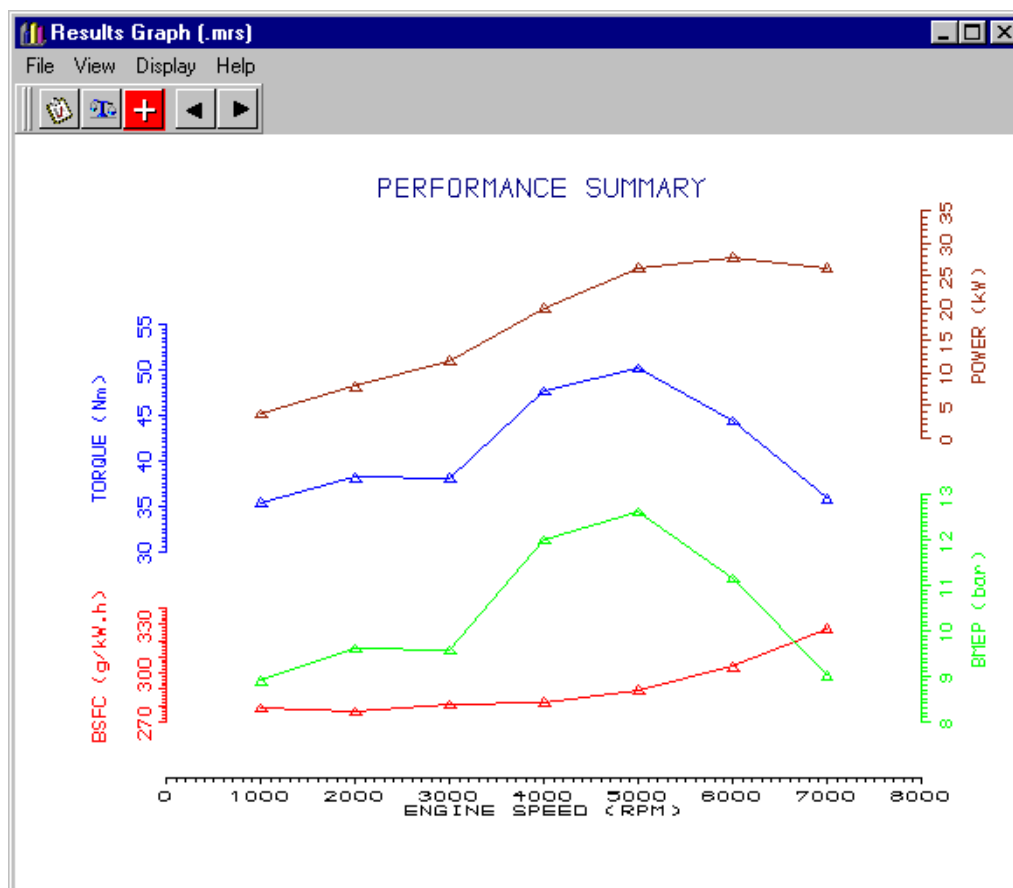


Figure 3.6. Results for model with inlet pipe.

Note that there is now a pronounced peak in the torque curve at 5000 rev/min. This is the effect of the intake system tuning phenomenon which increases the pressure across the intake valve at engine speeds which excite the gas in the inlet system at its resonant frequency.

### Comparing the Results with Tutorial 1

It is instructive to compare the results predicted by the code for this model with those produced in the previous tutorial. In order to do this click on the **File / Load Results (shuffle)** option from the top menu and select **tutorial\_1** from the browser. Then click on **Open**. This will place both sets of results on the same set of graphs as shown in Figure 3.7.

The large improvement in the performance of the engine at high engine speeds is immediately apparent. The peak BMEP value for the model with no pipes is 9.4 bar at 2000 / 3000 rev/min whereas the peak value for the model with the inlet runner pipe is 12.6 bar at 5000 rev/min – a 34 percent improvement!

The actual data values for the two sets of results can be accessed from the performance summary from the text file viewer or by listing lines from the .MRS file viewer. Before using this latter approach it is prudent to change the number of decimal points listed in the power data. This is done by clicking on the **Axis Scales** option from the **View** menu as shown in Figure 3.8. Type a 1 in the **Y-Axis 4** box. This will give the power values to one decimal place. Click **Refresh** and close the window.

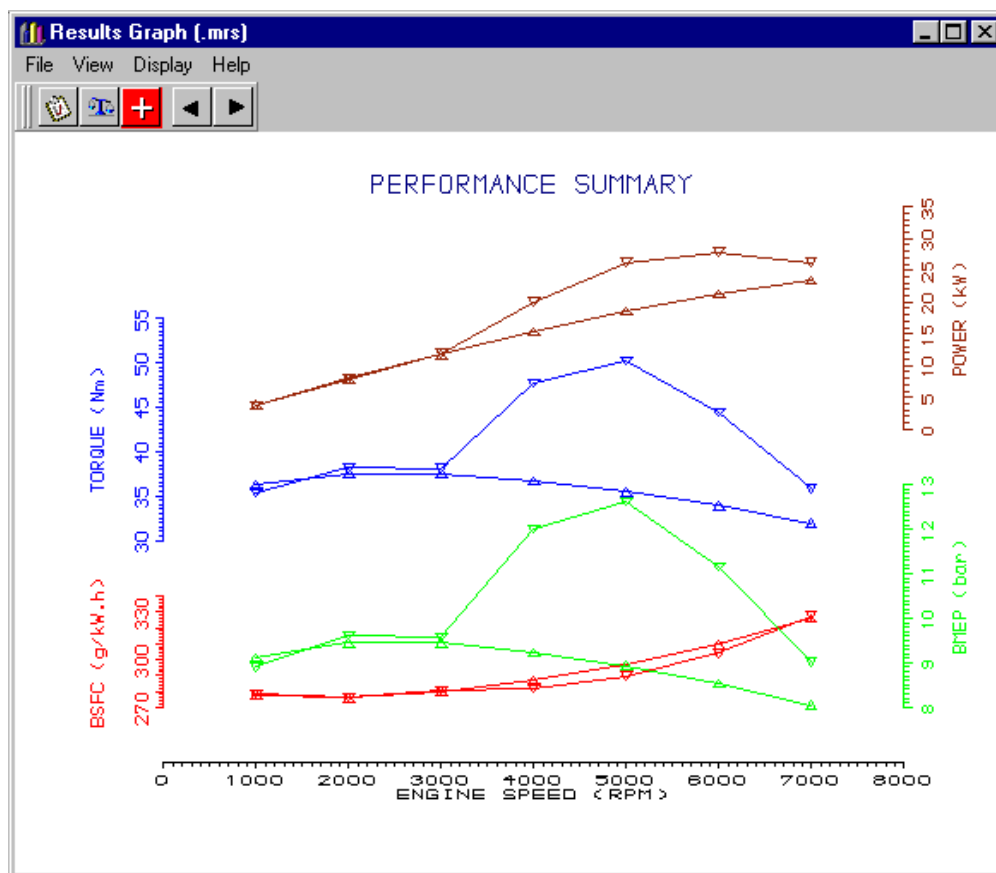


Figure 3.7. Comparison of .MRS results.

	X-Axis:	Y-Axis 1:	Y-Axis 2:	Y-Axis 3:	Y-Axis 4:
Axis Min:	0.000	270.000	8.000	30.000	0.000
Axis Max:	8000.000	340.000	13.000	55.000	35.000
No. Increments:	8	7	5	5	7
Decimal Pts:	0	0	1	0	1
Force Fit:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 3.8. Changing the number of decimal places on a line.

Now click the **View / List Line(s)** option to reveal the screen shown in Figure 3.9.

X = ENGINE SPEED (RPM)		Y4 = POWER (kW)	
Position 1			
X		Y	
1000.		3.8	
2000.		7.9	
3000.		11.8	
4000.		15.4	
5000.		18.6	
6000.		21.3	
7000.		23.4	

Figure 3.9. List line values.

Selecting the **List / Graph / Graph 4** option displays the power results from the model constructed in Tutorial 1.

Selecting the **List / Position / Position 2** option displays the power results from the model constructed in this tutorial.

Calculate the percentage increase in peak power.

Close the List line and .MRS viewer windows.

## Displaying the .PRS Results

We can examine some of the phenomena which contribute to the substantial improvement in engine performance when the intake pipe is added by displaying the .PRS results.

Switch the interface to the .PRS viewer by clicking on the icon shown below.



Type **Control A** to autoscale the model in the viewing environment.

Click on the **red dot (mesh point) closest to the intake valve** and move the cursor over the graphs on the right-hand side of the screen. Click the **right-mouse button** and select the **Autoscale All Graphs** option. Also from the **right-mouse button** menu select **Show Valve Event Lines**. The default graphs show the variation in pressure, temperature, mass flow rate, and velocity at the point selected (see Figure 3.10). The engine speed at which the results were obtained is displayed in the bottom left-hand corner of the model display window (in the case 1000 rev/min). The two orange vertical lines indicate the exhaust valve opening and closing points respectively (from left to right), and the blue vertical lines represent the intake valve opening and closing points respectively.

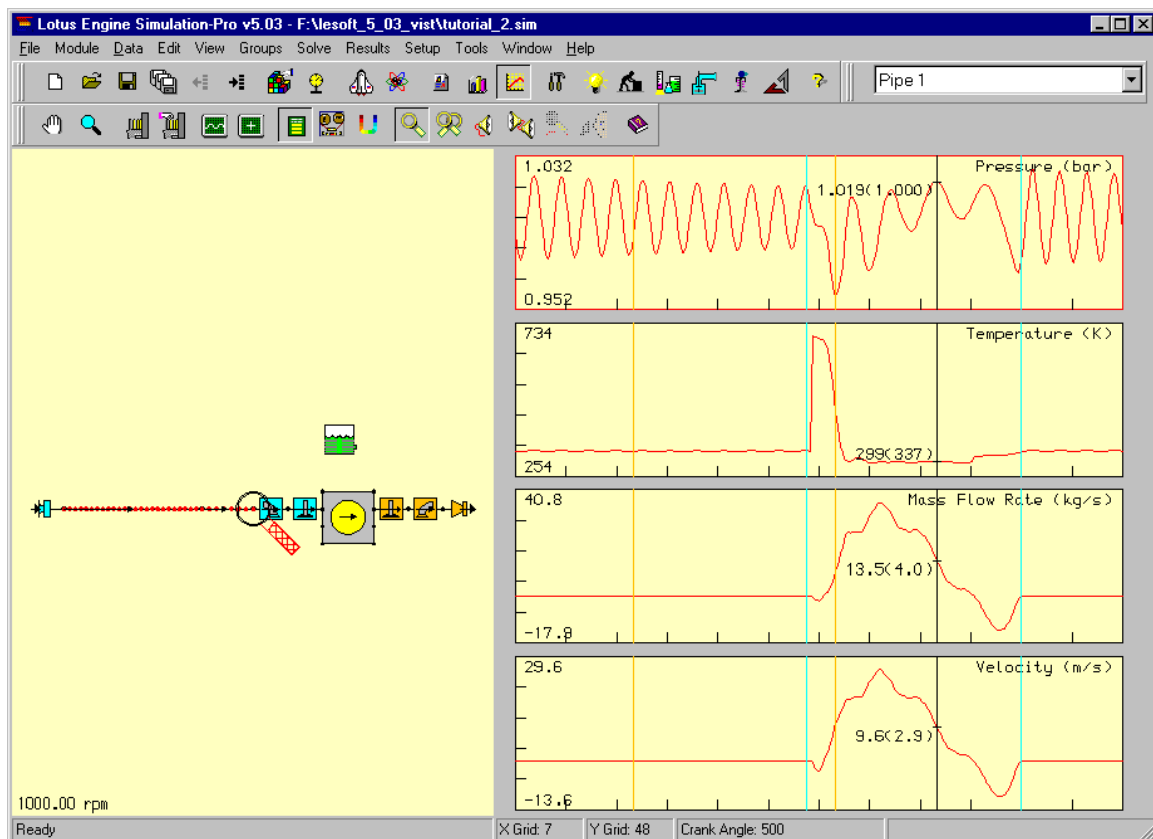


Figure 3.10. Gas property variations at the intake valve – 1000 rev/min.

Clicking on these graphs at various points moves the black vertical line, next to which is given the value of the results at the point (crank angle point) selected (indicated in the text box below the graphs). The mean values of each line can be displayed next to the current value by selecting **Display Graph Mean Values** from the right-mouse button menu.

The curves shown can be appreciated better with respect to the open and closed periods of the cylinder with respect to the intake valve. To the left of the left-hand blue line (IVO) and the right of the right-hand blue line (IVC) is the period during which the intake valve is closed. The pressure variation is characterised by an almost sinusoidal wave of decaying amplitude. In between these two lines the intake valve is open.

Note the large increase in the temperature at the intake valve in the valve overlap period (between IVO and EVC) – this is caused by hot gas flowing back from the cylinder into the intake pipe during this period, as can be seen from negative values in the mass flow rate and velocity diagrams.

We now wish to load in the results for 5000 rev/min (the peak torque speed). Click on the **PRS Results File Status** icon shown below.



This produces the window shown in Figure 3.11. The **Single Speed** option has been selected by default. Use the **right-arrow button** to move through the speeds until speed **5 of 7** is reached. Now click on **Apply** and **close** the PRS Results File Status window.

From the right-mouse button menu select Autoscale All Graphs. This will change the display to be similar to that shown in Figure 3.12.

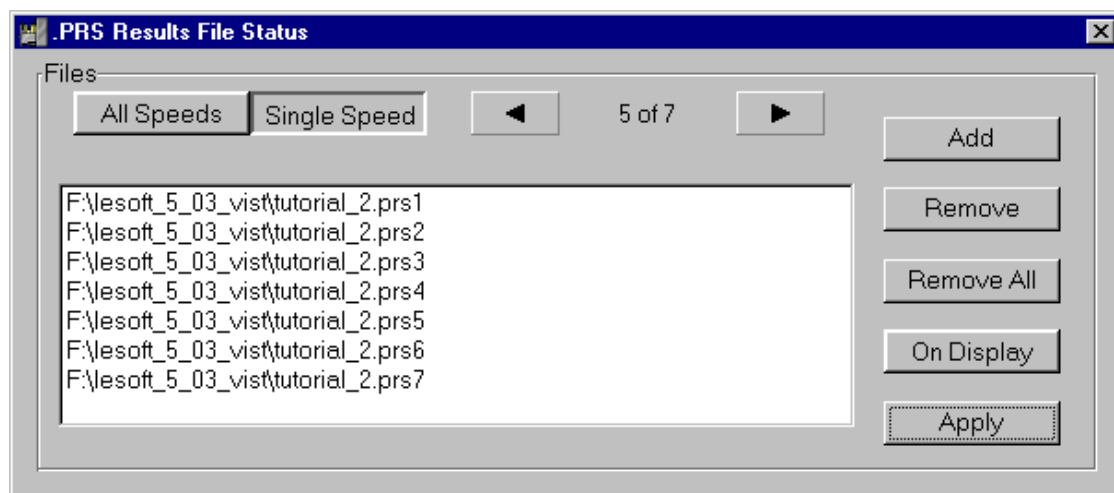


Figure 3.11. PRS results file status window.

Note that there are now fewer oscillations in the pressure trace because the time for the propagation of the pressure waves to occur is diminished. It is easier to gain an appreciation of the basic intake tuning mechanism by studying the gas property variations at this engine speed and this is explained below.

The periodic charging and discharging of the cylinders, together with the pressure increase generated during the combustion process, gives rise to highly unsteady flow in the manifolds of reciprocating engines. In a four-stroke engine, at the beginning of the induction process, the downward motion of the piston creates a reduction in cylinder pressure which gives rise to the propagation of a rarefaction wave into the intake runner pipe. This is manifested in the low pressure region between the IVO line and the black line in Figure 3.12. . The reflection of this wave at the end of the inlet runner pipe creates a pressure wave which, if it arrives back at the inlet valve in the period between bottom-dead centre and the point of inlet valve closure (as in Figure 3.12), can augment the trapped mass of air in the cylinder. By managing this mechanism the major features of the engine torque characteristic can be manipulated.

Note that the waves in the closed period are now distorted decaying sinusoids – the wave profiles have changed because they are non-linear. This means that the high pressure points on a wave front travel faster than the low pressure points. Hence the wave fronts steepen-up. Also note that there is now no reverse flow from the cylinder into the intake pipe at the start of the induction event.

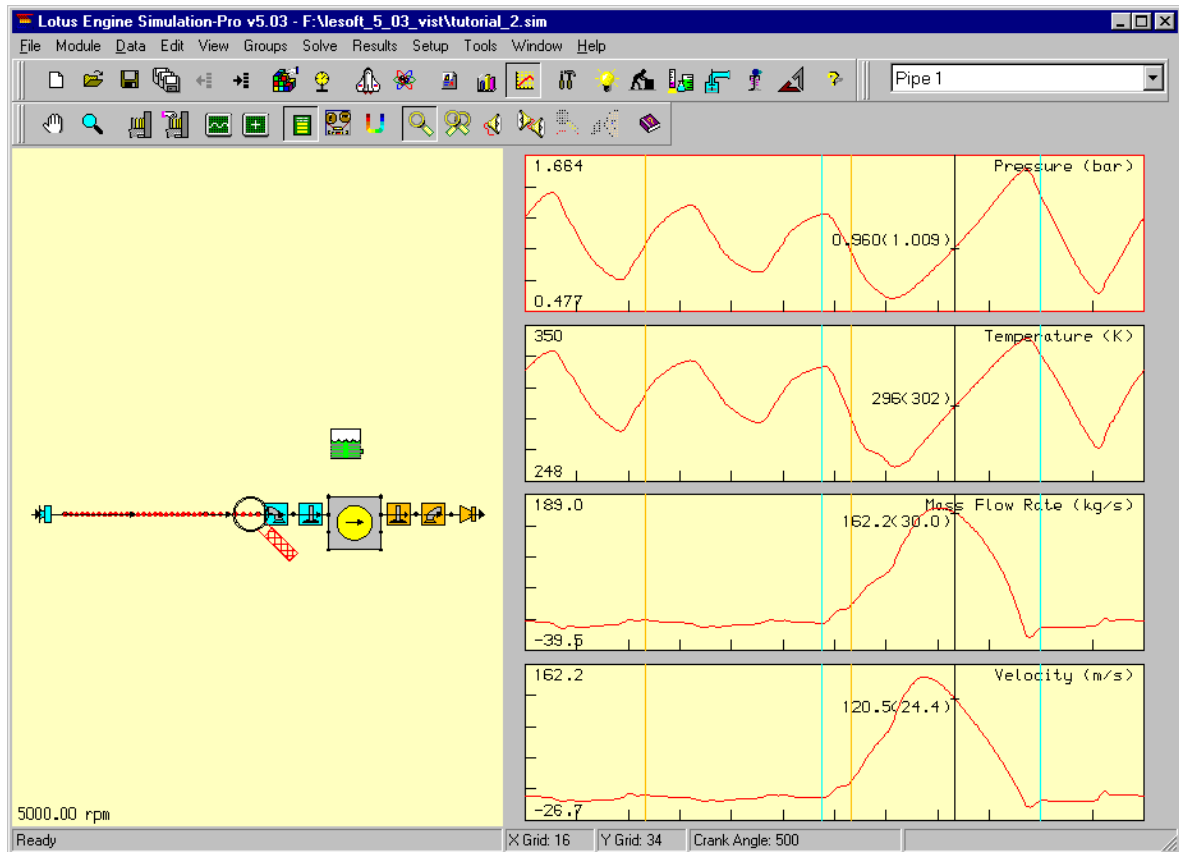


Figure 3.12. Gas property variations at the intake valve – 5000 rev/min.

## Animating .PRS Results

Two types of animation can be produced from the .PRS results. These are both described below.

To get a better appreciation of the wave propagation process within the intake pipe select **Display Pipe Graphic from the right-mouse button menu**. Click on the red sideways Y-icon at the top of the screen to display both pipes in the intake system. Click on the **Autoscale** option from the **View** menu. Click on the 'hand' icon at the top of the screen and position the pipe toward the bottom of the window.

This will generate the window shown in Figure 3.13.

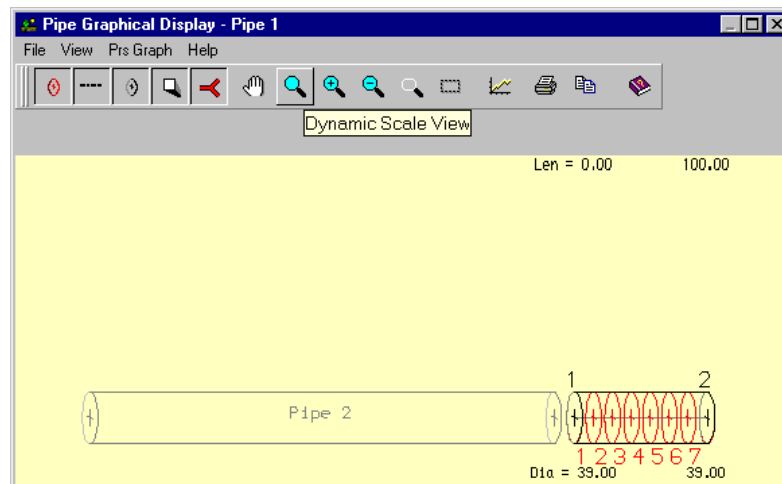


Figure 3.13. The Pipe Graphical Display window.

Now click on the graph icon and select the **Graph Y Length** option from the **Prs Graph** menu. Enter 120 in the value box and click **OK**. These actions will result in the window appearing as shown in Figure 3.14.

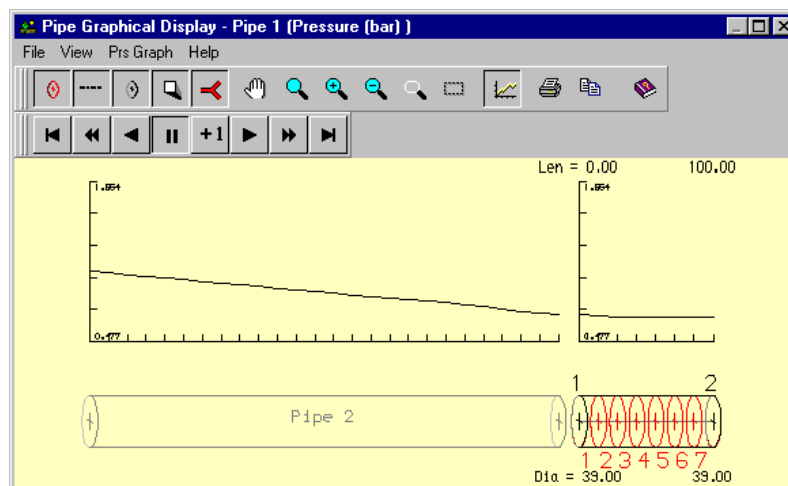


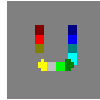
Figure 3.14. The Pipe Graphical Display window.

The animation control buttons can now be used to animate the variation of pressure along the pipe length.

Click on the rightward-pointing triangle to start the animation. Note that the position within the engine cycle is indicated by the moving black line on the .PRS graphs behind the window (the actual crank angle value is displayed in the value box at the bottom of the .PRS graphs – if you cannot see this reposition the .PRS Graphical Display window. The '+1' button enables the animation to be stepped through one time increment (2 degrees crank angle) at a time. Explore the use of the other functions on the animation control panel.

Stop the animation by clicking on the button with the two thick vertical black lines and close the pipe graphical display window.

Now click on the **Shaded Display Visibility** icon shown below.



This transforms the view of the engine model.

Now select the .PRS Video Controller Visibility icon shown below.



This produces a moveable animation control window as shown in Figure 3.15.

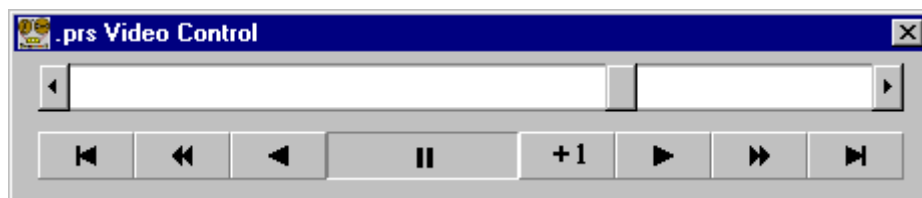


Figure 3.15. .PRS Video Control panel.

Clicking the rightward-pointing triangle animates the .PRS model as shown in Figure 3.16.

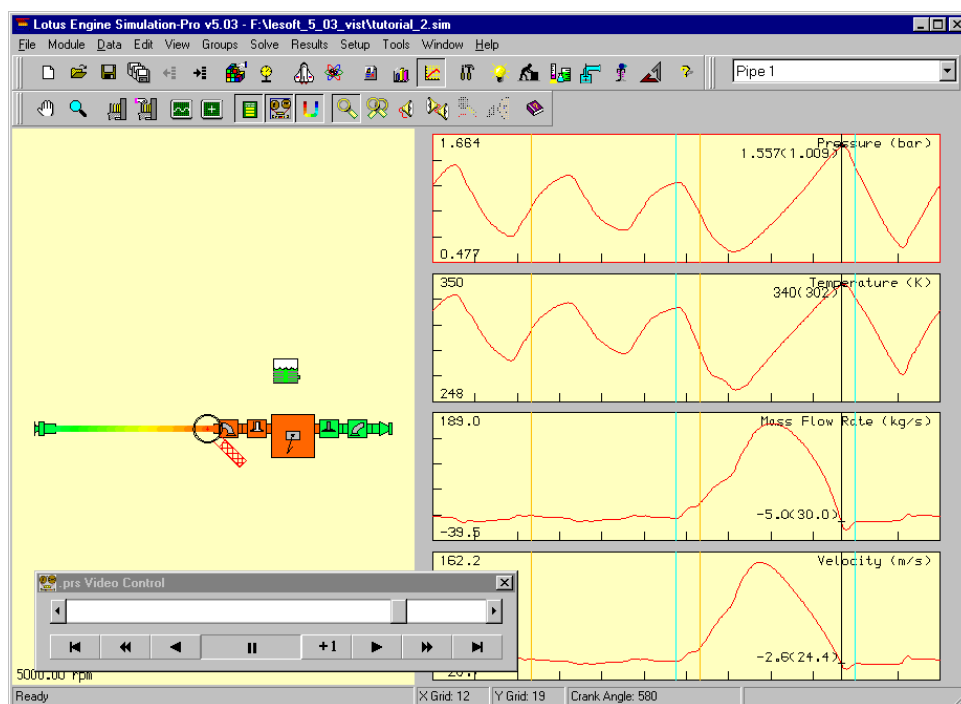


Figure 3.16. Still from animated .PRS display.

In this animation the piston and valves are phased correctly. When the valves are open they turn white.

Stop the animation.

Turn the .PRS graphs off by clicking on the Results Pane Visibility icon to the left of the .PRS Video Controller Visibility icon.



Zoom and manoeuvre the image using the magnifying glass and hand icons on the tool panel. This allows the user to study particular phenomena in detail.

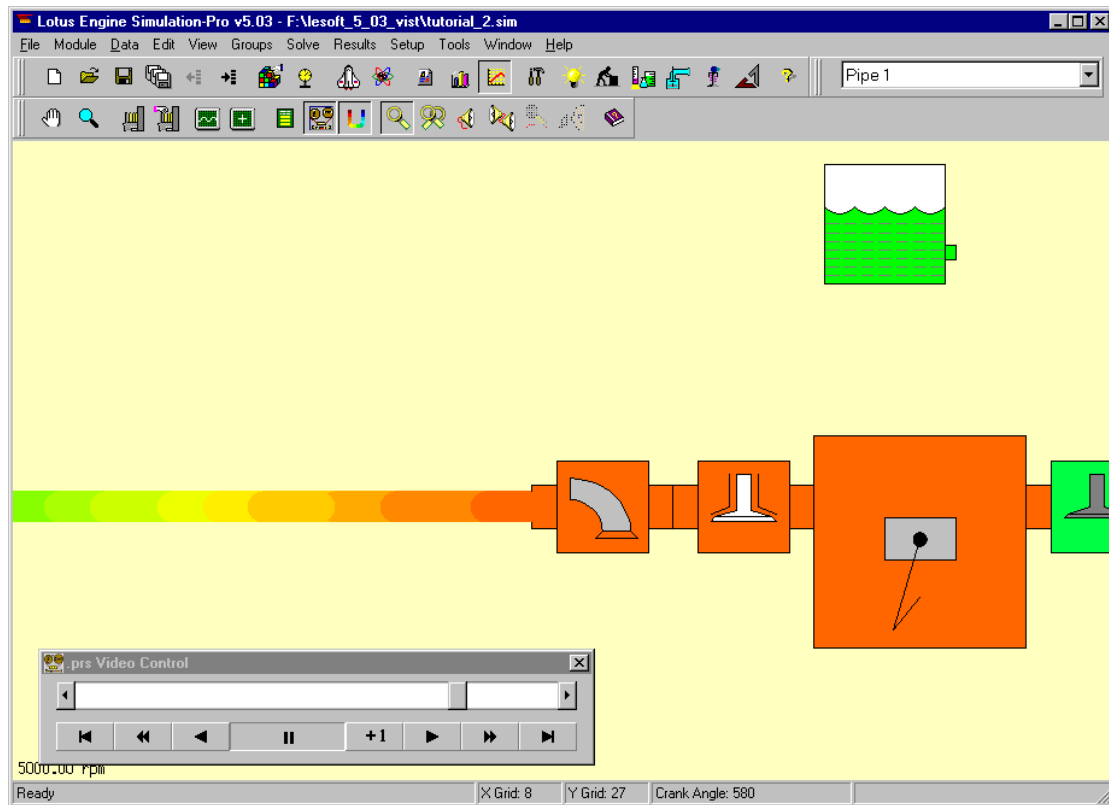


Figure 3.17. Zoomed area of the .PRS animation.

Replace the .PRS graphs and type **Control-A** to autoscale the model.

Now return to the builder environment by clicking on the icon shown below.



**This completes this tutorial.**

## Exercise

Change the length of the intake runner to 500 mm and re-run the model. Compare the results with those obtained using the 350 mm intake runner. What is the maximum torque speed? Investigate the pressure wave variation in the intake pipe at different engine speeds.



# 4

## Tutorial 3. 'Parametric Studies Using Basic Single-Cylinder Model with Intake Pipe'

### Overview

This tutorial shows the user how to perform an analysis using the parametric tool, using the single-cylinder engine model developed in Tutorial 2.

The Parametric / Optimizer Tool allows the user to run a series of tests without having to modify the initial engine model. Groups of components are created and the attributes of all the components within a group can be changed automatically using the Parametric Tool. The Parametric Tool allows the user to perform 1-D, 2-D or even up to 10-D parametric studies.

The chapter contains the following sections:

- Element Groups, 42
- Creating Element Groups, 42
- Adding Elements to a Group, 42
- Viewing Elements within a Group, 43
- Starting the Parametric Tool, 44
- Defining a Scoring System, 44
- Setting Up Parameters, 46
- Launching a 1D Parametric Run, 47
- Exercise - 1D parametric analysis, 49
- Running a 2D Parametric Run, 50
- Running the Optimizer, 52

## Element Groups

Before the Parametric / Optimization Tool can be invoked, the user must specify the engine test conditions and define element groups.

### Creating Element Groups

Begin by loading the model **tutorial\_2.sim** into LOTUS ENGINE SIMULATION. To create a group select **Groups** from the LOTUS ENGINE SIMULATION menubar. Then select **New** from the menu to create a new group. A pop-up window will appear where an identifying label for the group can be entered, as shown below.

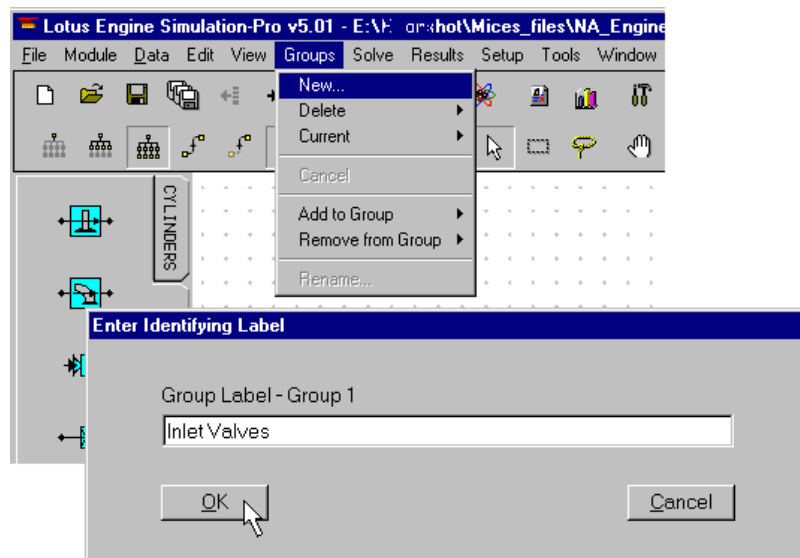


Figure 4.1. Creating a Group.

We wish to create two groups: one called '*Inlet Valves*' and another named '*Inlet Runner*'. So type '*Inlet Valves*' into the pop-up window, as shown in Figure 4.1. Then press **OK**. Select **Groups**, from the LOTUS ENGINE SIMULATION menubar again, followed by **New**. Now type '*Inlet Runner*' in the pop-up window. Then press **OK**.

These new groups contain no elements. Elements must now be added to them.

### Adding Elements to a Group

To add elements to an existing group requires the following two steps to be performed:

- The element to be added is selected;
- To add the selected element to a group select **Groups** from the LOTUS ENGINE SIMULATION menubar. Then select **Add to Group** from the menu. A pop-up window will appear enabling the group to which the elements are to be added to be selected.

Only elements of the same generic type can be added to a group, i.e. the elements within the group must all be pipes or all be plenums. The first element added to the group will set the type of elements that the group contains.

We wish to add the inlet valve to the 'Inlet Valves' group. Begin by selecting the inlet valve with the mouse, as shown in Figure 4.2.

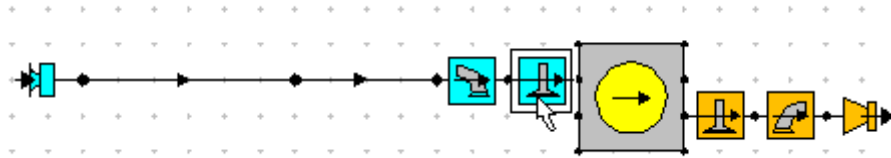


Figure 4.2. Selecting the inlet valve element.

Now select **Groups** from the LOTUS ENGINE SIMULATION menubar, followed by **Add to Group**. A list of the current groups will appear. Click on 'Inlet Valves' to add the inlet valve to this group, as shown in Figure 4.3. Elements can be removed from a group in exactly the same manner, except that **Remove from Group** should be selected from the **Groups** menu.

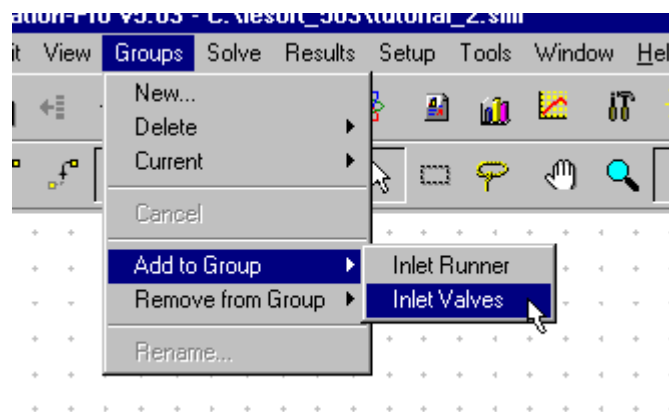


Figure 4.3. Adding an element to a group.

Now add the inlet runner pipe to the 'Inlet Runner' group. Click on the inlet runner pipe, so that it flashes, as shown in Figure 4.4. Then click on **Groups**, then **Add to Group**, then select the 'Inlet Runner' group.

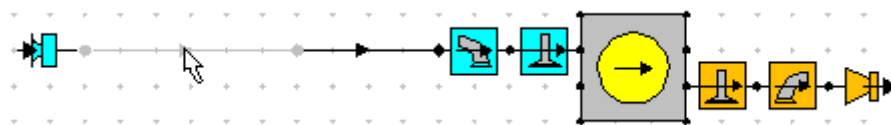


Figure 4.4. Selecting the inlet runner .

## Viewing the Elements within a Group

The elements contained within a group can be viewed by selecting **Groups** from THE LOTUS ENGINE SIMULATION menubar. Then select **Current** from the menu. A pop-up window will appear listing all of the current groups. The group to be displayed is selected from this list, as shown in Figure 4.5. Once a group has been selected, the builder window display will switch from displaying the current model, to displaying the contents of the selected group.

To revert the builder display to show the entire model, select **Groups** from THE LOTUS ENGINE SIMULATION menubar. Then select **Cancel** from the menu.

Save the modified model to **tutorial\_3.sim** having first changed the Test No. description to **tutorial\_3**.

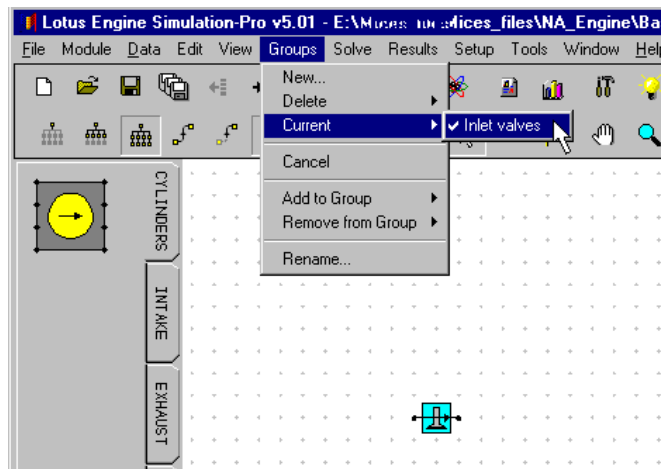


Figure 4.5. Viewing a group

## Starting the parametric tool

The parametric tool is started by clicking on the parametric tool icon from the LOTUS ENGINE SIMULATION toolbar, as shown in Figure 4.6. Groups must be defined before the parametric tool can be started.

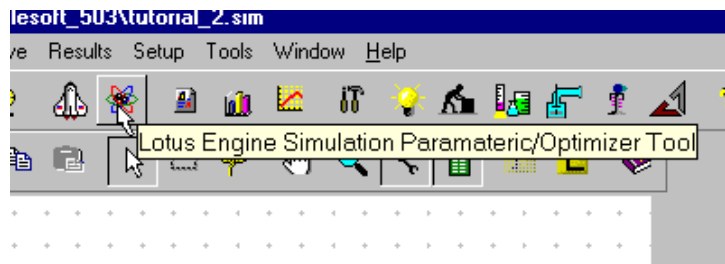


Figure 4.6. Invoking the parametric tool.

## Defining a Scoring System

When the parametric tool is invoked the user will be presented with the **Gates** menu screen, shown in Figure 4.7 below.

To aid the optimisation procedure and the analysis of results a scoring system has been devised:

- Gates of any performance metric (torque, BMEP, vol. Eff.) are specified at the engine speeds defined in the test data section of the engine model.
- Gates can be defined as:
  - a existing performance characteristic;
  - a target performance characteristic.
- Weighting coefficients multiply difference in the actual and gate values at each speed to produce a score:
  - Bonus and penalty weightings are applied to the difference between the achieved values and the target values.

The performance parameter used for the gate values can be altered by clicking the left mouse button, whilst the mouse pointer is positioned on the down arrow by the **Gate Variable** menu item, as shown in Figure 4.7. A list of parameters will appear, the desired parameter can be highlighted with the mouse pointer and selected with the left mouse button. In this tutorial we will select **Volumetric Efficiency** as the parameter to scrutinise in the parametric study.

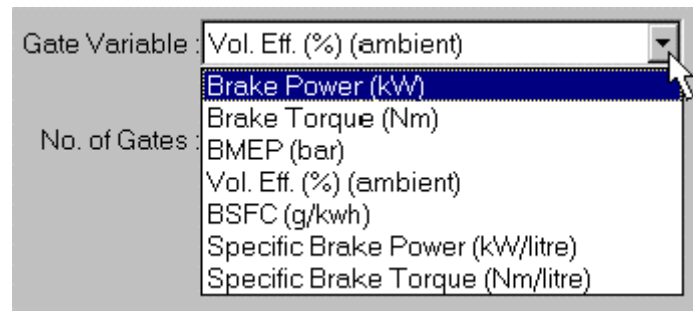


Figure 4.7. Setting the gate parameter

Volumetric efficiency is defined as

$$\eta_v = \frac{\text{mass flow rate of air into cylinder}}{\left[ \frac{\text{mass of air contained in swept volume at reference pressure and temperature} \times \text{no of induction cycles per second}}{\right]}$$

The volumetric efficiency is a measure of the 'breathing efficiency' of the engine and it is this parameter which is directly affected by changes to the manifold geometry. Studies involving air induction system, and exhaust system geometry should therefore be conducted initially using volumetric efficiency.

Now select **Run Baseline to fill Gates** from the **Gates** menu at the top of the screen, as shown in Figure 4.8.

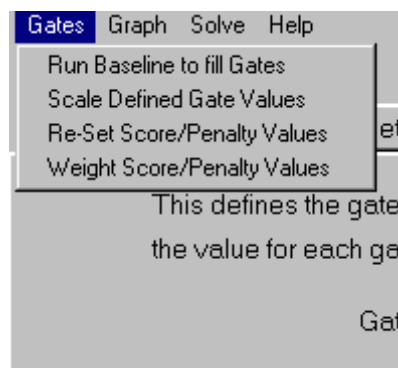


Figure 4.8. Selecting the Run Baseline to Fill Gates

This will run the baseline model to fill the gate values with the volumetric efficiency predictions. The parametric screen should now appear as shown in Figure 4.9

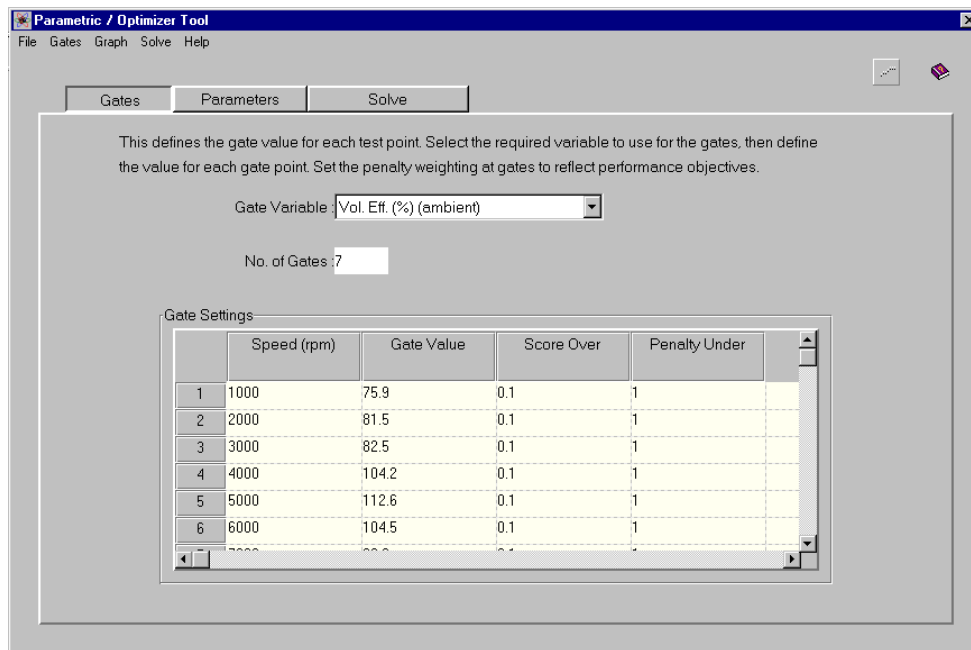


Figure 4.9. Parametric screen after running the baseline model to fill Gate Values with vol. Eff.

## Setting Up Parameters

Now click the **Parameters** tab. Having picked up a feel for the effects of inlet runner length on engine performance in the previous tutorial we will perform a one-dimensional parametric study of the effects of varying the intake valve closing angle (IVC). In the exercise of this tutorial a further 1D parametric run is required, varying the inlet pipe length.

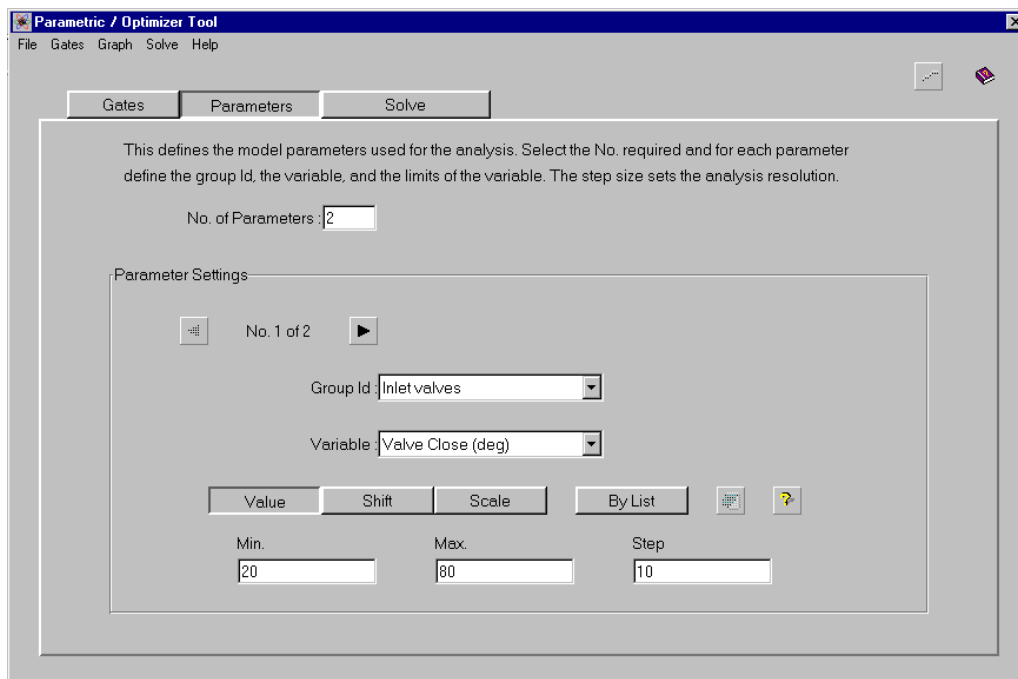


Figure 4.10. Setting up the IVC parameters.



In order to be able to perform both the proposed parametric runs set the **No. of Parameters** to 2 and press **Return**. Now select the **Inlet valve** group from the **Group Id** box and set the **Variable** to **Valve Close (deg)**. Click on the 'question mark' icon to enquire the value of IVC from the baseline model. This should be 60 deg. (after bdc induction stroke).

Click on the Value button as shown in Figure 4.10 and enter a **Minimum** value of **20**, a **Maximum** value **80**, at step sizes of **10** degrees.

## Launching a 1D Parametric Run

Now click on the **Solve** tab and select **1D parametric**. Click on the **Graph / Display / Volumetric Efficiency** option from the drop down menu at the top of the screen to change the output display parameter. Also click on the **Graph / Display / Autoscale** option. Now click on the Launch icon. This produces the window shown in Figure 4.11. Choose the 1<sup>st</sup> parameter as **inlet valves, valves close (deg)** as shown and click **OK**. This will launch the parametric run.

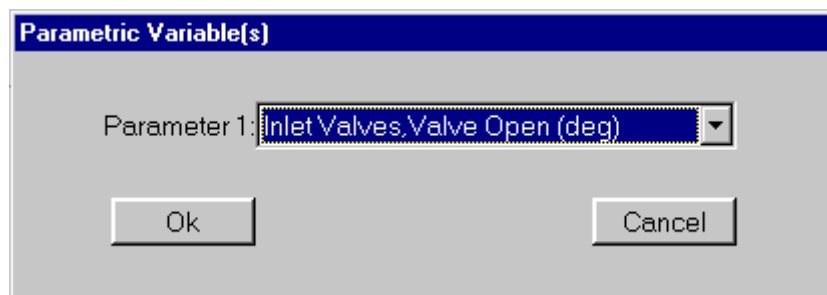


Figure 4.11. Parametric variable window.

The 1D parametric progress indicator will appear as shown in Figure 4.12. This window enables the user to monitor the progress of the runs. The top bar indicates the progress of the total job whereas the bottom bar indicates the progress of the current job.

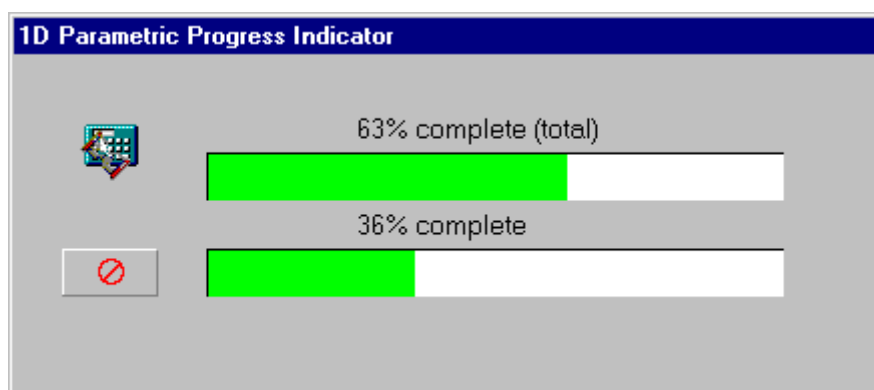


Figure 4.12. Progress Indicator window.

As the jobs run the volumetric efficiency curves will appear on the **Gates / Results** graph.

After the job has run the screen should appear as shown in Figure 4.13. Use the arrow keys to cycle through the results. The yellow curve is the one to which the values which appear in the **Picked Run Details** box refer. The red curve is the one selected by the scoring system defined in the Gates screen. Now the results have been obtained the weighting coefficients can be re-defined if desired.

