Development of Custom Workflows for Simulation and Analysis of Functional Mock-up Units

Sureshkumar Chandrasekar[1]
[1] Modelon Inc.
Hartford, CT
United States
chandrasekar.sureshkumar@modelon.com

Jesse Gohl[2]
[2] Modelon Inc.
Ann Arbor, MI
United States
jesse.gohl@modelon.com

Abstract

Development of customized workflows and interfaces to deploy Modelica Functional Mockup Units (FMUs) with the various FMU tools has been gaining traction in the industry – both with tool vendors as well as end users.

The FMI Add-in for Excel (FMIE) is a commercial product from Modelon AB that enables the deployment of FMUs in Microsoft Excel. FMIE enables the user to programmatically control the add-in through Visual Basic code in Excel. This allows the implementation of custom workflows and interfaces for the user to interact and automate the tasks involved in loading, simulation and analysis of FMUs.

In this paper we present a workflow in FMIE for an automobile thermal management model FMU. The workflow utilizes Visual Basic scripts for automation and user-forms for user interaction.

Keywords: FMU;Automated workflow;FMI Add-In for Excel; Visualization; Design of Experiments, Batch Simulations, Monte-Carlo Analysis

1 Introduction

The Functional Mock-up Interface (FMI) defines an interface to be implemented by an executable called the Functional Mock-up Unit (FMU) to describe, evaluate and simulate models of dynamic systems defined by differential, algebraic and discrete equations in different simulation environments [1][2]. The FMI standard has greatly facilitated the transfer, exchange of dynamic models between component suppliers and Original Equipment Manufacturers (OEMs) and enabled export and import of models between multiple simulation platforms that have FMI export/import functionality.

The FMI standard as a result, is gaining traction in the industry as the technology of choice in the exchange of dynamic models between suppliers and OEMs. As a result, there have been a large number of FMU deployment tools of which the FMI Add-In for Excel (FMIE) [3] developed by Modelon AB is a popular choice among customers.

The FMIE integrates FMI-based parallel simulations in Excel. The add-in offers the following features:

- Simulation of compiled dynamic models, FMUs generated by any FMI-compliant tool such as Dymola, OPTIMICA Studio or Simulation X.
- The add-in takes advantage of powerful features in Excel to set up and perform batch simulations for parameter sweeps and simulations driven by data series.
- Perform dynamic simulations or solve initialization problems, in parallel.
- The FMI add-in supports FMU for Model Exchange 1.0 and FMI for Co-Simulation 1.0.

Due to its application in multiple engineering domains and across several organizations, FMIE provides a general interactive environment through the Microsoft Ribbon interface. From version 1.2 onwards, FMIE enables the user to programmatically control the add-in through Visual Basic code in Excel. This has opened up interesting possibilities to create custom workflows to automate several tasks in the loading, simulation and analysis of FMUs in Excel with FMIE.

The workflow includes both scripting automation and a customized graphical user interface (GUI). The automation includes filtering of variables to set-up experiments, loading values for these filtered variables for simulation and scripts for plotting the output response to these experiments. Additionally, macros to do the same for Design of Experiments (DoE) and Monte-Carlo analysis where many hundreds of simulation cases are to be generated in a single experi-

DOI
10.3384/ECP140961265

ment possibly with statistical distributions for multiple parameters/variables. The workflow presented in this paper allows all of these steps to be performed in just a few mouse clicks interactively and improves efficiency and ease of use.

Further, for visualization of results, a customized GUI is presented as a Visual Basic for Applications user form within Excel. It presents a streamlined interface for plotting experiment trajectories to the end user. This allows the user to quickly find available trajectories and generate default charts for visualizing and comparing the results.

2 The FMIE Interface

The FMIE provides a general graphical interface through the Microsoft Ribbon interface and appears as one of the tabs in the Ribbon interface of Excel and labeled as ’FMI’ as shown in Figure 1. The FMI tab contains four different groups of buttons.

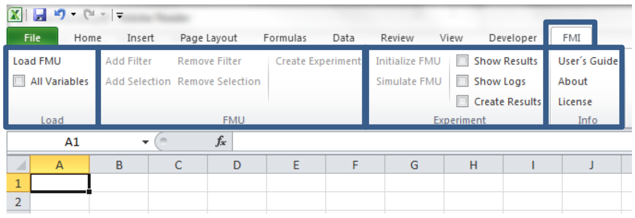


Figure 1: The FMIE Native Interface

The FMI tab in the ribbon is divided into four groups: **Load**, **FMU**, **Experiment** and **Info** as highlighted in Figure 1 and respectively have buttons to load, create and set-up, simulate and analyze results of FMU’s. In addition to these buttons, the functionalities of these buttons are mirrored with API function calls that are listed in Table 1. These functions are written in Visual Basic and the code for these is accessed from opening Visual Basic from the Developers tab.

FMIGetVersion	FMIUpdateSheet
FMILicenseCheckedOut	FMIGetCreateResults
FMISetCreateResults	FMIGetShowResults
FMISetShowResults	FMIGetShowLogs
FMIetShowLogs	FMICreateExperiment
FMILoadFMU	FMISetFilterButton
FMISetSelectionButton	FMISimulateButton
FMIInitializeButton	FMISimulate0
FMIInitialize0	FMISimulate1
FMIInitialize1	

Table 1: List of API functions in FMIE.

2.1 Structure of loaded FMU and experiment sheets

The FMIE follows a pre-defined structure to how the information about input, output variables and parameters from the FMU are displayed in Excel. The loaded FMU displays meta-data containing general information of the FMU model e.g. model name, FMU kind, number of state variables etc. The list of input, output variables and parameters are displayed under the Variables section in the same worksheet. The variables are identified by the name, variability, start value, unit, description etc. and there is a column to indicate if the variable in the corresponding row is to be included in an experiment. This Boolean is set to “TRUE” if the variable is to be selected for analysis in an experiment. Each of these columns has a search filter associated with which the variable list can be parsed to select the appropriate variable set for analysis. This process gets tedious and inefficient when the number of variables in an FMU or the number of variables to be selected in an analysis is huge. As a result, any workflow to be developed should provide an easy and efficient way to filter out these variables for analysis.

Once the analysis variables are filtered and an experiment sheet created, the input variables and parameters are separated automatically by FMIE from the output variables in the experiment sheet. The nominal values of the inputs and parameters are also displayed and 3 default simulation cases already set-up. The user is to then create as many simulation cases as needed within the experiment by creating new columns adjacent to the 3 already created. The next step before simulating the FMU is to load the parameter, input values for the experiment. This is also to be done manually in the native interface and can be automated to improve efficiency. The next section describes a template for entering the variable list and parameter values for experiments that would enable the workflow created to automatically load these into the experiment sheet.

3 Workflow

A typical workflow involved in the set-up, simulation and analysis of FMUs is illustrated in Figure 2.
At a higher level, the workflow follows a sequential procedure to load the FMU, filtering the relevant input, parameter and output variables for the study, create experiments and loading the input data for simulation, plotting and analysis of