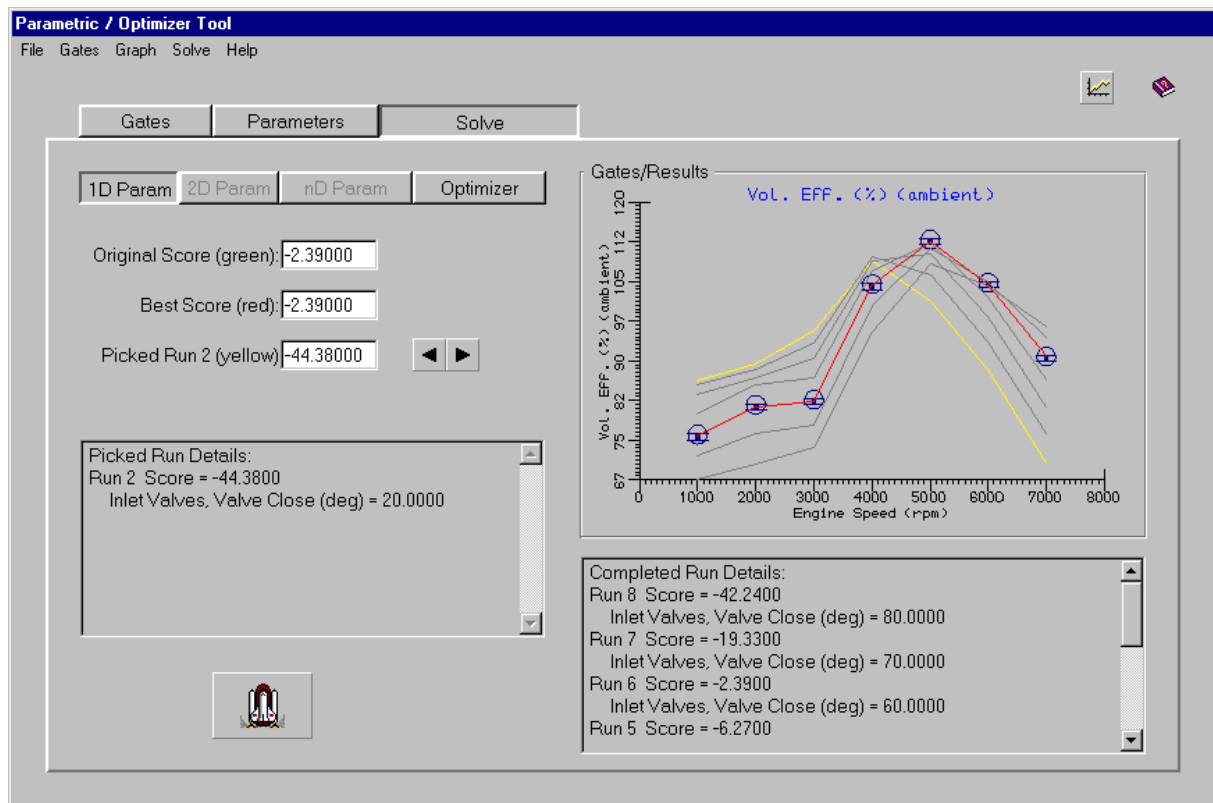
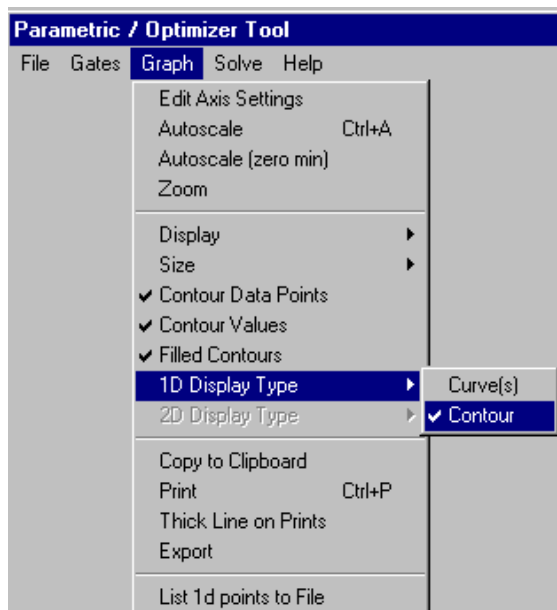


Figure 4.13. Results screen – Curve Display.

In the results above it can be seen that early IVC timing gives good volumetric efficiency at low engine speeds and poor volumetric efficiency at high engine speeds. Conversely late IVC timing gives high volumetric efficiency at high engine speeds and low volumetric efficiency at low engine speeds.

If you wish to save the results this must be done before exiting the Parametric window.



The 1D parametric results can also be displayed not only for over 130 different variables, but also in two basic styles. The alternative styles are as **curves** or as a **contour**. To display the 1D results as a contour plot select the **Graph / 1D Display Type / Contour** menu option.

The one result that can't be displayed as a contour for a 1D display is the **Curves Gate Score**. For a 1D parametric analysis this can only be displayed as a single line.

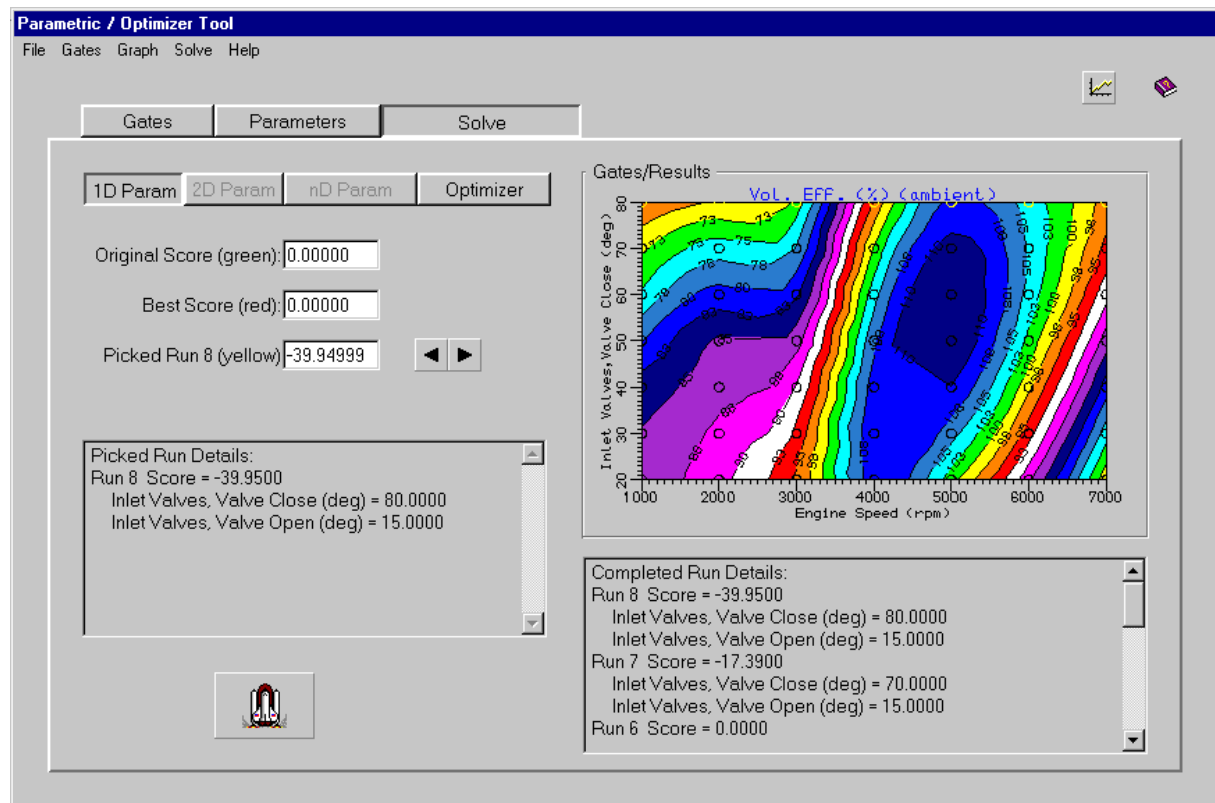


Figure 4.14. Results screen – Contour Display.

## Exercise

Set up a 1-D parametric to vary the intake pipe length between 100 mm and 600 mm with a 50 mm step size. (results shown below).

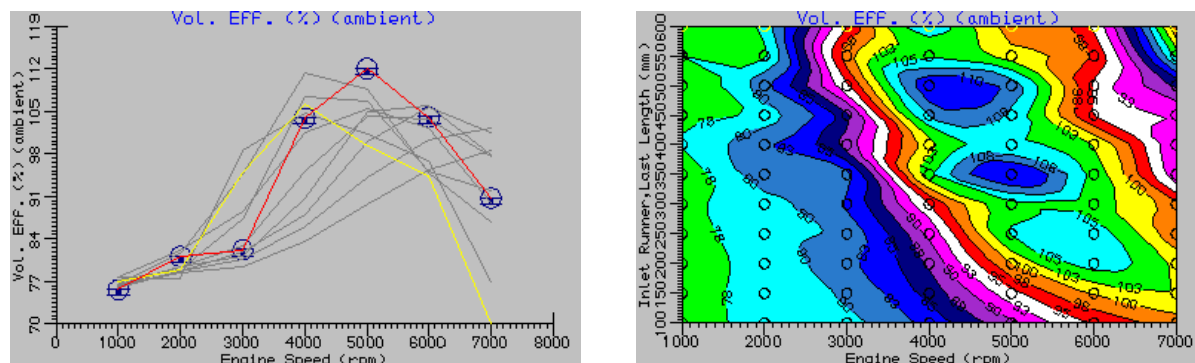


Figure 4.15. Example 1 Results Screen Shot

## Running a 2D Parametric Analysis

Ensure both parameters are set as for the previous 1D section and example 1, i.e.

Parameter 1, Group Id = **Inlet Valves**, Variable = **Valve close (deg)**,  
**By Value**, Min = **20**, Max = **80**, Step = **10**

Parameter 2, Group Id = **Inlet Runner**, Variable = **Last Length (mm)**,  
**By Value**, Min = **100**, Max = **600**, Step = **50**

A 2D parametric analysis will run a full matrix of each inlet valve closing value with each inlet runner length. For the settings as above this will be 77 (7valve x 11pipe) individual build specifications, each build specification being run for 7 speeds. Select the **Solve** tab and change to **2D Parametric**. Again set the display type to be **Vol Eff (%)** and then submit the job using the launch icon and select the **Inlet Valves, Valve Close (deg)** as the 1<sup>st</sup> parameter and **Inlet Runner, Last Length (mm)** as the 2<sup>nd</sup> parameter.

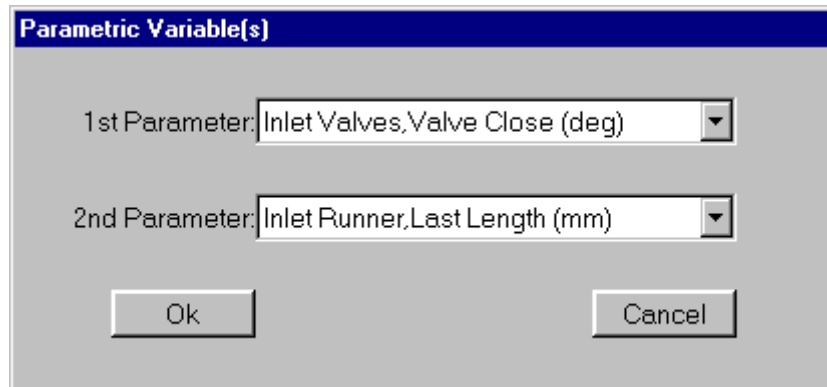


Figure 4.16. 2D parametric variable window

The 2D parametric progress indicator will appear allowing the progress of the analysis to be monitored. After the job has run, (this will take approx. 2-2<sup>1/2</sup> hours to run on a 500Mhz Pc), the screen should appear as shown in Figure 4.17.

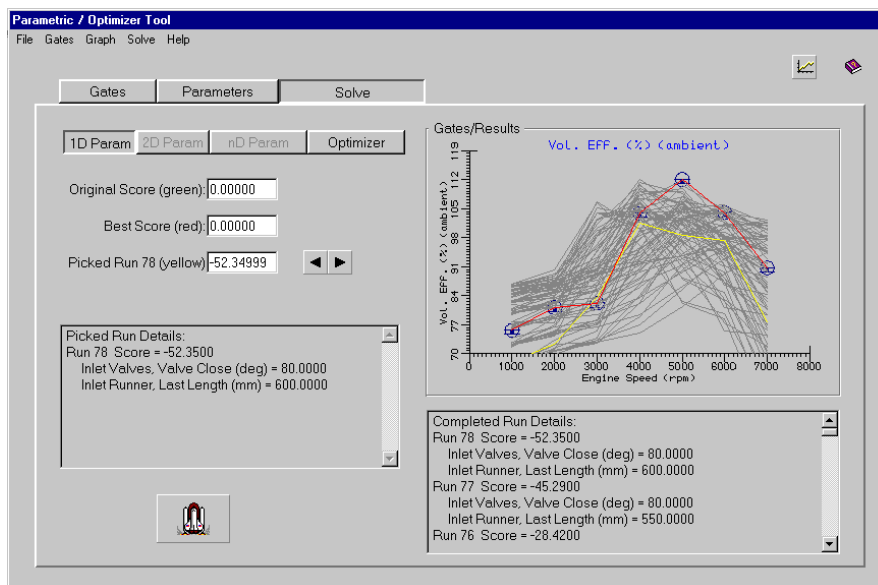
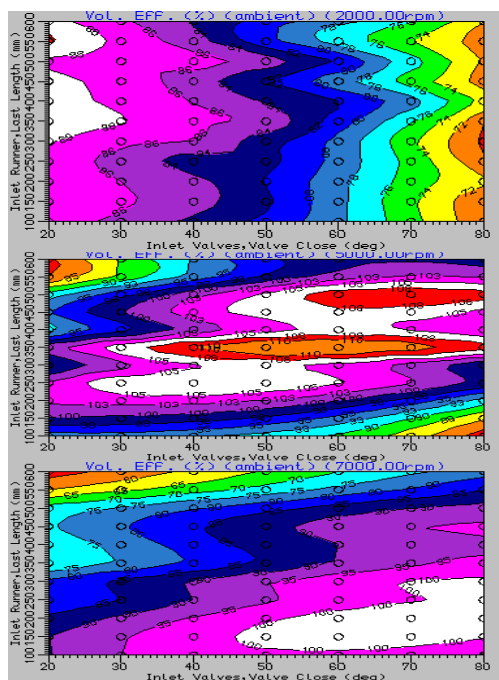


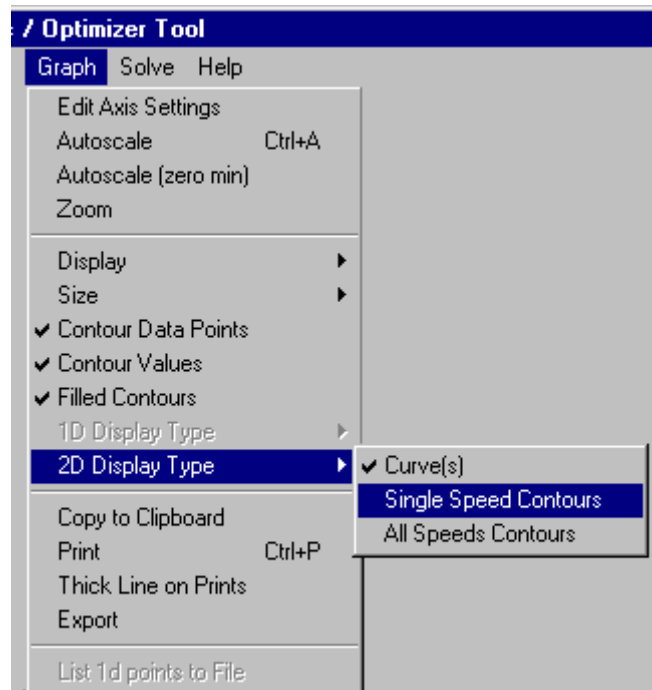
Figure 4.17. 2D Parametric Results Screen

Again the arrow keys can be cycled through to indicate the build specification of a specific run and show the Volumetric Efficiency curve for that build specification. As with the 1D run the weighting coefficients/gate values can be adjusted to identify different curve gate score scenarios.

The 2D display can also be changed to display different variables and in different styles. The 2D parametric results can be displayed as either **curves**, **single speed contours** or **all speed contours**. The **All speed Contours** option is only visually viable if the contour fill is turned **off**. Using individual speed contour plots of volumetric efficiency highlight the problem in identifying an optimum build specification over the complete speed range, since the contours vary widely between the speeds.



Example Single Speed Contours



2D Parametric Menu Options

As with the 1D parametric results plotting the **Curves Gate Score** produces slightly different output from the other variables. The graph is always displayed as a contour plot of the score against the two parametric variables. The contour plot for this example is shown in Figure 4.18.

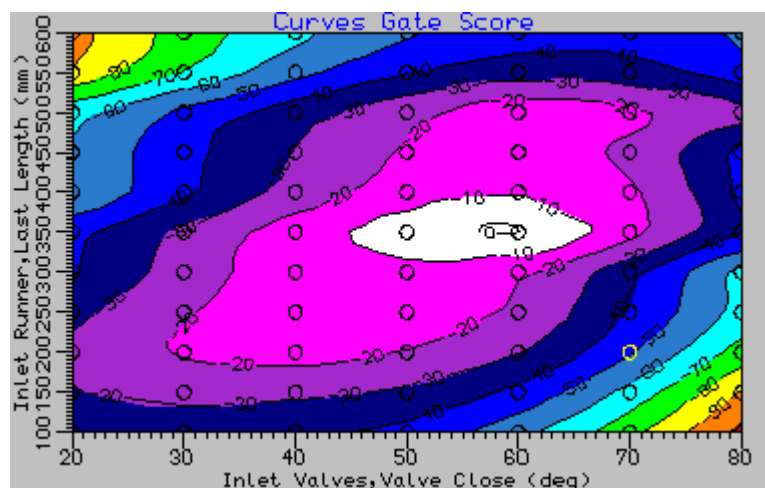


Figure 4.18. 2D Parametric Curves Gate Score Contour plot

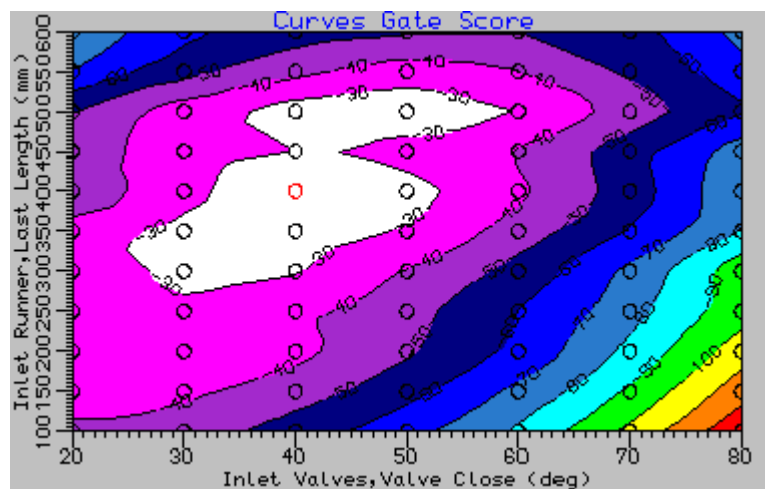
This contour plot identifies the combination of parametric variables, which achieve the best score for the defined gates/scoring. By coincidence this example shows that the original specification of Inlet valve closing at 60 degrees and Inlet runner length of 350 mm achieves the best score, although this is perhaps not so surprising since the gate values were derived from the original build specification.

If we were to change the gate values to represent an alternative performance requirement, the build specification identified as the optimum would change.

As an example let's reset the gate values to aim for an engine that has greater volumetric efficiency at lower engine speeds than that achieved with the standard model, we will need to sacrifice some high speed volumetric efficiency for this improved low speed benefit. Select the **Gates** tab and edit the gate values as below;

Rpm	Original Gate Value	New Gate Value
1000	75.5	85
2000	81.0	90
3000	82.1	110
4000	104.0	105
5000	112.2	100
6000	104.1	95
7000	90.7	85

Now select the **Solve** tab and note that the contour plot has changed to that shown below indicating a new maximum score point (marked with a red circle) at a valve timing of 40 degrees and a runner length of 400 mm.



Thus the advantage of running a full 2D parametric is that if the results are saved they can be subsequently used to identify differing optimum build specifications to suit varying targets without recourse to re-running the analysis.

## Running the Optimizer

An alternative approach to the parametric run is to use the **Optimizer** solution method. This has the advantage, over the parametric approach, in that it does not need to perform a full matrix of tests and can thus run quicker. The disadvantage of the optimize route is that since it uses the **Curves Gate Score** to decide the solution route if you need to review an alternative gate value/score setting you must re-run the analysis.

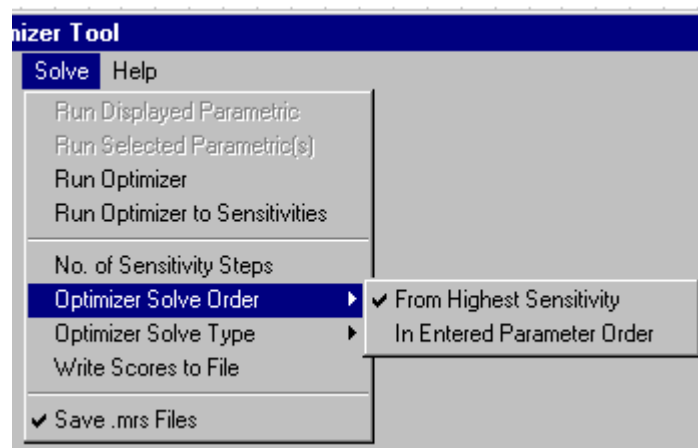
As a test case we will re-run the current example with the new gate values and identify how many solution steps are performed to hopefully achieve the same conclusion. An **Optimizer** solution will use the parameter settings in a slightly different way to the 1D and 2D parametric solution methods. The **min** and **max** values are still used to 'bound' the solution but the **step** value is now used to control the resolution of the optimized solution. Thus the current settings would only identify the optimum valve close position to within 10 degrees and the inlet runner length to within 50mm. For a direct run time comparison we can leave these settings but normally you would take advantage of the optimizer to identify a solution with smaller step settings.

Change back to the **Solve** tab, (having first checked that your gate values are still set to the 'new' values). Select the **Optimizer** solution type and accept any results data loss, (perhaps save them first if you haven't yet done so?).

The optimizer has a couple of solution control settings that can be changed by the user. They are;

- Number of sensitivity steps, (default 3)
- Parameter Solve Order, either in order of highest sensitivity or in the order entered, (default is in the order of highest sensitivity).
- Solve Type as either full wave or half wave, (default full wave).

All of these options can be accessed through the **Solve** menu, see below.



For this example we will stick with the default settings, (for further information on optimizer settings consult the help file).

Before we can proceed with the solution we have one more task to perform. Changing to an optimizer type solution will make visible some additional widgets on the **Parameters** panel. These are used by the optimizer to identify coupled parameter runs. If we do not identify that we want our two parameters to be solved as coupled, the optimizer will run them as individual 1D analyses, solving the most sensitive first before moving on to solve the next parameter. We require the optimum to mimic the previous 2D parametric run and thus require them to be optimized coupled.

Change back to the **Parameters** panel and with parameter one displayed select the **Coupled** button next to the toggle arrows, this should then display in the selection box our other parameter for inlet runner length. If you now use the arrow keys to move to the second parameter you will see the **coupled** button and selection box disabled to indicate that this parameter is already coupled and cannot be selected again, see Figure 4.19.

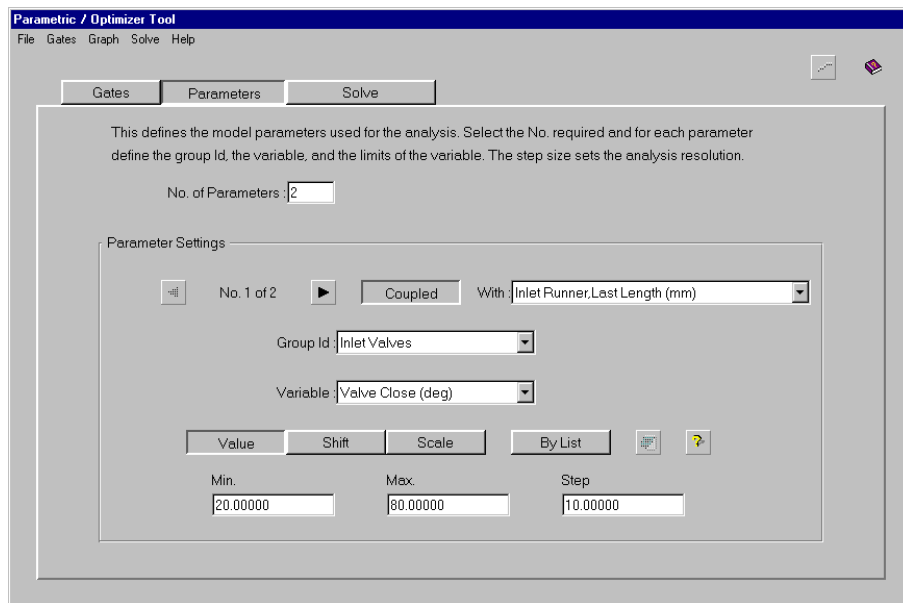


Figure 4.19. Parameter Setting for Coupled Optimize Run.

Return to the solve panel and set the display to show the **curves gate score** and select the autoscale option, we are now ready to run the optimizer. Selecting the launch icon will start the run displaying the normal progress display.

As the optimizer progresses the calculated scores are displayed in the graph. The first part of the optimization identifies the sensitivity of the score to the parameter over its defined range. The **number of sensitivity steps** setting is used to control how many points are used to identify the variation in score for each optimizer parameter step. The default setting of three, means that the optimizer will look at the score for then min. max and mid point values. (A higher setting whilst it may assist in identifying an intermediate peak in the sensitivity would lead to more runs and hence longer run times).

Once the optimizer has identified the sensitivities and the points at which the maximum values occurred, the solve order can be identified, (this is not applicable to our example since we only have one step in the optimizer routine because our two parameters are solved coupled). The individual optimizer steps' maximum value points are used to identify the initial setting for the parameters and later refined as that parameters step is reached.



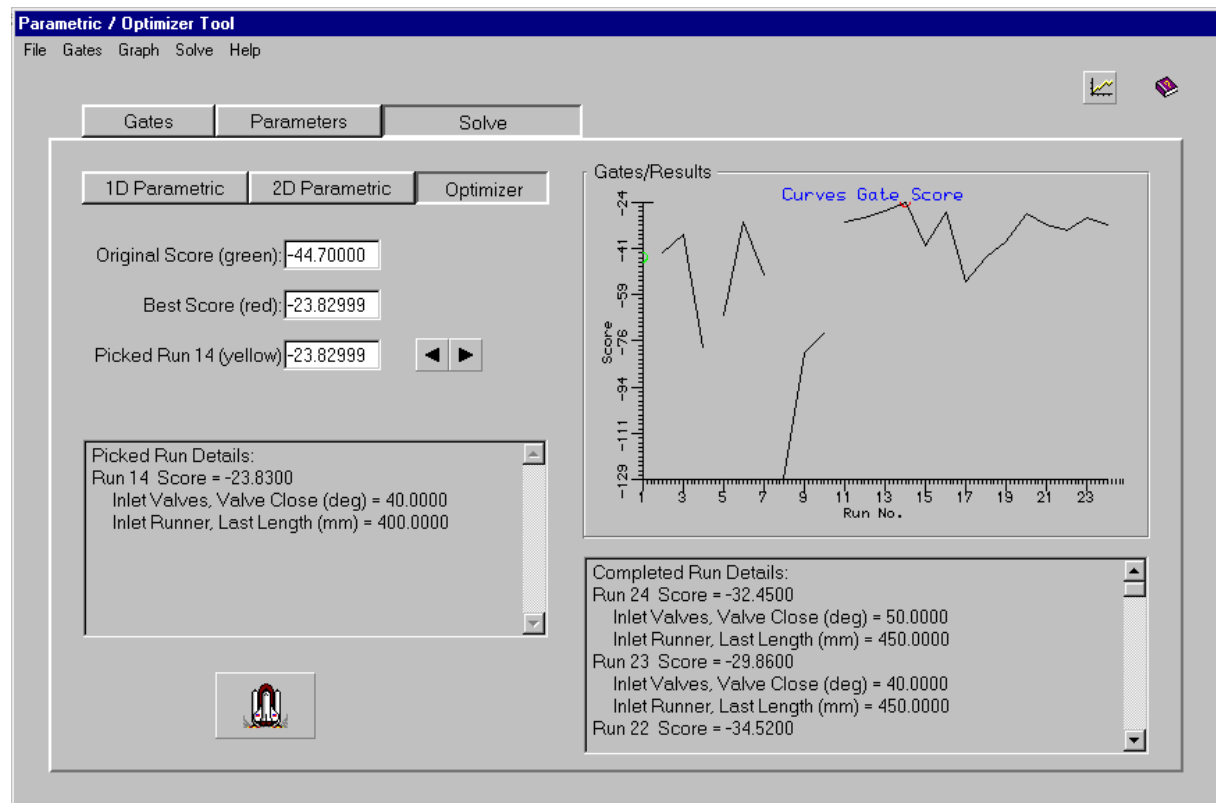


Figure 4.20. Optimizer Results Screen.

Once the run is complete the screen display should look as shown in Figure 4.20. The red circle again indicates the highest score point, which if we use the arrow keys to pick this run (point No. 14) the build specification is identified as a valve timing of 40 degrees and a runner length of 400 mm. This is the same solution point as identified by the full 2D parametric run in 77 runs, whilst the optimizer identified it after 14 runs and completed 24 runs to confirm the peak. This represents a considerable saving in the solution run time.



## Overview

This tutorial shows the user how to produce a multi-cylinder model, starting from the single cylinder model we have developed in the previous tutorials.

We will produce a 4 cylinder model using the existing single cylinder model and look at the influence of pipe junction geometry for our example 4-2-1 exhaust manifold.

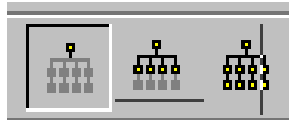
The tutorial includes the following sections:

- Adding the Exhaust pipes, 57
- Copying Model Segments, 59
- Creating the 4-2-1 Exhaust Manifold, 61
- Selecting the Firing Order, 64
- Running the Multi Cylinder Model, 65
- Using the Pressure Loss Junction, 66
- Running the Pressure Loss Junction Model, 68

## Adding the Exhaust Pipes

To produce the 4 cylinder model we will use the single cylinder engine created in the previous tutorial, we shall begin by loading the file **tutorial\_3.sim**.

In a similar manner that previously used to add the inlet pipes we now wish to insert exhaust pipes to the exhaust port icon. In order to be able to detach a single element from other elements the **Move Element Singularly** icon on the left-hand side of the group of icons shown below should be depressed.



Select the exhaust boundary element and move it out of the way, we will connect it back into the system once we have built the manifold. You may need to zoom out a bit to provide adequate space to add the pipes.

Pick the **Pipe** tab on the toolkit and select a pipe element. Add the first end of the pipe to the free end of the exhaust port and leave the pipes second end several grid squares away from the port. In the property sheet for the pipe change the length to **100 mm**, set the start and end diameters to **32 mm**, the wall thickness to **4 mm** and the cooling type to **Water cooled**. This pipe represents the exhaust port in the cylinder head.

Now add the primary exhaust pipe connect the first end to the free end of the exhaust port pipe and again leave the second end a few grid squares away. In the property sheet for the pipe change the length to **400 mm** and start and end diameters to **32 mm**, we will accept the other defaults. Your model should now look like that illustrated in Figure 5.1.

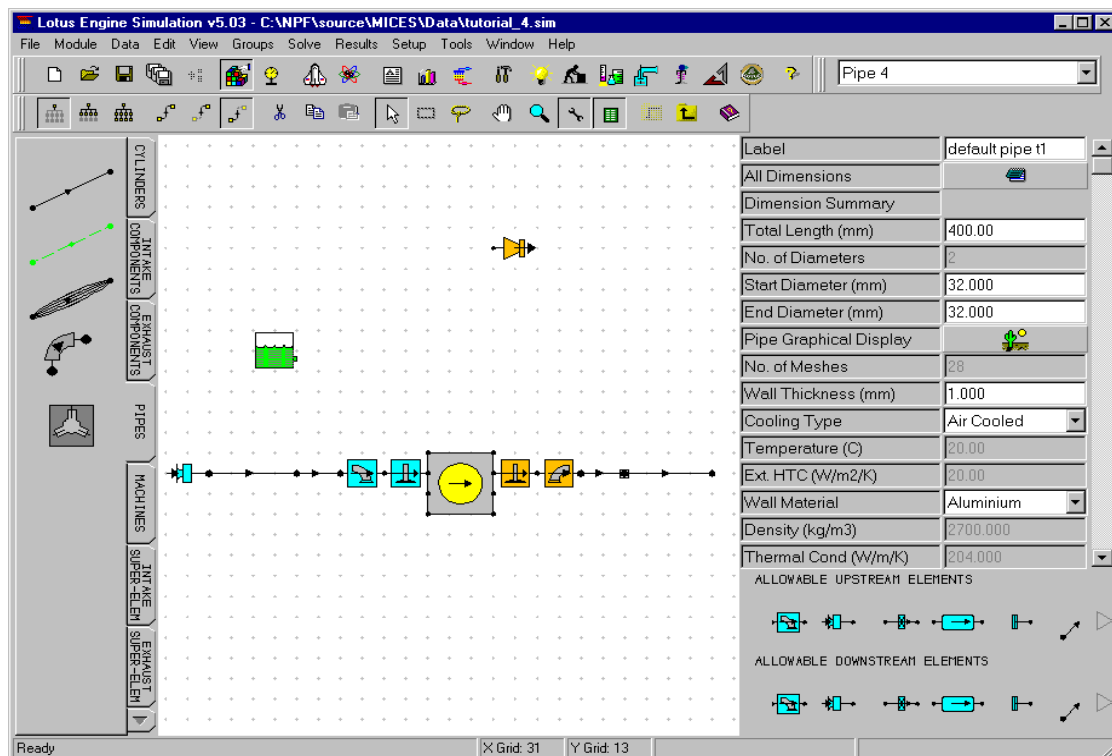


Figure 5.1. Adding the exhaust pipes

## Copying Model Segments

We will build up the multi cylinder model by copying the single cylinder and pasting it back in three times to give us our basic 4 cylinder model. Before we do this we need to move the inlet boundary so that we can copy only the portion of the model we require. Pick the inlet boundary and move it out of the way.

Select the **Edit Select** icon, as shown below,



Then select the complete single cylinder model with the mouse. The display will indicate the captured region, all elements contained wholly within this region will be subjected to any subsequent cut/copy paste type activity. Thus ensure that you have fully enclosed both ends of the pipes, see Figure 5.2.

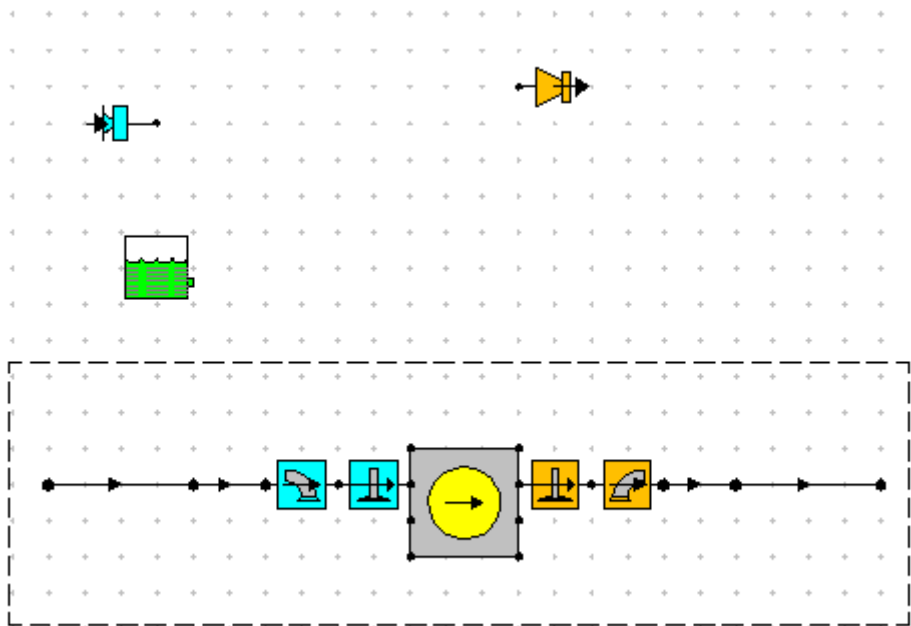


Figure 5.2. Capture of single cylinder portion

Now select **edit / copy** (or Ctrl+C), (the capture box will now disappear), then select **Edit / Paste** (or Ctrl+V) three times, each 'paste' event will add a new cylinder assembly below the previous one to produce the building blocks for our four cylinder model. The new model is displayed in Figure 5.3

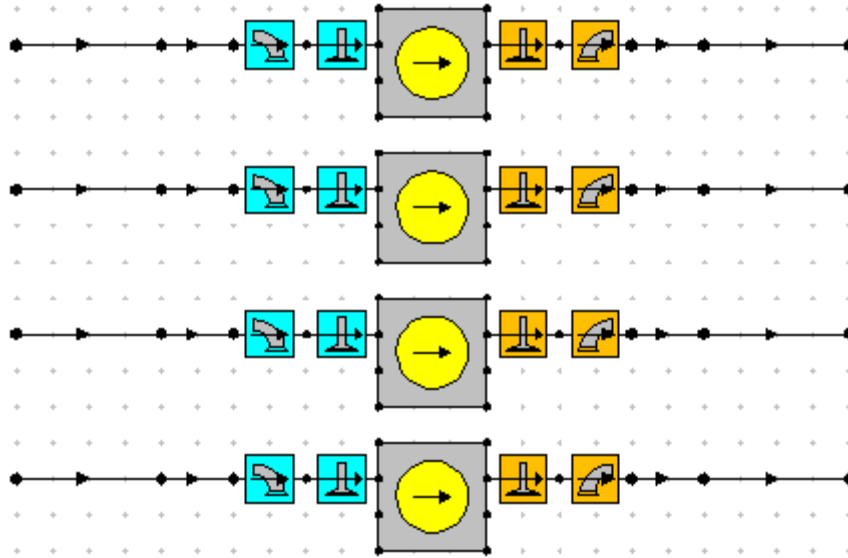


Figure 5.3. Model after 'pasting'.

To produce the four cylinder model we must first connect the free ends of the intake side together. Select in turn the free end of each of the top three inlet pipes and connect them to the free end of the lower inlet pipe. Select the **Intake Components** tab on the toolkit add select the plenum element. Connect the down stream end of the plenum to the free ends of the inlet pipes, (you may want to rotate the plenum round to do this, use either **rotate** option on the right mouse menu or use the Ctrl+PgUp/Ctrl+PgDn keys). Select the plenum and change its volume to **2.5** litres in its property sheet. From the **Intake Components** toolkit tab we now need to select a **throttle element**. Connect this to the upstream end of the plenum. You may want to rotate the throttle element. Now reconnect the inlet boundary to the upstream (free end) of the throttle. Your model should now look similar to that shown in Figure 5.4

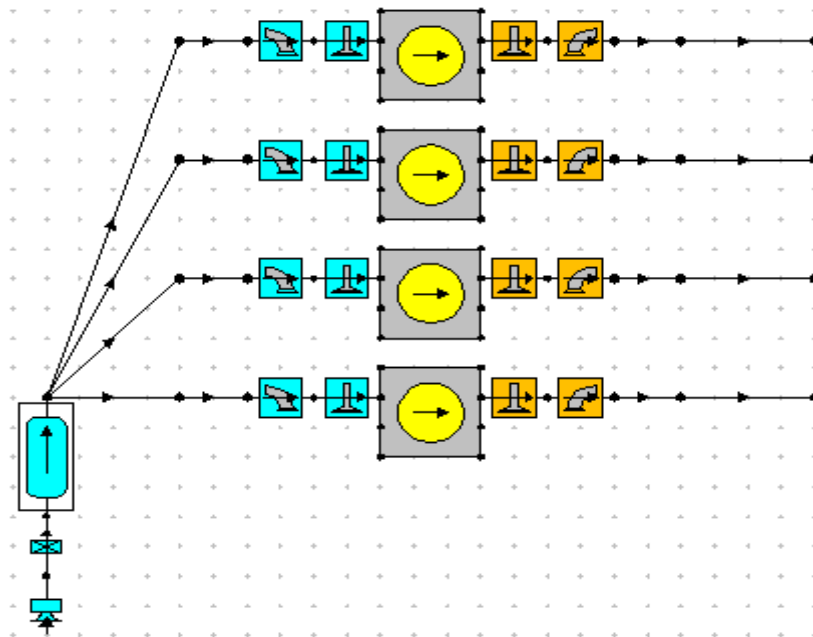


Figure 5.4. Inlet manifold connected.

Before leaving the inlet side we need to set some properties for the throttle. Select the throttle so that it is 'In-Focus', we will then set its cross sectional area in the property sheet. We can either enter the cross sectional area in directly, or define it by its 'equivalent diameter'. By default the area definition method is directly via the **Minimum C.S.A** value entry. This is indicated by the 'greying' out of the **Equiv. Diameter** value entry. To switch to defining by the equivalent diameter select the **Equiv. Diameter** descriptive text box, (note not the value entry). This will then enable the **Equiv. Diameter** entry and 'grey-out' the cross sectional area. Set the equivalent diameter to **45 mm**.

## Creating the 4-2-1 Exhaust Manifold

We will now produce the exhaust manifold. For this example we will simulate a conventional 4-2-1 style manifold. We must first connect the free ends of the existing exhaust pipes (primaries) together into two separate junctions. Take the free end of the pipe connected to cylinder 4 (the bottom cylinder) and connect it to the free end of the pipe connected to cylinder 2. Similarly connect the free end of the pipe attached to cylinder 3 to the free end of the pipe connected to cylinder 1, (the top cylinder). Your network should look similar to that shown in Figure 5.5

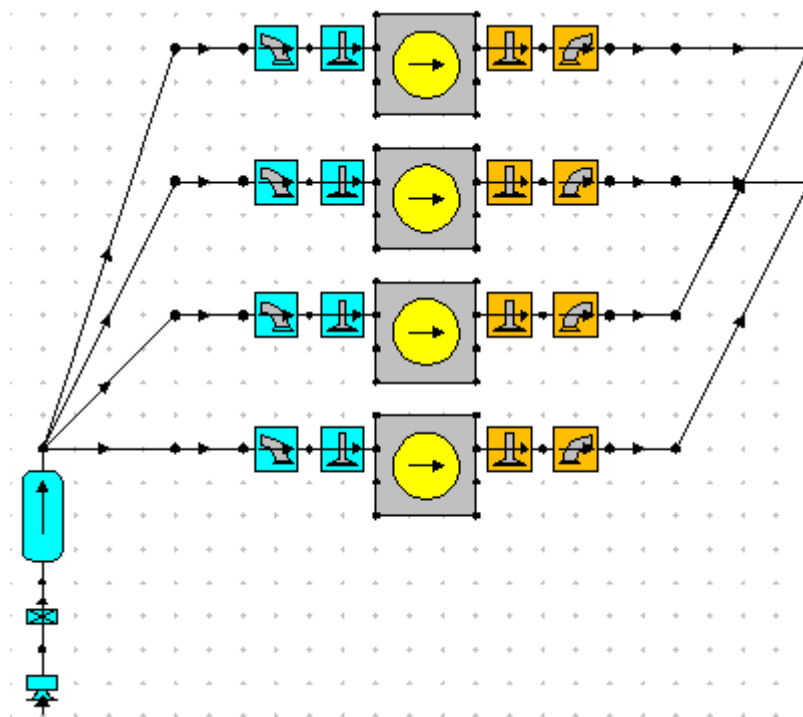


Figure 5.5. Connecting the exhaust primaries together.

We will now add the two secondary pipes to the model. From the **pipe** tab on the toolkit add two new pipes. Connect the upstream end of a pipe to each of these pipes to one of the primary junctions and then connect the two secondaries together at some point above and to the right of the primaries. You may need to manipulate the view using the **zoom** or **translate** options. We will now set the properties of the secondary pipes. Select one of the secondary pipes, to illustrate the use of the element labelling feature we will pick and edit **Pipe 17**. To do this use the element selection box in the top right of the interface to select the required pipe. The required pipe will now be 'in-focus' and we can define its properties. These component labels can also be displayed in either full or abbreviated form on the actual model. The component selection box uses the abbreviated labels whilst the property sheet lists the full label. To display the abbreviated labels on the model select the pull down menu item, **View / Visibilities / Abbrev. Label Visibility**. This will then show the labels for all the model components.

The 'Abbrev. Labels' are also used as the label by which copying of component data between elements is identified. We will use this feature later, so for now leave the labels visible.

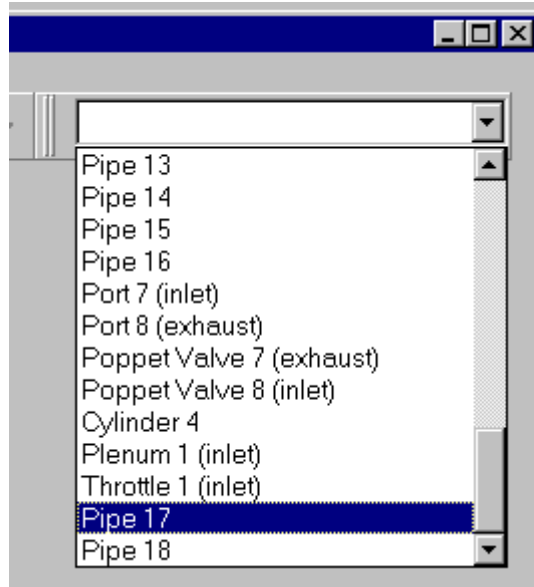
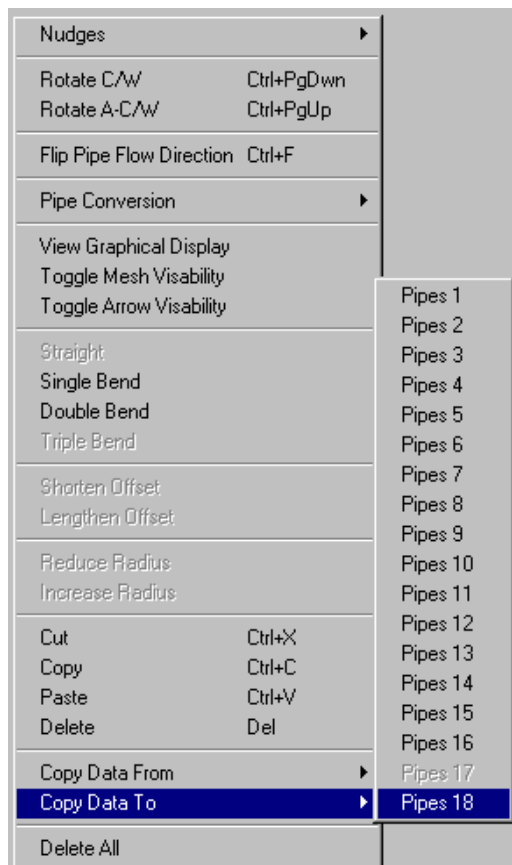


Figure 5.6. Component selection box.



With Pipe 17 in-focus set the values in its property sheet as, length **260 mm**, start and end diameters as **41 mm**. Leave all other values as the defaults. We can now copy the properties of this pipe to the other secondary pipe, pipe no. 18, (you can confirm this is the required pipe from the visible labels). To do this ensure pipe 17 is 'in-focus', (i.e. selected and its properties displayed), then use the right mouse (on the network display area) and select the **Copy Data To / Pipe 18** option. This will update the values of the second secondary pipe for us. You can confirm this copy has occurred by selecting pipe 18 and viewing its values for length and diameter.

Figure 5.7. Copying data between pipes.



The exhaust side of the model should now look like that illustrated in Figure 5.8.

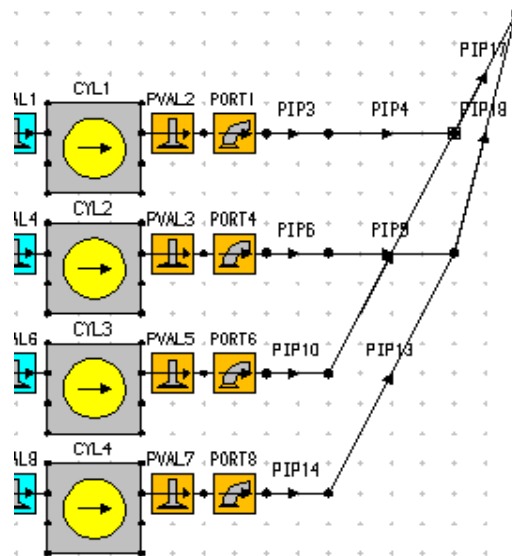


Figure 5.8. Model with exhaust secondary pipes added.

We now require to add the final pipe, a plenum to represent the silencer/catalyst and then the final throttle before connecting back in the existing exhaust boundary. From the pipes tab on the toolkit select a pipe and add its upstream end to the free end of the exhaust secondary pipes. Position the free end of this new pipe a couple of grid squares up from its other end. Set its length to **300 mm** and its start and end diameters to **53 mm**. Select the **Exhaust Components** tab on the toolkit and pick an exhaust plenum. Add this to the model connecting its upstream end to the manifold pipes free end. You may want to rotate this component to improve the visual appearance of the model and aid connection. Set the plenum's volume to **3.0l** and accept all the other data defaults. To the free end of the plenum connect a throttle element, (observing flow direction arrows as normal) and finally reconnect the existing exhaust boundary. Finally set the exhaust throttle area to **1345 mm<sup>2</sup>**. Your model should now look similar to that shown in Figure 5.9.

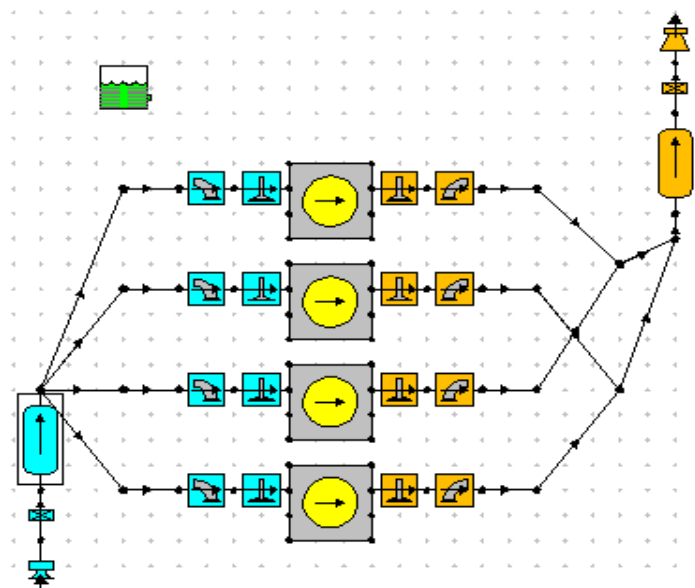


Figure 5.9. The completed network model.

## Setting the Firing Order

The act of 'copying' and 'pasting' cylinder groups as we have done, has the benefit of also copying all the associated component properties thus saving us re-entering properties. What this does mean is that all our cylinders currently have the same phase(firing) angle. This would be flagged as a 'warning' by the data checker when the job was submitted. To test this run the data checking wizard using the pull down menu **Tools / Data-check Wizard**. The display will list two warnings associated with firing order.

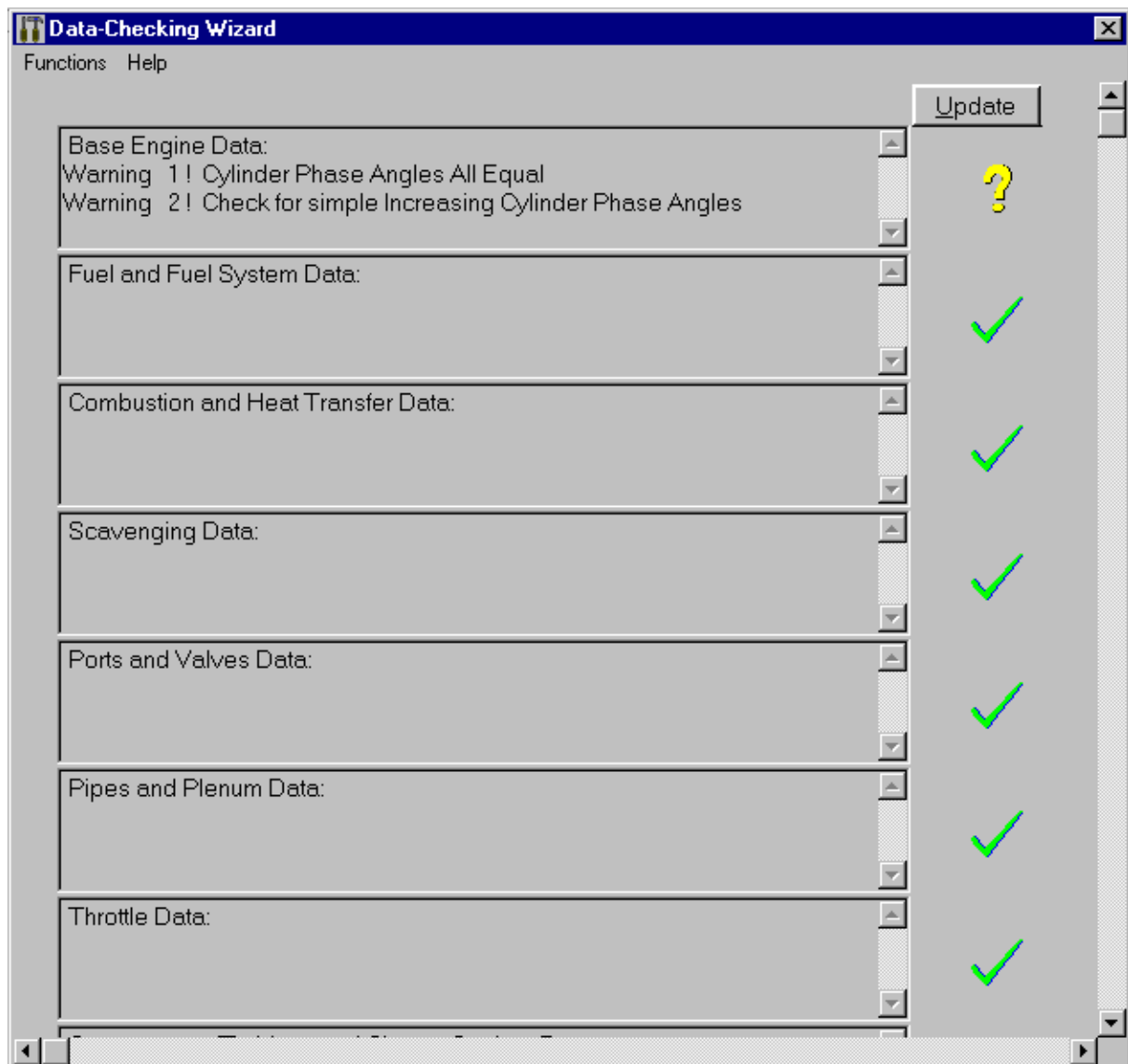
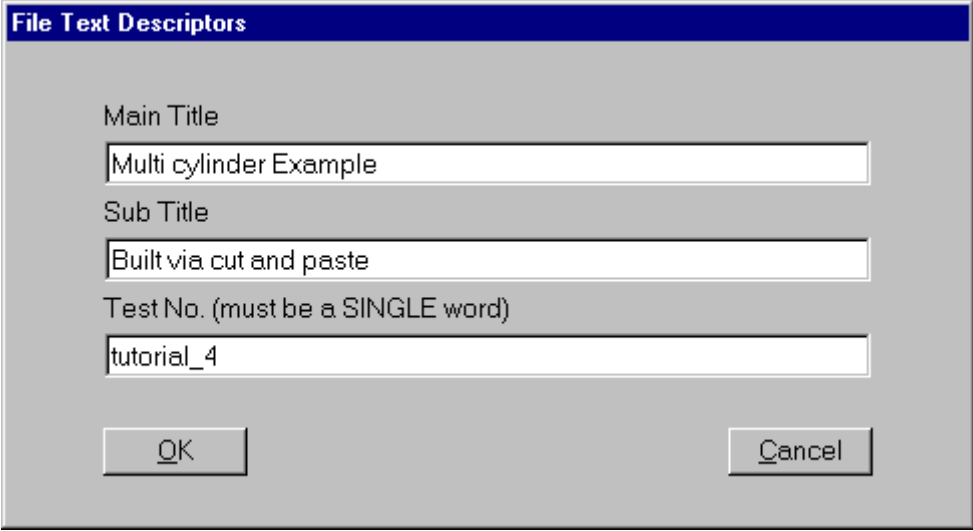


Figure 5.10. Data Checking – showing firing order warning.

We will set the firing order to 1-3-4-2, where the top cylinder (cyl1) is taken as cylinder 1. Select the individual cylinders and edit the phase value for cylinder 2 as **540**, cylinder 3 as **180** and cylinder 4 as **360**. Re-run the data checker to confirm that the phase angles are now acceptable.

## Running the Multi Cylinder Model

We will keep the test conditions that we defined in the earlier tutorials. Before we run the model change the model titles and test No. to reflect the model changes and identify the results as being for tutorial 4.



The screenshot shows a dialog box titled "File Text Descriptors". It has three text input fields. The first field, labeled "Main Title", contains the text "Multi cylinder Example". The second field, labeled "Sub Title", contains the text "Built via cut and paste". The third field, labeled "Test No. (must be a SINGLE word)", contains the text "tutorial\_4". At the bottom of the dialog are two buttons: "OK" and "Cancel".

Figure 5.11. Text descriptors.

Now run the analysis using the normal 'Launch' icon, having first 'saved' the model if you are using the running from '*existing saved data file*' option. It is not necessary to save the model before running if you are using the '*Currently Held Data*' option as the latest values will be extracted and used prior to job submission by the application, but it is good working practice to save the model changes before running.

This model will take 40 minutes to run on a 500 MHz machine, to get an approximation of the expected run time for your machine, you can scale this run time by the ratio of the processor speeds. The job status window will give you an indication of the run time (elapsed and remaining) as it progresses.

Once the job has run, load the graphical mrs results and display the resultant graphs.

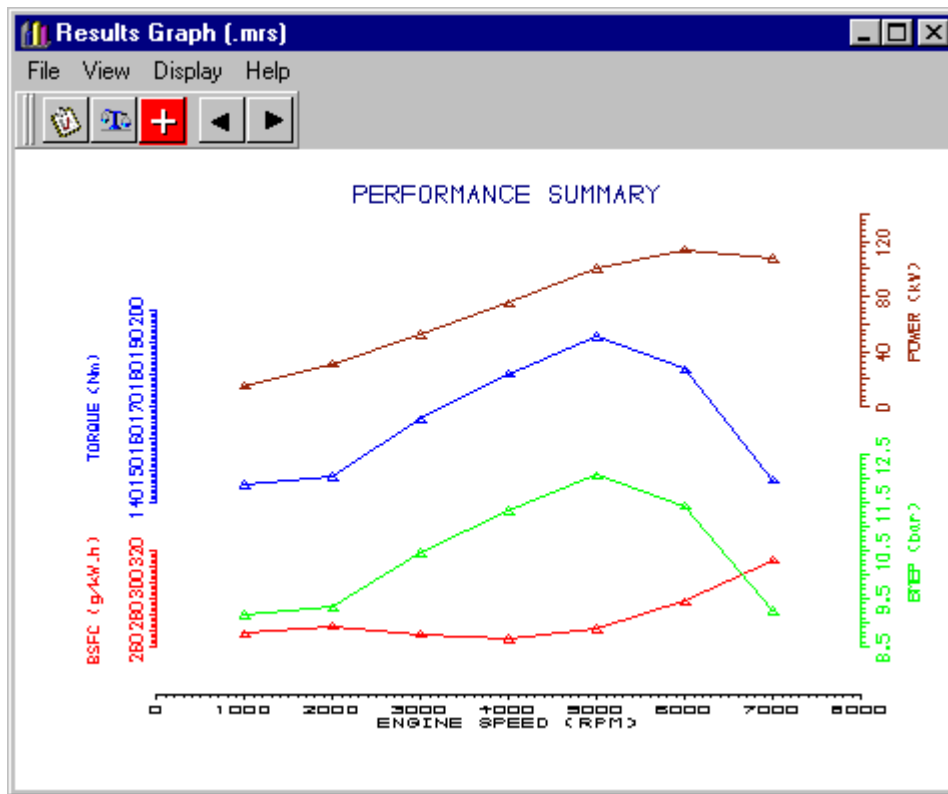


Figure 5.12. Tutorial 4 – standard junction results.

## Using the Pressure Loss Junction

The process of connecting the exhaust pipes together imply a default pipe to pipe junction type, the 'constant pressure junction'. This particular junction model type does not include any attempt to model the angular geometry of the pipe intersections. We can easily change to an alternative pipe junction model type that includes the effect of pipe angles on the pressure loss characteristics. This alternative junction model is the 'Pressure loss junction' and is added to the model as a 'draggable' element.

Select the pipes toolkit panel and locate the pressure loss junction element, (ringed in the figure below). Pick the pressure loss junction and add it to both of the 4-2 junctions.

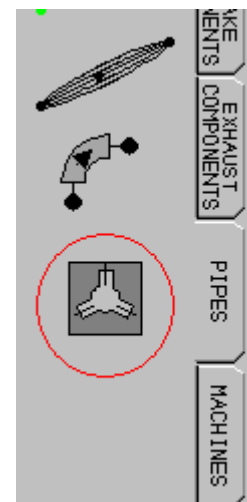
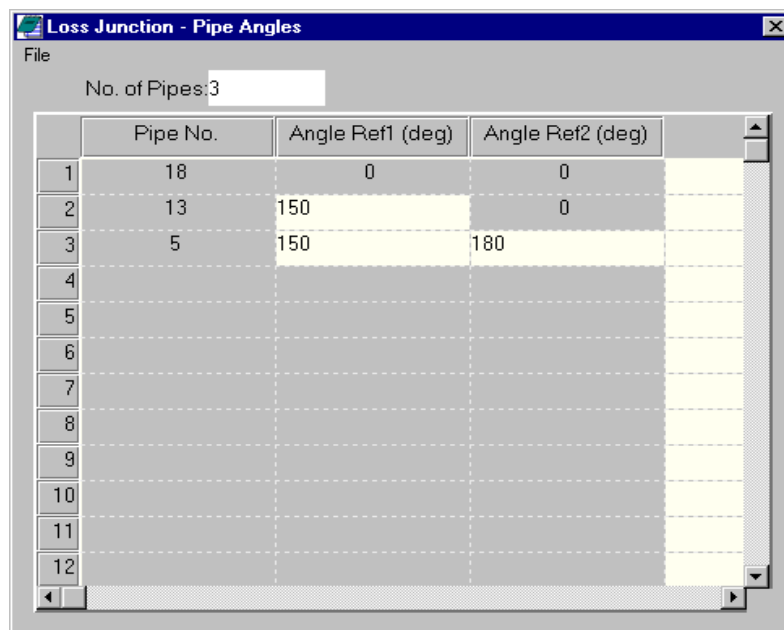


Figure 5.13. Loss junction element.

The pipe angle data for the pressure loss junctions are relative to reference pipes. You must identify which pipes should be used for these reference pipes. In our case we require pipes 17 and 18 to be the reference pipe 1 on loss junction 1 and 2 respectively. In this instance we do not care which of the other two pipes is used for reference pipe 2.

We now need to set the actual pipe angles used to represent the junction geometry. Select each loss junction in turn and select the *Angle Data* edit icon on the property sheet. Enter the angles relative to Ref1 as **150** degrees and relative to Ref2 as **180** degrees, as shown in Figure 5.14.



	Pipe No.	Angle Ref1 (deg)	Angle Ref2 (deg)
1	18	0	0
2	13	150	0
3	5	150	180
4			
5			
6			
7			
8			
9			
10			
11			
12			

Figure 5.14. Loss junction pipe angle data.

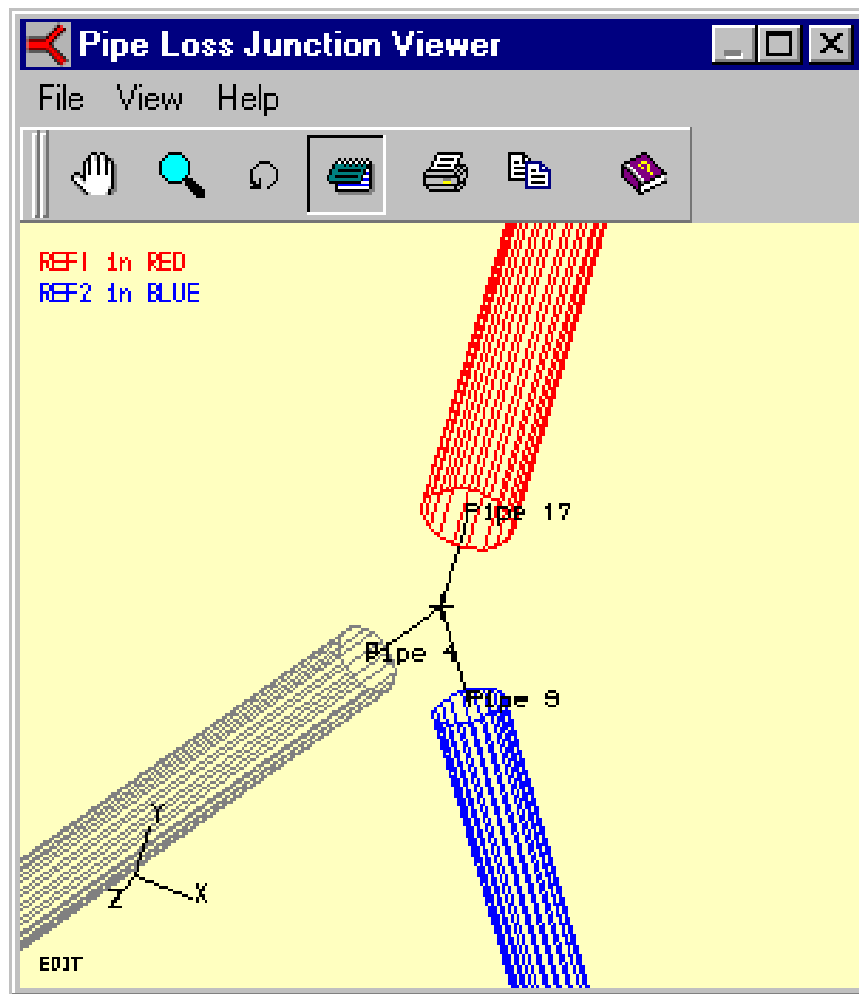


Figure 5.15. 3D Loss junction dynamic viewer.

To visualise the geometry we have just defined select the *Pipe angle Display* from the property sheet. The graphical display allows us to view the junction in 3d, (in edit mode we can also re-define the angle values).

## Running the Pressure Loss Junction Model

Before re-running the model, edit the *Test No.* in the 'File Text Descriptors' and change it to **tutorial\_4b**, then save the model file as **tutorial\_4b.sim**. Submit the job using the *Launch* icon. When the job has finished, (again approx. 40 minutes on a 500 MHz machine), load the mrs results and cross plot them against the original model. (Hint: the easiest way to cross plot files is to use the menu items on the *Results Graph (.mrs)* dialogue box, these include **File / load latest (position) ...** ).

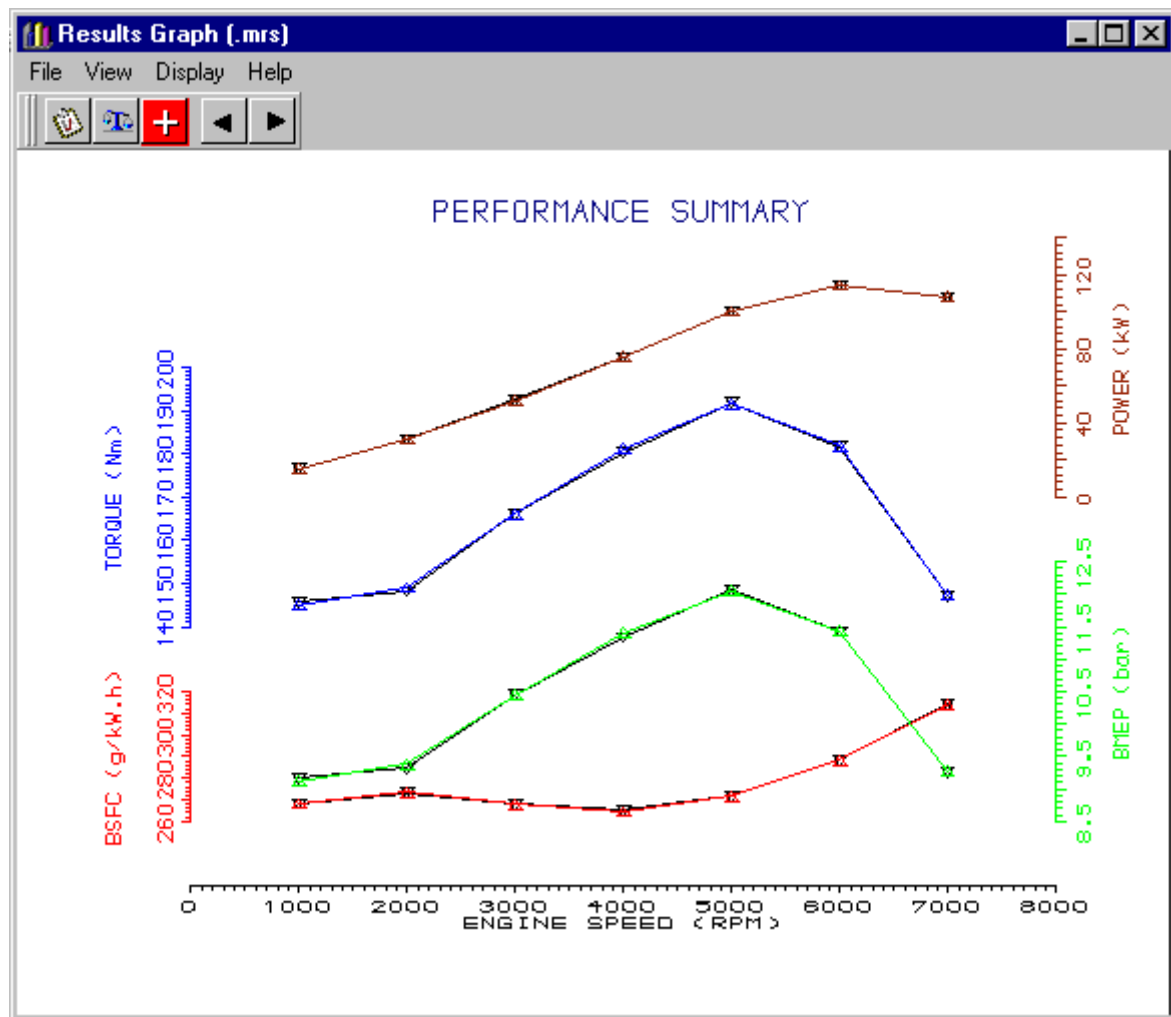


Figure 5.16. Comparison of pipe junction cycle averaged results.

The comparison of the overall performance levels shows relatively small changes in the predicted performance. There are however significant changes in the gas dynamics within the exhaust pipes, these can be investigated using the prs results viewer by cross plotting the two sets of prs results for a particular speed.

Figure 5.17 is for the 7000 rpm speed condition for *Loss Junction 1*. To display the pipe animation change to the prs display module, (**Module / Results Viewer**) load the two required prs files (if not already loaded). Select pipe 9 either from the element selection box or directly from the network model. Ensure that the required animation variable (in the example this is pressure) is '*on display*' by selecting the required graph with the right mouse button and pick **Display Selected Graph** from the menu options. Again with the right mouse select from the menu options **Display Pipe Graphic**. This opens the dialogue box illustrated below.

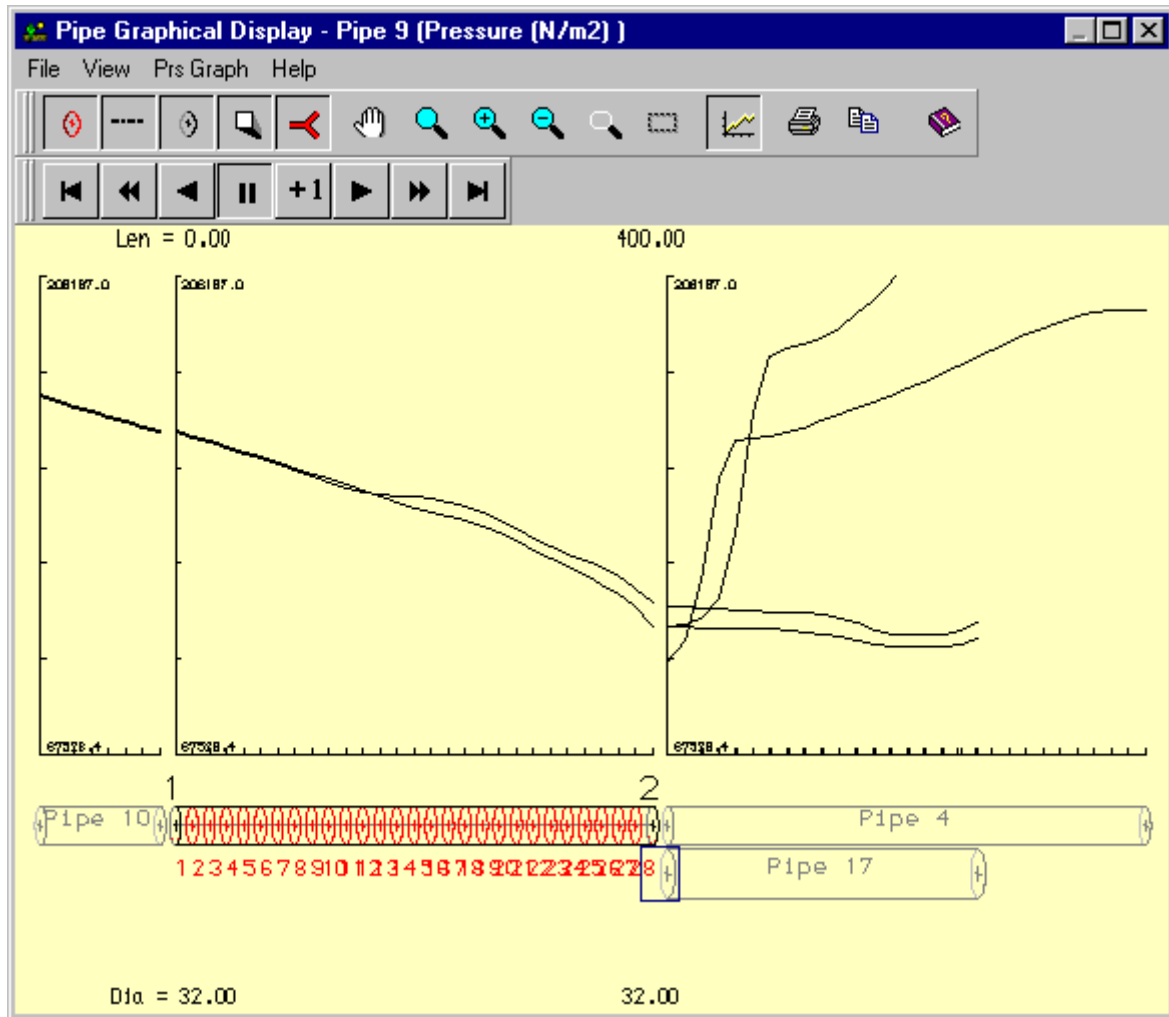


Figure 5.17. Effect of loss junction model - example of instantaneous pipe pressure results.



# 6

## Tutorial 5. 'Using the Concept Tool'

### Overview

This tutorial shows the user how to produce a multi cylinder model directly from the concept tool.

We will produce a 4cyl model similar to that created in the previous tutorial but based on the standard rules built in to the concept tool.

The tutorial includes the following sections:

- Opening the Concept Tool, 72
- Entering the minimum Data requirements, 74
- Setting Data Values Directly, 74
- Editing the User Defined Rules, 75
- Changing the Warning Limits, 76
- Selecting Inlet and Exhaust Options, 77
- Creating the Simulation Model from the Concept Tool, 78
- Concept Tool Test Points, 79
- Running the Concept Tool Model, 79

## Opening the Concept Tool

The concept tool can be opened either directly from the opening start-up wizard or from within the main engine simulation application. To open the concept tool from the start-up wizard check the **Concept Tool** toggle in the *Simulation tools* panel, then select **Ok**.



Figure 6.1. Selecting the Concept Tool from Start-Up.

If already in the main engine simulation application select the pull down menu item **Tools / Concept Tool...** or from the toolbar the Concept tool Icon, (ringed in the figure below).

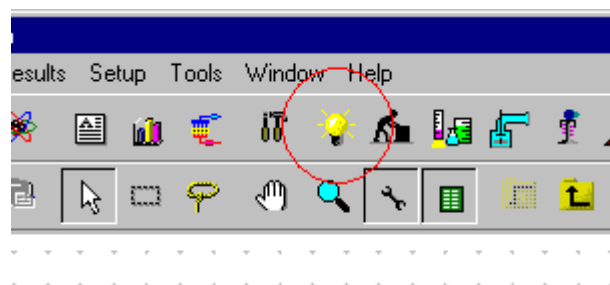


Figure 6.2. Opening the Concept Tool from the Toolbar.

The concept tool dialogue box will then be opened. The first time that you open the concept tool in a session you will be presented with an information box giving a brief description of the process involved with using the concept tool, see Figure 6.3.

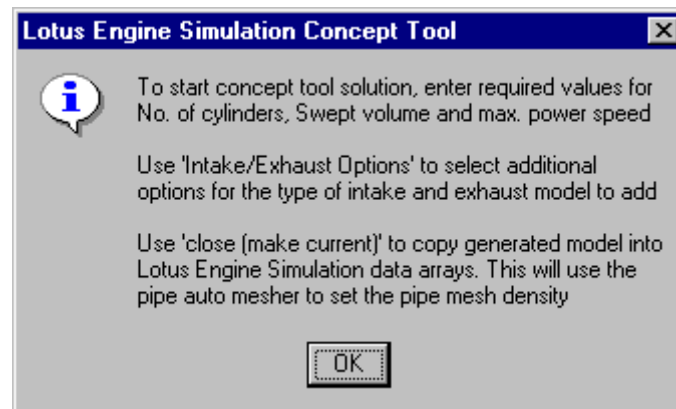


Figure 6.3. Concept Tool start up message.

Simply select **OK** to close this information box. The concept tool consists of two tabbed panels, we will primarily be concerned with the data in the **Standard Data** tab and in particular the dimensional data variables towards the top of this tab.

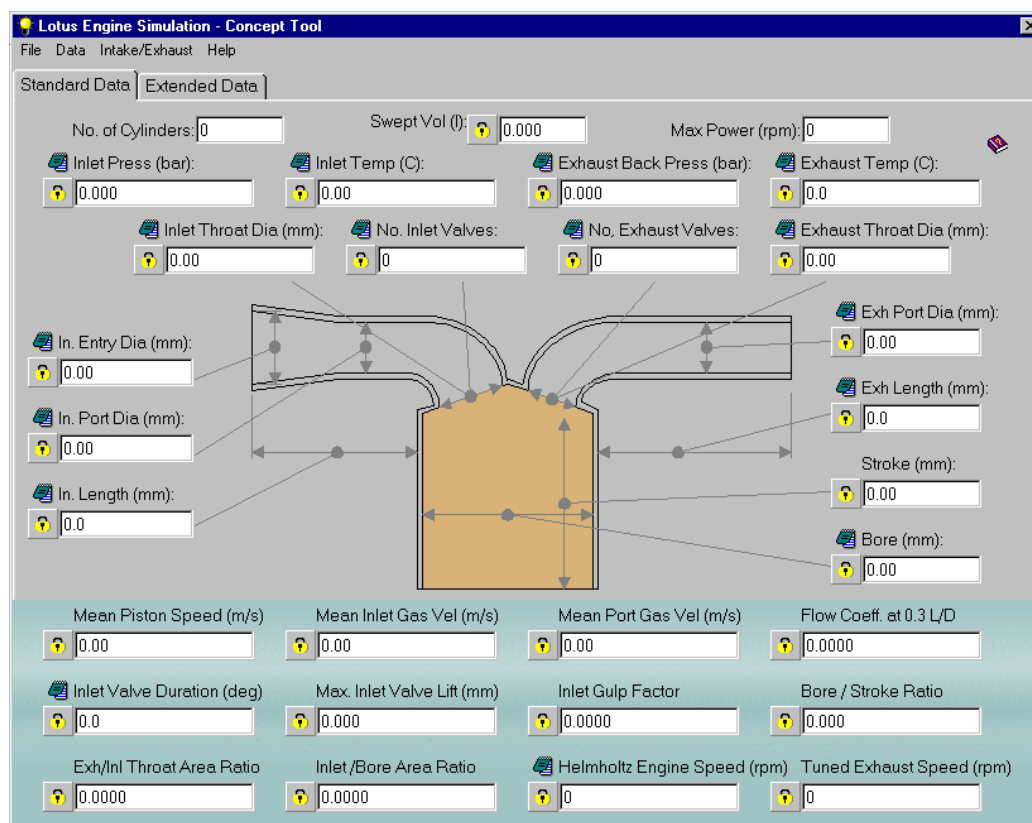


Figure 6.4. Concept Tool dialogue box.

The concept tool uses a number of empirical rules and classical relationships to produce the cylinder geometry from three basic numbers. The user must provide as a minimum these three values of **No. of Cylinders**, **Swept Volume (l)** and **Maximum Power Speed (rpm)** to enable the cylinder model to be created. All other data variables are optional and the user can define these directly or take the calculated values. The overall layout is shown below prior to any data being entered.

We will use the concept tool to create a similar model to that generated in the previous tutorial. Set the No. of cylinders to **4**, the swept volume to **2.0** litres and the maximum power speed to **6000** rpm. Once the maximum power speed has been entered pressing return will perform the initial calculation populating all the data fields with the default values.

Any of the data fields can be set directly by the user. To illustrate this we will change the default value given for the bore of 86.0 mm to be the same as in our previous model. Change the bore to **87** mm and press return. Note that background colour of the *bore* data field is now coloured in mouve to indicate that it has been 'user defined' and the padlock icon next to it has been selected. To revert back to the default value we can simply un-select the padlock icon. With the bore value set to 87.0mm the display should appear as shown in Figure 6.5.

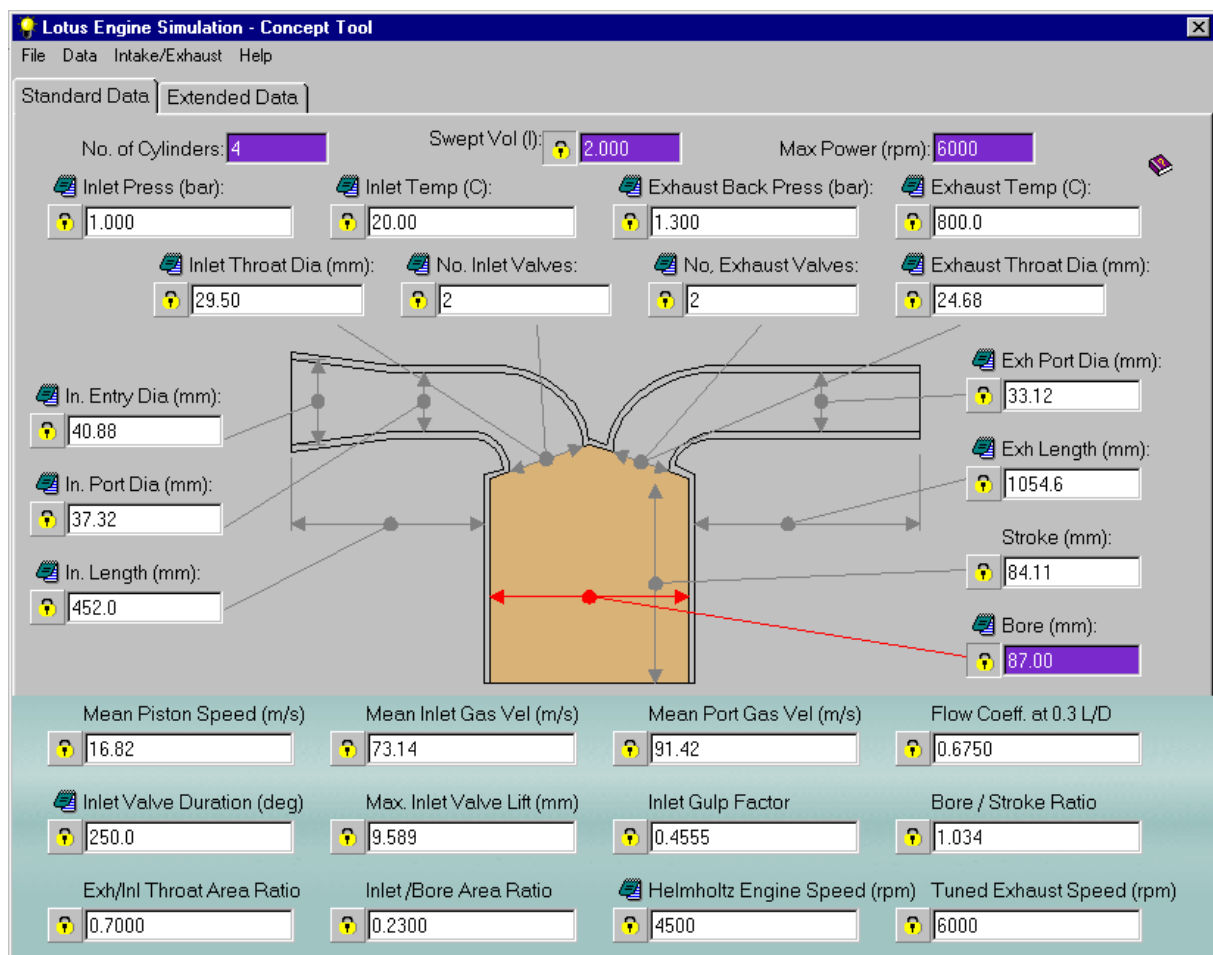


Figure 6.5. Setting the cylinder bore value directly.

## Editing the User Defined Rules

Each of the data fields that have a 'notepad' type icon next to them are defined by a *rule*. The notepad icon allows the user to edit that rule from the standard Lotus formula provided. This allows users to tailor the concept tool to their own company *standards*. We will change the user rule for the *Inlet Throat Diameter* to illustrate this process. Select the edit icon next to the inlet throat diameter data entry.

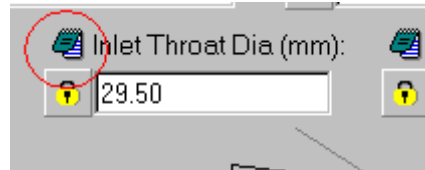


Figure 6.6. Icon to open user definable rules.

This opens the user definable function edit box as shown below. The function is defined using Fortran syntax based around using a number of available data fields. Each *field* is identified within the function by a F1, F2 type string, where the key for each is given in the *Available Fields* list. The Fortran functions supported by this tool are given in the *Supported Functions* list. The current is displayed in an editable text box with the default setting shown above it. We can see that currently we are using a function that is the same as the default. The default function is based on the cylinder bore (F4) and the number of inlet valves (F13). Edit the function to change the ratio value from 0.23 to **0.2222**. We can test the effect this has on the calculated value by using the **Test String** button. This will evaluate the string using the current values for F4 and F13 to determine the new value for inlet throat diameter. Select **Save** to save our change and return to the main concept tool window. The Inlet throat diameter should now display a value of 29.0 mm.

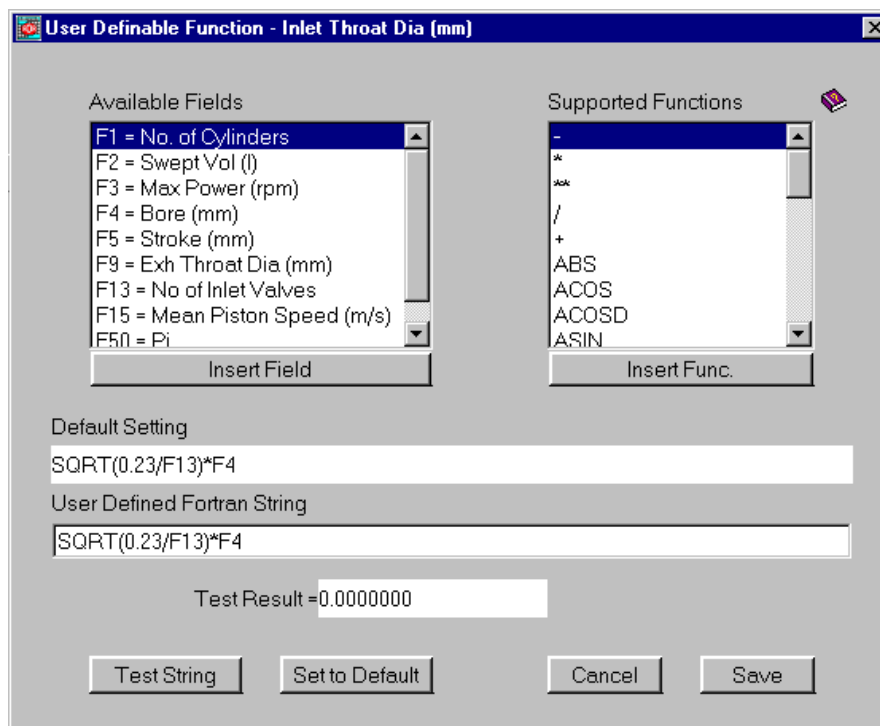


Figure 6.7. User definable functions editing.

## Changing the Warning Limits

The values displayed at the bottom of the window are calculated parameters rather than dimensional properties. These parameter fields will be highlighted in red if they exceed the current target limit. The program comes with a set of *Lotus* default values that can be changed by the user. To illustrate the use of warning limits we will deliberately change a data value to produce a warning.

Change the value for cylinder bore to **70** mm and press return. The display will now show the values of *Mean Piston Speed* and *Inlet/Bore Area Ratio* with a red fill background. The flagging of errors is controlled by a set of min/max limits, for which default values are provided. These defaults can be edited through **Data / Data Limits** menu. This opens the limit editor shown in the figure below. Scroll down through the list until you find *Mean Piston Speed (m/s)*, selecting this item will then display the default warning settings and the current user settings for this particular parameter. You can see that the default warning limit for mean piston speed is 20 m/s.

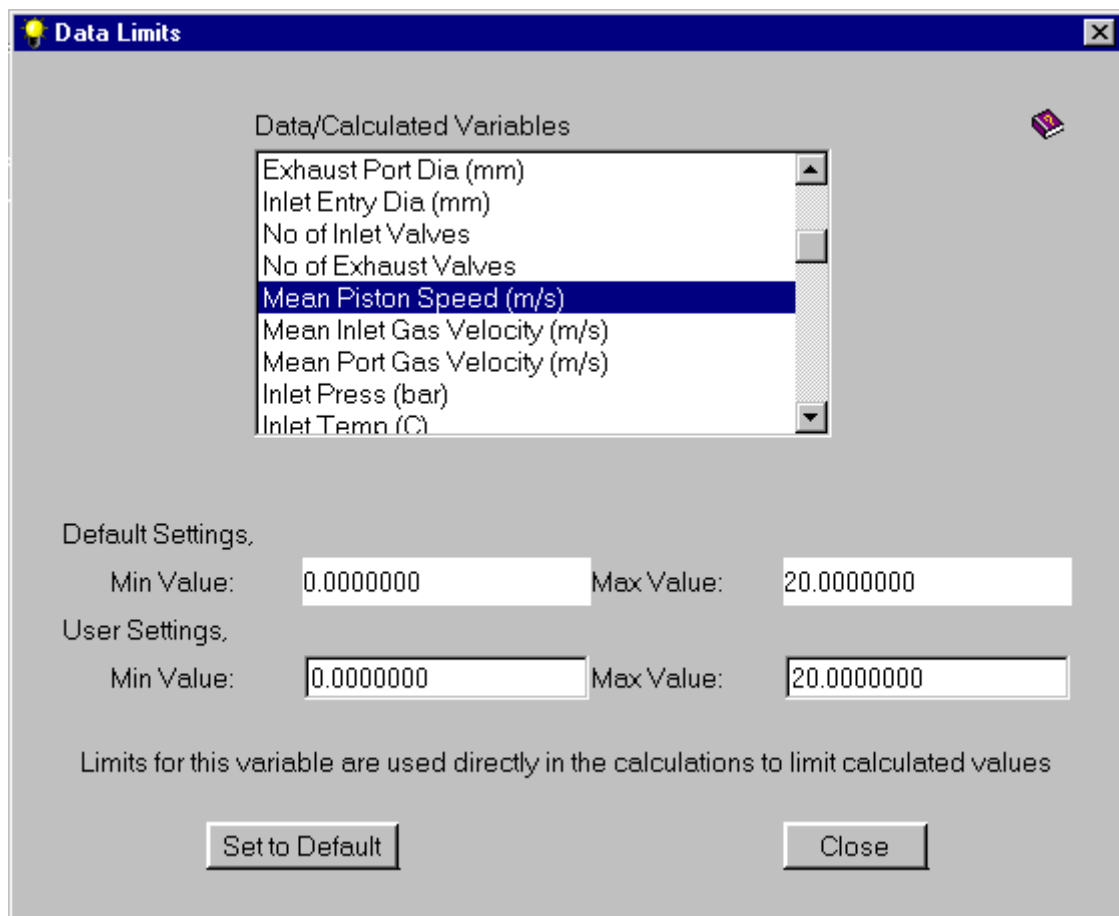


Figure 6.8. Setting and testing data limits.

Close the *Data Limits* dialogue box and return to the main concept tool window. Change the value for the *Bore* back to **87.0** mm and press return.

## Selecting the Inlet and Exhaust Manifold Options

Before we create our simulation model we can select the type of inlet and exhaust options that are applied to our defined *cylinder* model. The intake and exhaust options are set through a manifold wizard. To open this wizard select the **Intake/Exhaust / Intake/Exhaust Options** pull down menu, as shown in Figure 6.9.

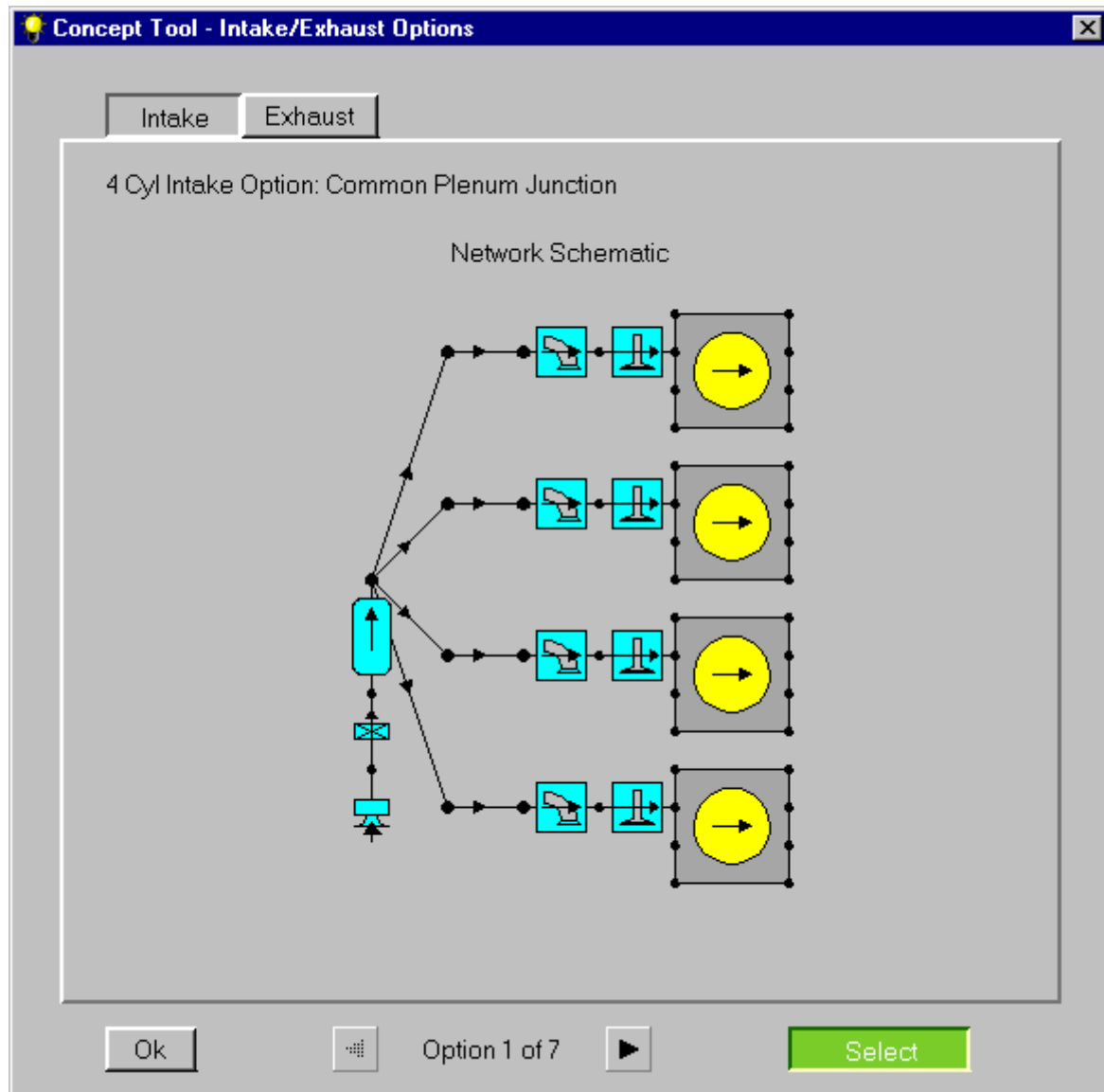


Figure 6.9. Selecting the intake option.

This wizard displays the intake and exhaust types available. The current selection is indicated by the **Select** button being indented, toggle through the inlet options using the arrow keys at the bottom of the window. Confirm that option 1 is set for the inlet. Change to the exhaust options by selecting the **Exhaust** tab at the top of the window and again toggle through the exhaust options to confirm that option 1 is selected. Select the **Ok** button to accept the choices.

## Creating the Simulation Model from the Concept Tool

We can now create an engine simulation model from the concept tool data. From the top menu bar select the **File / Close (make current)** menu item.

Accept the data loss warning. This will close the concept tool down and create the simulation model in the main interface. The display will autoscale to fit the newly created model, as shown in Figure 6.10.

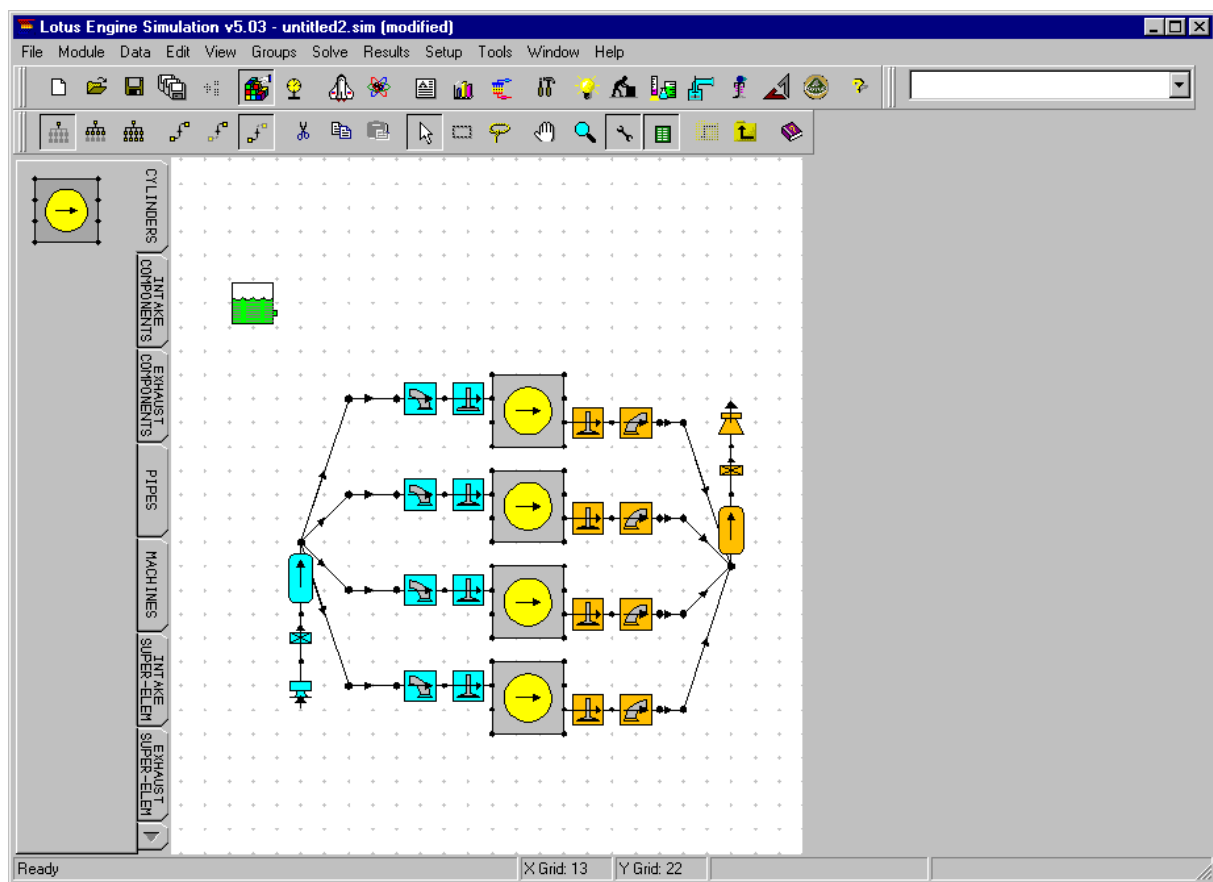
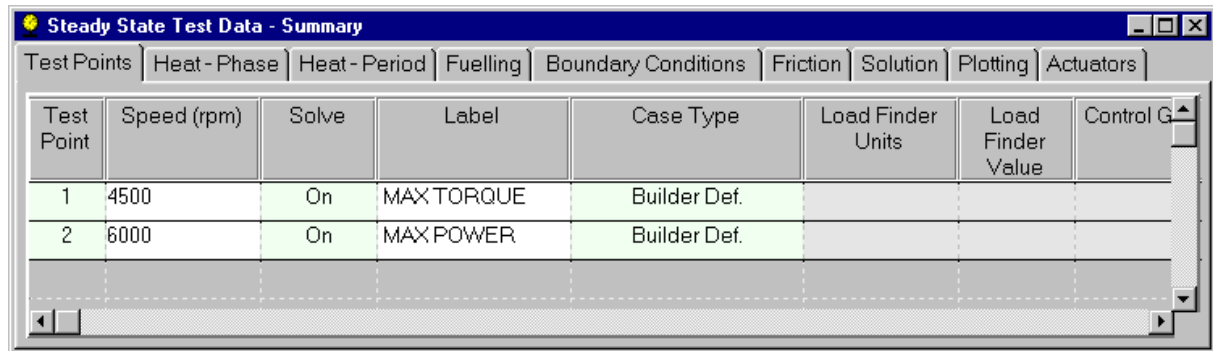


Figure 6.10. Concept Tool generated model.



## Concept Tool Test Points

The concept tool automatically creates two test points for us, the first for a maximum torque point and the second at our defined maximum speed condition. The maximum torque speed point is controlled by the **Helmholtz Engine Speed** data entry in the concept tool.



Test Point	Speed (rpm)	Solve	Label	Case Type	Load Finder Units	Load Finder Value	Control G
1	4500	On	MAX TORQUE	Builder Def.			
2	6000	On	MAX POWER	Builder Def.			

Figure 6.11. Concept Tool generated test points.

## Running the Concept Tool Model

Because of the limitations on exhaust options available with the concept tool the created model is not identical to that we created in the previous tutorial with a 4-2-1 exhaust. We could now modify the concept tool produced model to have the 4-2-1 exhaust as previously, but for the purpose of this exercise we will continue with the 'as-created' model.

Before we run this model open the text descriptors (**Data / File Descriptions...**) and set the *Test No.* to **tutorial\_5**, then save the model to **tutorial\_5.sim**.

We can now run our concept tool generated model in the normal way using the launch icon. Because of the reduced number of test points and fewer pipes this model will run in approx 6 minutes on a 500 MHz machine.

Once run load the mrs results into the graph display, (see figure below).

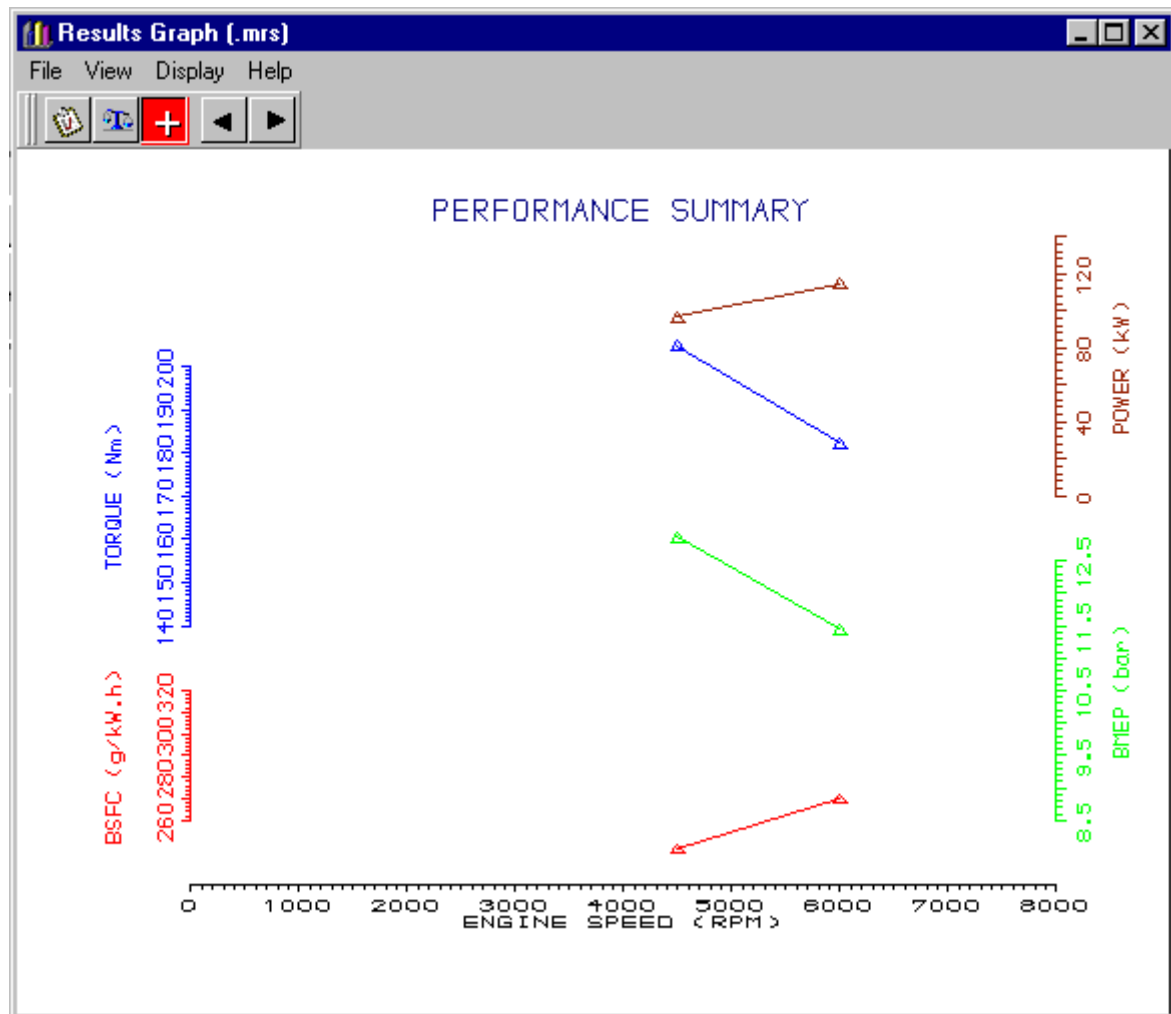


Figure 6.12. Concept Tool model .Mrs results.

# 7

## Tutorial 6. 'Using Sensors and Actuators'

### Overview

This chapter introduces you to Sensors and Actuators. We will use sensors and actuators to simulate an engine that has variable valve timing (VVT).

This chapter contains the following sections:

- Loading the Model, 82
- The Importance of Groups, 82
- Adding Sensors and Actuators to a Model, 84
- Using Control Elements, 86
- Running the VVT Model, 88
- Running the Model with Fixed Geometry, 89
- Exercise – Adding an Exhaust Cam Phaser, 89

## Loading the Model

For this tutorial we will use the existing model created in tutorial 4. Load the previously saved file **tutorial\_4.sim** and check it's the same as shown in the figure below.

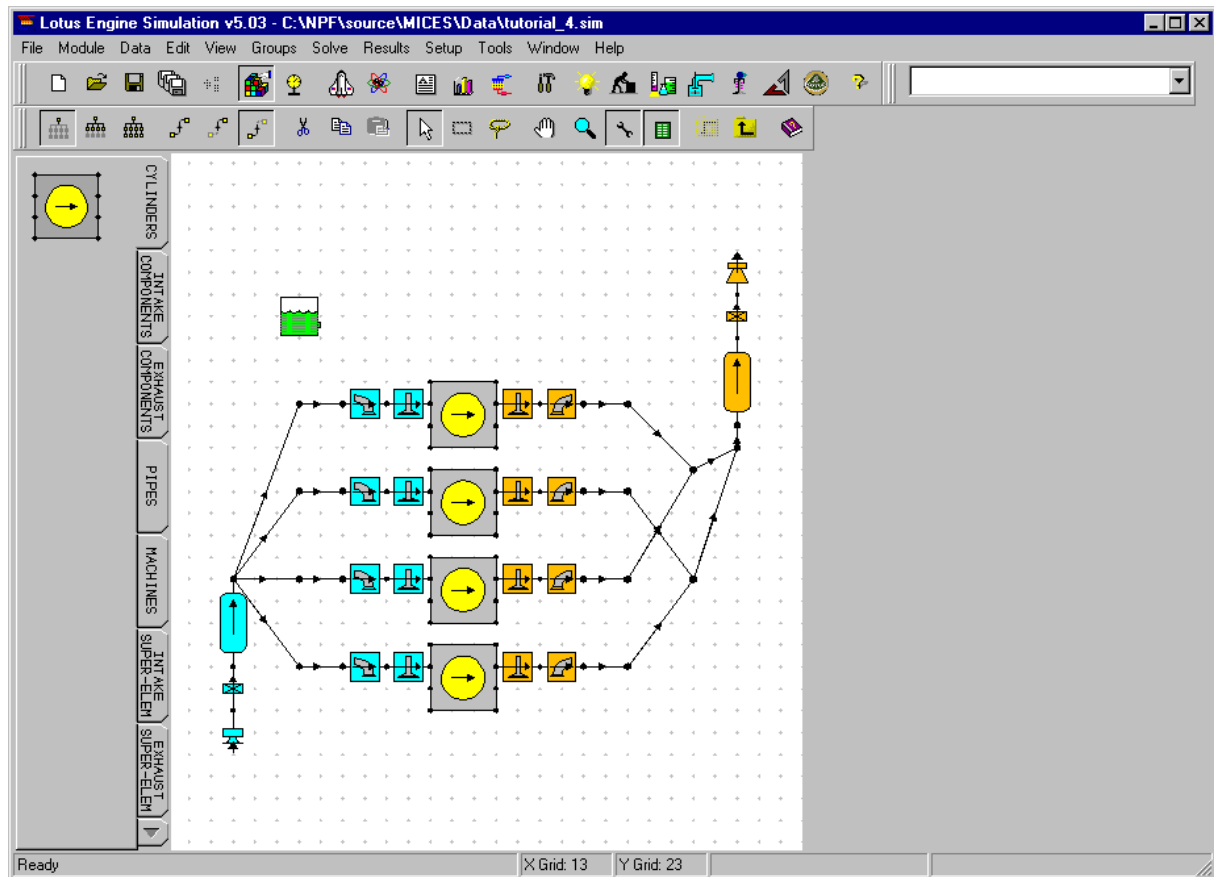


Figure 7.1. Loaded **tutorial\_4.sim** model file.

## The Importance of Groups

Sensors and actuators are added to the model in a similar manner to all the other components. To simplify the use of sensors and actuators they are normally applied to an element via its group association. Thus before we start to add the sensors and actuators we need to create the required groups.

To build our VVT model we will change the inlet valve timing as a function of engine speed, thus we need to be able to apply an actuator to each inlet valve element. This would need four separate actuators unless we make a group out of the inlet valves.

To create the group, select from the top menu bar **Groups / New**, enter the label for the new group as **Inlet Valves VVT** and select **Ok**.

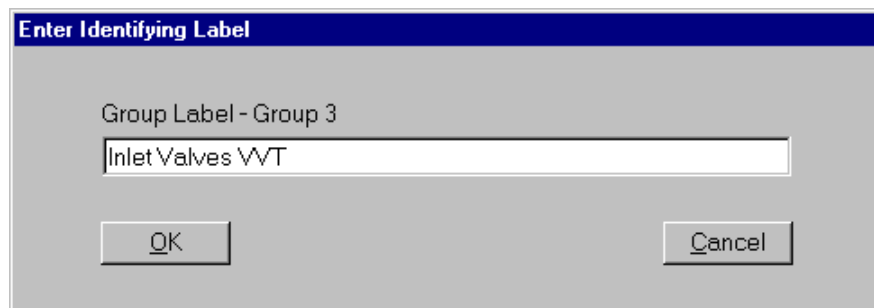


Figure 7.2. Defining the group label.

Select the **Edit / Pick Area** menu item or icon and pick an area that encompasses all of the inlet valves, (see below). We could pick and add each individual valve separately, but picking an area allows us to add them all in one go.

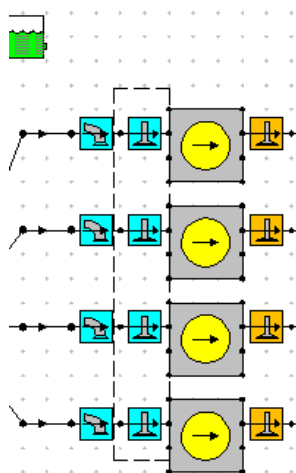
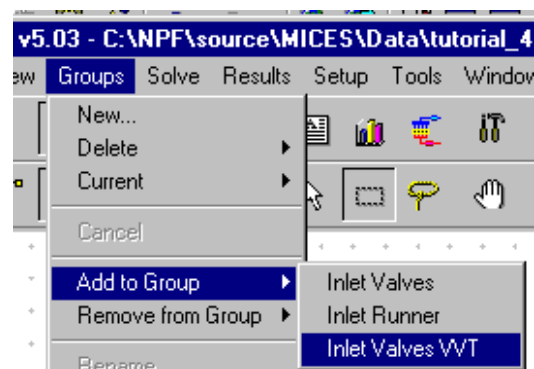


Figure 7.3. Selecting a group of elements.



To add the selected inlet valves to our group select the group menu item **Groups / Add to Group / Inlet Valves VVT**

To confirm successful creation of the group display the group using **Groups / Current / Inlet Valves VVT**. This should then just display our four inlet valves. To return to displaying the complete model select **Groups/Cancel**

### Adding Sensors and Actuators to a Model

We can now add the required sensors and actuators. We will attach a sensor to a cylinder to sense engine speed and an actuator to one of the inlet valves to vary inlet valve timing as a function of engine speed.

Sensors and actuators are not attached to the same connection points we use for linking the engine components together with, instead they are connected to the elements *harness connector*'. By default all element harness connectors are not visible, their visibility is turned 'on' individually by element as required. You can only connect to a harness connector if it is visible. The visibility switch can be found in the elements property sheet.

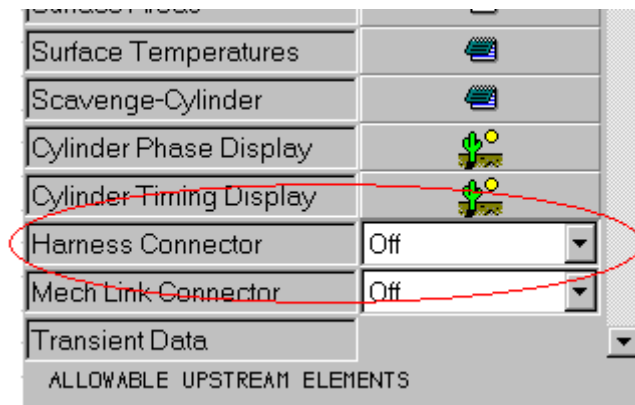


Figure 7.4. Setting the harness connector visibility.

Select **Cylinder 1** and make its harness connector **visible**. Select **Poppet Valve 1 (inlet)** and make its harness connector visible, see Figure 7.5 below.

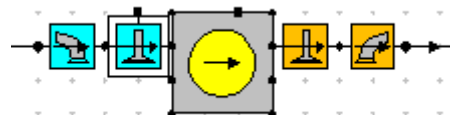
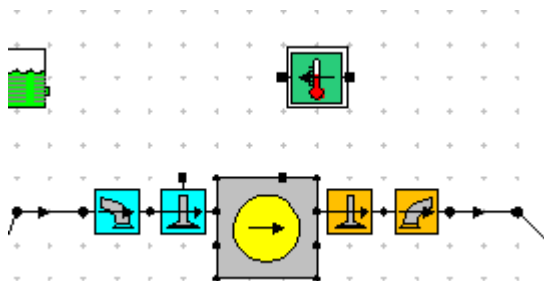
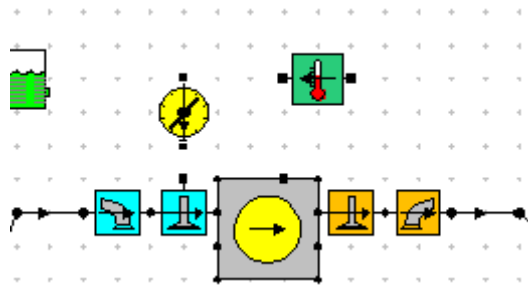


Figure 7.5. Harness Connectors Visible

To connect sensors and actuators to the model and to each other we use harness wires. These can be selected from the **Sensors and Actuators** toolkit tab.



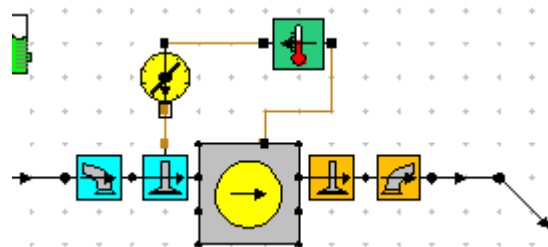
First we will add the required sensor and actuator, then we will connect them together with the harness wires. From the Sensors and Actuators toolkit tab select the sensor element and place it just above cylinder 1. Flip the sensor direction so that its output faces towards the inlet valve (hint to flip an element flow select **Ctrl+f** when the required element is 'in-focus').



Now select an actuator element and place it just above Poppet valve 1, again rotate the element so that it points towards the target valve's harness connector.

We can now connect the elements together using three harness wires.

Connect the cylinder harness connector to the input of the sensor. With the second harness wire connect the output from the sensor to the input of the actuator. Finally with a third harness wire connect the output from the actuator to the harness connector on the poppet valve. Your model should now look like that shown below.



Note that the first harness wire has been displayed as a 'double bend' style, as with ordinary pipes and virtual links this makes no difference to the properties of the element, it is purely to assist in element connection and improving the visual appearance.

We must now define which particular property of the cylinder we wish to sense and which valve property we wish to define with the actuator. Select the sensor and select from the *Sensed Parameter* list **Crank Speed (rpm)**. For this particular parameter we can leave the *Sensor Group Type* as **Single**.

Actuator Label	default actuator
Actuator Variable	
Actuator Group Type	Valve Open (deg) Valve Close (deg) Dwell at Max (deg) Max Lift (mm) <b>MOP (deg)</b> Opening Lash (mm) Closing Lash (mm) Valve Lift (mm)
Actuator Apply Type	
Actuator Solve Type	

Sensor Label	default sensor
Sensed Parameter	
Sensor Group Type	Pressure (N/m <sup>2</sup> ) Temperature (K) Volume (m <sup>3</sup> ) Mass (Kg) Gas Constant (J/Kg.K) Ratio spec. Heats Gas Viscosity (Kg/s.r) <b>Crank Speed (rpm)</b> Crank Angle (deg) Cycle No.
Sensed Element	

Now select the actuator and select from the *Actuator Variable* list **MOP (deg)**. Set the *Actuator Group Type* to **Group**. Set the *Actuator Apply Type* to **By Value** and set the *Actuator Solve Type* to **Per Cycle**.

We now have a sensor that detects engine speed and an actuator that updates all the inlet valves MOP (maximum opening point), at each cycle. We need to define the relationship between engine speed and valve MOP. This is done using control elements that are created as children of sensors and actuators and define how an input signal(s) is used to calculate the output signal.

## Using Control Elements

Control elements are children of sensors and actuators. To be able to see the children of a particular element we will introduce the concept of **Levels**. All of our existing model components can be considered to exist in one level (considered as level 1), with a single parent (taken as at level 0). Sensors and actuators, unlike the components we have worked with up to now, can have child elements associated with them at a lower level (considered as level 2). We can move between levels where relevant using the menu items, **Data / Down a Data Level** or **Data / Up a Data Level**, alternatively the equivalent icons can be used, (it is currently not possible to move to level 0).

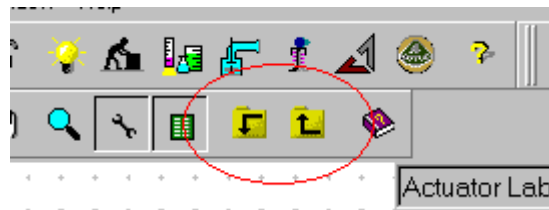


Figure 7.6. Level icons.

You can move down to level 2 with either sensors or actuators. For our particular example we need only add a control element to the actuator and allow the sensor to simply pass its sensed speed value unchanged.

If we select the sensor and move down to level 2, we will see two control components in inlet boundary and an outlet boundary. These get added automatically as children to a sensor whenever you add a new sensor. For our simple case of passing the value unchanged the inlet is left connected directly to the outlet. At this level the toolkit contents change to reflect the new parent, (only elements that can be children of the current parent are displayed). The toolkit now list 1D and 2D control elements. Return back to level 1.



Figure 7.7. Default sensor children.

Select the actuator and move down to level 2, again we have an inlet boundary and an outlet boundary that were added by default with the actuator. For our requirements with this example the simplest and most appropriate control element to use is the 1D lookup table. We use the 1D table as we only require the valve timing to vary as a function of 1 parameter, (engine speed). We need to insert the 1D lookup table control element between the inlet and outlet boundaries. Separate the boundary elements and connect in a 1D lookup table element, (see figure below). (Hint; remember to use the **Edit / Move By / Single** option to enable connected elements to be moved independently).



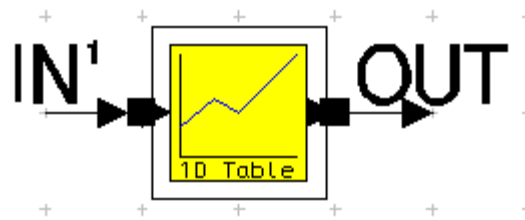


Figure 7.8. Actuator control elements after addition of 1D look-up table.

Next we need to define the properties of the 1D table. Select the 1D table component and pick the **LookUp Data** option from the property sheet. This opens the data spread sheet. Modify the default data supplied to have 8 points with X values (speed) of 1000, 2000, 3000, 4000, 5000, 6000, 6500 and 7000 and Y values (MOP) of 90, 90, 90, 95, 105, 120, 125, 125 (see table below).

1D Control Look-Up Table			
File			
No of Values: 8			
	X	Y	
1	1000	90	
2	2000	90	
3	3000	90	
4	4000	95	
5	5000	105	
6	6000	120	
7	6500	125	
8	7000	125	
9			
10			
11			
12			

Figure 7.9. Actuator 1D look-up table data.

Once the data has been entered close the table by selecting **File / Close**. In the property sheet set the *LookUp type* for the actuator to **Interpolate Only**, this uses linear interpolation within the defined range, but won't extrapolate beyond the defined range instead using the first or last value as appropriate.

Return back up to level 1. In the text descriptors (**Data / File Descriptors**), change the *Test No.* to **tutorial\_6**. Close the text descriptors box and now save the file as **tutorial\_6.sim**. We are now ready to run the VVT model.

## Running the VVT Model

Submit the job in the usual way, again as in tutorial 4, this will take about 40 minutes to run on a 500 MHz machine. Once the run has finished load the mrs results into the mrs graphs.

Open the mrs results graph and load into position 2 the results from tutorial 4. We can now identify the benefits of variable inlet valve timing compared to a fixed valve geometry engine.

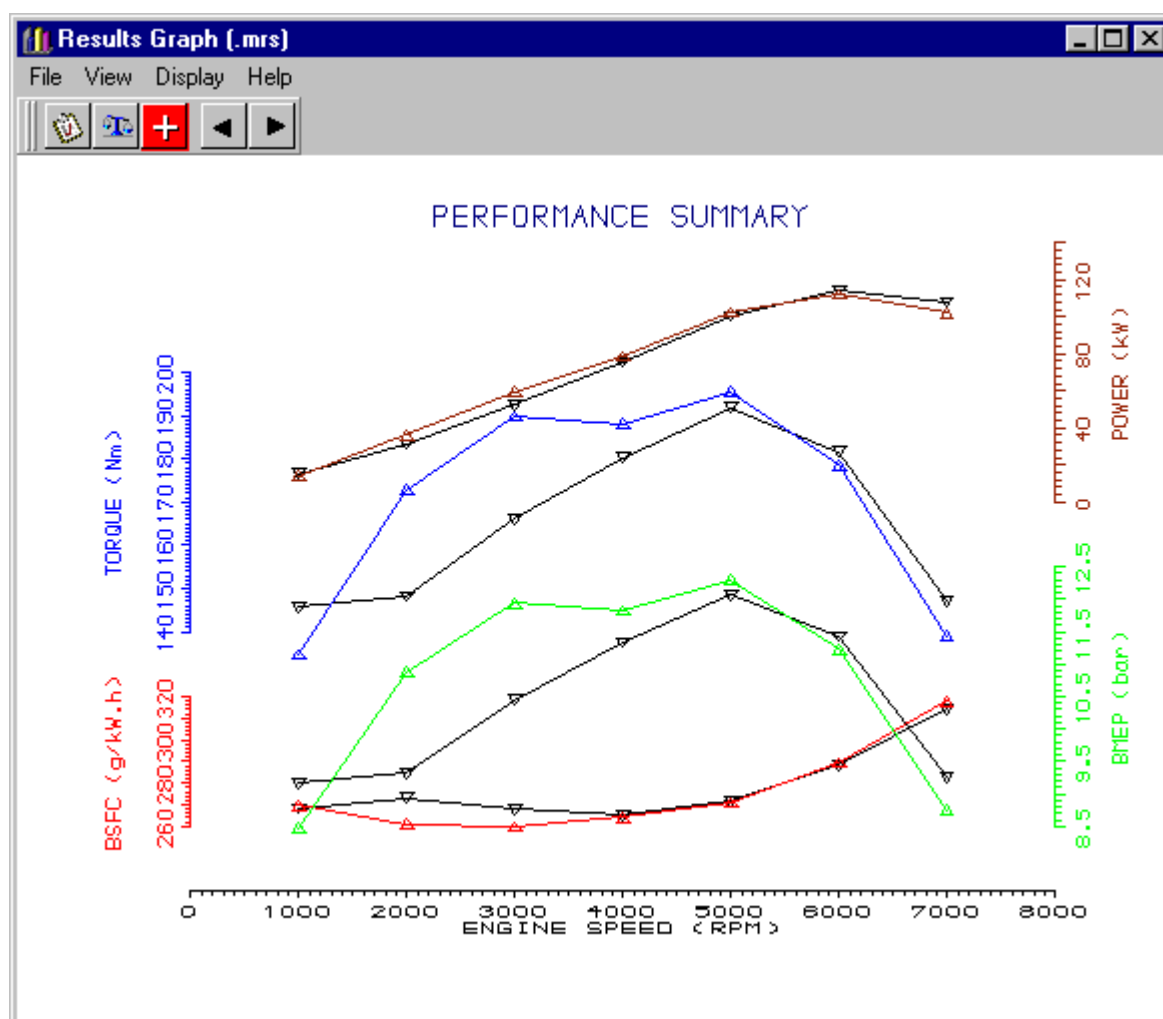


Figure 7.10. Comparison of VVT model with Tutorial 4 fixed geometry model.

The VVT model shows significant improvements in the low speed region, but loses out slightly in the upper speed region, this implies we have not picked an overall 'optimum' timing curve for the VVT engine.

## Running the Model with Fixed Geometry

It is possible to enable/disable individual actuators for different steady state test points. Thus although we have now built a VVT model, we could run the original 'fixed' geometry engine without needing to delete the sensors and actuators. These settings are controlled through the test conditions summary spreadsheet. From the menus select **Data / Test Conditions / Steady State Test Data Summary....**, or alternatively use the F12 function key. This opens the test conditions summary used previously. Select the **Actuators** tab.

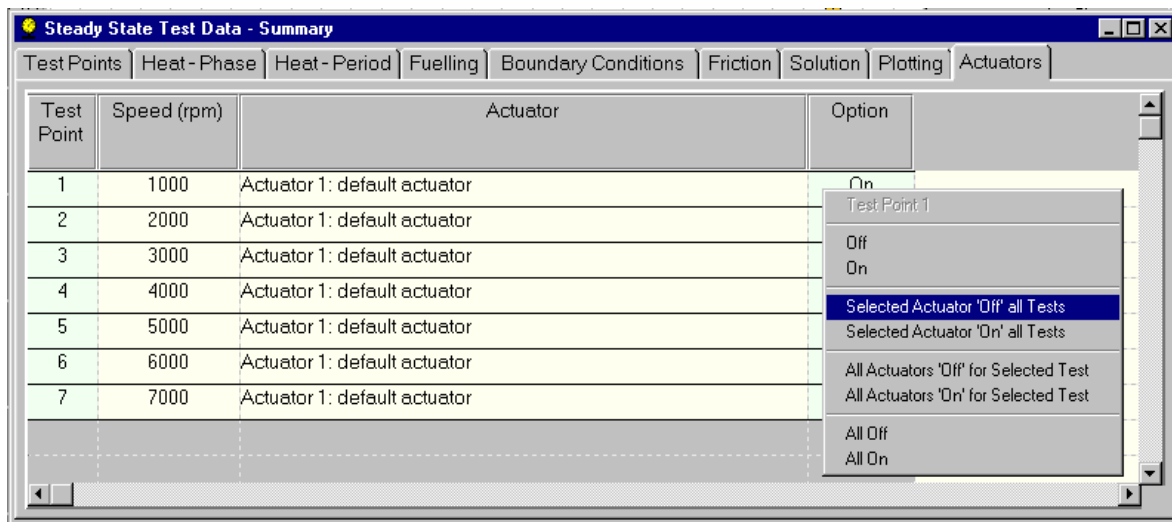


Figure 7.11. Steady State Test Conditions – Actuators Tab

Each actuator is listed against each test point. Selecting the **Option** column with the left mouse button provides the option to switch the individual actuators on or off. If you select **All Off** this will disable the actuators for all test points and revert back to our original fixed geometry valve data.

### Exercise Adding an Exhaust Cam Phaser

As a further example add an exhaust cam phaser. You will need to create a new group for the exhaust valves, add a second actuator, (you can use the same sensor for engine speed). Add the required control element and define suitable values. Remember that exhaust MOP's are –ve.

You may want to try improving the inlet cam phaser, such that it is at least as good as the fixed geometry engine, in the upper speed region.



# 8

## Tutorial 7. 'Using the Load Finder'

### Overview

This chapter introduces you to the concept of the Load Finder. A utility that modifies the properties of an element or group of elements to match the output load to a target value.

This chapter contains the following sections:

- Loading the Model 92
- 'Load Finder' Objectives 92
- Adding the Throttle to a New Group 93
- Setting Up the 'Load Finder' Cases 93
- Monitoring the 'Load Finder' 96
- Running the 'Load Finder' Job 98
- Reviewing the 'Load Finder' Results 99
- Exercise – Virtual Links 99

## Loading the Model

For this tutorial we will again use the existing model created in tutorial 4. Load the previously saved file **tutorial\_4.sim** and check it's the same as shown in Figure 8.1.

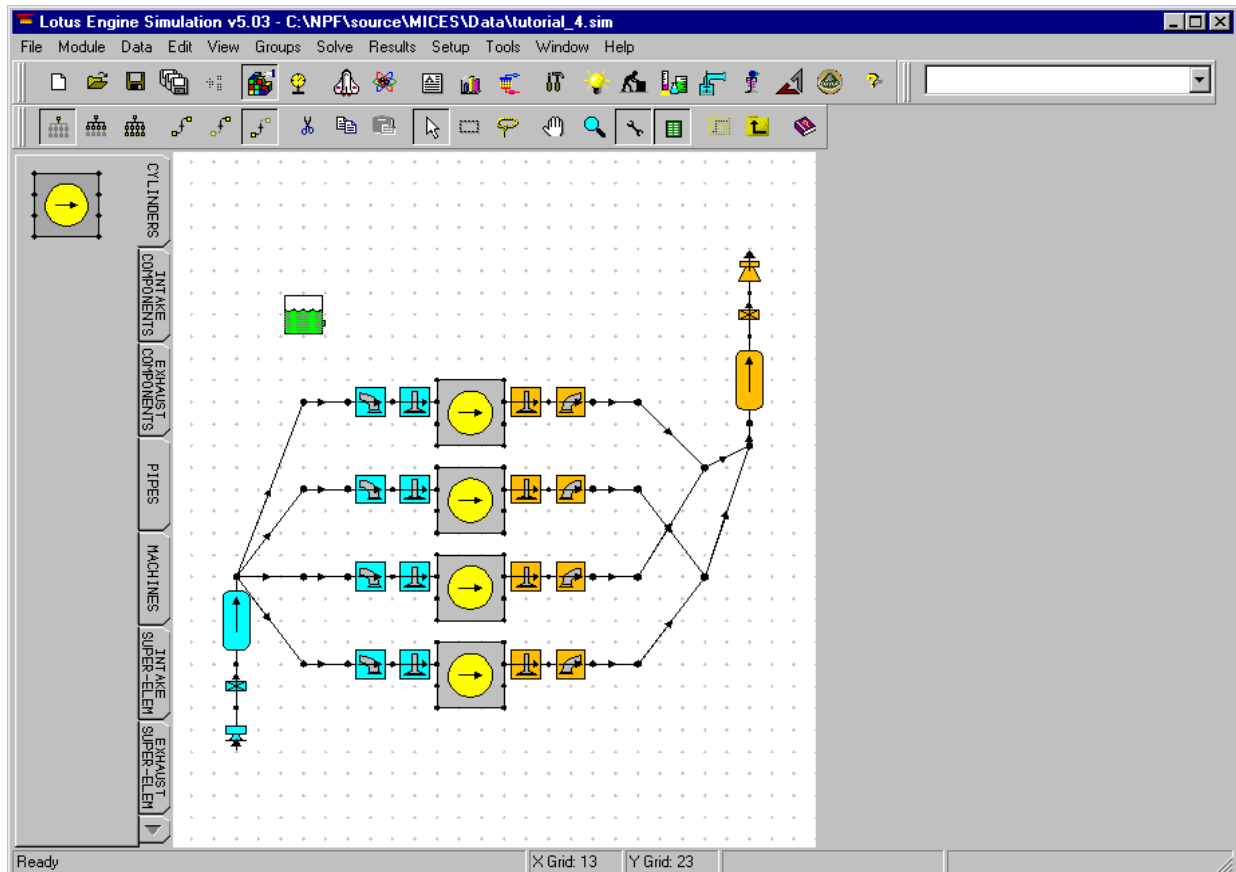


Figure 8.1. Loaded model.

## Load Finder Objectives

Until now, all our tutorials have been based on building models to run wide open throttle (WOT)/full load type analyses. The objective of the load finder is to take our model and run it a load condition other than that achieved with the 'as-built' engine model. Normally this would mean running our WOT model at some part load condition. To do this the Load Finder uses an element, or group of elements, as the load control and varies a particular property to set the load. Perhaps the simplest case that can be envisaged is that of a simple intake throttle component, and varying its minimum cross sectional area. A slightly more complex load controller could be the simulation of a throttle-less engine using poppet valve timing as the control.

All our normal sensor and actuator functionality is available to use during a 'load finder' run, so we can build additional control inter-dependency into our part load 'load finder' models.

It needs to be remembered that as you change the output load the validity of using fixed values for some of the models properties is debatable. This is particularly true of combustion data where the combustion phase and duration may vary with load. This variation of the combustion parameters can

be added to the model through the use of sensor and actuators attached to the cylinders. For the purpose of this exercise we will ignore any influence of operating condition on the combustion.

## Adding the Throttle to a New Group

To demonstrate the use of the Load Finder we will create a new group and add the inlet throttle to it. Create a new group using **Groups / New...**, set the group label as **Throttle**, as shown in Figure 8.2.

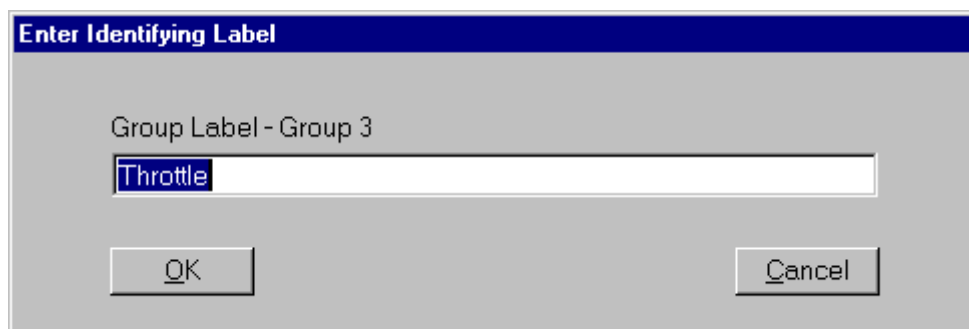


Figure 8.2. New throttle group.

Select the intake throttle with the mouse, and with it in focus add it to our new group using **Groups / Add to Group / Throttle**.

## Setting Up the 'Load Finder' Cases

The load finder run is initiated from the test data summary spread sheet. Open this spread sheet up using either the **F12** function key or the pull down menu **Data / Test Conditions / Steady State Test Conditions Summary...**

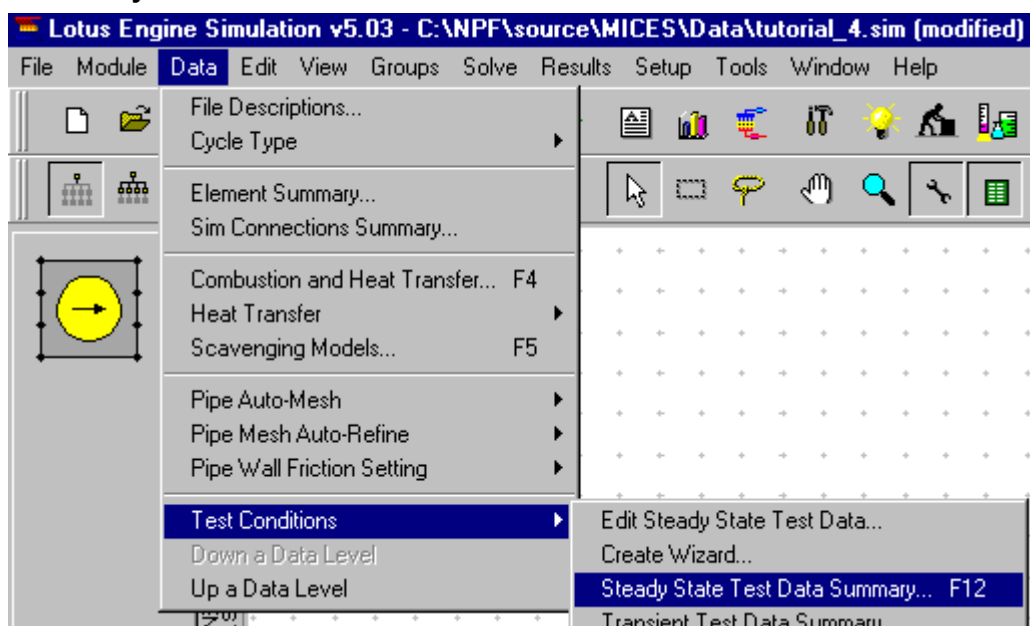


Figure 8.3. Opening the Steady-State Test Data Summary.

The load finder is switched on from the **Test Points** tab under the **Case Type** column, by default all test points are created as **Builder Def.**. This means that the simulation is run using data exactly as defined in the model.

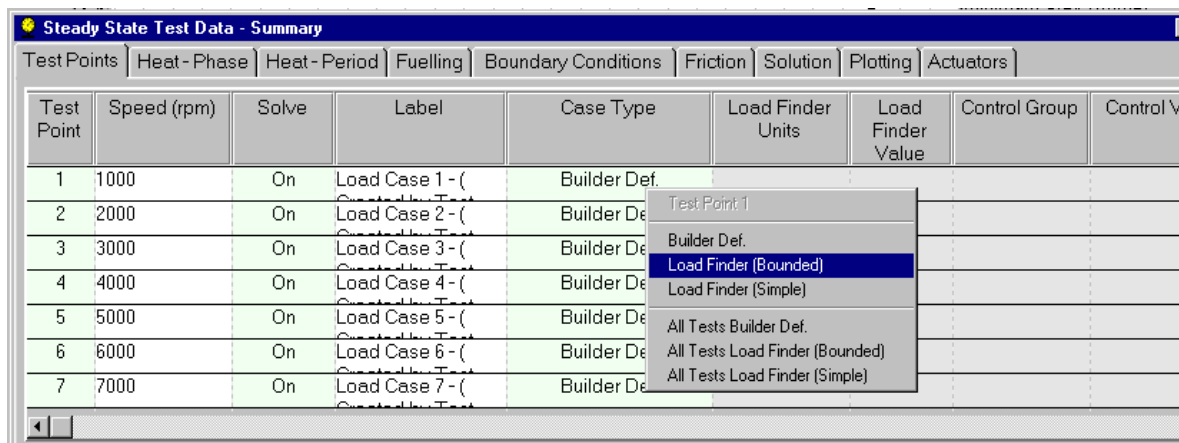


Figure 8.4. Changing to a 'load Finder' run.

To change a test point to run a 'load finder' case, select in the **Case Type** column for the required test point and select one of the two load finder options from the left mouse menu. From this menu you can also set all test points to 'load finder' at the same time.

The two types of load finder, *Bounded* and *Simple* represent alternative approaches to controlling the iterative process used to match the element property against the required load. The *Bounded* approach as the name suggests requires the user to specify the upper and lower limit of any variation in the element property, this can help reduce the number of cycles required to match to the required load. For our example we will use the *Simple* approach. From the left mouse menu on the **Case Type** column select **All Tests Load Finder (Simple)**.

Having selected a simple 'load finder' run the next four columns are enabled, (we will ignore the two columns on the far right *No. of Cycles 1* and *No. of Cycles 2*). We need to define the units and value for the load finder, what group will be used to control it and what variable. From the left mouse menu on the **load finder units** column select **BMEP (bar)**. (note: just set this on the first test point and we will copy it to all test points later. In the **Load Finder Value** column for test point 1 enter **4.0**. In the **Control group** column for test point 1 select from the left mouse menu the group we created for the throttle, select **Throttle**, as shown in Figure 8.5.

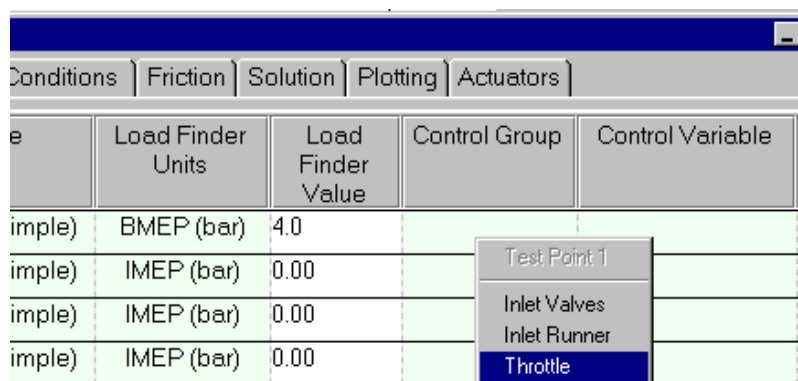


Figure 8.5. Setting the Load Finder Control Group.



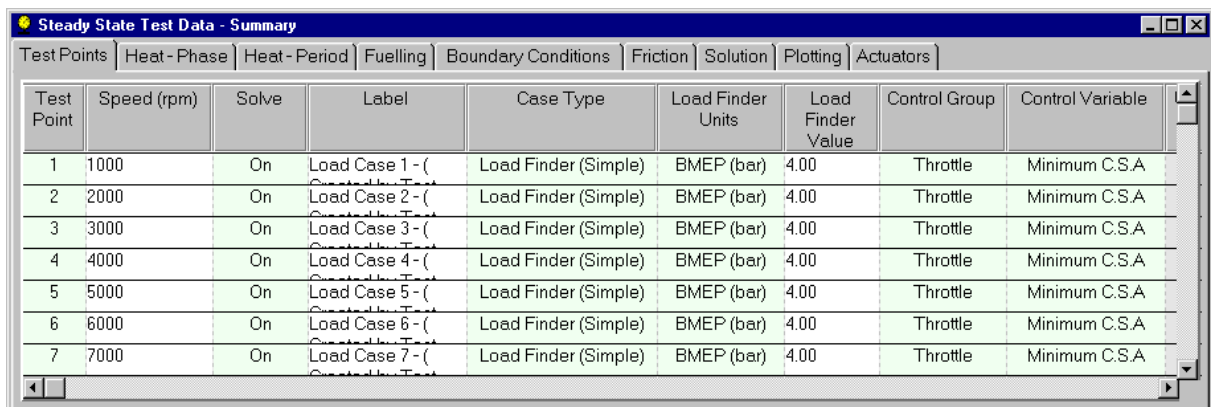
We can now set the *Control Variable* by selecting from the left mouse menu in this column on test point 1 the required variable. We shall use **Minimum C.S.A.** as our control variable.



Case Type	Load Finder Units	Load Finder Value	Control Group	Control Variable	Upper Bound
Load Finder (Simple)	BMEP (bar)	4.00	Throttle		
Load Finder (Simple)	IMEP (bar)	0.00			
Load Finder (Simple)	IMEP (bar)	0.00			
Load Finder (Simple)	IMEP (bar)	0.00			
Load Finder (Simple)	IMEP (bar)	0.00			
Load Finder (Simple)	IMEP (bar)	0.00			
Load Finder (Simple)	IMEP (bar)	0.00			

Figure 8.6. Selecting the Load Finder control variable.

We now need to copy these settings to all the other test points. We could simply repeat the above for all the other test points or, more efficiently, we can select from the left mouse menu on the *Test Point* column in test point 1 (far left of the spread sheet) the **Copy Picked Test to All Test Points**. The load finder portion for all the other tests will be updated to reflect our entered settings for test point 1 (see Figure 8.7)



Test Point	Speed (rpm)	Solve	Label	Case Type	Load Finder Units	Load Finder Value	Control Group	Control Variable
1	1000	On	Load Case 1 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.
2	2000	On	Load Case 2 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.
3	3000	On	Load Case 3 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.
4	4000	On	Load Case 4 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.
5	5000	On	Load Case 5 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.
6	6000	On	Load Case 6 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.
7	7000	On	Load Case 7 - (Created by Test)	Load Finder (Simple)	BMEP (bar)	4.00	Throttle	Minimum C.S.A.

Figure 8.7. Final Load Finder settings.

For this example we will run at a constant 4 bar BMEP across the entire speed range, but we could just as easily have defined a *Load finder value* that varied between test points.

Before we leave the Test Data summary sheet we need to change one other solver setting. Because of the iterative nature of the Load Finder we need to increase the maximum number of engine cycles to give the solver sufficient cycles to converge to the required control variable value. The load finder will also run a number of cycles before it will start to change the model data such that a certain level of stability is achieved before the model is modified, further adding to the requirement for an increase in the number of cycles.

We also need to disable the solver convergence check, as the convergence check is not a check on the load finder solution but a check on mass flow convergence between adjacent cycles. One does not necessarily imply the other.

Change to the **Solution** tab on the test data spread sheet. In the *Cycle Limits* column for test point 1 change from *Default Cycle limits* to **User Defined Cycle Limits**. We can now edit the next three columns for test point 1 to be all set to **25**. As before we can from this tab copy test point 1's values to the other test points.

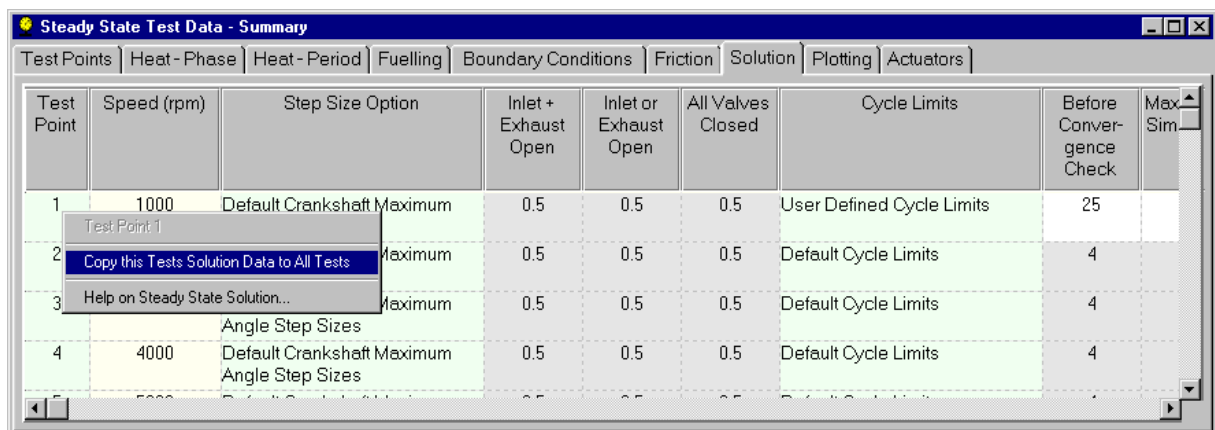
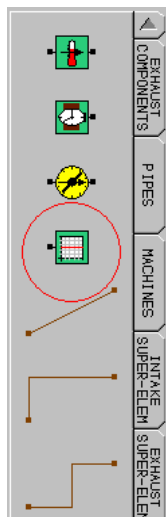


Figure 8.8. Copying solution parameters to all test points.

We can now **close** the test data summary sheet and return to the main display.

## Monitoring the 'Load Finder'

During a Load Finder run, or in fact for any run type, we can make use of a specific sensor element to monitor how the load finder solution is progressing. This sensor element is the *Sensor Plot* and is located on the *Sensors and Actuators* tab of the toolkit. This sensor not only creates an output file independent of the 'mrs' and 'prs' files but can also be viewed dynamically on the job status display as the run proceeds.



We will use the sensor plot to display the average BMEP of the cylinders and the throttle minimum cross sectional area. To this we first need to create another new group. This group should be called **Cylinders** and contain all four cylinders. (If in doubt how to do this refer back to the previous tutorial where we created a group for the inlet poppet valves).

Change the toolkit tab to **Sensors and Actuators**. Select the plot sensor from the toolkit and place just above cylinder 1. Turn on the harness connector for Cylinder 1 and for the inlet throttle.

Using a harness wire connect the cylinder to the input of the plot sensor, and similarly connect the throttle's harness connector to the input of the plot sensor. Your model should now appear as shown in Figure 8.9.

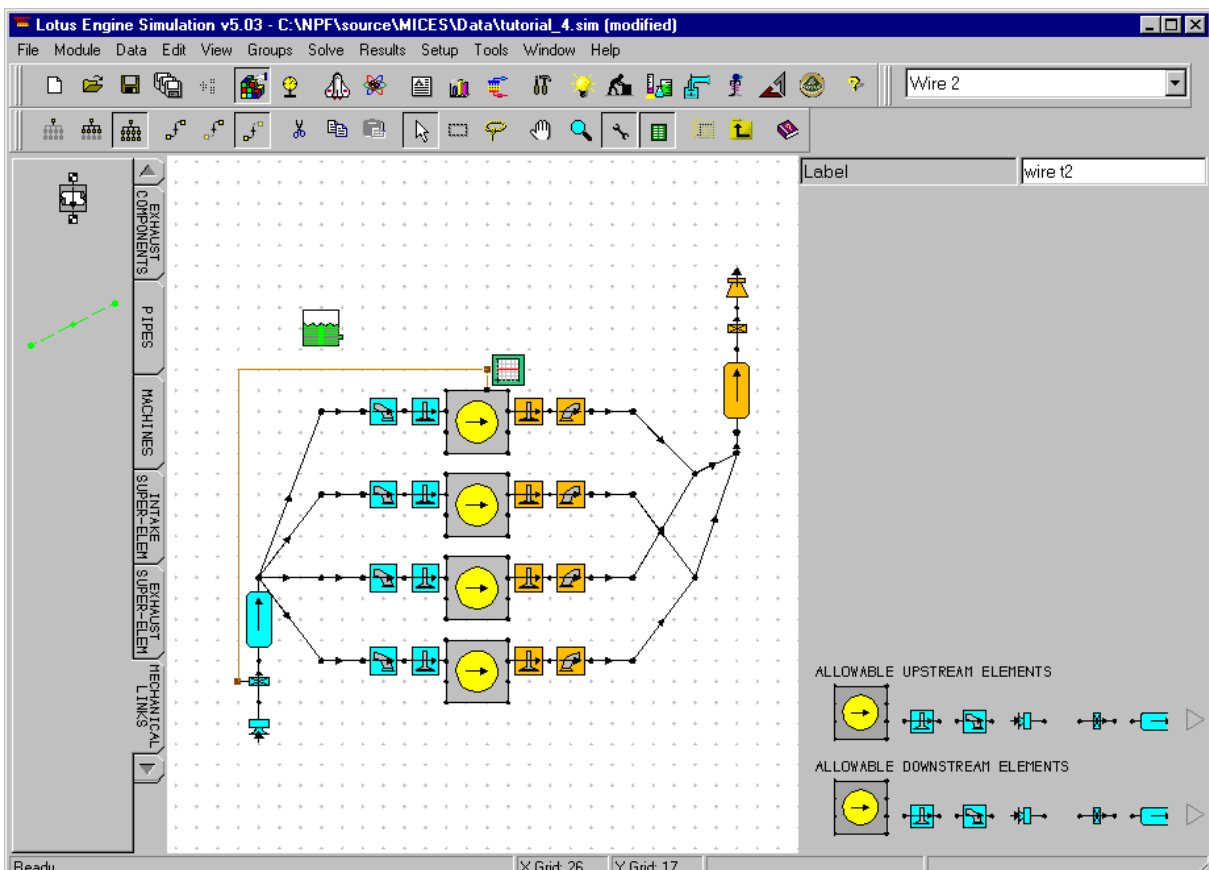


Figure 8.9. Model after addition of plot sensor connections.

Having made the connections we now need to define how and what the plot sensor will store. Select the plot sensor element so that we can view the elements properties. We will first define the name of the output file. Select the *browse* icon next to the *File Name* entry in the property sheet and enter the filename as **tutorial7.trs** (note: for a multiple speed run a separate file is produced for each speed and an index added to the defined file name i.e. tutorial7.trs1, tutorial7.trs2 ....).

Label	output plot file
File Name - C:\NPF\source	
Plot for Run Type	Steady State
Channel Select	Wire 1
Channel Parameter	Pressure (N/m2)
Channel Group Type	Liner Avg. Temp. (K)
	IMEP (complete) (bar)
	<b>BMEP (bar)</b>
Channel Element	Indicated Power (kW)
	Brake Power (kW)
	Brake Torque (Nm)
Plot Associate Type	Volumetric Eff. (%)
Plot Sample Size (s)	Bore (mm)
Plot File Format Type	Stroke (mm)
	Throw (mm)
Add File Headings	Omit

We now select which particular property we require to save for each wire connection. In the *Channel Select* entry ensure this is set to **Wire 1** and then in the *Channel Parameter* menu select **BMEP (bar)**. Change the *Channel Group Type* to **Group** and *Group Apply Type* to **Average of Group**.

Repeat this process for the throttle, by setting the *Channel Select* to **Wire 2** and the *Channel Parameter* to **Min. CSA (mm2)**. For this wire we do not need to change the *Channel Group Type* as we are only interested in the one element.

The sensor plot files can be saved in ASCII or Binary format, include headings, and be stored on either a time or crank angle basis. We will accept the defaults for all these.

## Running the 'Load Finder' Job

This job will take approx. 108 minutes on a 500 MHz PC.

Before we run the job edit the file descriptions and set the *Test No.* to **Tutorial7** then save the data file to **Tutorial\_7.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to set the .MRS and .PRS file names (Tip: always use the data checking wizard to validate any new models).

Once the job is successfully running we can monitor its progress using the *job status* tab on the solver control window. From the *Select Required Display* selection box change from the normal *Standard 5x Bar Charts* selection and pick **Sensor Plot 1 (steady state)**. The normal bar chart display will change to an x-y plot with two lines on, one for each sensor wire attached to the plot sensor. The graph will display on the y-scale the values associated with the first wire. To change the displayed values use the right mouse button on the graph and select **Channel 2** from the **Show Y-axis values for** item in the menu.

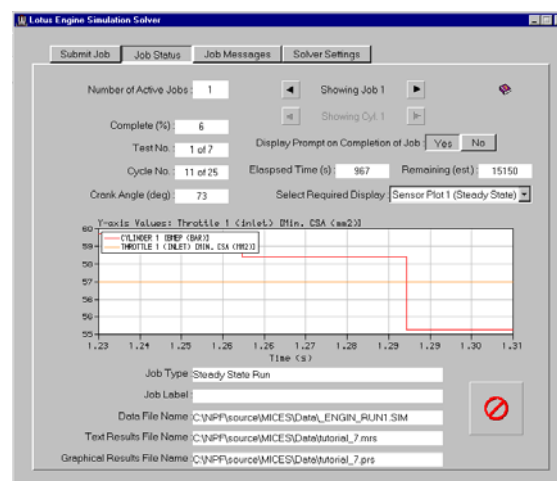


Figure 8.10. Plots sensor dynamic display.

## Reviewing the Load Finder Results

When the job has run, load the mrs results into the mrs graph display. The first thing you will notice is that the BMEP value does not make exactly 4.0 bar BMEP throughout the speed range. This is because 25 cycles was insufficient to exactly match the target. The BMEP varies from 3.3 bar to 4.2 bar.

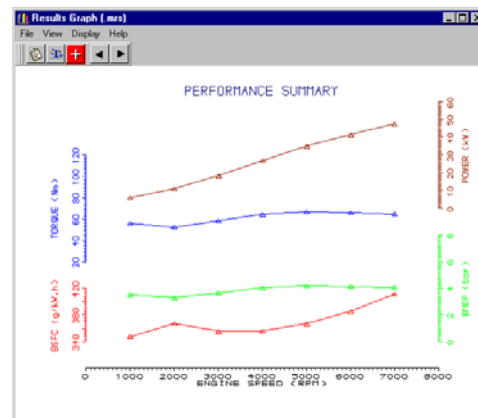


Figure 8.11. 4.0 bar BMEP Load Finder Results – 25 cycles

## Exercise – Virtual Links

We have seen from the previous load finder run that the increase in the required No. of cycles leads to significant increases in run times. One way of reducing run times is to de-refine the model by replacing the pipes with virtual links. Virtual links have no associated mass flow or properties they simply imply element connections, they can also be used to just help layout our conventional pipe based model. Whilst it is not normally appropriate to de-refine a model in this way, as removing the pipes will lose there associated wave dynamics, there are occasions when it is acceptable to do this. This exercise will illustrate the run time benefits that can be achieved when replacing the pipes with virtual links.

First delete all the pipes from the model then directly connect the free end of each inlet port to the free end of the inlet plenum with a virtual link. Repeat this for the exhaust side. Your model should look like that shown in Figure 8.12.

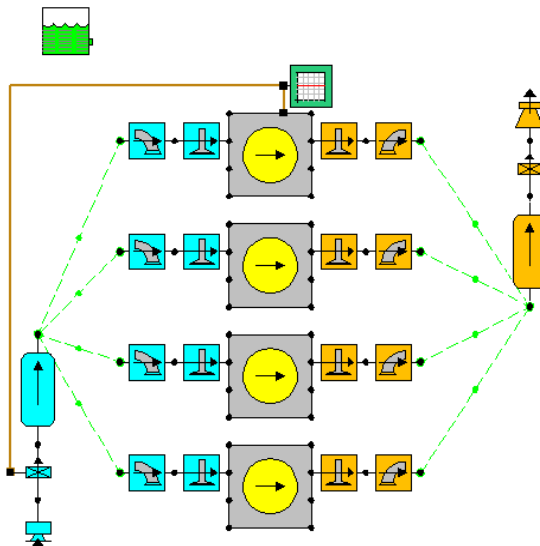


Figure 8.12. Virtual link version of the model.

Re submit the job, (you may want to change the Test No string and save it to tutorial\_7b.sim so that you don't lose the existing results). You may want to increase the number of simulation cycles to assist in the load finder convergence, say 40 cycles. At 25 cycles you should find that the run time is reduced by a factor of 10. The results are shown for 40 cycles compared to the full pipe model results for 25 cycles, (note the improved load finder convergence with 40 cycles).

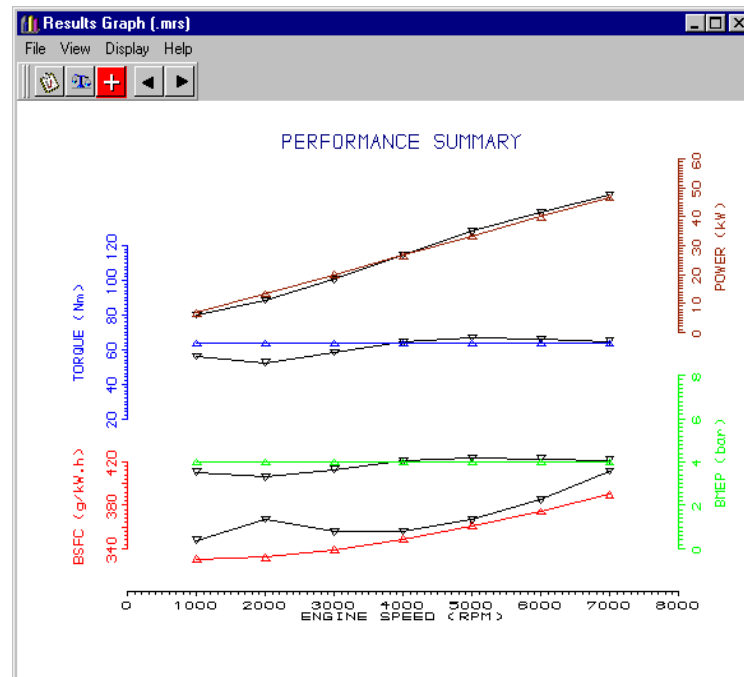


Figure 8.13. Comparison of results.

# 9

## Tutorial 8. 'Multi-Cylinder Turbocharged Model'

### Overview

This chapter uses one of our previous naturally aspirated multi-cylinder models and takes you through the steps to convert it into a turbocharged model.

This chapter contains the following sections:

■ Loading the Original Model	102
■ Adding the Turbocharger	102
■ Modifying the Turbocharger Data	104
■ Monitoring the Turbocharger Speed	106
■ Running the Turbocharged Model	107
■ Viewing the Results	109
■ Exercise	112

## Loading the Original Model

We will again use the existing model created in tutorial 4. Load the previously saved file **tutorial\_4.sim**. We will need to modify the intake and exhaust pipe arrangements to enable us to connect the turbocharger into the system.

## Adding the Turbocharger

Change the toolkit tab to show **Machines**, select the turbocharger element at the top of the toolkit and add it to your model just below cylinder 4. To aid connectivity we will rotate the turbocharger round 180 degrees, (use Ctrl+Page Down), such that the turbine portion of the turbocharger, (shown in orange) is uppermost. Finally we need to *flip* the flow directions of both the compressor and the turbine. To do this with the turbocharger in focus select from the right mouse menu, **Flip Compressor** and **Flip Turbine**. Your model should now look like that shown in figure 9.1.

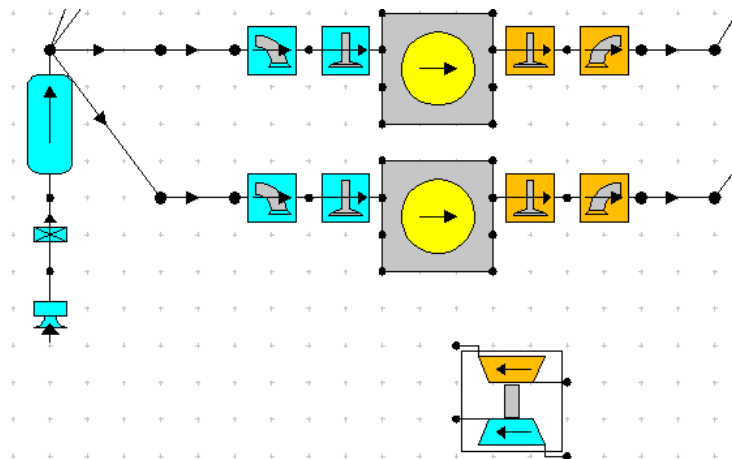


Figure 9.1 - Adding the Turbocharger.

With the turbocharger in focus we can see that the allowable connections are limited to plenums and the inlet boundary, (assuming we ignore the virtual link).

We will move the existing inlet boundary on the inlet (upstream) side of the compressor (the compressor is coloured blue). You will probably want to rotate it so that it lies in the horizontal direction to aid layout. Now add a second inlet plenum to the downstream end of the compressor, simply copy and paste the existing inlet plenum and move it to make the required connection. Change the volume of the new plenum to 1.0 litres. Connect the outlet of this plenum to the throttle using a virtual pipe (coloured green), you may want to change the visual appearance of the virtual pipe to **single bend** to assist in the layout.

Turbocharged engines tend not to have tuned inlet systems, so we will shorten the length of the primary inlet runners (the 4 pipes connected immediately downstream of the original inlet plenum) from 350 mm to 150 mm. Select each of these pipes in turn and alter their length in their property sheet. This will have the benefit of reducing the number of meshes in these pipes, and hence will reduce the calculation run times. Your model should now look similar to that shown in Figure 9.2.



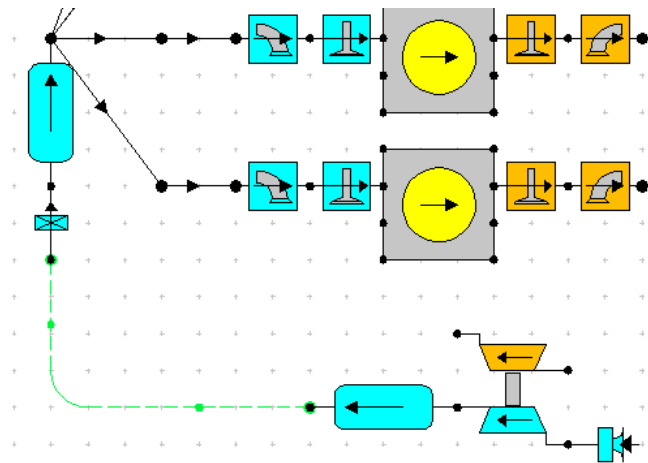


Figure 9.2. Connecting the compressor.

We will now perform a similar series of changes, system to those we have implemented to the inlet, to the exhaust. Move the existing exhaust plenum, throttle and exhaust boundary to the downstream end of the turbine. Copy the existing exhaust plenum and add it to the upstream end of the turbine. Select the exhaust throttle and alter its diameter to 60 mm in the property sheet. Your model should now look similar to the one shown in Figure 9.3.

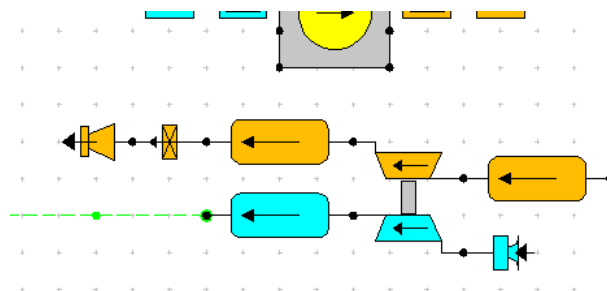


Figure 9.3. Connecting the turbine.

Now delete the secondary and tertiary pipes from the exhaust system, just leaving the exhaust ports and primary runners (Hint: If you delete a component unintentionally, pressing **Ctrl+Z** will undo the action. Alternatively selecting **Edit / Undo** from the menubar will perform the same action). Connect the free ends of the exhaust primary runners and connect them to the free end of the new exhaust plenum.

We will shorten the length of the exhaust primary runners from 400 mm to 100 mm. Select each of these pipes in turn and alter their length in their property sheet. You may want to change the visual appearance of these pipes by altering them to **double bend** to assist in the layout. The location of the bends within the pipe can be altered by selecting **Lengthen Offset** or **Shorten Offset**, using the right mouse button menu, whilst the pipe is selected.

Finally, we shall change the compression ratio of our engine from 11:1 to 8:1. Select each cylinder in turn and alter the compression ratio in the cylinder property sheet.

You should now have a fully connected turbocharger model. To verify your connections switch on the connectivity error visibility using **View / Show Connectivity Errors**. If you have any elements shown in red check your model with that shown in Figure 9.4 for the missing connections.

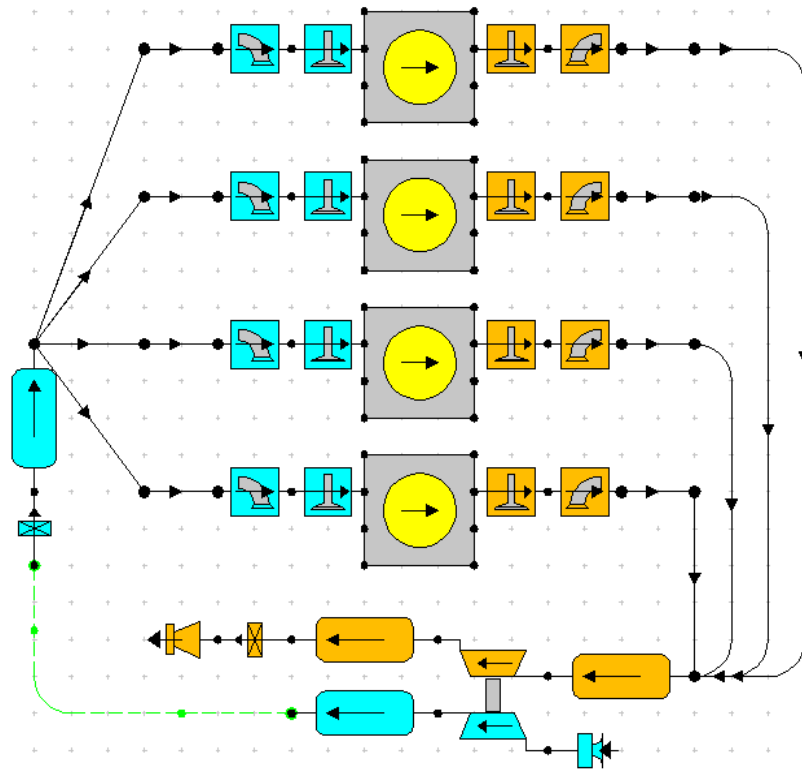


Figure 9.4 Turbocharger Fully Connected

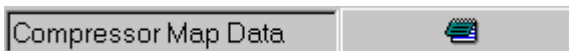
## Modifying the Turbocharger Data

We will now change the properties of the turbocharger data. The data contained within the default turbine and compressor maps are from a large diesel engine and need to be changed before we run our model. Select the turbocharger and edit the following parameters in the properties menu:

Compressor Inlet Dia. (mm)	50
Compressor Outlet Dia (mm)	40
Rot. Inertia (kg.m2)	4E-6
Turbine Inlet Dia. (mm)	40
Turbine Outlet Dia (mm)	60
Rot. Inertia (kg.m2)	4E-6

The property sheet for the turbocharger should now appear as shown in Figure 9.5. The inlet and outlet diameters are simply the diameters of the entry and exits to the compressor and turbine. The rotating inertia is used in the calculation for updating the turbocharger speed. The torque balance between the compressor and the turbine is calculated, and the acceleration/deceleration of the device is limited by its inertia. The speed calculation is either performed only one per engine cycle, or at every calculation time-step, depending on the **Update Speed** option selected in the turbocharger property sheet. We shall leave this at the default setting of **By Time Step**.

Rather than entering alternative compressor and turbine maps we will simply scale the existing data. To view the compressor map data click on the pad icon, shown below, in the properties sheet menu.



Compressor Label	default compresso
Compressor Type	Full Map
Inlet Dia. (mm)	50.00
Outlet Dia. (mm)	40.00
Rot. Inertia (kg.m <sup>2</sup> )	4.0000e-006
Gear Ratio to Shaft	1.00
Drive Gear Mech Eff. (0-1)	1.000
Compressor Map Data	
Turbine Label	default turbine
Turbine Type	Full Map
Inlet Dia. (mm)	40.00
Outlet Dia. (mm)	60.00
Rot. Inertia (kg.m <sup>2</sup> )	4.0000e-006
Gear Ratio to Shaft	1.00
Drive Gear Mech Eff. (0-1)	1.000
Turbine Map Data	

Figure 9.5. Turbocharger property sheet.

A spreadsheet, which contains all the compressor data, will appear. The top left-hand box in this window allows the user to specify the number of constant speed lines used to define the compressor map. The default value is 6 (maximum = 20). Data for each turbine can be displayed by using the arrow keys next to the value entry box. The map consists of mass flow versus pressure ratio and mass flow versus efficiency values.

**Compressor Map Data**

Data

No of Speeds:  ◀ No. 1 of 6 ▶

Speed (rpm):  No of Points:  Speed Scale Factor:

Display:

	Mass Flow	Pressure Ratio	Efficiency (0-1)
1	0.032165	1.0222	0.5727
2	0.04067	1.0188	0.6361
3	0.050365	1.0151	0.6912
4	0.062055	0.99904	0.7376
5	0.075635	0.96901	0.7364
6	0.091945	0.92232	0.6843
7	0.10822	0.85764	0.525
8			
9			
10			

Scale Factors

Figure 9.6. Compressor map data window.

Change the **Speed Scale Factor** from 1.0 to 1.5, the **Mass Flow Scale Factor** to 0.35 and the **Pressure Ratio Scale Factor** to 0.7, as shown in Figure 9.6. This will apply these scale factors to all of the constant speed lines defined in the compressor map. Clicking on the graph icon in the compressor map data window will open a window displaying the compressor map. This should now look like the map shown in Figure 9.7.

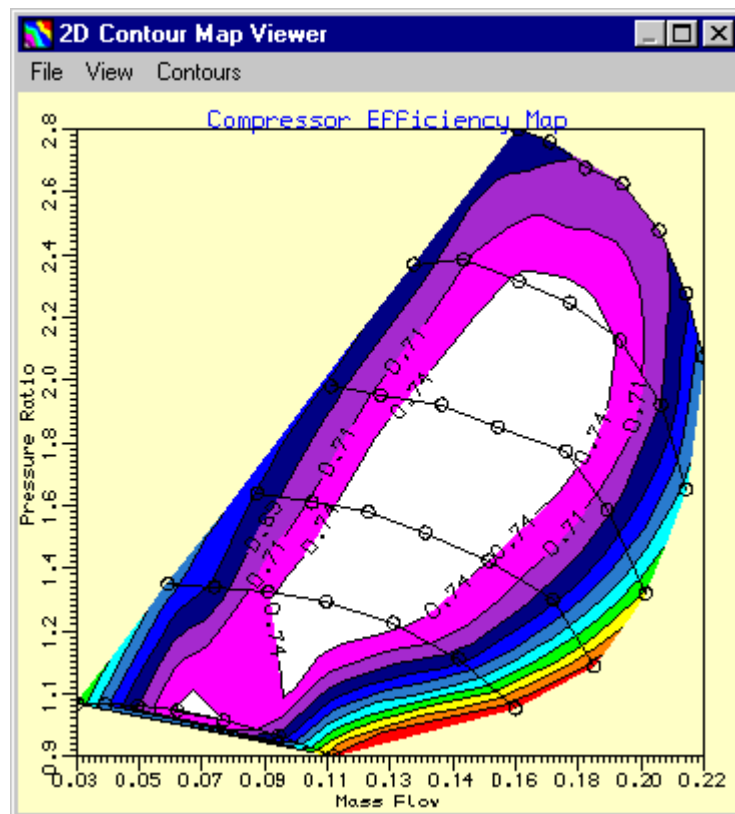


Figure 9.7. Compressor map.

We need to similarly scale the turbine map data. Close the compressor map data window and click on the pad icon, shown below, in the turbocharger properties sheet menu. Change the **Speed Scale Factor** to 1.5, the **Mass Flow Scale Factor** to 0.25 and the **Pressure Ratio Scale Factor** to 0.9.



## Monitoring the Turbocharger Speed

It is often useful, when running a turbocharged engine model, to use the sensor element to monitor how the simulation is progressing. We will use a sensor plot to display the turbocharger speed (If in doubt how to do this refer back to the previous tutorial where we used a plot sensor to view BMEP and inlet throttle CF).

Change the toolkit tab to **Sensors and Actuators**. Select the plot sensor from the toolkit and place it just below the turbocharger element. Turn on the harness connector of the turbocharger. Using a harness wire, connect the harness connector tab of the turbocharger (the turbocharger element has two harness connectors, one attached to the compressor, the other to the turbine, these access the same element properties, so it doesn't matter which tab you connect to) to the input of the plot sensor, as shown in Figure 9.8.

Having connected the plot sensor, we now need define to what the plot sensor will store and to which file it will be written to. Select the plot sensor element so that we can view the element properties. We will first define the file name for the output. Select the *browse* icon next to the *File Name* entry in the property sheet and enter the filename as **tutorial8.trs**, (note for a multiple speed run a separate file is produced for each speed and an index added to the defined file name i.e. tutorial8.trs1, tutorial8.trs2 ....etc.).

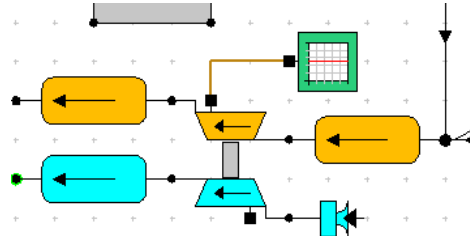


Figure 9.8. Model after addition of a plot sensor.

We now need to select which particular property we will store in the file, we can store one parameter for each wire connected to a plot sensor, as we only have one wire connected we only need to define one parameter. Ensure that the *Channel Select* entry box is set to **Wire 1** and then in the *Channel Parameter* menu select **Comp Speed (rpm/K<sup>0.5</sup>/T)**. Change the *Channel Group Type* to **Single**.

The sensor plot files can be in ASCII or Binary, include headings, and be against either time or crank angle. Set the *Plot Associate Type* to **Crank Angle** and the *Plot Sample Size (deg)* to **20**.

## Running the Turbo-charged Model

Before we run our turbocharged model, we shall change the maximum number of engine cycles, as turbocharged simulations can take longer to achieve a converged solution than naturally aspirated simulations, as several cycles are required for the turbocharger speed to stabilise. Press the F12 key to open the **Steady State Test Data - Summary** and click on the **Solution** tab. For the first test point change the **Cycle Limits** from **Default Cycle Limits** to **User Defined Cycle Limits**. Change the **Max No for Simulation** to 20 and the **From which Results are Written** to 20, as shown in Figure 9.9.

Steady State Test Data - Summary					
Boundary Conditions   Friction   Solution   Plotting   Actuators					
All Valves Closed	Cycle Limits	Before Convergence Check	Max No for Simulation	From which Results are Written	
0.5	User Defined Cycle Limits	4	20	20	
0.5	Default Cycle Limits	4	10	10	
0.5	Default Cycle Limits	4	10	10	

Figure 9.9. Edit the cycle limits.

Click on the **Test Point** column of the first test point and then select **Copy this Tests Solution Data to all Tests**, as shown in Figure 9.10. This will change the maximum number of cycles to 20 for all of the defined test conditions.

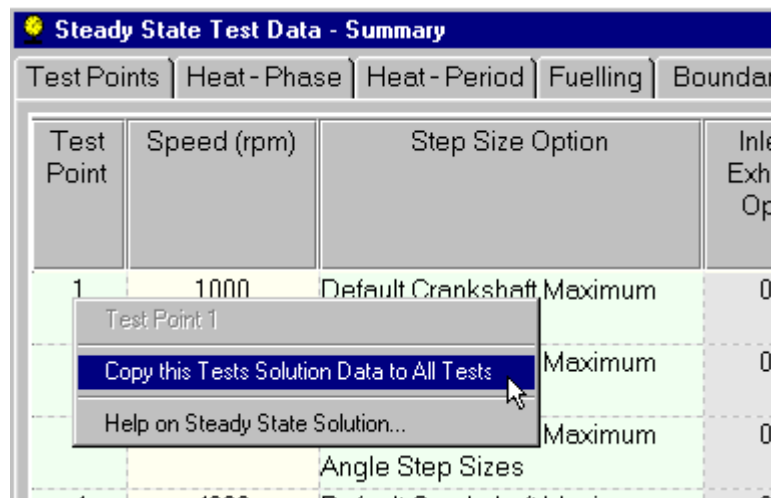


Figure 9.10. Copy new cycle limits to all tests.

Now click on the **Plotting** tab and ensure that the *Plotting Option* is set to **User Defined Plotting Options**. Now change the *Cylinder Options* to **Cylinder Stored**, the *Plenum Options* to **Plenum Stored**, the *Pipe Options* to **All Pipe Data Stored**, the *Flow Options* to **All Flow Data Stored** and the *Turbine Options* to **All Turbine Data Stored**, for all of the test conditions, as shown in Figure 9.11.

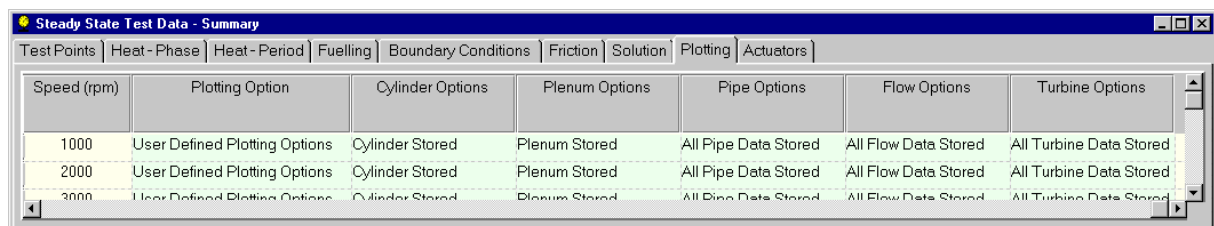


Figure 9.11. Copy new cycle limits to all tests.

Close the Steady State Test Conditions Summary Window and change the file descriptions and set the *Test No.* to **Tutorial8** then save the data file to **Tutorial\_8.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to set the .mrs and .prs file names.

Once the job is successfully running we can monitor its progress using the *job status* tab on the solver control window. From the *Select Required Display* selection box change from the normal *Standard 5x Bar Charts* selection and pick **Sensor Plot 1 (steady state)**, as shown in Figure 9.12. The normal bar chart display will change to an x-y plot. This graph will display on the compressor speed. Note: the turbocharger speed is reset at the beginning of each steady state test point to be half way between the maximum and minimum compressor speed curves defined in the compressor map.

You may observe the following message in the solver run-time window: *Warning compressor match point not found*. This generally indicates that the instantaneous pressure ratio across the compressor is higher than any point on the compressor map, for the current compressor speed. Provided that this message only occurs a few times per cycle then it can be safely ignored. Persistent occurrences of this message generally indicate a problem with the turbocharger map data or the model, and should be investigated further.

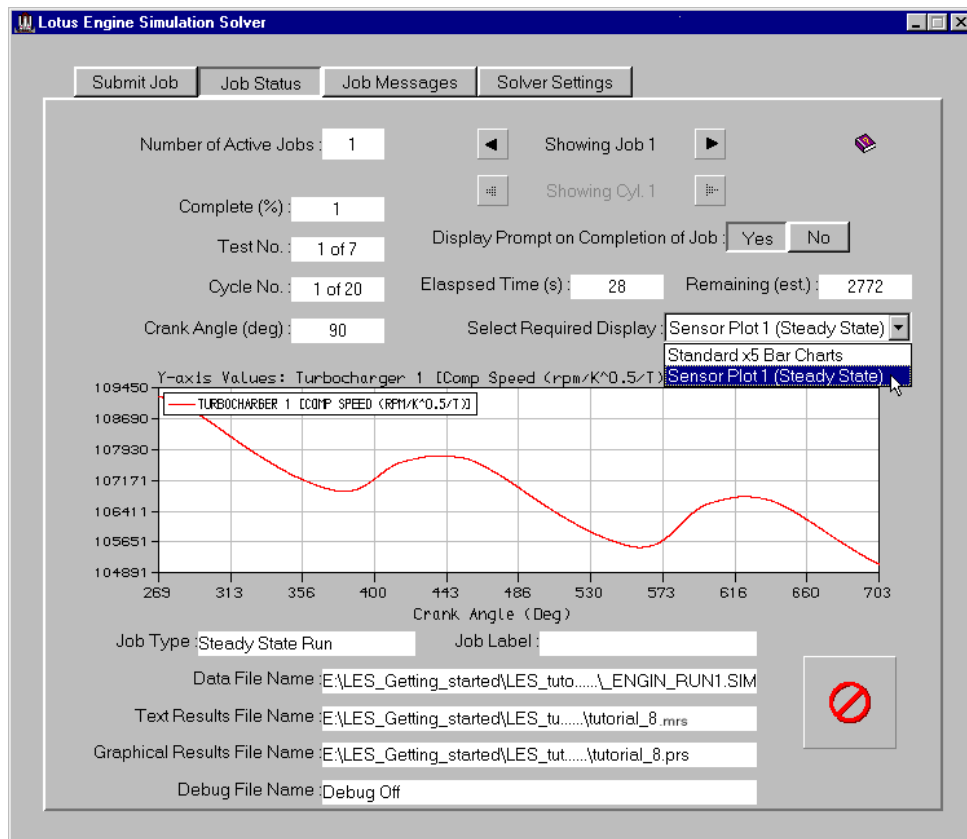
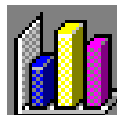


Figure 9.12. Runtime monitoring of turbocharger speed using sensor plot.

## Viewing the Results

Once the job has finished, the Batch Job Finished window will appear. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. Tick the boxes to **Load Text Results (.mrs)**, **Load Graphics Results (.mrs)** and **Load Graphics Results (.prs)**. This places the .MRS results file, which contains all the relevant cycle-averaged data, and the .PRS file, which contains all the intra-cycle data, into the dedicated text reader and graphical viewing facilities.

Use the graphical viewer to display the .MRS results. Open the .MRS file viewer by clicking on the icon shown below.



This produces the window shown in Figure 9.13. Select the **Autoscale** option from the **View** menu within this window.

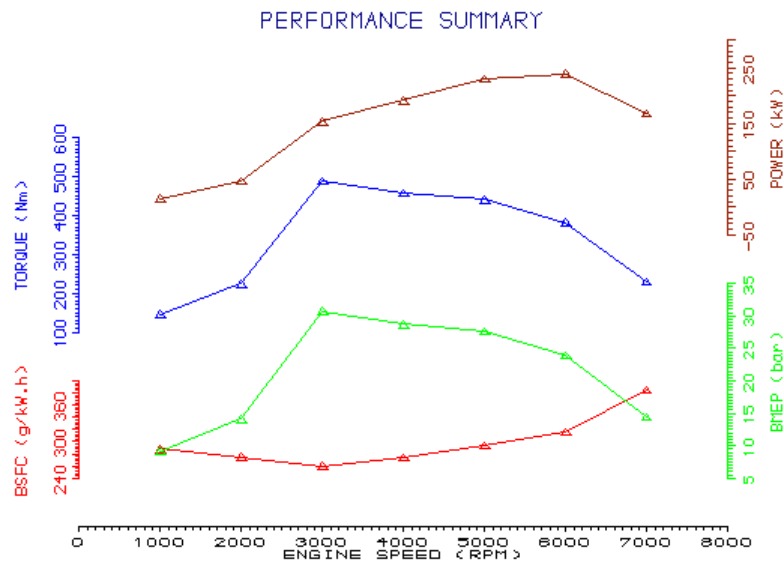


Figure 9.13. .MRS Results Graphical viewer.

The first plot which can be viewed (see Fig. 9.13) shows the BSFC (brake specific fuel consumption), BMEP (brake mean effective pressure), Torque, and Power produced by the engine. It can be seen that the torque produced by the engine increases sharply between 2000 and 3000 rev/min. Peak torque is at 3000 rev/min, and corresponds to a BMEP of about 30 bar. This BMEP is exceptionally high for a turbocharged automotive engine, which typically have maximum BMEP values of less than 20 bar. Further examination of the results shown in the .MRS graph viewer reveals that the predicted PMAx values for this engine are about 150 bar from 3000 to 5000 rev/min, which is unrealistically high for a spark ignition engine running on gasoline.

Select **Display / Plot / Plot 8** from the .MRS graph viewer menu bar. The viewer window should now change to display a blank screen, as only the first seven plots are defined by default. If the screen isn't blank use the arrow keys to move along until you find an unspecified plot. Now select **View / Specify Graph** from the menu bar. The *Specify Graph menu* will appear allowing the variables displayed on each graph to be specified. Up to four graphs can be specified for each plot screen.

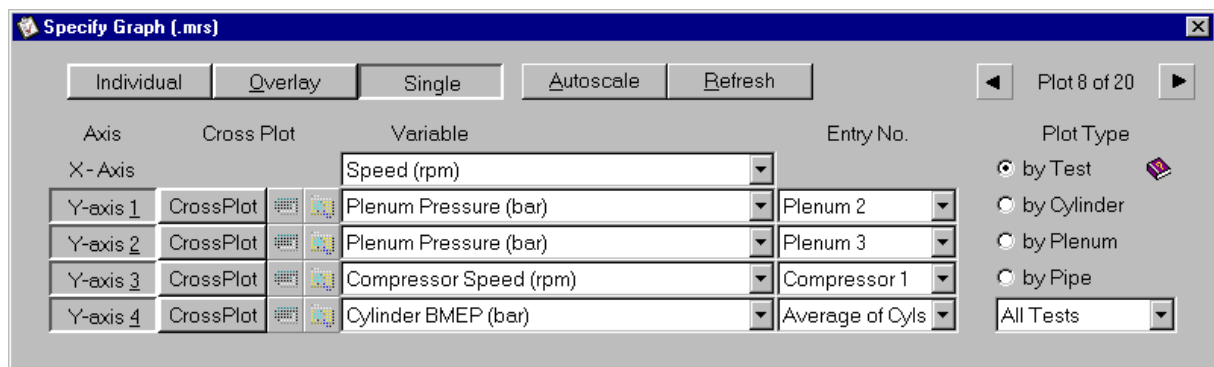


Figure 9.14. Specifying new .MRS plots



We shall specify four new graphs for our plot screen. There are 3 selection boxes in the top left hand corner of the *Specify Graph menu*, **Individual**, **Overlay** and **Single**. These alter the way our graphs will appear, click on the **Single** selection box so that it appears depressed. We shall plot the parameters we wish to view, against engine speed, so select **Speed (rpm)** as the X-Axis variable. For the Y-Axis variables select **Plenum Pressure (bar)** for Y-Axis 1 and Y-Axis 2, **Compressor Speed (rpm)** for Y-Axis 3 and **Cylinder BMEP (bar)** for Y-Axis 4. Select **by Test** for the *Plot Type* option, from the right-hand side of the *Specify Graph menu*. In the bottom right-hand corner of the window select **All Tests**.



Now close the *Specify Graph menu* window and click on the plenum directly downstream of the compressor. Make a note of the plenum number that appears in the component list box, shown above. Click on the plenum directly upstream of the turbine, and make a note of its number. Return to the .MRS graph viewer and re-open the *Specify Graph menu* window. Use the left and right arrows to return to our new plot. In the *Entry No.* column we need to specify which component the parameters, that we specified in the *Variable* column, will be taken from. For the Y-Axis 1 Plenum pressure, select the plenum number corresponding to the plenum upstream of your turbine. For the Y-Axis 2 Plenum pressure, select the plenum number corresponding to the plenum downstream of your compressor. For the Y-Axis 3 compressor speed, select **Compressor 1**. For the Y-Axis 4 Cylinder BMEP plot select **Average of Cyls**. Your *Specify Graph menu* should now look similar to the one shown in Figure 9.14. Click on the **Autoscale** button and then the **Refresh** button. Your .MRS plot should now look similar to the one shown in Figure 9.15. You may want to change the line colours and alter the appearance of the grid on the graphs. To do this close the *Specify Graph menu* window and select **View / Setup** from the .MRS viewer menubar. You might like to experiment with the effect that the various options in this menu have on the appearance of the plots.

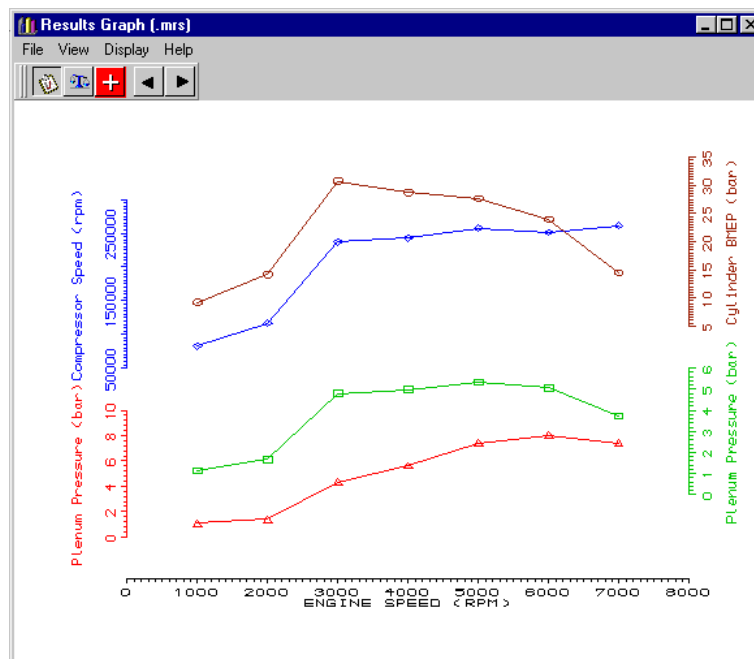


Figure 9.15. Viewing the turbocharger operating parameters.

Examination of the four graphs we have just created reveals that the pressure in the plenum directly upstream of the turbine reaches a pressure in excess of 8 bar at 6000 rev/min. Also, the compressor delivery pressure is over 5 bar for a large proportion of the speed range of the engine. The compressor speed is also very high, being about 250000 rev/min for the majority of the speed range of the engine. You will notice that this is in excess of the highest speed line specified in our compressor

map data. LOTUS ENGINE SIMULATION has extrapolated the data in compressor map. You can also see that the BMEP, and hence torque, of the engine is very strongly coupled to the compressor delivery pressure. Typically, the boost pressure of an automotive spark ignited gasoline engine is limited to 2 bar, or less, using a waste-gate.

We will examine how a simple waste-gate may be implemented, using the sensors & actuators elements, in the following tutorial.

## Exercise

Use the \*.PRS viewer to examine the variation in pressures around the pipe network and the cylinder pressure diagrams. Use the *Graph Status* menu, shown in Figure 9.16, (**Results / .prs Results / Graphs / Graph Status...** from the LOTUS ENGINE SIMULATION menubar, or simply **Graph Status** from the right mouse button menu) to change the graphs to display turbine and compressor data, such as compressor/turbine speed or pressure ratio.

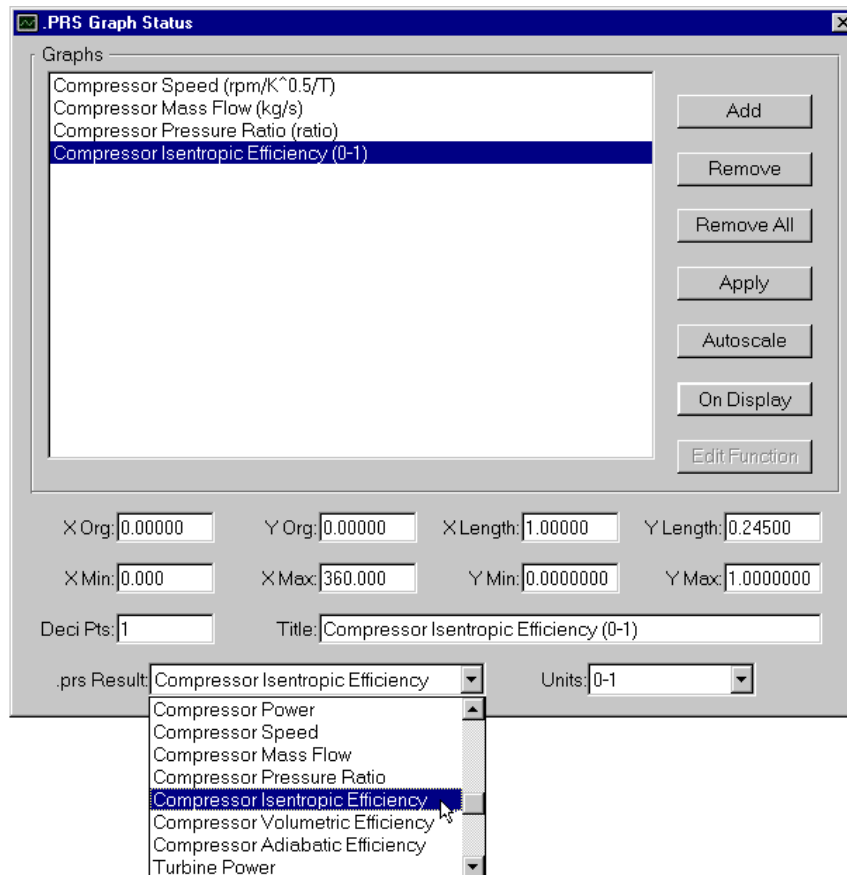


Figure 9.16. The *Graph Status* menu.

# 10

## Tutorial 9. 'Turbocharged Model with a Waste-gate and Charge-cooler'

### Overview

This chapter uses our turbocharged multi-cylinder engine, which we developed in the previous tutorial, and takes you through the steps to add a simple waste-gate. A simple control system for this waste-gate is constructed, using the sensors and actuators elements.

This chapter contains the following sections:

- Loading the Original Model, 114
- Adding the Waste-gate, 1164
- Adding Control Elements to the Waste-gate, 1175
- Running the Turbocharged Model with a Waste-gate, 118
- Viewing the Results, 119
- Waste-gate for Variable Boost pressure, 120
- Damping the Waste-gate Motion, 124
- Exercise, 125

## Loading the Original Model

We will use the existing turbocharged model created in tutorial 8. Load the previously saved file **tutorial\_8.sim**.

A waste-gate is essentially a valve that allows some of the exhaust gases to by-pass the turbocharger turbine. In simple mechanical systems the waste-gate consists of a simple spring-loaded valve and the inlet system pressure actuates the opening of the waste-gate via a controlling diaphragm. We shall model the waste-gate using a simple throttle element. The sensors and actuators elements will be used to adjust the flow area of this throttle based on the inlet system pressure.

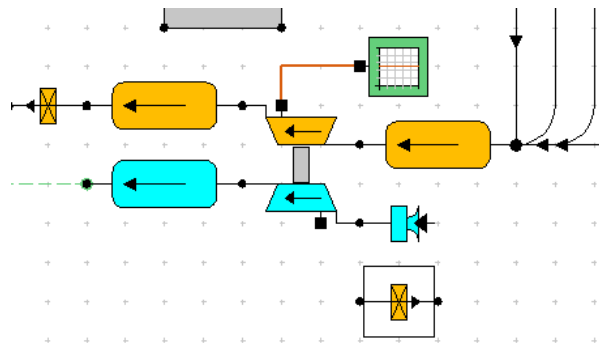


Figure 10.1. Adding the waste-gate throttle.

## Adding a Simple Waste-gate

Select a **throttle** from the **Exhaust Components** toolkit and place it just below our inlet boundary, as shown in Figure 10.1. Ensure that the *Throttle Type* is set to **Simple Throttle** and the *Discharge Data Type* is set to **CF Fixed**. Set the *Throttle CF* to **1.0** and the *Minimum C.S.A. (mm<sup>2</sup>)* to **700.0**. Now select a **pipe** from the **Pipes** toolkit. Connect the upstream end of the pipe to the plenum downstream of the turbocharger and the downstream end to our new throttle element (you may want to shorten the plenum downstream of the compressor, to stop the new pipe over lapping the end of the plenum. To do this select the plenum and click the right mouse button, then select **Shorten**), as shown in Figure 10.2. Change the length of the pipe we have just added to **50mm** and ensure that the diameter is set to **35mm**. Change the wall material to **Cast Iron** and the wall thickness to **3mm**.

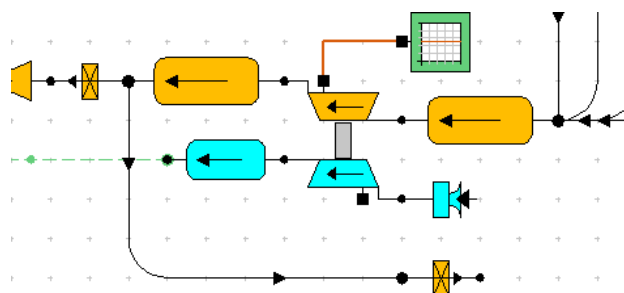


Figure 10.2. Add a pipe to the waste-gate throttle.

Select a **virtual pipe** from the **Pipes** toolkit and connect it from our waste-gate throttle to the plenum upstream of the turbine, as shown in Figure 10.3. The mechanical connections for our waste-gate are now complete. We now need to add the control elements.

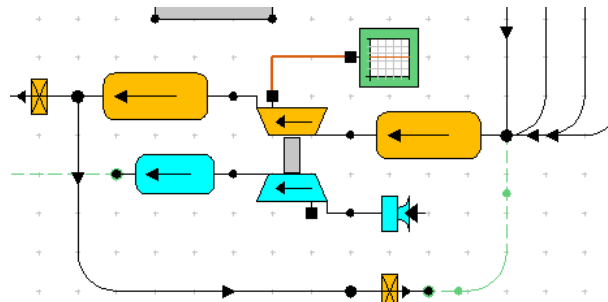


Figure 10.3. Add a virtual pipe to the waste-gate throttle.

### Adding Control Elements to the Waste-gate

Select the plenum downstream of the compressor and toggle the *Harness Connector* option to **On**. Now do the same thing to our waste-gate throttle element. Click on the *Sensors & Actuators* toolkit tab and select a **sensor** element. Place this sensor element just beneath the pipe connected to the waste-gate. Now pick an **actuator** element from the *Sensors & Actuators* toolkit and place it just to the right of the sensor element. Use the harness wires from the *Sensors & Actuators* toolkit to connect the inlet plenum harness connector to the sensor element, the sensor to the actuator, and the actuator to the harness connector of the waste-gate throttle. Your model should now look similar to the one shown in Figure 10.4.

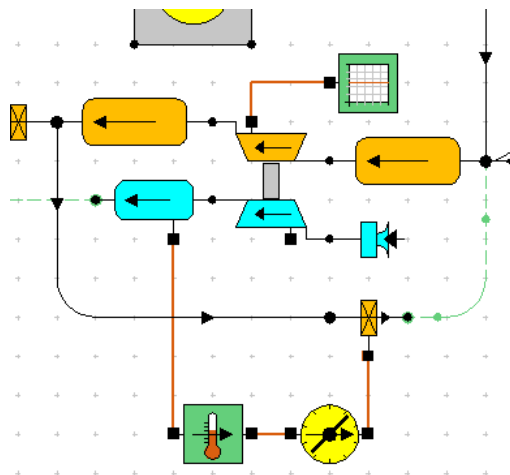


Figure 10.4. Adding control elements to the waste-gate.

Select the **sensor** element and change the properties so that the *Sensed Parameter* is **Pressure (N/m<sup>2</sup>)** and the Sensor Group Type is **Single**, as shown in Figure 10.5.

Sensor Label	default sensor
Sensed Parameter	Pressure (N/m <sup>2</sup> )
Sensor Group Type	Single

Figure 10.5. Selecting the *Sensed Parameter*.

Now we need to define the control strategy for the actuation of our waste-gate. We will attempt to model a simple mechanical waste-gate, which is held shut by a spring, and actuated by the inlet pressure acting on a diaphragm.

If we assume that the diameter of our diaphragm is 50 mm, then the cross-sectional area is roughly  $0.002 \text{ m}^2$ . If we wish our waste-gate to begin opening at an inlet pressure of 1.8 bar, then the preload we require in the spring that holds our waste-gate shut is

$$F = p.A = 180000[\text{Pa}] \cdot 0.002[\text{m}^2] = 360 \text{ N} . \quad (10.1)$$

If we want our waste-gate to be fully open by the time the inlet pressure has reached 2.0 bar, and we want the maximum lift of our waste-gate to be 10 mm, then the spring stiffness required is given by

$$\text{spring stiffness} = \frac{\Delta F}{l} = \frac{(200000 - 180000)[\text{Pa}] \cdot 0.002[\text{m}^2]}{0.01[\text{m}]} = 4000 \text{ N/m} . \quad (10.2)$$

If we assume that the maximum flow area of our waste-gate is  $700 \text{ mm}^2$ , and that this area varies linearly between fully closed and fully open (at 10 mm lift), we can obtain an expression for the instantaneous area of our waste-gate at a given valve lift

$$\text{flow area}[\text{mm}^2] = l[\text{m}] \cdot 70000[\text{mm}^2 / \text{m}] . \quad (10.3)$$

The instantaneous lift is obtained from

$$l[\text{m}] = \frac{\text{force} - \text{preload}}{\text{spring stiffness}} = \frac{\text{force}[\text{N}] - 360[\text{N}]}{4000[\text{N/m}]} . \quad (10.4)$$

Denoting the inlet pressure as F1 calculate the instantaneous force on the spring as

$$\text{force}[\text{N}] = p.A = F1[\text{pa}] \cdot 0.002[\text{m}^2] . \quad (10.5)$$

Combing equations (10.3), (10.4) and (10.5), we can write an expression for the instantaneous flow area of the waste-gate, based on the inlet pressure (F1), giving

$$\text{flow area}[\text{mm}^2] = \left( \frac{(F1[\text{Pa}] \cdot 0.002[\text{m}^2]) - 360[\text{N}]}{4000[\text{N/m}]} \right) * 70000[\text{mm}^2/\text{m}] . \quad (10.6)$$

This can obviously be simplified to give

$$\text{flow area} = \left( \frac{(F1 - F2)}{(\Delta F)} \right) * A_f , \quad (10.7)$$

where:

F1	= Instantaneous inlet system pressure;
F2	= Waste-gate opening pressure;
$\Delta F$	= Pressure difference between waste-gate opening and fully open;
$A_f$	= Waste-gate flow area when fully open.

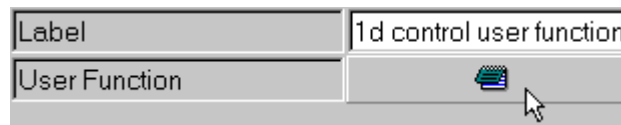
Select the actuator element and use the level control toggle, shown below, to drop into the control element level.



Now select a **User Function** element from the *1D Control Elements* toolkit and place it between the **IN<sup>1</sup>** and **OUT** signal boundaries, as shown in Figure 10.6.



Figure 10.6. Inserting the user function element.



Select the **User Function** element and click on the pad icon, shown below, to open the user defined function menu. The data entry window shown in Figure 10.7 should now appear.

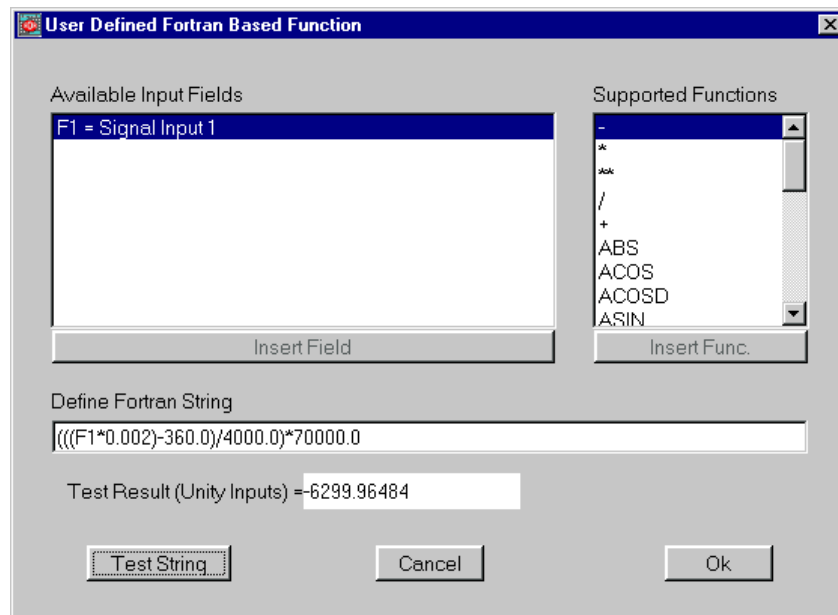


Figure 10.7. Entering the actuator control string.

The **User Function** control element allows the entry of a generic user defined maths function based on Fortran syntax. This can be used to perform any maths-based action on the input signal provided it can be written as a single Fortran string using the supported intrinsic functions. The input signal is represented in the string as 'F1'. We will use this function to enter the relationship that we have developed above for waste-gate flow area versus inlet pressure, given in equation 10.6. We can simply enter this equation in the **Define Fortran String** entry box, using the string 'F1' to represent the input signal  $IN^1$ . In this box type  $(((F1*0.002)-360.0)/4000.0)*70000.0$ , as shown in Figure 10.7. We can test the integrity of the equation we have just entered by pressing the **Test String** button. This will solve the equation we have entered in the **Define Fortran String** entry box, using a value of 1.0 for F1. The **Test Result** should be  $-6299.96484$ .

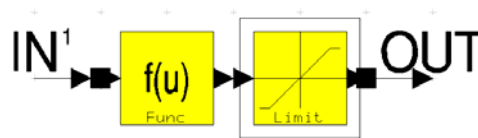


Figure 10.8. Adding the control limit element.

The equation that we have entered in the **User Function** control element will give us a value between 0 and  $700 \text{ mm}^2$  for the waste-gate flow area when the inlet system pressure is between 1.8 and 2.0 bar. However, when the inlet system pressure falls outside this range the waste-gate area returned will be either negative or greater than  $700 \text{ mm}^2$ . We can limit the output from the actuator using the **Limit** control function. Select a **Limit** control element from the *1D Control Elements* toolkit and place it between the **User Function** control element and the **OUT** signal boundary, as shown in Figure 10.8. In the **Limit** control element property sheet, alter the *Maximum Value* to **700.0** and the *Minimum Value* to **0.0**, as shown in Figure 10.9.

Label	1d control limit
Maximum Value	700.0000
Minimum Value	0.0000

Figure 10.9. Entering the control limits.

Use the level control toggle, shown below, to return to engine model element level.



Select the actuator element and alter the *Actuator Variable* to **Minimum C.S.A.**, the *Actuator Group Type* to **Single**, the *Actuator Apply Type* to **By Value**, and the *Actuator Solve Type* to **Per Time Step**, as shown in Figure 10.10.

Actuator Label	default actuator
Actuator Variable	Minimum C.S.A
Actuator Group Type	Single
Actuator Apply Type	By Value
Actuator Solve Type	Per Time Step

Figure 10.10. Setting the actuator parameters.

We have now fully defined the control system for our waste-gate. You may like to add two additional .TRS plot sensors, so that the inlet plenum pressure and waste-gate cross-sectional area can be monitored during the simulation run. If you can't remember how to add a plot sensor then refer to the *Monitoring Turbocharger Speed* section of the previous tutorial. If .TRS plot sensors are added it is recommended that the *Plot Associate Type* is set to **Crank Angle** and the *Plot Sample Size (deg)* is set to **20**, so that the plot files are in the same format as the existing compressor speed .TRS plot created in the previous tutorial. Remember to give the plot files a name.

## Running the Turbocharged Model with a Waste-gate

Change the file descriptions and set the *Test No.* to **Tutorial9** then save the data file to **Tutorial\_9.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to set the .mrs and .prs file names.

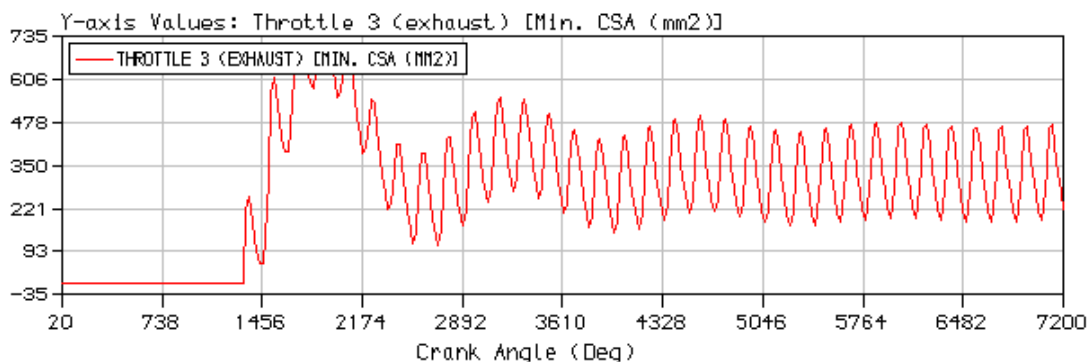


Figure 10.11. TRS output of waste-gate flow area (5000 rev/min).

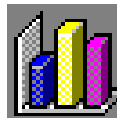


Once the job is successfully running we can monitor its progress using the *job status* tab on the solver control window and by viewing the .TRS plots that we have defined. You will notice that the waste-gate flow area fluctuates with a frequency of about  $180^\circ$  (i.e. the induction event of each cylinder causes a fluctuation in inlet pressure which in turn causes the waste-gate area to change). In reality the motion of the waste-gate is damped because the mass of the valve prevents it responding instantly to pressure changes. We shall examine how to damp the motion of the waste-gate later in this tutorial.

## Viewing the Results

Once the job has finished, the Batch Job Finished window will appear. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. We can press the Cancel button, without selecting any of the options.

We shall use the graphical viewer to compare the .MRS results obtained from Tutorial 8 with or model with the waste-gate. Open the .MRS file viewer by clicking on the icon shown below.



Select **File / Load Results Exclusive** from the .MRS viewer menubar. The file browser window will appear. Now select the file **Tutorial8.mrs** and click on **Open** to load the .MRS results from the previous tutorial. Now select **File / Load Results Shuffle** from the .MRS viewer menubar (the shuffle option will move the currently held .MRS file along one position and add the newly opened file at position 1). The file browser window will appear again. Now select the file **Tutorial9.mrs** and click on **Open** to load the .MRS results from the present tutorial.

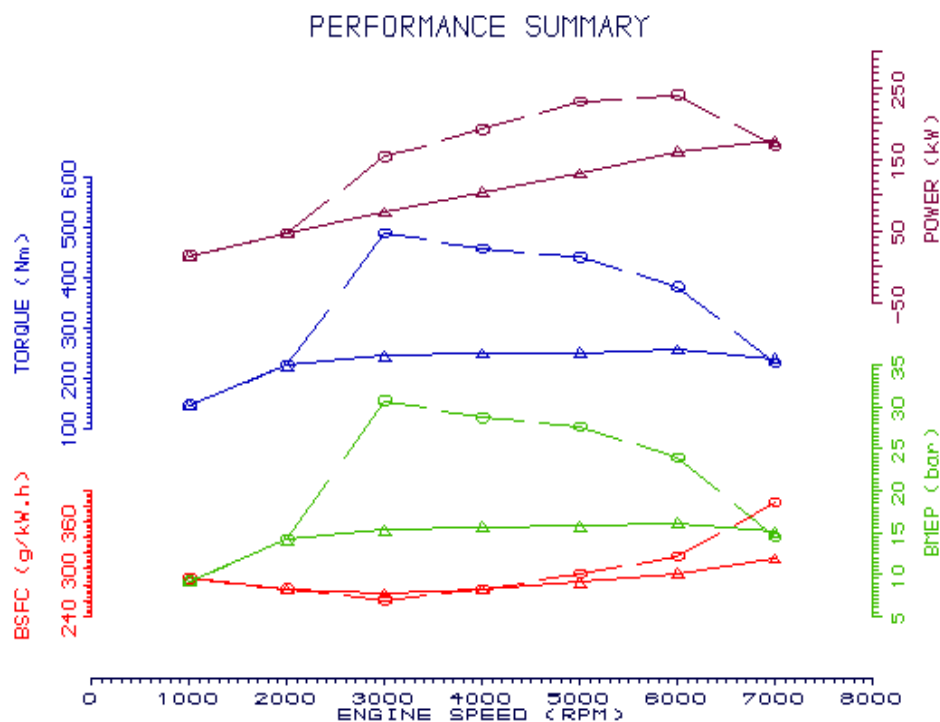


Figure 10.12. .MRS Results Graphical viewer.

Select **View / Autoscale** option from .MRS viewer menubar. Your .MRS graph display should now look similar to the one shown in Figure 10.12 (you may want to change the line style or colour so it is easier to distinguish between the results of tutorial 8 from those of this tutorial).

This plot (see Fig. 10.12) shows the BSFC (brake specific fuel consumption), BMEP (brake mean effective pressure), Torque, and Power produced by the engine. It is clear from the performance summary results, shown in Figure 10.12, that the addition of our waste-gate has limited the BMEP of the engine, between 3000 and 6000 rev/min, to about 16 bar. It is interesting to note that the waste-gate has had no effect on the engine performance below 2000 rev/min.

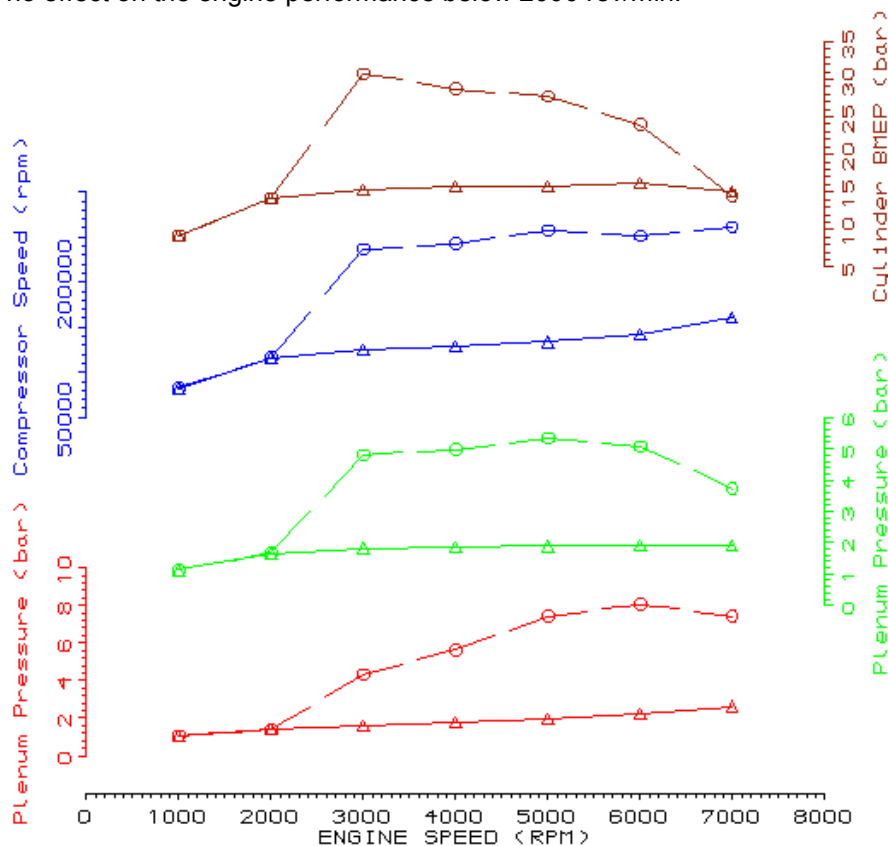


Figure 10.13. Turbocharger operating parameters.

Select **Display / Plot / Plot 8** from the .MRS graph viewer menu bar. The viewer window should now change to display the plots shown in Figure 10.13. If the screen appears blank, see the section entitled *Viewing the Results* in tutorial 8, to find out how to specify the .MRS plotting options to display the turbocharger operating parameters. Comparison of the results for tutorial 8 with those for tutorial 9 reveals that the pressure in the plenum directly upstream of the turbine is now limited to about 2.6 bar. Also, the inlet system pressure is below 2.0 bar across the entire engine speed range. We can also see that at 1000 and 2000 rev/min the inlet system pressure is below 1.8 bar, thus our waste-gate doesn't open at these two engine speeds. This is the reason that engine performance, at these two speeds, remains unaffected by the addition of a waste-gate. The addition of the waste-gate has also limited compressor speed.

## Waste-gate for Variable Boost Pressure

Modern electronic control systems allow the maximum boost pressure to vary with engine speed. We will modify our simple waste-gate, so that we can vary the pressure at which the valve begins to open. We will be simply altering the spring preload, given by equation (10.1). We could just as easily use the technique employed here to vary any other property of our waste gate, it is probably easier to start

from equation (10.7). Close the *.MRS Viewer* window and return to the *Builder Interface*, if you haven't done so already. We need to add a sensor to detect crankshaft speed. Click on cylinder number 4 and then select **On** for the *Harness Connector* option in the element properties menu. Pick a **Sensor** from the *Sensors & Actuators* toolkit and place it in the model window. Using the harness wires, connect this sensor to cylinder number 4 and the waste-gate actuator, as shown in Figure 10.14. Click on the sensor and select **Crank Speed (rpm)** as the *Sensed Parameter* and set the *Sensor Group Type* to **Single**.

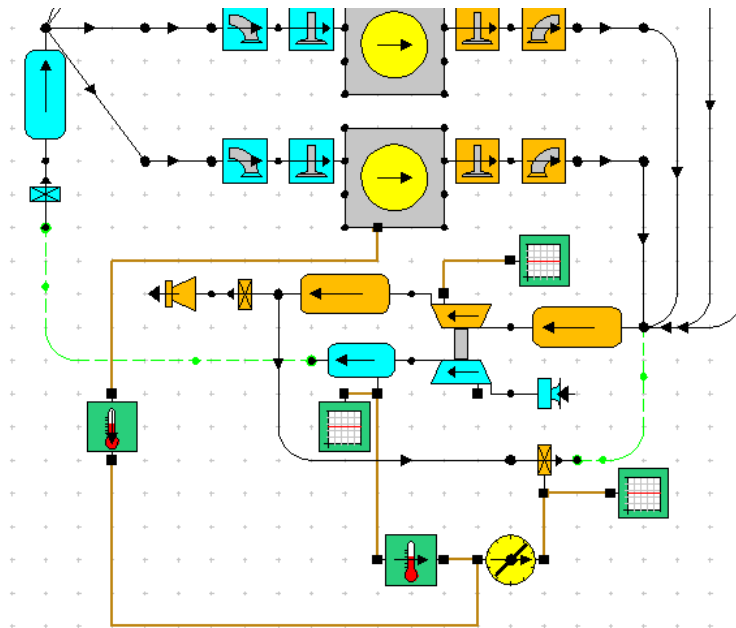


Figure 10.14. Adding a speed sensor to the waste-gate actuator.

Select the actuator element and use the level control toggle, shown below, to drop into the control element level.



Now select a **1D Table** element from the *1D Control Elements* toolkit and place it next to the  $IN^2$  signal boundary. Delete the existing 1D User Function element. Now select a **2D User Function** element from the *2D Control Elements* toolkit and place it next to the **1D Table** so that the output signal of the **1D Table** is connected to the **Y** input of the **2D User Function** element. Connect the existing **Limit** element to the output of the **2D User Function** element. Now connect the **OUT** signal boundary to the output of the **Limit** element. Finally connect the  $IN^1$  signal boundary to the **X** input of the **2D User Function** element using a **Virtual Link**. Your control element network should now appear similar to the one shown in Figure 10.15.

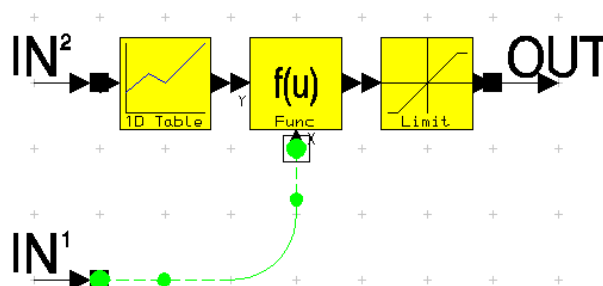
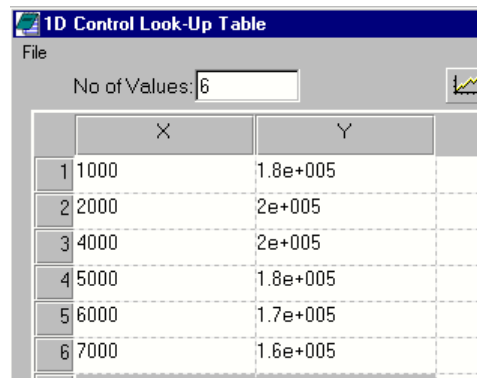


Figure 10.15. Adding a speed sensor to the waste-gate actuator.

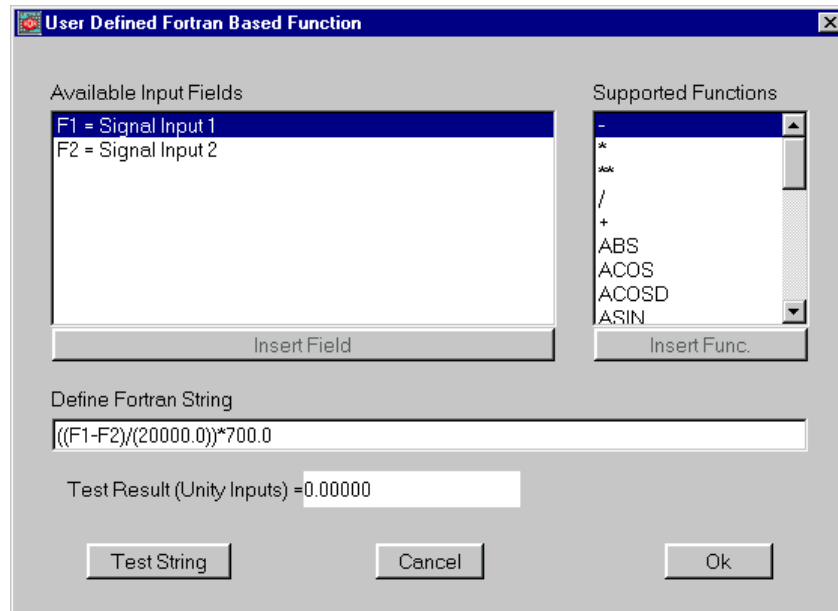


The dialog box titled '1D Control Look-Up Table' has a 'File' menu and a 'No of Values' input field set to 6. It contains a table with two columns, X and Y, and six rows of data.

	X	Y
1	1000	1.8e+005
2	2000	2e+005
3	4000	2e+005
4	5000	1.8e+005
5	6000	1.7e+005
6	7000	1.6e+005

Figure 10.16. Entering the variation in waste-gate opening pressure.

Select the **1D Table** element and click on *LookUp Data* to open the data entry window, shown in Figure 10.16. In this table we need to enter the pressure at which the waste-gate valve will begin to open. Enter **6** in the *No. of Values* box. In the **X** column we shall type a number of engine speeds (note that we do not need to enter speeds corresponding to test points, as the Look-Up Table elements will linearly interpolate the data entered between speed points. They can also be set to extrapolate the data if the current speed is beyond the range of the data entered, or simply use the closest speed point data). Enter the speed points as **1000, 2000, 4000, 5000, 6000 and 7000**. In the **Y** column we shall enter the opening pressures (in N/m<sup>2</sup>) for each of the speeds in column **X**. Enter **180000, 200000, 200000, 180000, 170000 and 160000** for the **Y** column data. Close the data entry window and ensure that the *LookUp Type* option is set to **Extrapolate Only**.



The dialog box titled 'User Defined Fortran Based Function' has two main sections: 'Available Input Fields' and 'Supported Functions'. The 'Available Input Fields' section lists 'F1 = Signal Input 1' and 'F2 = Signal Input 2'. The 'Supported Functions' section lists various mathematical functions including ABS, ACOS, ACOSD, and ASIN. Below these sections are buttons for 'Insert Field' and 'Insert Func.'. At the bottom, there is a 'Define Fortran String' text box containing the equation  $((F1-F2)/(20000.0))*700.0$ , a 'Test Result (Unity Inputs) = 0.00000' display, and buttons for 'Test String', 'Cancel', and 'Ok'.

Figure 10.17. Adding a speed sensor to the waste-gate actuator.

Select the **2D User Function** element and click on the pad icon to open the data entry window, shown in Figure 10.17. We need to replace the previous value we had for the spring preload, which was 360 N, with our **1D Table** output multiplied by the diaphragm area. The equation we need to type into the **Define Fortran String** entry box becomes  $((F1-F2)/(20000.0))*700.0$ , as shown in Figure 10.17. The string 'F1' represents the input signal **IN<sup>1</sup>**, which is the inlet system pressure, and the string 'F2' represents the output signal from our **1D Table**. We can test the integrity of the equation we have just entered by pressing the **Test String** button. This will solve the equation we have entered in the **Define**

**Fortran String** entry box, using a value of 1.0 for both F1 and F2. The **Test Result** should be 0.00000.

Use the level control toggle, shown below, to return to engine model element level.



Change the file descriptions and set the *Test No.* to **Tutorial9b** then save the data file to **Tutorial\_9b.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to update the .mrs and .prs file names.

Once the job has finished, the Batch Job Finished window will appear. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. Tick the box to **Load Graphics Results (.mrs) / Load Results Shuffle** (the shuffle option will move the currently held .MRS file along one position and add the newly opened file at position 1). Open the .MRS graphical viewer. Select the **Autoscale** option from the **View** menu within this window. If you still have the **Tutorial9.mrs** file loaded, this will produce the window shown in Figure 10.18 (if you haven't got the **Tutorial9.mrs** file open, select **File / Load Results (position) / Position 2** from the .MRS viewer menubar then select the file **Tutorial9.mrs** from the file browser window).

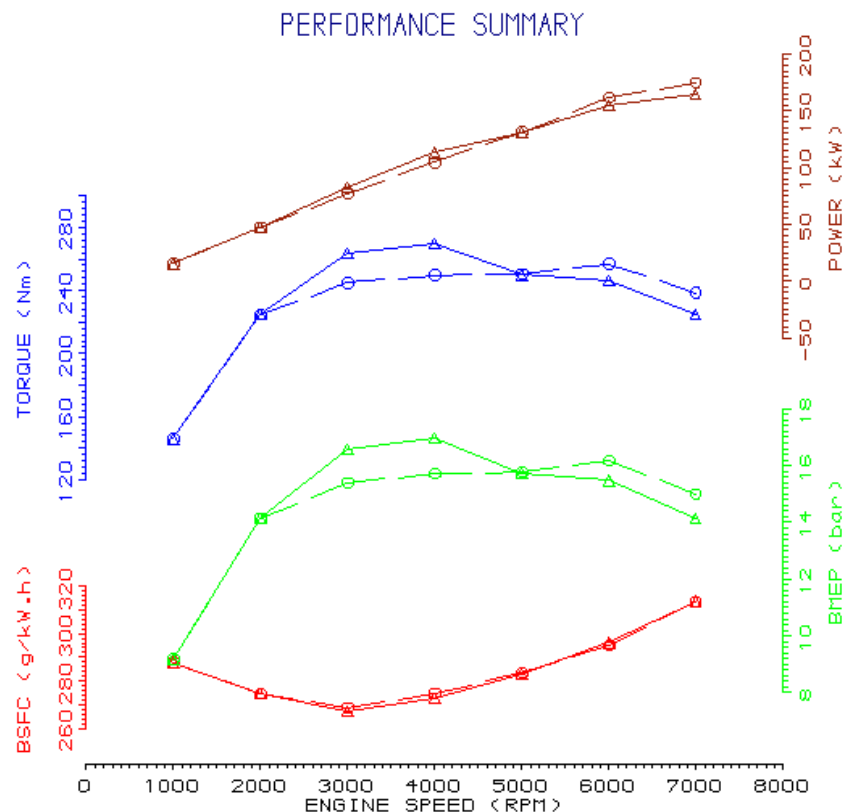


Figure 10.18. Comparison of the results for Tutorial\_9 and Tutorial\_9b.

We can see, from the results shown in the .MRS file, that our modified waste-gate has had no effect on the engine performance at 1000, 2000 and 5000 rev/min. This would be anticipated at 1000 and 5000 rev/min, as we have set the opening pressure to be equal to the original opening pressure at these speeds. The performance of the engine at 2000 rev/min remains unaltered, despite us changing the opening pressure of the waste-gate, because at 2000 rev/min and below, the inlet system

pressure is insufficient to open the waste-gate. It is clear that the increased opening pressure at 3000 and 4000 rev/min has increased the BMEP of the engine at these speeds. Similarly, the reduced opening pressure at 6000 and 7000 rev/min has reduced the BMEP of the engine. If we examine the boost pressure across the speed range we will see that our modified waste-gate is working as expected.

## Damping the Waste-gate Motion

We noted earlier, whilst viewing the instantaneous waste-gate flow area in the TRS plot file output, that it oscillated with a period of roughly  $180^\circ$ . This occurs because the simple control system we have constructed contains no damping. In reality the waste-gate valve cannot respond instantly to variations in inlet system pressure as a finite time is required to accelerate, and decelerate, the valve mass. Also, the inlet system pressure is usually fed to the actuator diaphragm via a relatively long pipe with a small diameter, which acts to damp the inlet pressure fluctuations 'felt' by the waste-gate. There are a number of ways we can damp the motion of the waste-gate valve using the *Sensors & Actuators* elements within LOTUS ENGINE SIMULATION.

We shall damp the waste-gate motion by damping the inlet pressure signal passed to the waste-gate actuator. Click on the inlet pressure Sensor element and then use the level control toggle to drop into the control element level. Add an **Integrator** control element and **User Function** control element between the **IN<sup>1</sup>** and **OUT** signal boundaries, as shown in Figure 10.18.

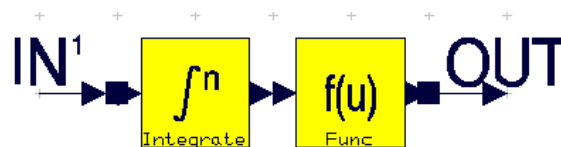


Figure 10.18. TRS output of waste-gate flow area (5000 rev/min).

Select the **Integrator** element and change the *Integrate w.r.t.* option to **Crank Degrees** and the *Integral Period (crank deg)* to **180** in the element property sheet, as shown in Figure 10.19. This element integrates the input signal over the specified period.

Label	1d control Integrator
Integrate w.r.t	Crank Degrees
Integral Period (crank deg)	180.0000

Figure 10.19. Property sheet for the **Integrator** control element.

Now click on the **User Function** element and open the data entry window. We need to divide the output signal from the **Integrator** element by the number of degrees the signal has been summed over, otherwise the signal will be 180 times too high! Enter **F1/180.0** in *Define Fortran String* entry box. Test the integrity of the equation we have just entered by pressing the **Test String** button. This will solve the equation we have entered in the *Define Fortran String* entry box, using a value of 1.0 for F1. The **Test Result** should be 0.00556.

Use the level control toggle, shown below, to return to engine model element level.

Change the file descriptions and set the *Test No.* to **Tutorial9c** then save the data file to **Tutorial\_9c.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to set the .mrs and .prs file names.

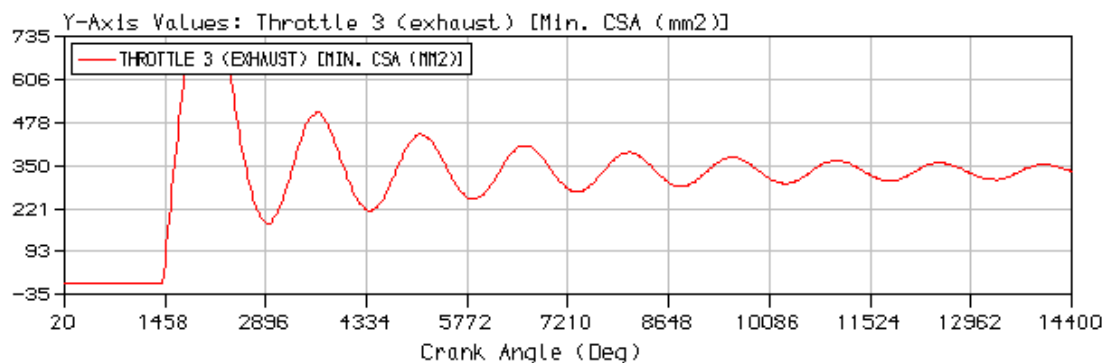


Figure 10.20. TRS output of waste-gate flow area (5000 rev/min).

Once the job is successfully running we can monitor its progress using the *job status* tab on the solver control window and by viewing the .TRS plots that we have defined, as shown in Figure 10.20. You will now notice that the frequency of the waste-gate flow area variation is lower than before (c.f. Figure 10.20 with Figure 10.11). However, whilst the motion of the waste-gate is damped, more engine cycles need to be simulated to achieve cyclic convergence.

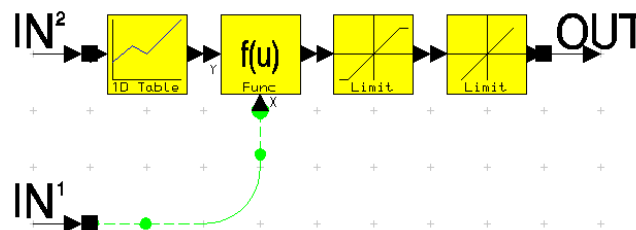


Figure 10.21. Add a **Limit Change** element to the actuator.

## Exercise

Instead of using the **Integrator** control element to damp the waste-gate motion, try adding a **Limit Change** element prior to the output of the waste-gate actuator element. Begin by either deleting the two control elements we added to the pressure sensor, or re-load the **Tutorial\_9b.sim** file. Add a **Limit Change** element to the end of actuator control network, as shown in Figure 10.21. Alter the *Limit Change* w.r.t. to **Crank Degrees**, the *Rate of Change* to **0.2** and the *Limit Type* to **-ve direction only**. This will have the effect of only allowing the waste-gate area to decrease by 0.2 mm² per degree CA, without limiting the opening rate. A comparison of Figure 10.22 with Figure 10.11 shows the effect that this has on the waste-gate area during the simulation. Try altering the waste-gate control strategy and damping mechanism to see the effect that they have on the waste-gate. To save time you may prefer to investigate the effect at just one or two engine speeds.

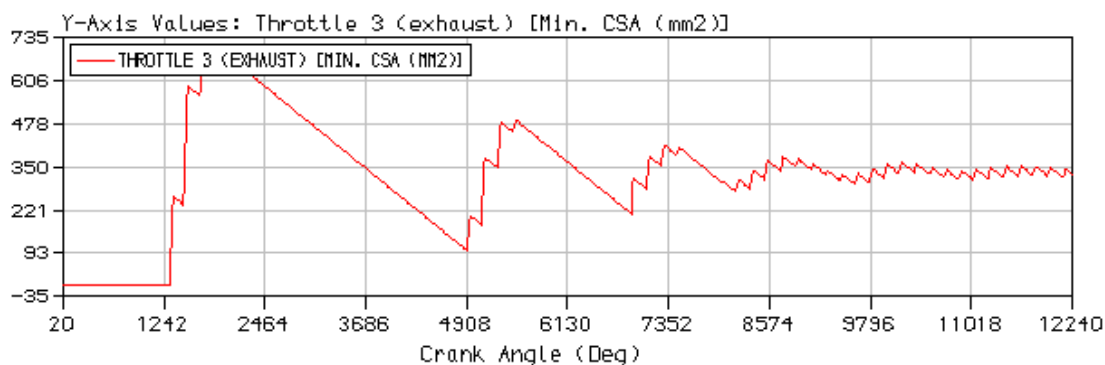


Figure 10.22. TRS output of waste-gate flow area (5000 rev/min).





# 11

## Tutorial 10. Modelling a Multi-Cylinder Turbocharged Diesel Engine

### Overview

This chapter introduces the user to modelling high-speed diesel engines. Due to the large amount of data which needs to be entered an example of a small turbocharged direct-injection diesel engine is studied. Examples describing the building of most of the individual features of this model, such as the pipe network, the turbocharger, and the waste-gate, and the use of .trs plots, have been described in the earlier tutorials. In this chapter it is assumed that the user has completed the earlier tutorials. The new features of this model are the use of a diesel fuel injection system, the definition of a diesel combustion model, and the use of a charge-cooler.

The chapter contains the following sections:

- Description of the Model, 128
- Running the Model, 133
- Discussion of Results, 134
- Exercise, 135

## Description of the Model

Load the example model **tutorial\_10.sim** from the **Examples** sub-folder created during the install process. If the model does not appear use the **Control A** command to centre the model. The builder window should appear as shown in Figure. 11.1.

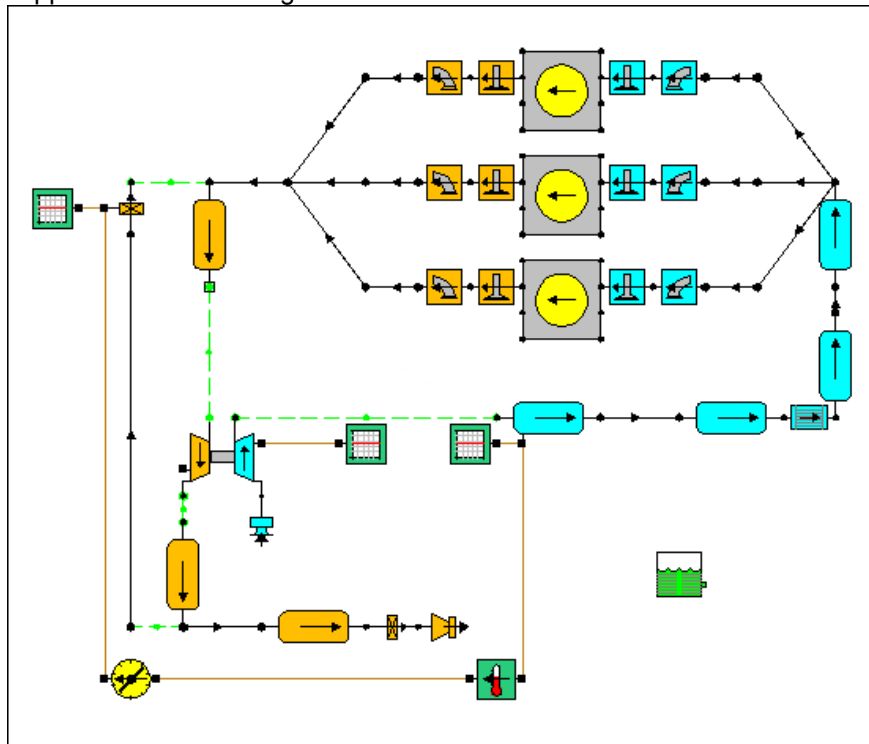


Figure 11.1. Small high-speed DI diesel engine.

The model represents a 1.45 litre three-cylinder direct-injection diesel engine fitted with a turbocharger, waste-gate and charge-cooler. Clicking on the fuel tank icon shown below



reveals the 'fuel system' and 'fuel type' property sheet shown in Figure 11.2.

Label	
Fuel System	Direct Injection
Fuel Type	Diesel
Calorific Value (kJ/kg)	42700.0
Density (kg/litre)	0.8400
H/C Ratio Fuel (molar)	1.900
O/C Ratio Fuel (molar)	0.000
Molecular Mass (kg/k.mol)	170.000
Maldistribution Factor	1.000

Figure 11.2. Property sheet for 'fuel system' and 'fuel type' definition.

A *direct-injection* fuel system has been specified so that the fuel will be injected directly into the cylinder. The fuel type has been set to *Diesel* and the default properties for that fuel have been assumed – these are shown in the greyed-out boxes of the property sheet.

The nature of the combustion process in diesel engines differs from that in spark-ignition engines and different options are available to cater for this in Lotus Engine Simulation. The energy release in a cylinder is characterised by the fraction of the mass of the fuel burnt at any particular crank angle position. This parameter is referred to as the 'mass fraction burnt' and is defined as

$$\text{MFB}_\theta = \frac{\text{Mass of fuel burnt up to crank angle position } \theta}{\text{Total mass of fuel trapped in the cylinder}}.$$

In Lotus Engine Simulation the variation of mass fraction burnt during the combustion event can be specified directly at each engine operating condition or it can be modelled using a Wiebe function. If the latter approach is adopted a two-part Wiebe function is the most suitable way of modelling the combustion event in direct injection diesel engines. This model includes separate terms for the pre-mixed and diffusion controlled periods of the combustion process. The mathematical details of this and the other models available in Lotus Engine Simulation can be found in the **On-line Help** (*Help Topics/Contents/Theory/Combustion Models*).

The heat release model using the two-part Wiebe function is defined from the data screen shown in Figure 11.3 which is generated by clicking on the **Combustion Model** button of the cylinder property sheet.

**Combustion Data**

Help

Combustion Model

Type: **Two-Part Wiebe**

Two-part Wiebe function

☐ Default

A: 6.900 M: 0.500 CP1: 1.00

CP2: 1.0 Fract.: 0.00 Delay: 0.0

☒ User

A: 8.900 M: 0.600 CP1: 1.00

CP2: 1.0 Fract.: 0.20 Delay: 0.0

Figure 11.3. Setting up a two-part Wiebe function.

The shape of the cumulative mass fraction burnt curve and the rate of heat release (burn rate) curve, defined by the coefficients entered, can be seen by clicking on the graph icon on this screen. This generated the plots shown in Figure 11.4.

The significance of the coefficients used in this model are described in the **On-line Help** (*Help Topics/Contents/Theory/Combustion Models*).

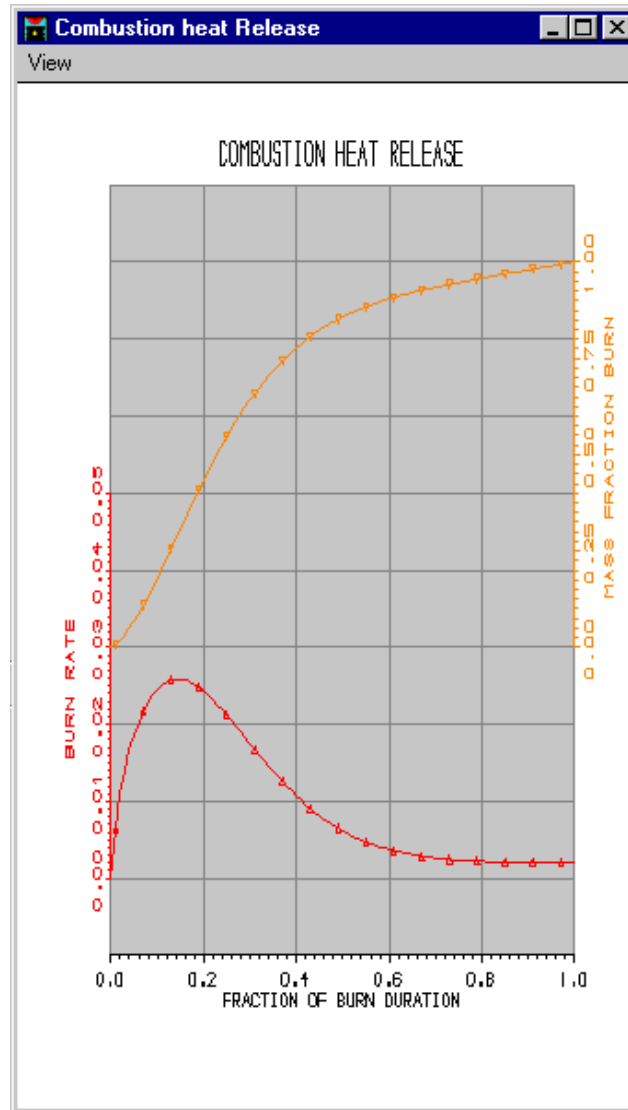
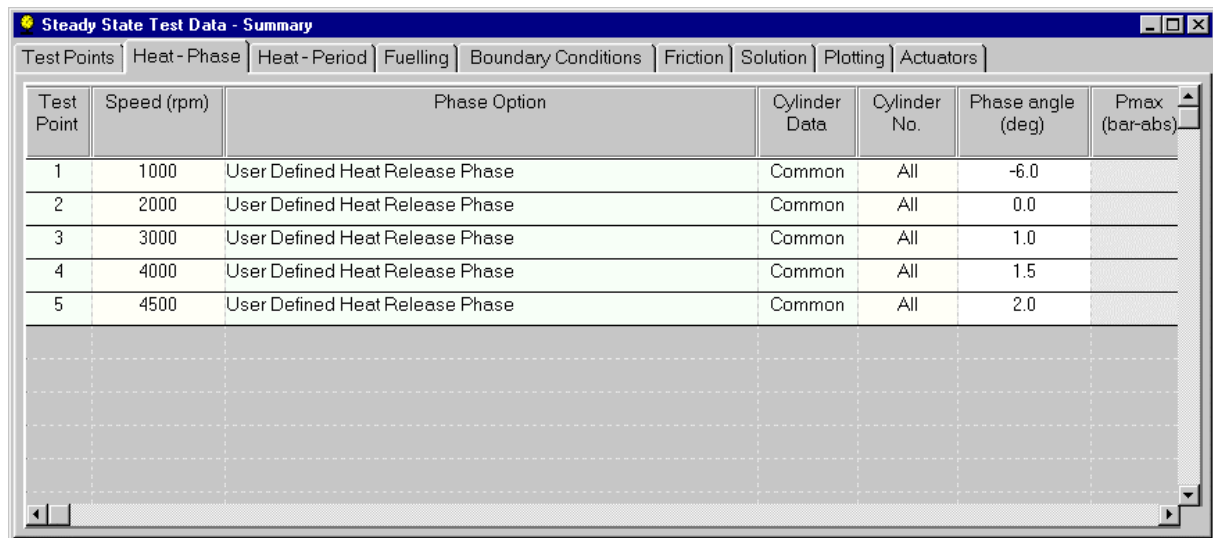


Figure 11.4. Variation of cumulative mass fraction burnt and heat release rate with fraction of the combustion period.

To view the data defining the phasing and period of the combustion event click on the **Steady-State Test Data** icon shown below.



Clicking on the **Heat - Phase** tab reveals the data screen shown in Figure. 11.5.



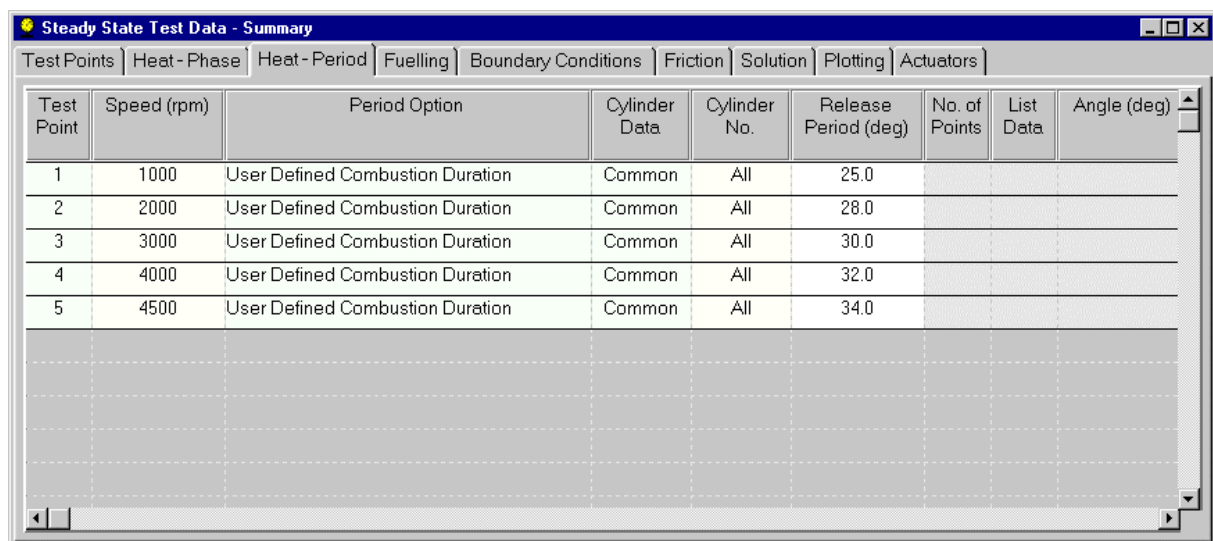
Test Point	Speed (rpm)	Phase Option	Cylinder Data	Cylinder No.	Phase angle (deg)	Pmax (bar-abs)
1	1000	User Defined Heat Release Phase	Common	All	-6.0	
2	2000	User Defined Heat Release Phase	Common	All	0.0	
3	3000	User Defined Heat Release Phase	Common	All	1.0	
4	4000	User Defined Heat Release Phase	Common	All	1.5	
5	4500	User Defined Heat Release Phase	Common	All	2.0	

Figure 11.5. Heat release phase data specifying combustion event.

It can be seen that the **User Defined Heat Release Phase** option has been selected and the phase angle, which denotes the start of combustion in a diesel engine, has been set-up to vary with engine speed. Use the **On-line Help** (*Help Topics/Contents/Theory/Combustion Models*) to clarify the different definitions of **Heat Phase** for the different combustion systems available in Lotus Engine Simulation.

Note that a *positive* value of the **Phase angle** indicates that the start of combustion will occur *before* top-dead-centre whereas a *negative* value indicates that the start of combustion will occur *after* top-dead-centre. The start of combustion has been set as *Common* to all cylinders in this model but can be varied between cylinders if desired.

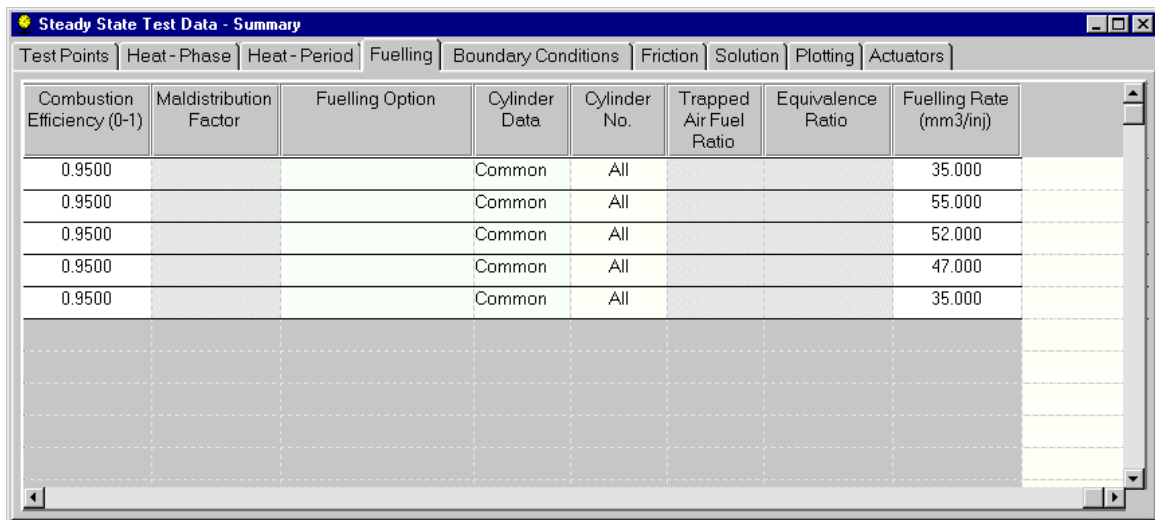
Clicking on the **Heat – Period** tab produces the data screen shown in Figure 11.6. In this table the **Period Option** has been set to *User Defined Combustion Duration*. The **Release Period** has been set for the individual engine speed as shown in the table.



Test Point	Speed (rpm)	Period Option	Cylinder Data	Cylinder No.	Release Period (deg)	No. of Points	List Data	Angle (deg)
1	1000	User Defined Combustion Duration	Common	All	25.0			
2	2000	User Defined Combustion Duration	Common	All	28.0			
3	3000	User Defined Combustion Duration	Common	All	30.0			
4	4000	User Defined Combustion Duration	Common	All	32.0			
5	4500	User Defined Combustion Duration	Common	All	34.0			

Figure 11.6. Heat release period data specifying combustion event.

The final pieces of data required to define the diesel combustion event are the **Combustion Efficiency** and the quantity of fuel injected to each cylinder per cycle (**Fuelling Rate**). The data window in which this information can be specified is accessed by clicking on the **Fuelling** tab, and is shown in Figure 11.7.



Combustion Efficiency (0-1)	Maldistribution Factor	Fuelling Option	Cylinder Data	Cylinder No.	Trapped Air Fuel Ratio	Equivalence Ratio	Fuelling Rate (mm <sup>3</sup> /in)
0.9500			Common	All			35.000
0.9500			Common	All			55.000
0.9500			Common	All			52.000
0.9500			Common	All			47.000
0.9500			Common	All			35.000

Figure 11.7. Specifying combustion efficiency and fuelling rate.

Click on the other tabs in the **Steady-State Test Data** to examine the rest of the parameters that have been set. Now close the Steady-State Test Data window.

Clicking on the icon shown below



reveals the top-level property sheet of the **Charge-Cooler** data, shown in Figure 11.8.


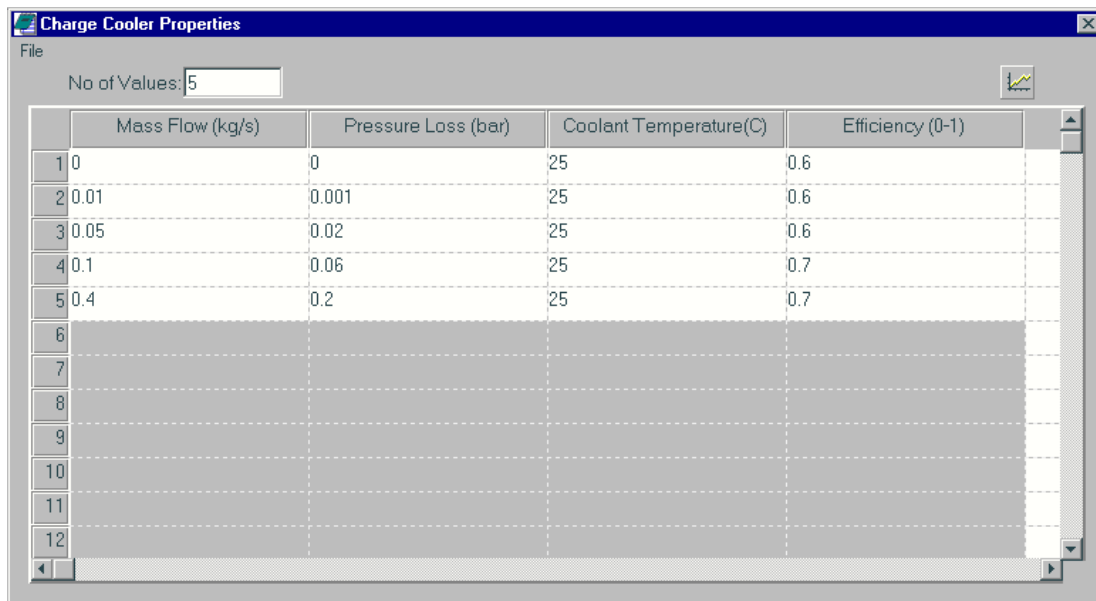
Label	charge cooler
Properties	
Harness Connector	Off

Figure 11.8. Top-level property sheet for charge-cooler.

Clicking on the **Properties** icon produces the data window in which the information characterising the charge-cooler performance is given, as shown in Fig. 11.9.



	Mass Flow (kg/s)	Pressure Loss (bar)	Coolant Temperature (C)	Efficiency (0-1)
1	0	0	25	0.6
2	0.01	0.001	25	0.6
3	0.05	0.02	25	0.6
4	0.1	0.06	25	0.7
5	0.4	0.2	25	0.7
6				
7				
8				
9				
10				
11				
12				

Figure 11.9. Charge-cooler property data.

The performance of the cooler is defined by its efficiency, or effectiveness, as a heat exchanger, and by the pressure-drop / mass flow rate characteristic of the device. The effectiveness of the cooler is defined as

$$\varepsilon = \frac{\text{actual temperature drop}}{\text{maximum theoretical temperature drop}} = \frac{T_{\text{air out}} - T_{\text{air in}}}{T_{\text{air out}} - T_{\text{cool in}}},$$

where  $T_{\text{air out}}$  is the compressor air outlet temperature,  $T_{\text{air in}}$  is the compressor inlet temperature, and  $T_{\text{cool in}}$  is the coolant inlet temperature. This data is usually available from the charge-cooler manufacturer.

The waste-gate model has been set up in a similar way to that described in Chapter 10. Click on the sensor and the actuator to investigate the waste-gate control structure. The maximum opening area of the device is 200 mm<sup>2</sup>.

Familiarise yourself with the rest of the model data by viewing the property sheets for the other elements.

## Running the Model

Filenames should now be assigned to the output from the .trs plot sensors in order to run the model.

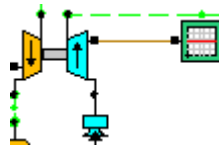


Figure 11.10. Plot sensor for compressor speed.

The plot sensor attached to the right-hand side of the turbocharger, shown in Figure 11.10, is sensing compressor speed, defined as  $N / \sqrt{T / T_{\text{ref}}}$ , where  $T$  is the compressor inlet temperature and  $T_{\text{ref}}$  is the reference temperature (298 K). The plot sensor attached to the inlet plenum, shown in Figure

11.11, is sensing compressor outlet pressure, while that attached to the turbocharger waste-gate is sensing the flow area of the restriction (Figure 11.12).

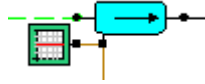


Figure 11.11. Plot sensor for compressor outlet pressure.

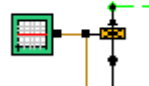


Figure 11.12. Plot sensor for compressor outlet pressure.

The model can now be run by submitting the job in the normal way (see Chapter 2 – Tutorial 1). Remember to use the .trs sensor plots to monitor the progress of the run, as described in Chapter 10.

### Discussion of Results

Steady state test points have been set up at 1000, 2000, 3000, 4000, and 4500 rev/min. By monitoring the output from the .trs plot sensors the user will see that only a minimal amount of boost is produced by the compressor (about 1.2 bar absolute – compressor outlet pressure) at 1000 rev/min. and the waste-gate does not open.

At 2000 rev/min the compressor outlet pressure has increased considerably to about 1.85 bar – the waste-gate is now open. Plot the performance results for the engine as described in Chapter 2 – they should appear as shown in Figure 11.13.

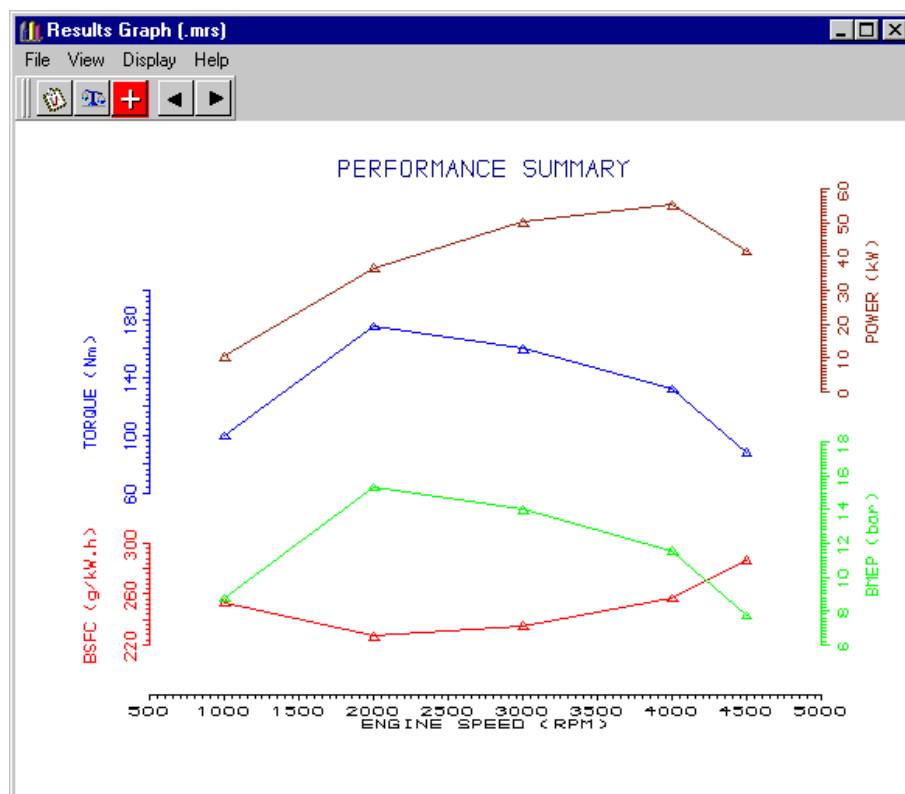


Figure 11.13. Variation of power, torque, BMEP, and BSFC with engine speed.

The torque curve is characteristic of a small high-speed direct-injection diesel engine, rising rapidly as the engine to a maximum value of 174 Nm at around 2000 rev/min, and then gradually dropping as



the engine speed increases. The maximum power produced by the engine is about 55 kW (74 hp) at 4000 rev/min.

### **Exercise**

Plot the output data from the .trs sensors at 2000 and 4500 rev/min.



# 12

## Tutorial 11. Modelling the Transient Response of a Turbocharged Diesel Engine

### Overview

This chapter introduces the user to the modelling of engine transient performance using the model considered in Tutorial 10 as the starting point. The model is first run at reduced load with the engine speed set at 2000 rev/min. The set-up procedure for a transient event is then described starting from this operating condition.

The chapter contains the following sections:

- Setting Up the Part-Load Condition, 138
- Entering Mechanical Property Data and Attaching a Shaft, 139
- Defining a Transient Event, 141
- Running the Model, 144
- Discussion of Results, 145
- Exercise, 145

## Setting Up the Part Load Condition

Load the model **tutorial\_10.sim** and save the file as **tutorial\_11.sim**. Open the **Steady State Test Data** icon and turn all the **Test Points** off by clicking on the **Solve** column and selecting *All Off* from the pop-up menu which appears. Now turn on Test Point 2 by clicking on the 2000 rev/min row of the **Solve** column and using the same pop-up menu to select *On*.

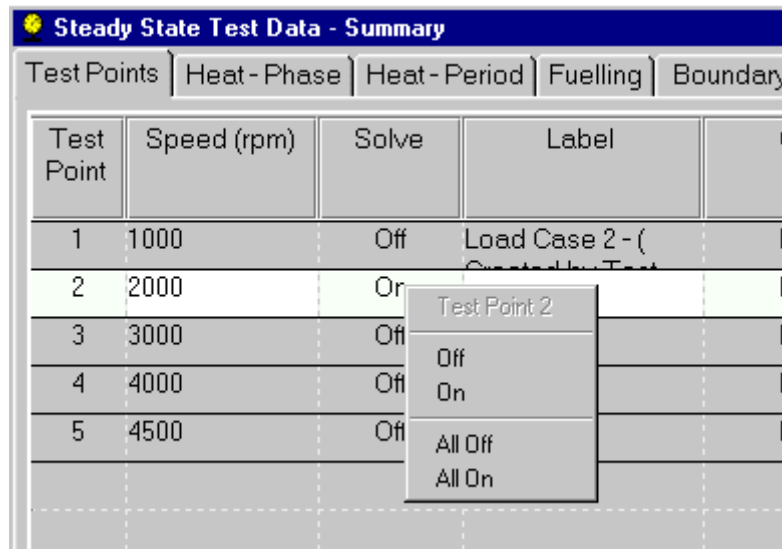


Figure 12.1. Turning the 2000 rev/min test point on.

Now click on the **Fuelling** tab and reduce the **Fuelling Rate** at 2000 rev/min to 40 mm<sup>3</sup>/inj. as shown in Figure 12.2. Now close the **Steady State Test Data**. Open the **Text File Descriptions** data window from the **Data/File Descriptions** option on the drop-down menus at the top of the interface window and change the **Test No.** entry to *tutorial\_11* as shown in Figure 12.3. Click **OK** and save the file. Now run the job, clicking on the **Use Test No.** option in the job submission window to set the name of the output files.

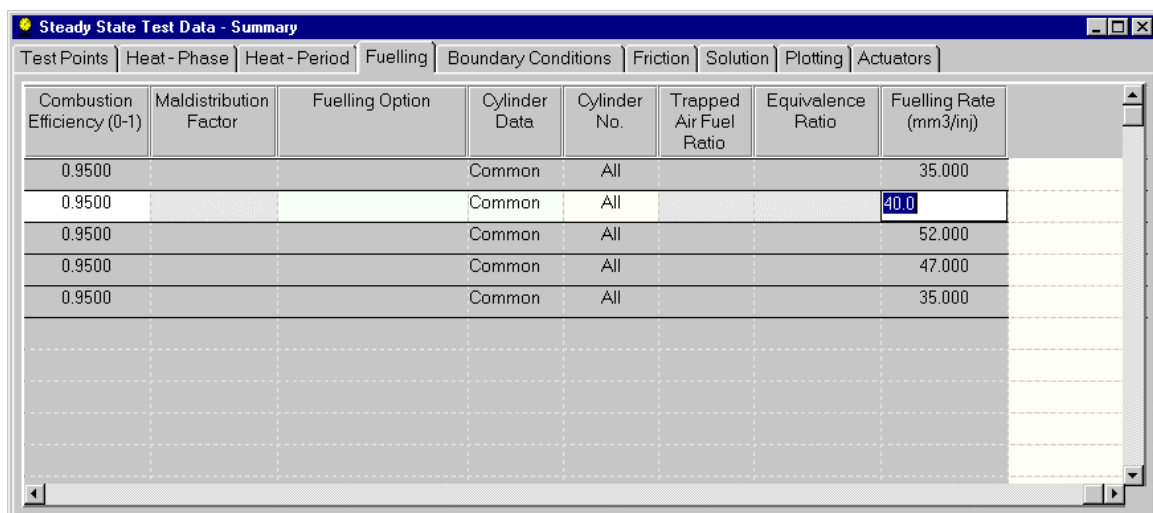


Figure 12.2. Changing the quantity of fuel injected per cycle.

**File Text Descriptors**

Main Title  
example small TDI diesel

Sub Title

Test No. (must be a SINGLE word)  
tutorial\_11

OK Cancel

Figure 12.3. Changing the Test No. text entry.

When the job has run open the .mrs file in the text viewer (as described in Chapter 2) and find the BMEP values for the cylinders (shown in Figure 12.4). The average value of BMEP should be about 10.8 and this can be used to set the load level in the transient event which will be defined later. Close the text file viewer.

```

PERFORMANCE
~~~~~
Cylinder 1
IMEP (Power bdc-bdc)      12.88 bar    IMEP (Complete Cycle)    12.24 bar
IMEP (Pumping bdc-bdc)   -0.64 bar    BMEP . . . . .          10.83 bar
Mechanical Efficiency     88.5 %      FMEP . . . . .          1.41 bar
Indicated Power . . .     9.74 kW     Brake Power . . . . .    8.62 kW
    
```

Figure 12.4. BMEP data in .MRS file.

## Entering Mechanical Property Data and Attaching a Shaft

Click on each cylinder of the model and enter the mass properties of the piston / connecting rod components, as shown in Figure 12.5. Note that if the connecting rod inertia is set to zero an inertia value is calculated from the lumped rotating and reciprocating masses of the connecting rod. If an inertia value is set for the rod a residual inertia value is calculated as the difference between the lumped inertia and the actual inertia value specified.

The next task is to add a shaft to the model in order to represent the engine crankshaft. First click on the cylinder at the top of the model (cylinder 3) and turn on the **Mechanical Link Connector** by choosing this option from the data property sheet for the cylinder, as shown in Figure 12.5. The user will need to use the vertical scroll bar on the right-hand side of the cylinder data property sheet in order to access this list box. The cylinder should now appear as shown in Fig. 12.6.

Cylinder Phase Display		▲
Cylinder Timing Display		
Harness Connector	Off	▼
Mech Link Connector	On	▼
Transient Data		
Cyl Axis Angle (deg)	0.00	
Piston Mass (kg)	0.34000	
Piston-Pin Mass (kg)	0.11000	
Con-Rod Rot Mass (kg)	0.40000	
Con-Rod Recip Mass (kg)	0.16000	
Con-Rod Inertia (kg.m <sup>2</sup> )	0.00000	
Piston Motion		
Motion Type	Std Crank Slider	▼
User Sub Id No.	0	
User Sub Args		
User Sub DLL Type	Fortran	▼

Figure 12.5. Mechanical property data for piston / connecting rod assembly.

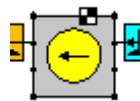
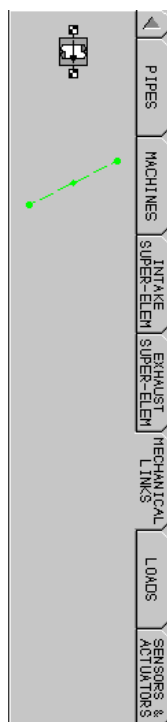


Figure 12.6. Mechanical link connector on cylinder.



Now use the down-arrow at the bottom of the builder tool box to move down the menu tabs until the **Mechanical Links** tab can be seen, as shown in Figure 12.7. Select a virtual link from the tool box and attach it to the mechanical link connector on cylinder 3.

Now select a shaft, flip the connectivity arrow by selecting the **Flip Flow Direction** option from the right-mouse button menu for the shaft, and attach it to the mechanical link connector. Set the rotational inertia of the shaft as  $0.12 \text{ kg m}^2$  – this value represents the inertia of the crankshaft, valve train and accessory system attached to the engine. Because the speed ratio of these systems differs from that of the crankshaft their referred inertia needs to be calculated and added to that of the crankshaft. The referred inertia can be calculated using the square of the speed ratio between the systems multiplied by the actual inertia.

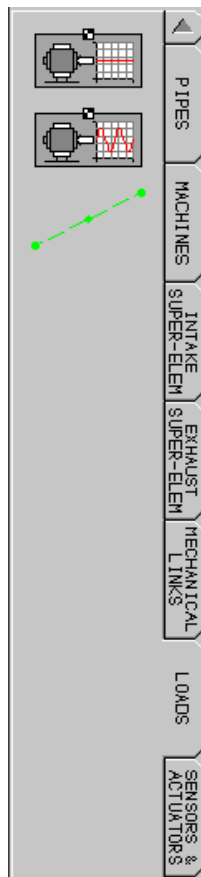
Attach another mechanical link connector to the shaft.

Figure 12.7. Selecting the Mechanical Links tab.

Label	default conn. shaft
Rot. Inertia (kg.m <sup>2</sup> )	0.12

Figure 12.8. Specifying the engine inertia.

## Defining a Transient Event



Select the **Loads** tab from the tool kit menu, as shown in Figure 12.9, and select the transient load icon. Attach it to the top of the mechanical link connector after using the **Flip Flow Direction** option to move the connection point to the bottom of the icon.

The mechanical link / cylinder assembly should now appear as shown in Figure 12.10.

The transient event can now be defined by clicking on the transient load icon to reveal the data property sheet shown in Figure 12.11.

Set the **Initial Steady State Test Point** to *Test Point 2*.

Now click on the **Transient Case Data** button in order to generate the data window shown in Figure 12.12.

Set the **Length Type** as *Cycles* and **Length** to *80* cycles. Now define the **Load Inertia** as *0.25 kg m<sup>2</sup>*.

Set the **Update Speed** option as *By Time Step* – this will cause the program to perform the engine acceleration calculation at every calculation time step. This option is not really necessary for the type of simulation performed in this tutorial but has been selected in order to introduce this feature which can resolve the intra-cycle speed fluctuation of the engine during a transient event.

Figure 12.9. Selecting the Mechanical Links tab.

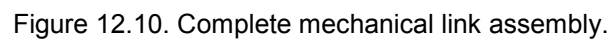


Figure 12.11. Transient load property sheet.

Figure 12.12. Transient test data.



The transient **Event Type** can now be set as *Load vs Cycles* and the **Event Units** set as *MEP* (mean effective pressure). Now set the **No of Points** to 2 and define the start of the event (**Start Point**) to be point 1. Turning the **List Data** option to *On* enter the values in the spreadsheet as shown in Figure 12.13. In this case the load has been set as constant throughout the transient event. The value used here is that obtained from the initial steady state run at the start of the tutorial.

Event Type	Event Units	No of Points	Start Point	List Data	X	Y
Load vs Cycles	MEP (bar)	2	1	On	1	10.8300
					2000	10.8300

Figure 12.13. Transient load definition.

Create a component **Group** containing all the cylinders and turn on the **Harness Connector** for cylinder 3 and click on the **Sensors and Actuators** tab on the tool kit (use the down-arrow until this becomes visible). Attach an actuator and timer as shown in Figure 12.14. Set the **Timer Type** to *Transient (cycle)* in order to record the transient cycle number and set the actuator properties as shown in Figure 12.15.

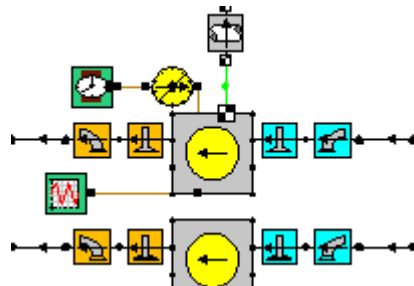


Figure 12.14. Attaching a transient timer, fuelling actuator, and engine speed sensor.

Actuator Label	default actuator
Actuator Variable	Fuelling (mm3/inj)
Actuator Group Type	Group
Actuator Apply Type	By Value
Actuator Solve Type	Per Cycle

Figure 12.15. Actuator properties.

Now set up a one-dimensional look-up table as shown in Figure 12.16 in order to define the schedule for the volume of fuel injected to the cylinders per cycle. Also attach a transient plot sensor to sense engine speed during the transient event, as shown in Figure 12.14, and set its properties as shown in Figure 12.17, using an appropriate file name.

**Also change all the steady state plot sensors to be transient sensors and rename the files they produce so as not to overwrite the files produced by running tutorial\_10.sim.**

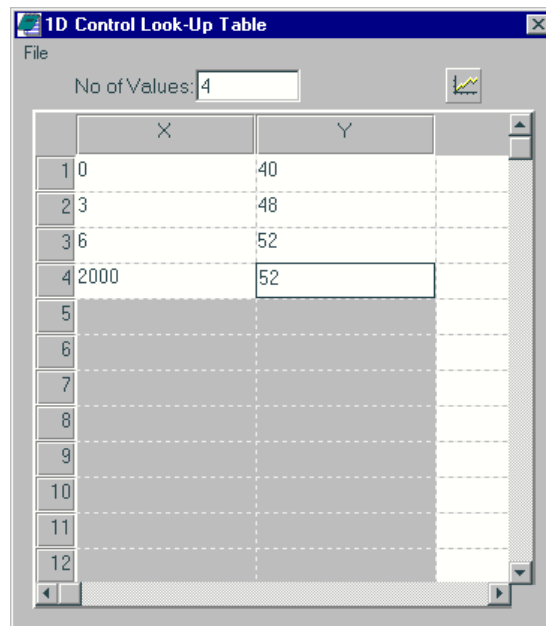


Figure 12.16. Actuator fuelling schedule.

Label	output plot file
File Name - F:\RJP\Source	
Plot for Run Type	Transient
Channel Select	Wire 9
Channel Parameter	Crank Speed (rpm)
Channel Group Type	Single
Channel Element	Cylinder 3
Plot Associate Type	Crank Angle (deg)
Plot Sample Size (deg)	20.000
Plot File Format Type	ASCII
Add File Headings	Omit

Figure 12.17. Transient plot sensor properties attached to cylinder.

Now save the model.

## Running the Model

The user is now in a position to run the model. Note that ideally maps for defining the combustion phasing and duration as functions of engine speed and load should be used with sensors and actuators in order to simulate the changes occurring to the combustion process during a transient event.

Start the simulation. The model will run at the 2000 rev/min point at part load as a steady state simulation in order to initialise the transient event. When the transient event starts [as indicated by (*trans*) appearing in the **Cycle No.** box] use the .trs plots to monitor the progress of the job. Figure 12.18 shows the boost pressure rising as the quantity of fuel injected is raised during the transient.

## Discussion of Results

Figure 12.19 shows that the waste-gate does not open during the part-load operating period at the start of the transient and begins to open as the fuelling rate is built up and the boost pressure increases.

Figure 12.20 shows the engine speed increasing as the fuelling is increased. Because The **Speed Update By time step** option was selected when defining the transient event (see Figure 12.8) the intra-cycle speed variation can be obtained – this is plotted in Figure 12.21.

## Exercise

Decrease the load inertia and re-run the model.

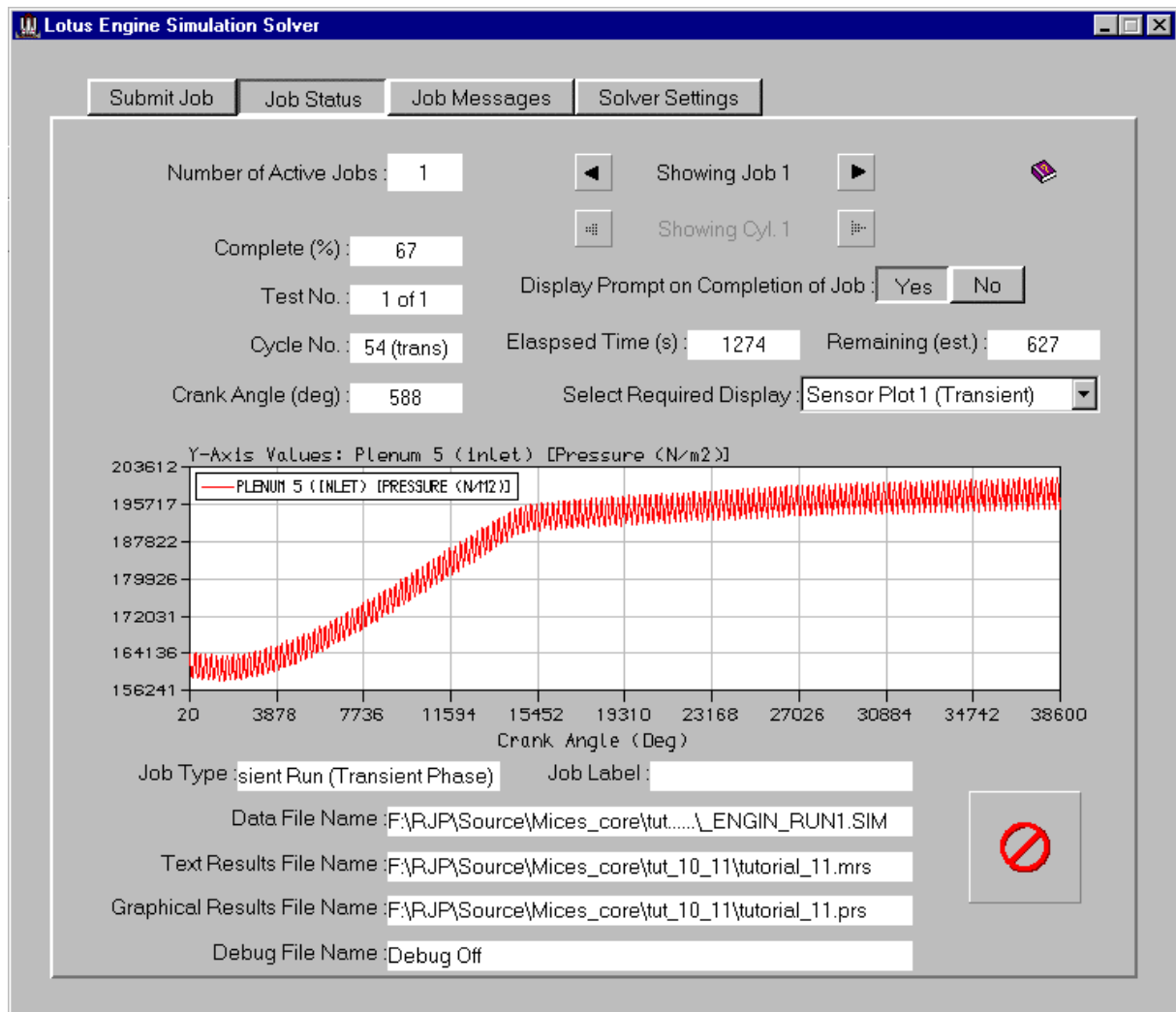


Figure 12.18. Job status window during the transient.

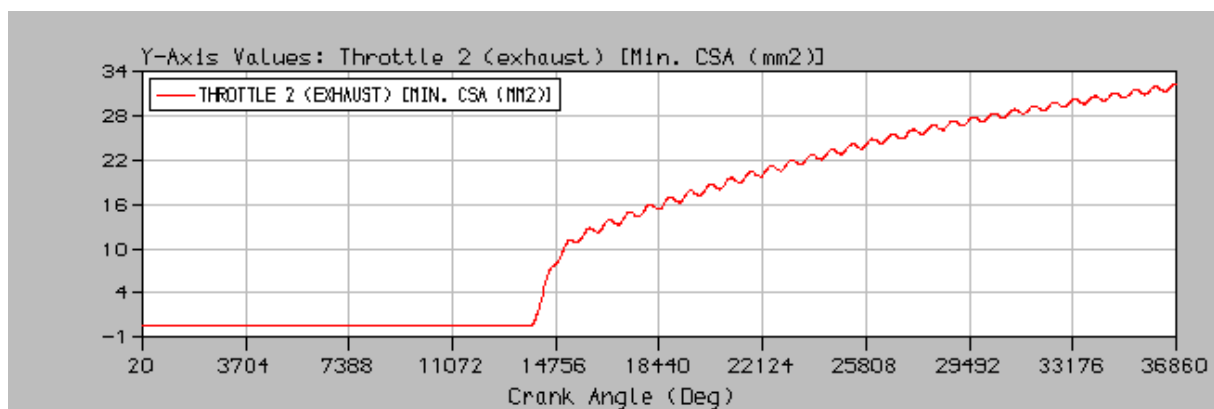


Figure 12.19. Variation of waste-gate cross-sectional area with crank angle.

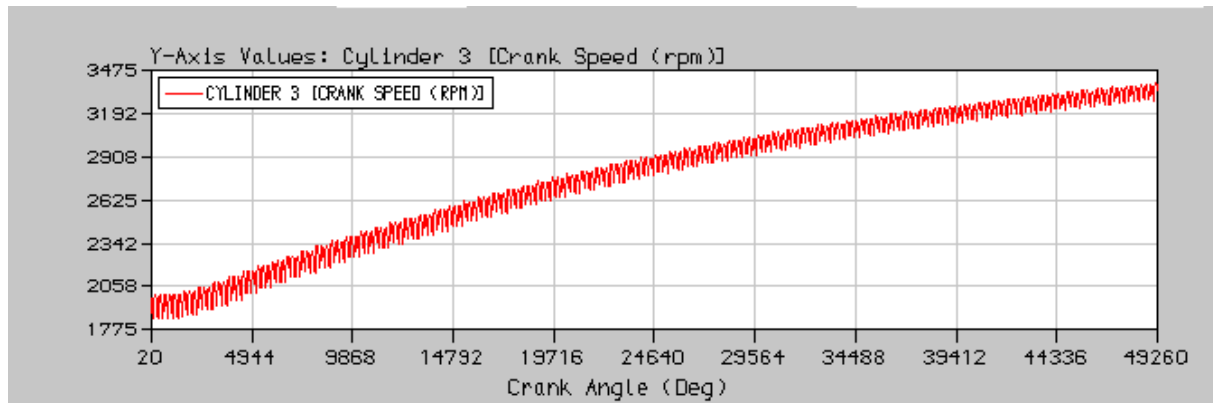


Figure 12.20. Variation of engine speed during the transient event.

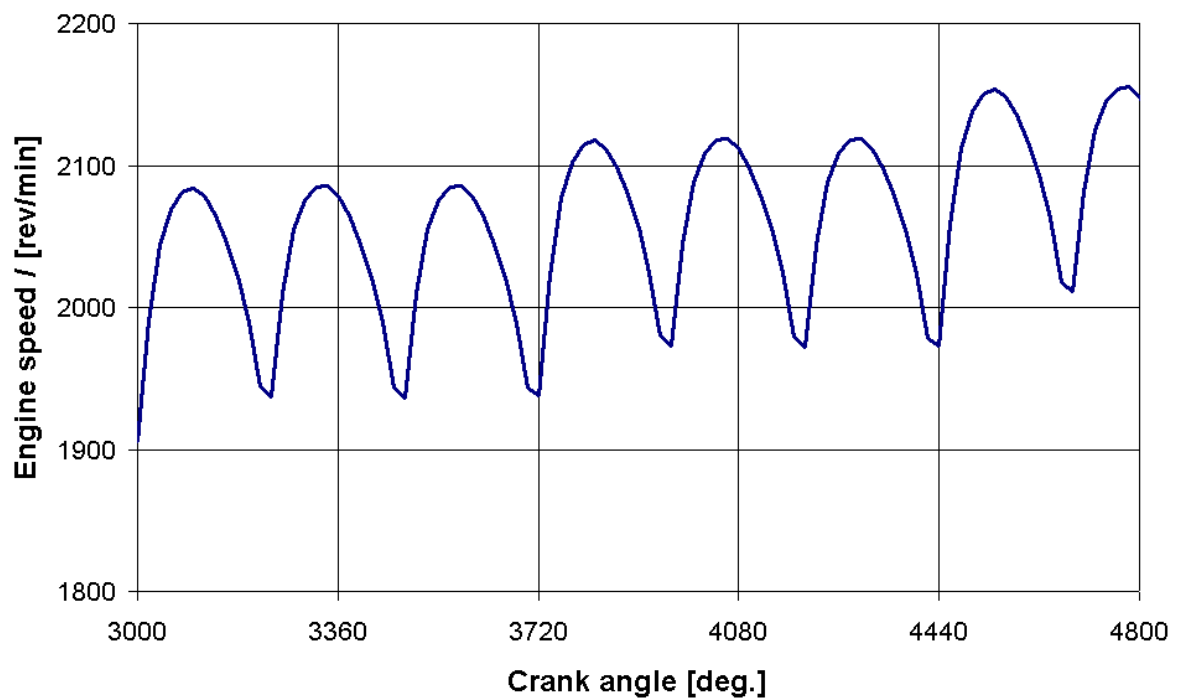


Figure 12.21. Detailed variation of engine speed with crank angle.



# 13

## Tutorial 12, 'Two-stroke Engine Model'

### Overview

This tutorial shows the user how to produce a model of a two-stroke engine.

This tutorial assumes that the user is now familiar with the LOTUS ENGINE SIMULATION interface, and thus many of actions required to complete this tutorial are only briefly described. It is recommended that the earlier tutorials should be completed before this tutorial is attempted.

This chapter contains the following sections:

- Building the Model, 150
- Creating Test Points, 154
- Running the Model, 156
- Exercise, 156

## Building the Model

We will begin by creating a new model. To ensure that all existing data is removed select **File / New** from the LOTUS ENGINE SIMULATION menubar, as shown in Figure 13.1.

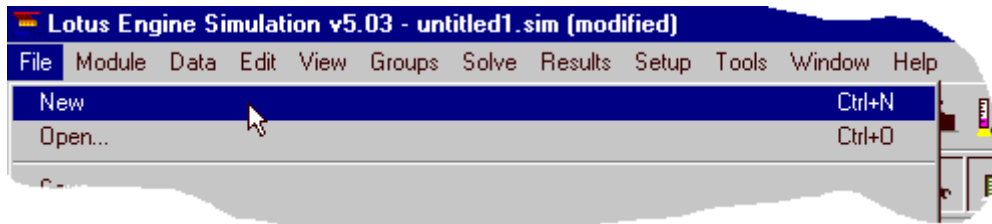


Figure 13.1. Creating a new model.

We shall construct a model of a 125cc, single cylinder, crankcase scavenged, two-stroke engine. Begin by setting the **Cycle Type** option. Select **Data / Cycle Type / 2-Stroke** from the LOTUS ENGINE SIMULATION menubar, as shown in Figure 13.2.

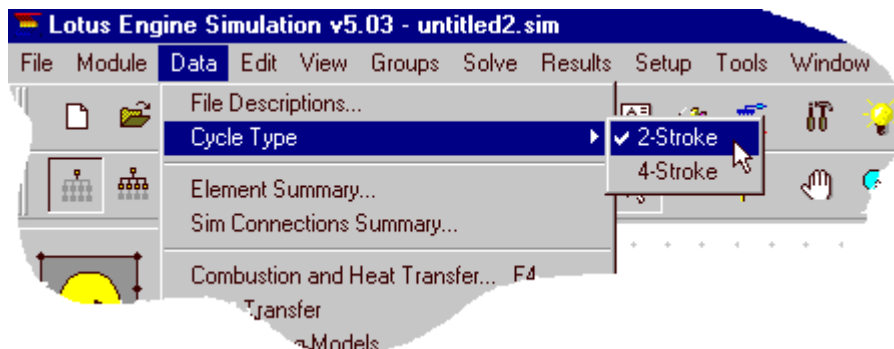


Figure 13.2. Setting the cycle type.

Now select a Cylinder from the toolkit and place it in the builder window. Change both the **Bore** and the **Stroke** of this cylinder **54.0mm** in the cylinder property sheet. Also, change the **Con-rod Length** to **110.0mm** and the **Compression Ratio** to **14.0**. Change the Open Cycle heat

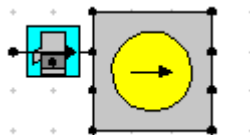


Figure 13.3. Adding piston ported inlet valve.

Select a **Piston Ported Valve** from the *Intake Components* toolkit and connect it to the inlet side of the cylinder, as shown in Figure 13.3. Change the properties of the piston ported inlet valve to those shown in Figure 13.4.

Now select a **Piston Ported Valve** from the *Exhaust Components* toolkit and connect it to the exhaust side of the cylinder. Change the properties of the piston ported exhaust valve to those shown in Figure 13.4.



Label	default piston ported in	Label	default piston ported e
Valve Option	Standard	Valve Option	Standard
Port Width (mm)	49.00	Port Width (mm)	36.00
Max. Port Height (mm)	16.00	Max. Port Height (mm)	28.00
Stroke (mm)	54.00	Stroke (mm)	54.00
Rod Length (mm)	110.00	Rod Length (mm)	110.00
Valve Open (deg)	113.00	Valve Open (deg)	82.00
Max Area CD Coeff	0.900	Max Area CD Coeff	0.900
Harness Connector	Off	Harness Connector	Off
Piston Ported Inlet Valve Properties		Piston Ported Exhaust Valve Properties	

Figure 13.4. Setting the properties of the piston ported inlet valve.

Now select a **Variable Volume Plenum** element from the *Intake Components* toolkit and connect it to the piston ported inlet valve, as shown in Figure 13.5. We will use the variable volume plenum element to represent the crankcase volume of the engine. The speed ratio of the variable plenum element can be set, relative to the speed of the crankshaft. Care should be taken when using this facility, as non-integer speed ratios may cause the phasing of the plenum, with respect to the crankshaft, to change between engine cycles, thus preventing cyclic convergence to be achievable. The phasing of the variable volume plenum element, with respect to cylinder number 1, can be set. A phase angle of 0° would correspond to the minimum volume of the plenum coinciding with T.D.C. of cylinder 1.

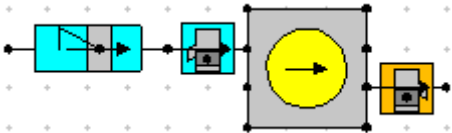


Figure 13.5. Adding a variable volume plenum to the intake system.

Ensure that the properties of the variable volume plenum element, so that they correspond to those given in Figure 13.6.

Label	default inlet vary. vol. p
Equiv. Bore (mm)	54.00
Equiv. Stroke (mm)	54.00
Equiv. Rod Length (mm)	110.00
Equiv. Compression Ratio	1.60
TDC Angle (deg)	180.00
Wall Temperature (°C)	100.0000
Plenum HTC (W/mm²K)	5
Speed Ratio	1.00
Harness Connector	Off

Figure 13.6. Setting the properties of the variable volume plenum.

Select a **Disc Valve** element from the *Intake Components* toolkit and connect it to the variable volume plenum, as shown in Figure 13.7. We shall accept the default parameters for the disc valve flow area and timing.

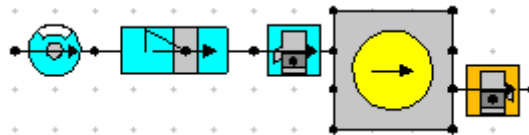


Figure 13.7. Adding a disc valve to the intake system.

Select a **Pipe** element from the *Pipes* toolkit and connect the downstream end to the variable volume plenum. Change the length of the pipe to **100 mm** and the diameter to **38.0 mm**. Add an **Inlet Boundary** to the upstream end of the pipe, as shown in Figure 13.8.

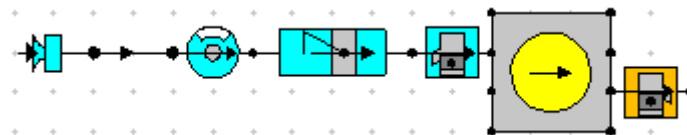
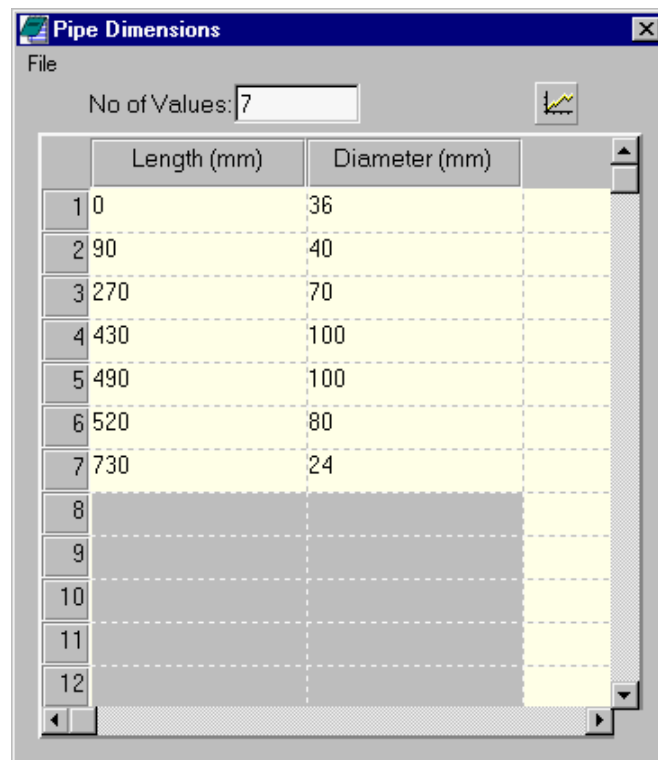


Figure 13.8. Complete inlet system.

Now add a pipe to the exhaust port. Click on the *All Dimensions* pad icon to open the pipe dimensions menu shown in Figure 13.9. Enter **7** in the *No of Values* box, then enter the lengths and diameters given in Figure 13.9.



	Length (mm)	Diameter (mm)
1	0	36
2	90	40
3	270	70
4	430	100
5	490	100
6	520	80
7	730	24
8		
9		
10		
11		
12		

Figure 13.9. Adding a disc valve to the intake system.

Close the *Pipe Dimensions* window and click on the **Pipe Graphical Display** icon, which will open the window shown in Figure 13.10.

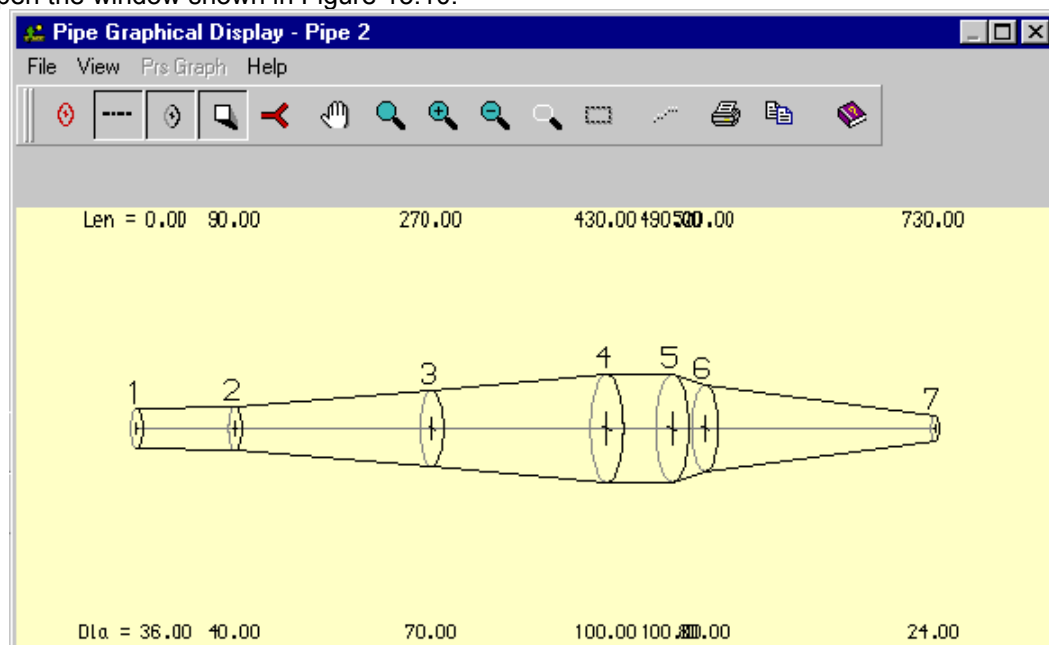


Figure 13.10. Viewing the pipe graphical display.

To complete the model, add an Exit Boundary to the end of the exhaust pipe. Your model should now look similar to the one shown in Figure 13.11.

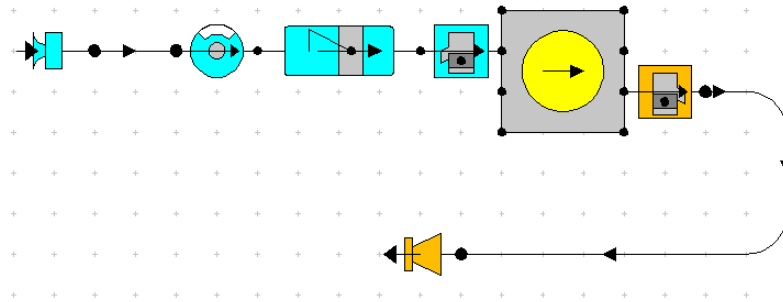


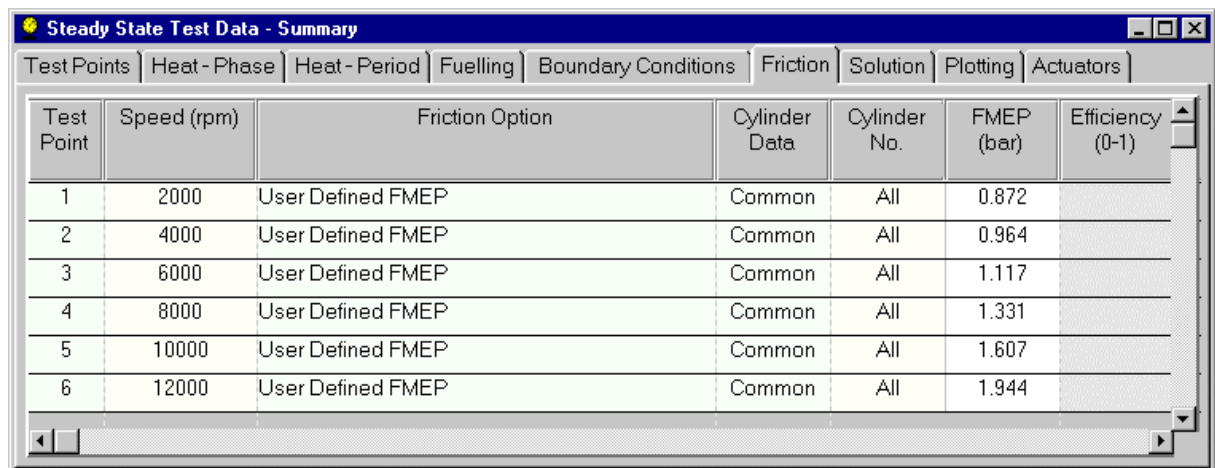
Figure 13.11. The completed model.

## Creating Test Points

Now we need to specify the test points at which the model will be run. We shall use the *Steady State Test Data Wizard* to create these automatically for us. Select **Data / Test Conditions / Steady State Create Wizard** from the LOTUS ENGINE SIMULATION menubar. This will open the *Steady State Test Data Wizard* window, shown in Figure 13.12. Create **6** new test points, **2000, 4000, 6000, 8000, 10000** and **12000** rev/min. Set the inlet and exhaust boundary conditions to **15 °C** and **1.01 bar**. Also set the Fuel/Air ratio to **1.1**. Click on **Apply** to create the new test points and close the window.

Figure 13.12. Opening the Steady State Test Point Wizard.

Open the *Steady State Test Data Summary* spreadsheet (Hint: this can be opened using the **F12** shortcut key). Change the *Heat Release Phase* to **12.0** and the *Heat Release Period* to **23.0** for all of the test points. Set the *Combustion Option* to in the **Fuelling** menu to **None**. Also set the friction data to the values given in Figure 13.13.



Test Point	Speed (rpm)	Friction Option	Cylinder Data	Cylinder No.	FMEP (bar)	Efficiency (0-1)
1	2000	User Defined FMEP	Common	All	0.872	
2	4000	User Defined FMEP	Common	All	0.964	
3	6000	User Defined FMEP	Common	All	1.117	
4	8000	User Defined FMEP	Common	All	1.331	
5	10000	User Defined FMEP	Common	All	1.607	
6	12000	User Defined FMEP	Common	All	1.944	

Figure 13.13. Steady state test point friction data.

Change the maximum number of calculation cycles for each test to **20**. Also change the *Plotting Options* so that the **Cylinder** data, **Penum** data and all **Pipe** data are all stored.

## Running the Model

Change the file descriptions and set the *Test No.* to **Tutorial12** then save the data file to **Tutorial\_12.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to set the .mrs and .prs file names.

Once the job has finished, the Batch Job Finished window will appear. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. Load the .MRS results and .PRS results into their respective graphical viewers. We can now view the predicted performance of the engine using the .MRS viewer, as shown in Figure 13.14, and the instantaneous pressures, mass flow etc. using the .PRS viewer.

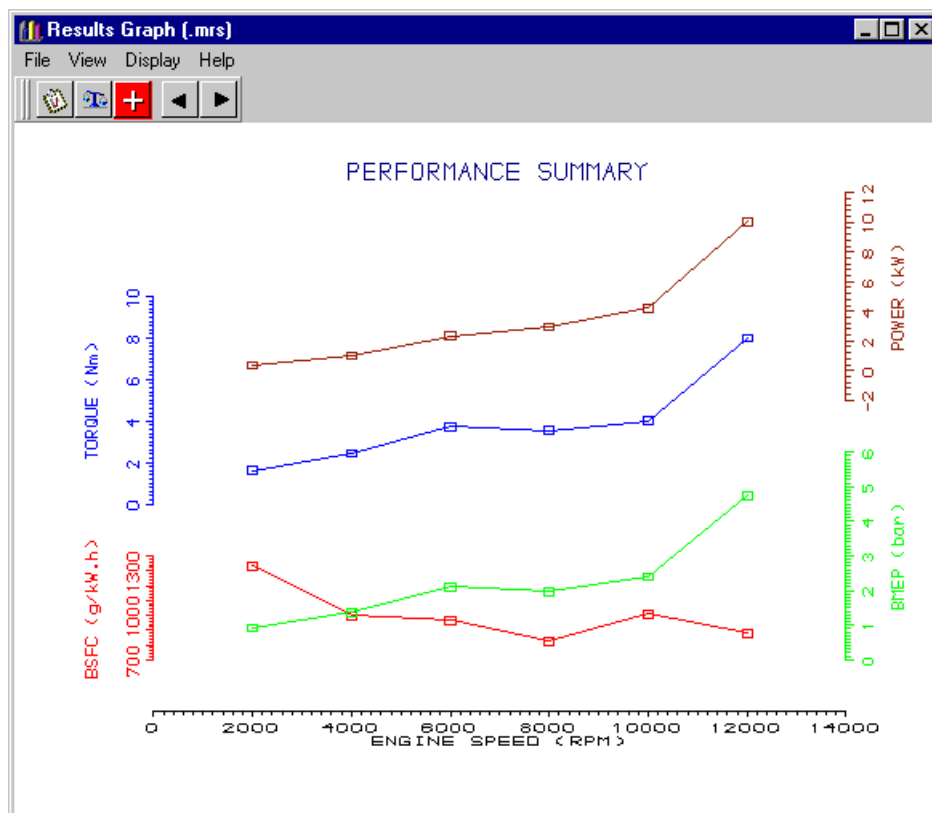


Figure 13.14. Predicted performance of the engine.

## Exercise

Delete the disc valve and replace it with a reed valve, as shown in Figure 13.15.

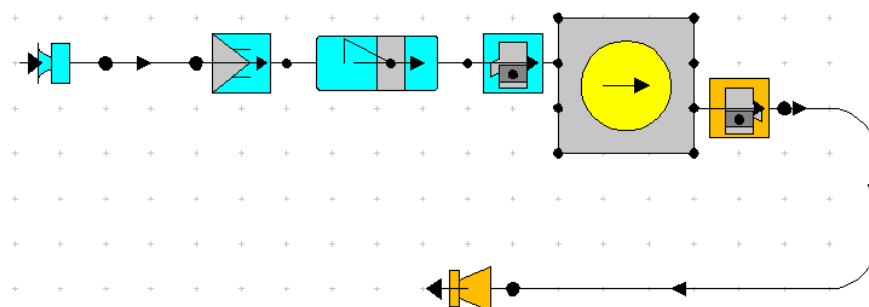


Figure 13.15. Adding a reed valve.

Change the properties of the reed valve element to those given in Figure 13.16.

Label	default reed valve
Valve Lift Option	Dynamic Lift
No of Petals	4
Mass of Petal (g)	0.200
Petal Stiffness (N/mm)	0.250
Area of Petal (mm <sup>2</sup> )	300.00
Passage Length (mm)	60.00
Max Lift CD Coeff	0.950
Max Lift (mm)	5.000
Harness Connector	Off

Figure 13.16 Reed valve property sheet.

Change the file descriptions and set the *Test No.* to **Tutorial12b** then save the data file to **Tutorial\_12b.sim**.

Submit the job in the normal way using the launch icon from the solver control dialogue box, having remembered to set the .mrs and .prs file names.

Once the job has finished, the Batch Job Finished window will appear. This prompts the user to load the simulation results into the various post-processing options within LOTUS ENGINE SIMULATION. Load the .MRS results (use the shuffle option to retain the results from **Tutorial12.sim**) and .PRS results into their respective graphical viewers.

Compare the results of the reed valve engine with those for the disc valve controlled version, as shown in Figure 13.17.

Use the .PRS file viewer to observe the variation of the reed valve flow area at the various engine speeds, as shown in Figure 13.18 (you may like to change the X-axis range of the PRS plots to **from 0 to 360 to -180 to 180**, which will move the TDC point to the centre of the X-axis).

Alter the parameters of the reed valve and observe the influence on the flow area of the valve and the engine performance.

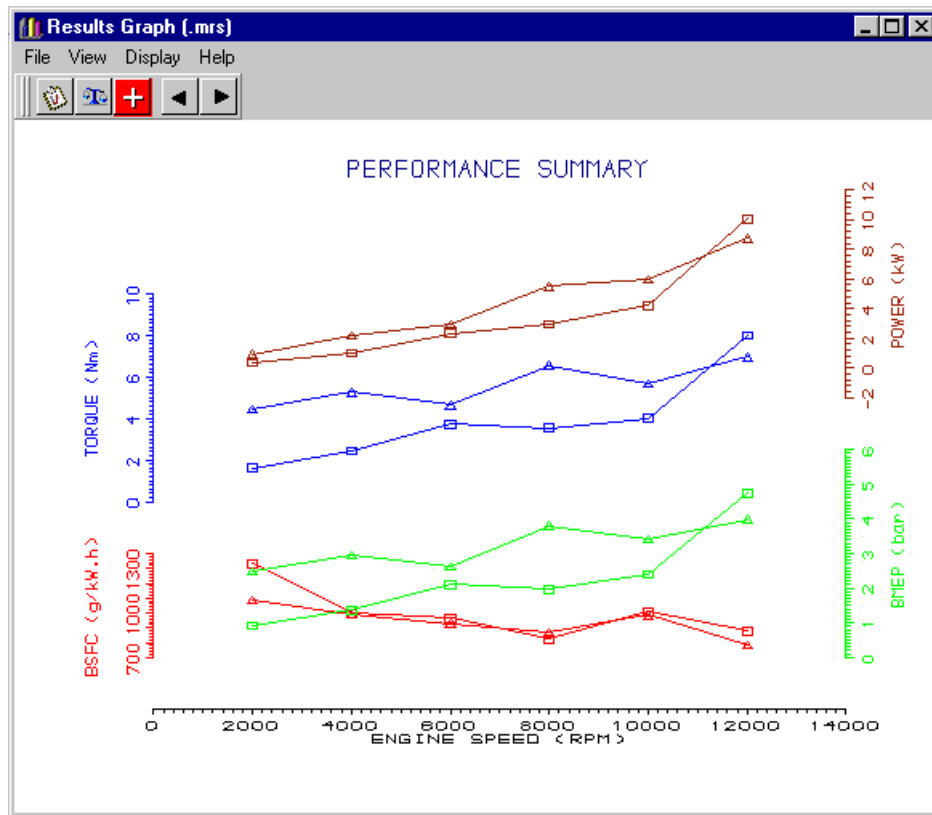


Figure 13.17. Comparison between the predicted performance of the engine with the disc valve and with the reed valve.

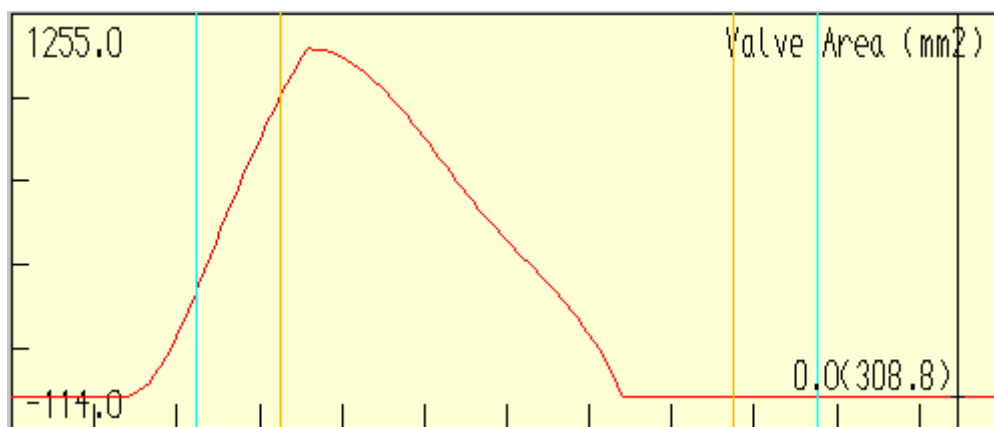


Figure 13.18. Flow area variation of the reed valve at 8000 rev/min.



## Overview

This tutorial shows the user how to produce an LES model which co-simulates with Simulink™.

This simplistic example will be based on the LES model created in Tutorial 6. The control functions, previously performed by the LES actuator elements, will be implemented in Simulink. This tutorial assumes that the user is familiar with Simulink and only provides a detailed description of the link to LES. It is also assumed that the user is now familiar with the LES interface and it is recommended that the earlier tutorials should be completed before this tutorial is attempted.

This chapter contains the following sections:

- Building the Simulink Model, 160
- Linking the LES model to Simulink, 165
- Registering the LES Solver, 168
- Running the Simulink Model, 169

## Building the Simulink Model

We will begin by creating a new Simulink™ model. From the Matlab™ window select **File / New** from the LOTUS ENGINE SIMULATION menubar, as shown in Figure 14.1. A new Simulink model window will appear.

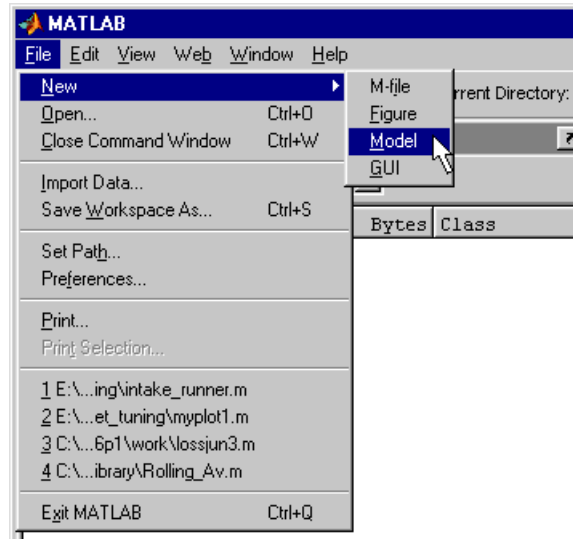


Figure 14.1. Creating a new Simulink model.

From the simulink library, select the LES Solver link mask (Fixed time-step), as shown in Figure 14.2. Drag this component into the simulink model window.

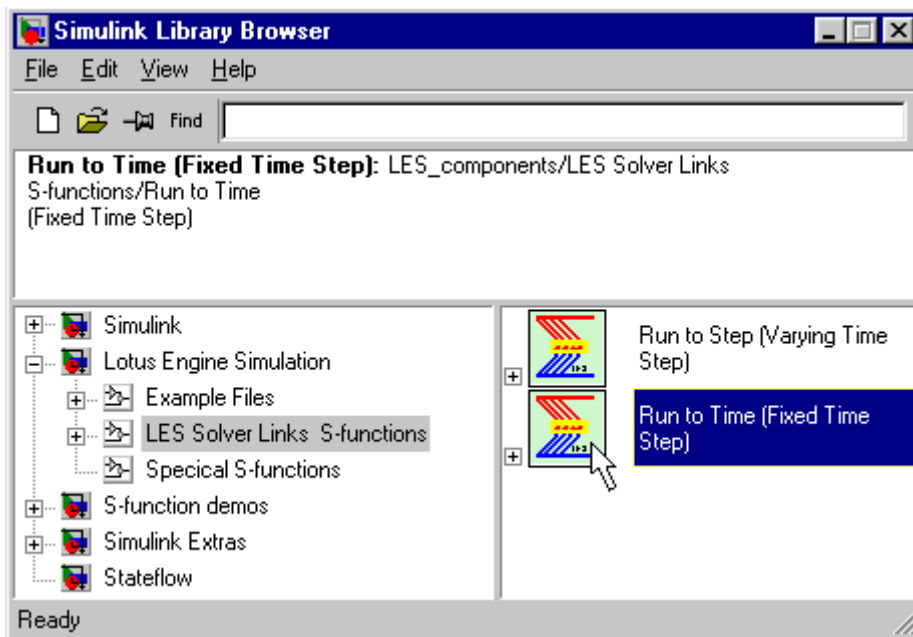


Figure 14.2. Select the LES Solver link mask (Fixed time-step).

The fundamental difference in the operation between the fixed time-step and the varying time-step elements is the number of times the LES solver loops through before returning values to Simulink. The LES solver does not operate with a constant calculation time-step size, the time step continuously being refined to balance run time against solution accuracy. Thus, if you choose to run Simulink in fixed time-step mode Simulink passes to the LES solver the

required time at which it requires LES to next return the sensor values. LES will run till it reaches this time, modifying the LES solution time step if necessary to match the target time. In this mode Simulink controls the time-step increment at which the Simulink model is updated.

If you choose to run in variable time-step mode LES will modify its time-step as per a standalone calculation at each calculation step it will return sensor values back to Simulink and set the next time-step value based on the current LES solver time-step. Thus in this mode the Simulink Model is updated every LES solver time-step and the LES solver defines each incremental time-step size based on its internal calculation rules. Typically the 'varying time-step' mode takes significantly longer to run because of the increased amount of com 'traffic' between the two applications.

The LES library items should automatically be added to the Simulink library during the LES installation. If this section does not appear in the library browser they can be added by adding the path in which they are located to the Matlab path file. This can be done from the Matlab command line using the 'addpath' command. You will need to add to the front of the path definition string the path to the Lotus Engineering software install folder on your local machine and the subfolder that contains the LES supplied Simulink components. This would typically be;

```
C:\Lesoft\matlab_components
```

Thus the Matlab path statement can be modified by typing the following at the Matlab commoand prompt

```
addpath ('C:\Lesoft\matlab_components')
```

To check that this has modified the Matlab path file simply type 'path' at the Matlab command prompt. This will invoke a list of the current directories specified in the file. The newly added directory should appear at the top of this list.

Double click on the LES Solver link mask. This will open the LES solver link mask subsystem window, as shown in Figure 14.3.

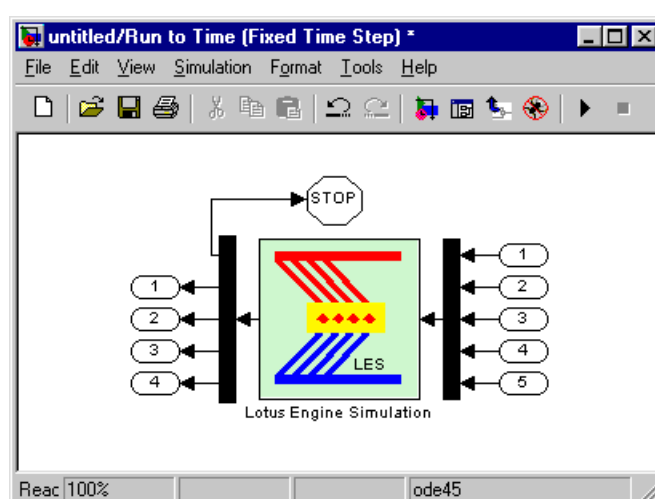


Figure 14.3. LES Solver Link Mask Subsystem Window.

Double clicking on the LES solver link mask will open the LES link block parameters menu, as shown in Figure 14.4. The first parameter entered into this menu defines the Lotus Engine Simulation model file to use, this should include the full pathname and would normally have a \*.sim extension. We shall call our \*.sim file 'Tutorial6\_simulink.sim'. Enter this in the first box of the block parameters menu. You need to add the your directory structure (pathnames) to the beginning of this filename.

The second field defines the output \*.mrs file name. The full pathname is not compulsory for this field. If omitted it will be created in the same folder as the data file. Enter 'Tutorial6\_simulink.mrs' in this box.

The third field defines the output \*.prs file name. Again, the full pathname is not compulsory for this field. If omitted it will be created in the same folder as the data file. As with a normal LES run if multiple speed points are calculated a \*.prs file will be created for each speed with the appropriate integer number appended to the supplied file name. Enter 'Tutorial6\_simulink.prs' in this box.

The forth and fifth fields together define which test point(s) will be run. The options include a steady-state single speed point, steady-state power curve, or transient. Obviously for a particular test to be selected it must exist in the LES model file. To run a single steady-state speed point enter the required LES test point number in field 4 and enter a zero in field 5. To run a complete steady-state speed sweep set the test point number in field 4 to zero and enter a zero in field 5. To run a transient analysis enter the required steady state start test point No. in field 4 and the required transient test case No. in field 5. We shall run a single speed, so enter 6 in the forth field and 0 in the fifth field.

**Block Parameters: Lotus Engine Simulation**

Lotus Engine Simulation (mask)

S-Function COM Link to Lotus Engine Simulation Solver.

The supplied S-function will perform the necessary COM initialization and send and receive data values between your Simulink model and Lotus Engine Simulation for a fixed time step analysis.

Provide the full path name of your engine simulation model. Provide output filenames, (if output path is omitted the same as provided for the model file is used).

Set the required steady state test point No. to run. For a transient test this will be the initial steady state test prior to the transient event. To run the complete steady state power curve set the steady state test point No. to zero.

Set the required transient test point No. to run. For a steady state test set this to zero.

**Parameters**

Supply Lotus Engine Simulation Model File Name  
C:\Lesoft\LES\_Examples\Tutorial6\_simulink.sim

Supply Lotus Engine Simulation \*.mrs Results File Name  
Tutorial6\_simulink.mrs

Supply Lotus Engine Simulation \*.prs File name  
Tutorial6\_simulink.prs

Set Steady State test point No. to run, (or Transient Start Test)  
6

Set Transient test point No. to run (0=steady state run)  
0

OK Cancel Help Apply

Figure 14.4. LES solver link block parameters menu.

Select a scope from the Simulink component library (Simulink/Sinks/Scope) and place this in the Simulink model window. Open the scope window by double clicking on the scope icon. Click on the scope parameters symbol and change the number of axes to 3, as shown in Figure 14.5.

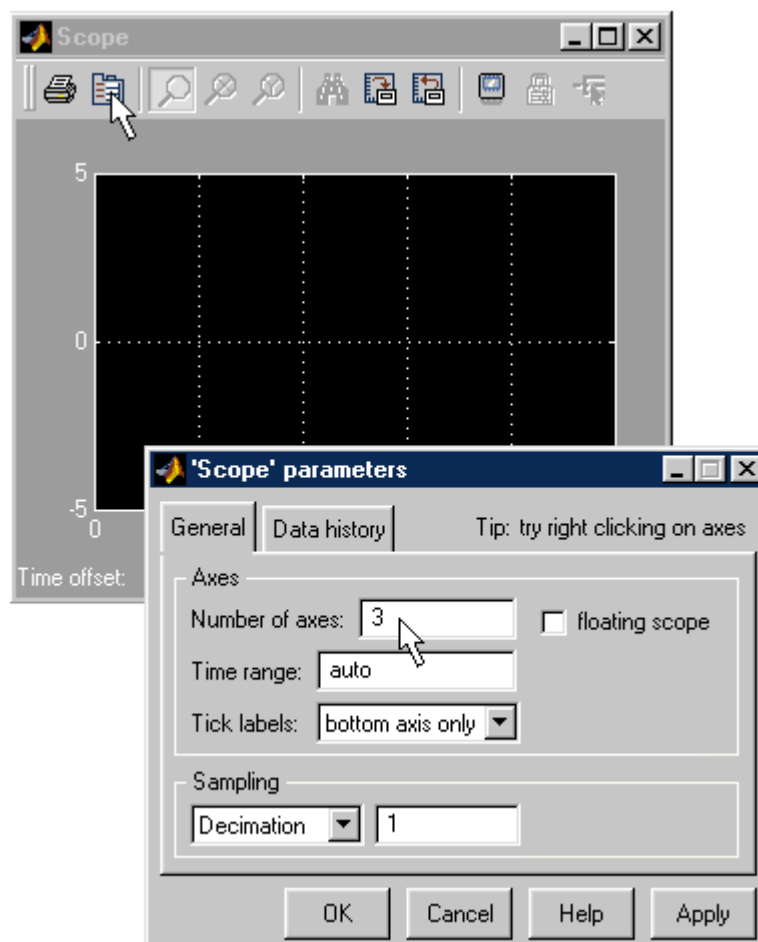


Figure 14.5. Change the number of scope axes.

Now select a look-up table from the Simulink component library (Simulink/Functions & Tables/Look-Up Table) and place this in the simulink model window. Double click on the look-up table element. This will open the data entry window for the look-up table element. We will enter the same MOP data that we entered into the 1D table for tutorial 6. Enter the input data (speed) as 1000, 2000, 3000, 4000, 5000, 6000, 6500 and 7000 and the output values (MOP) as 90, 90, 90, 95, 105, 120, 125, 125 (see Figure 14.6).

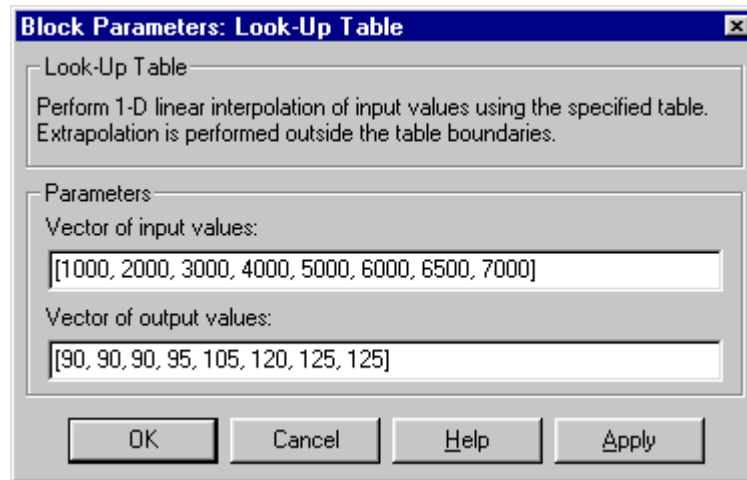


Figure 14.6. Change the number of scope axis.

We can connect the model elements together. Connect the first output of the LES Solver link mask to the first input of the scope element. Connect the second output of LES Solver link mask to the second input of the scope element and to the input of the look-up table element. Connect the output of the scope element to the first input of LES Solver link mask and to the third input of the scope element. The Simulink model should appear similar to that shown in Figure 14.7.

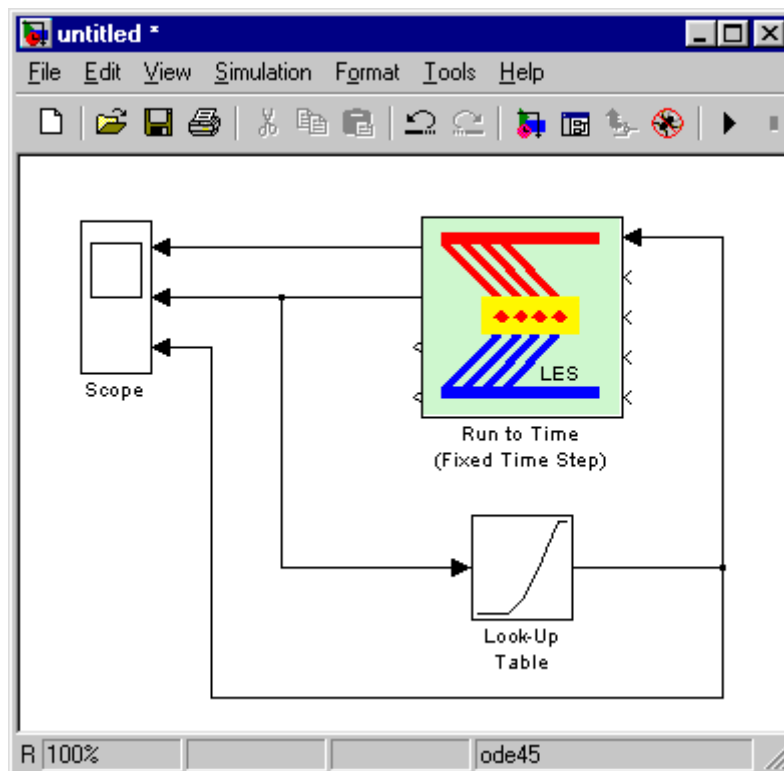


Figure 14.7. Simulink model window.

Finally we need to specify the Simulink simulation parameters. Select Simulation / Simulation parameters from the Simulink model window menubar. This will open the Simulink simulation parameters window, shown in Figure 14.8. Set the calculation start time to 0.0 seconds and the stop time to 1.0 seconds. The Simulation 'Stop Time' should be set sufficiently large that the run is stopped by the LES solver rather than by Simulink. If the Simulink 'Stop-time' stops the run before the LES has finished a warning is given in the Matlab command window. Set

the Simulink solver option to 'Fixed-step' and 'discrete (no continuous states)'. Set the fixed step size to '0.0001' seconds and ensure that the mode is set to 'Auto'.

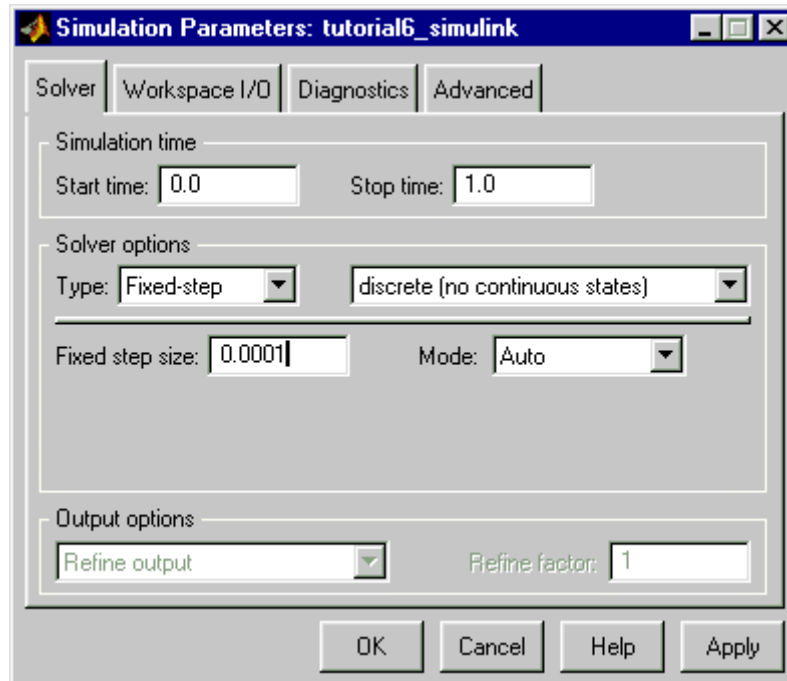


Figure 14.8. Setting the simulation parameters.

Now save the Simulink model file as 'Tutorial6\_simulink.mdl'.

## Linking the LES model to Simulink

We will use the model created in Tutorial 6 as the basis for this LES model. Open this file (if you have not got a copy of Tutorial\_6.sim then follow the directions given in Chapter 7 and construct the model).

Now select the Simulink interface element from the 'External Interfaces' tab on the toolkit and place it in the model window, as shown in Figure 14.9. The Simulink interface element can be moved and connected to the model just like any other builder element. The Engine Model treats the Simulink interface element as a separate processor that sits between sensors and actuators. Thus, the only allowed connections to the Simulink interface element are wires from sensors on the signal in-side the and wires to actuators on the signal out-side.

Due to the orientation of the existing sensor and actuator elements within the LES model it is worth flipping the flow direction of the Simulink interface element, to do this select the Simulink interface element and press the right mouse button, then select 'Flip Flow Direction' from the pop-up menu. Now select the actuator element and move down to data level 2 and delete the 1-D look-up table (if you are unsure how to do this refer to Tutorial 6). Reconnect the IN and OUT signal boundaries together and return to level 1.

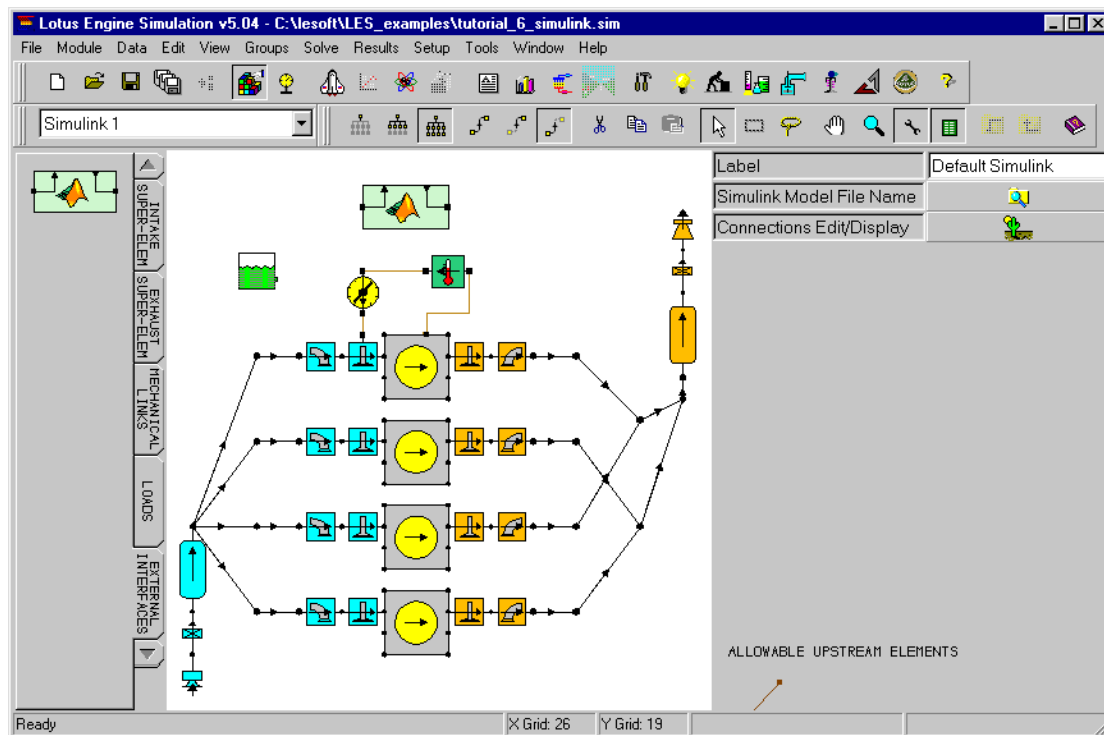


Figure 14.9. Adding a Simulink interface element to the model.

Now connect the output signal of the Simulink interface element to the input of the actuator and connect the output of the sensor to the input signal boundary of the Simulink interface element, as shown in Figure 14.10.

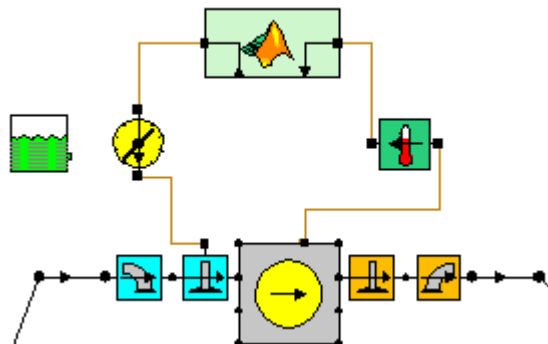


Figure 14.10. Connecting the Simulink interface element.

Add an additional sensor to the model and connect this from the cylinder 1 and the input signal boundary of the Simulink interface element. Modify the properties of the sensor to sense cylinder pressure.

Select the Simulink interface element and from the Simulink interface element property sheet, shown in Figure 14.11, select the file browser. From the file browser window select the Simulink model file 'Tutorial6\_simulink.mdl', as shown in Figure 14.12, then select open.

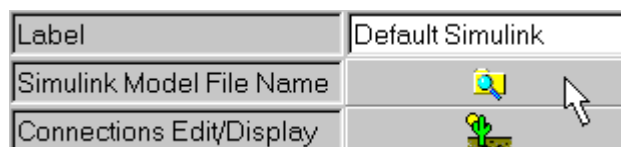


Figure 14.11. Simulink interface element property sheet.



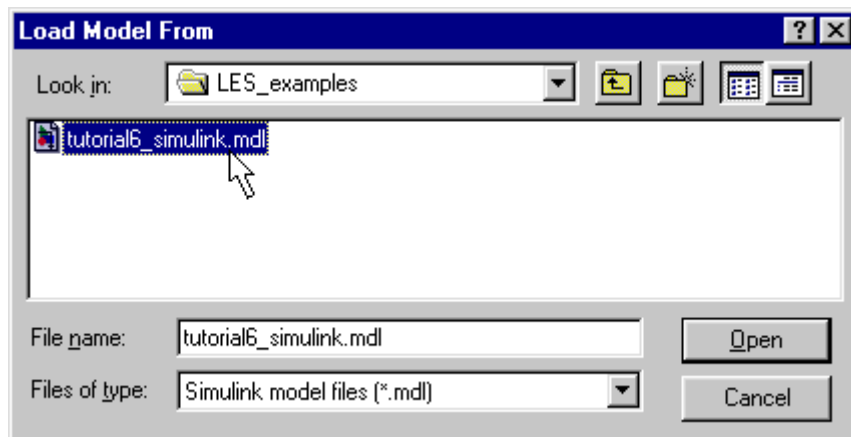


Figure 14.12. Select the Simulink model file.

Press the 'Connections Edit/Display' from the Simulink interface element property sheet. The 'External Connection (SIMULINK)' window will open, as shown in Figure 14.13. Now connect the signal from the cylinder pressure sensor to the Out1-Port on the 'demux' (connector No. 2 on the 'demux' – connector No. 1 is used by the 'stop' signal). To make the required connections select the arrowhead of the required sensor and drag it to the required demux port. To remove a sensor to demux connection selected the connected arrowhead and move it back to the unconnected position. In this way all of the sensors can be connected to the required demux ports. Connect the crank speed signal to the Out2-Port.

The actuator connections are made by selecting the arrowhead of the required mux port and dragging it to the required actuator connection. Connect the In1-Port 'mux' signal to the valve MOP actuator input signal. Your connections

It is not necessary to fill all the ports or fill in any particular order provided that you connect to the ones linked in the Simulink model.

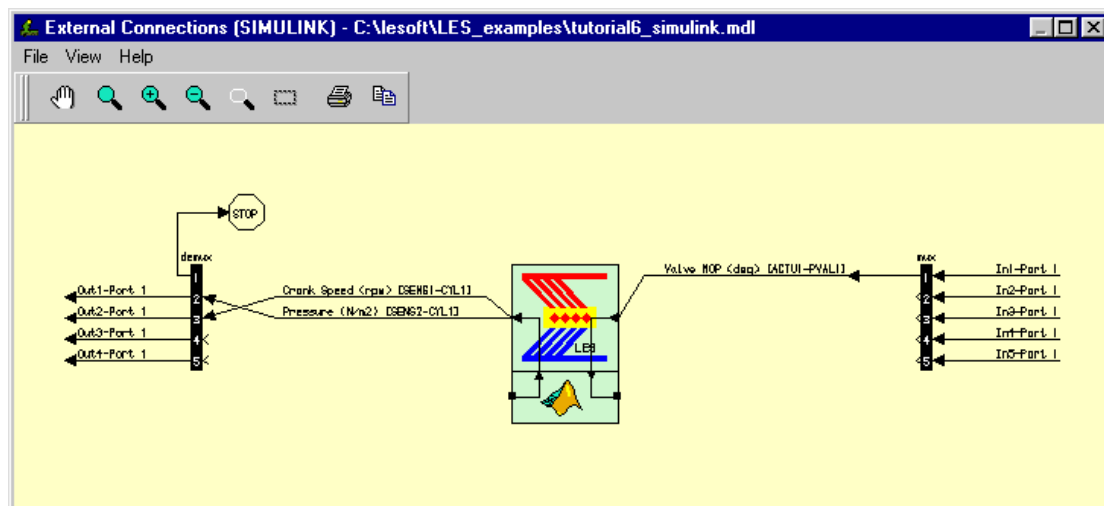


Figure 14.13. Define the sensor and actuator connections to the Simulink model file.

Once complete exit the connections display and save your model as 'Tutorial6\_simulink.sim' (the file name and location must correspond with the text entered in the first field of the LES solver link block parameters menu of the Simulink model file). You must save the model since Simulink runs the saved file and not that in the LES interface memory.

## Registering the LES Solver

The LES solver used for the co-simulation is a modification of the standard solver. An additional C++ wrapper has been added to provide the necessary interfaces and procedure entry points for the com standard. This alternative solver (filename `lesolveCpp.exe`) needs to be added to the system registry in-order for the executable to be identified by the com events. If LES C++ version of the solver has not previously been registered on your machine it will be necessary to register it now. Instructions of how this is done are given below. If your solver is already registered you can omit the remainder of this section.

To register the solver open the Solver Control display in LES and select the 'Solver Settings' tab. The executable that needs to be identified in the registry is `lesolveCpp.exe`. This will be in your Lesoft Install folder. Standard users will have the toggle next to 'Default Executable File' checked, this then allows the interface to look for the solver in the same folder that it was started from. This also allows the interface to look for the Com solver in the same way when you register it. Thus if the 'Default Executable File' toggle is checked simply select the 'Register Solver' button to carry out the Registry update. If you are using an alternative default solver location and have the 'User Defined Executable File' option set. Then you must first point the user executable at the required `lesolveCpp.exe` file.

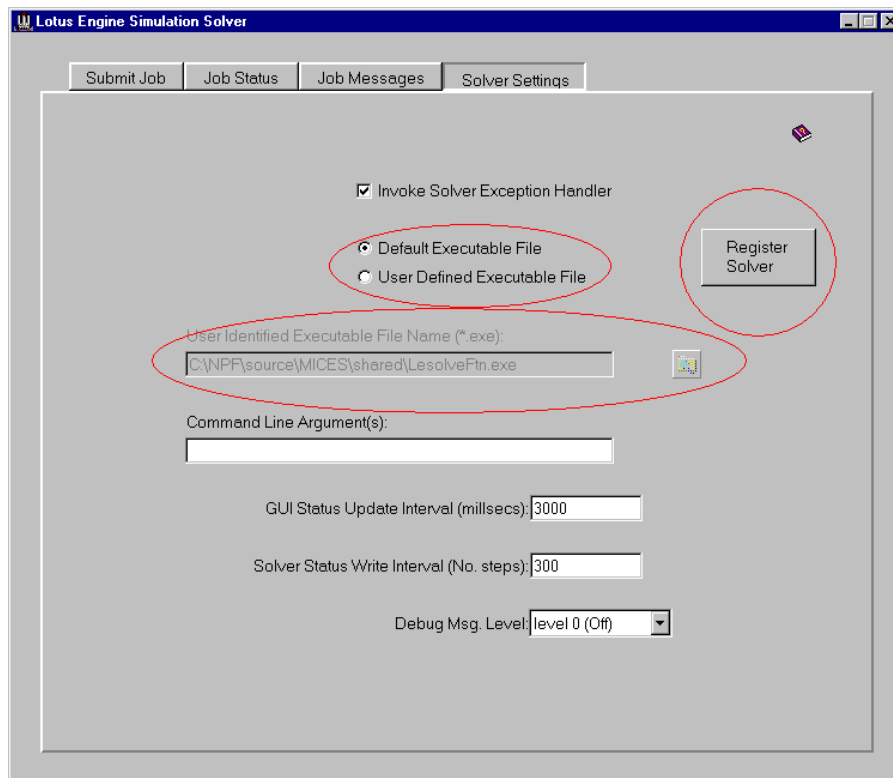


Figure 14.14. Registering the Solver.

When you select the 'Register Solver' button you will be asked to check that the file name and path is correct before carrying out the Registry update. You will be informed of a successful completion by the dialogue box shown in Figure 14.15. If you do not receive the 'success' message check with your local IT support for specific site variations and to check you have the necessary privileges. You will only need to register the solver once on a particular machine. Subsequent solver updates will need to be re-registered to be effected. This registration is only necessary if you want to use the Com interface link. The standard solver (`LesolveFtn.exe`) does not require to be entered into the registry as it does not support com interfaces.

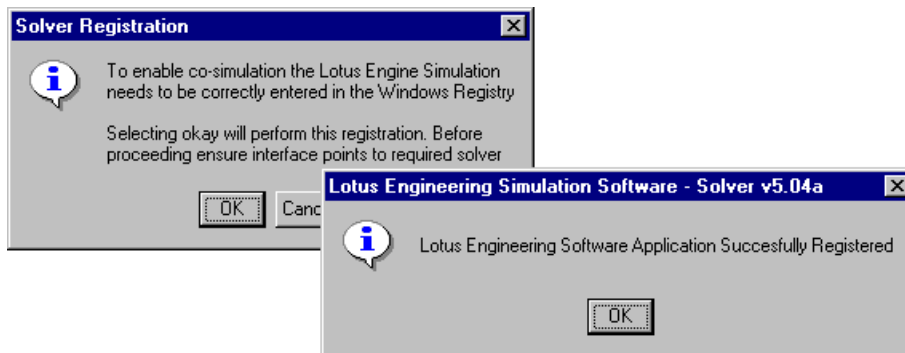


Figure 14.15. Successful Registration of the Solver.

The LES com interface is licensed separately from the standard solver and users wishing to use this interface should check for the relevant licensed feature, (solver-external). To check view your licence file (normally lotuspass.lic) in any text editor such as notepad, ensure you don't change any of the text. The external com interface requires the FEATURE 'solver-external' to be present. If you are not licensed on this feature refer to your local support.

The com enabled solver requires a significantly higher machine memory than the standard solver. The recommended amount is 256 mbytes of RAM.

## Running the Simulink Model

Before running the Simulink model, ensure that the current directory setting of Matlab is the in which the LES model and the Simulink model are stored in, as shown in Figure 14.16.

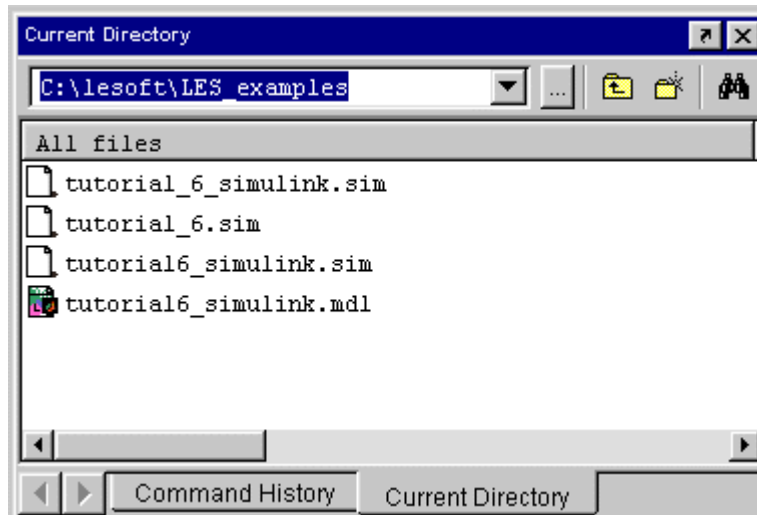


Figure 14.16. Setting the current Matlab directory.

To run the model, return to the Simulink model window and select 'Start' from the Simulation menu on the Simulink model toolbar or click on the 'Start simulation' icon. The simulation should then proceed displaying the % complete along the bottom. Note that because of the Stop time setting being greater than the required LES run time (to ensure LES controls the end point), the % bar will not normally reach 100% before the Simulink run ends.

If your simulink model fails to start then the following is suggested as a troubleshooting procedure:

1. Does the LES filename and location correspond to with the text entered in the first field of the LES solver link block parameters menu of the Simulink model file.
2. Is the solver registered.
3. Are you licensed to run LES and the LES external interface.
4. Can Simulink find the required S-function m-files, (ensure that the folder containing them is listed in the Matlab path file. Type 'Path' at the matlab command prompt to view the current directories present. If the directory is not listed add it by typing 'Addpath ('directory')' at the Matlab command prompt).
5. Does the LES model run standalone, (i.e. without the Simulink co-simulation).
6. Try starting Matlab from (or 'changing directory to' via the Matlab command window) the same folder as your target LES \*.sim file.

If the co-simulation job crashes the LES solver can sometimes be left as a live process. As only one Com version of the LES solver can run at a time subsequent attempts to run the LES Com solver will fail until the old process has been deleted. To check for this and to delete the process depends on the version of Windows being used. The example given in Figure 14.17 for Windows NT, other versions of Windows follow similar route but with slight differences in display.

To check for a hung LES solver process, open the task manager using Ctrl+Alt+Del and selecting the 'Task Manager' option. Select the 'Processes' tab and look for 'LESOLV~1.exe' (or 'lesolveCpp.exe'). If located select and using the right mouse select from the options menu 'End Process'.

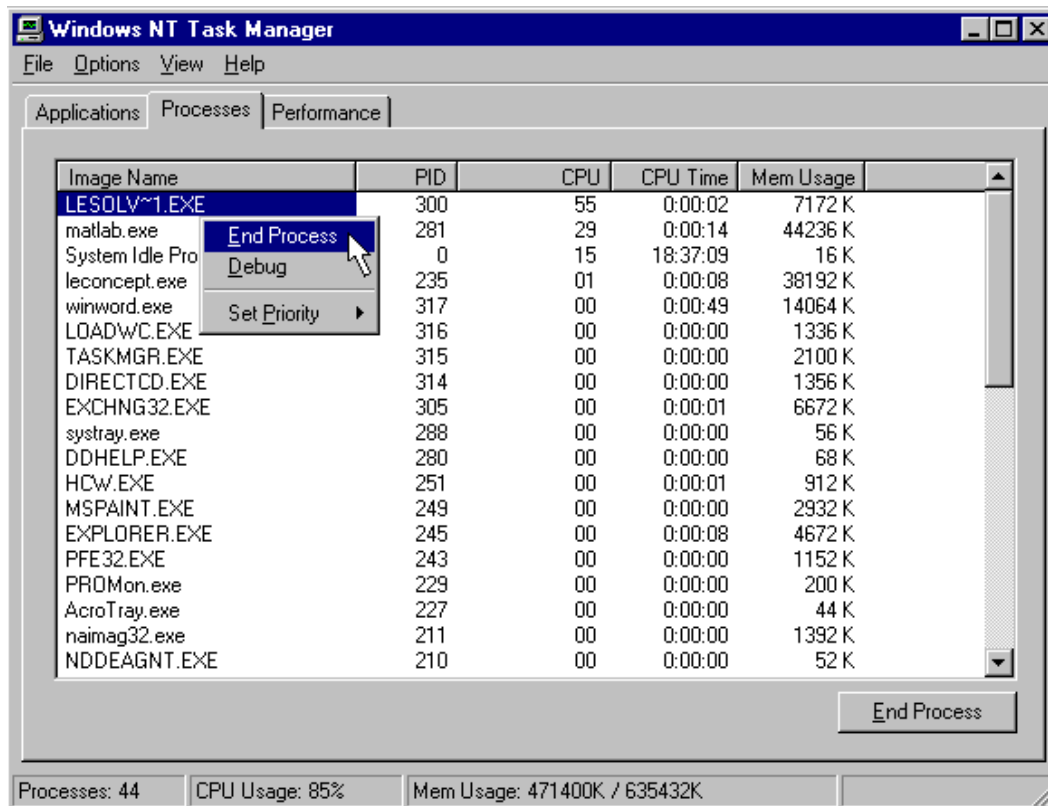


Figure 14.17. failed run – using Task Manager to end the process.

Once the simulation is under way the signals passed to the Simulink scope element can be viewed by double clicking on the cope element to open the scope window, which is shown in Figure 14.18. It may be necessary to autoscale the scope axes.

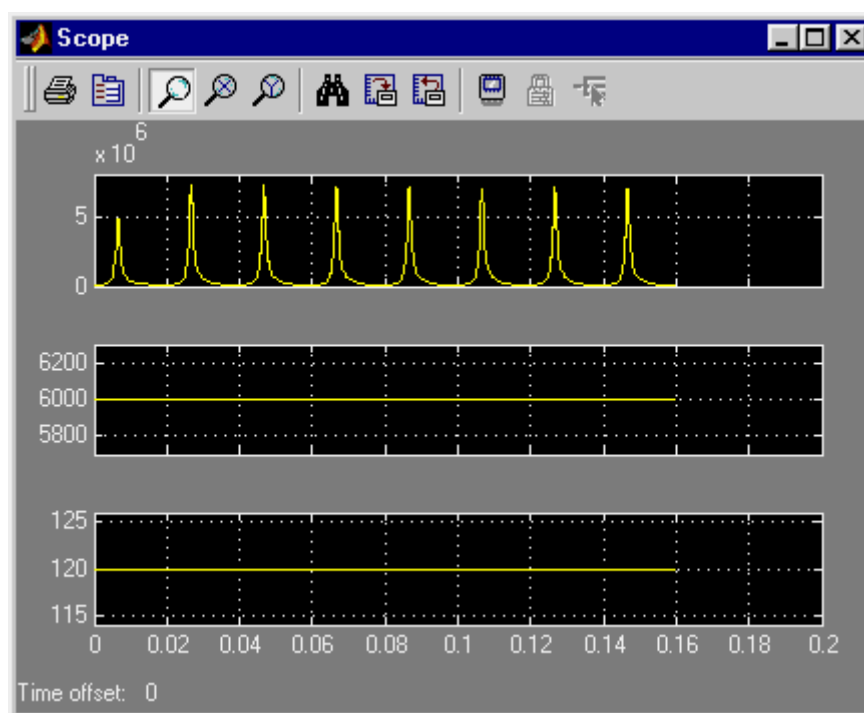


Figure 14.18. Example of monitoring LES run using the Simulink scope.

The monitoring of a LES – Simulink co-Simulation can be carried out using the same tools in LES as for a standalone LES simulation. The 'Job Status' panel on the LES Solver Control display can be used to the LES side of the co-simulation using either the conventional bar chart display or any available trs plots. The 'job Messages' section can also be used in exactly the same way as for a standalone run.

Because the analysis is not started from within LES the 'prompt on completion of job' option will not function.

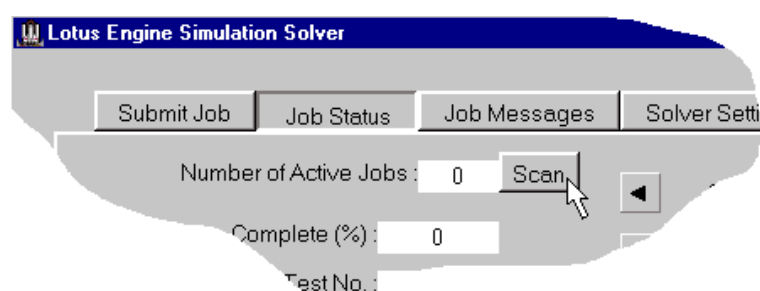


Figure 14.19. Scanning for active LES processes.

The LES job status monitoring uses the simulation log file to track progress. For a standalone LES run the monitoring of this log is initiated as part of the job submission. For a co-simulation run, because it is initiated by Simulink, LES does not know to start monitoring the log file. Thus to monitor a co-simulation run you need to point the interface at the log file. To do this once the job has been started from Simulink, return to the job status panel in LES and select the 'Scan' button, as shown in Figure 14.19. The file browser will open in your Windows 'Temp' folder where the log file is located. The co-simulation log file is always given the index 99 to indicate the difference between it and the standalone simulation log files that increment from '1'. Select the log file '\_Engin\_Batch99.log' from the browser. The status display will now show the status of the current co-simulation job.

Once the job has finished the \*.mrs files and \*.prs files can be opened in exactly the same way as for a standalone LES simulation.





## GROUP HEADQUARTERS

**Group Lotus Plc**  
Potash Lane, Hethel  
Norwich  
Norfolk, NR14 8EZ  
United Kingdom  
Phone: 01953 608000  
Fax: 01953 608000  
Email Cars: carsales@lotuscars.co.uk  
Email Eng: engsales@lotuscars.co.uk  
Email Group: group@lotuscars.co.uk

## INTERNATIONAL OFFICES

**Lotus Engineering Inc.**  
PO Box 7209  
1254 N. Main St.  
Ann Arbor MI 48104  
USA  
Phone: 734-995-2544  
Fax: 734-995-9301

**Lotus Engineering Malaysia Sdn. Bhd.**  
Lot G-5 Enterprise 3  
Technology Park Malaysia  
Lebuhraya Puchong-Sungia Besi  
Bukit Jalil  
57000 Kuala Lumpur, Malaysia  
Phone: 00 603 8994 3621  
Fax: 00 603 8994 1172

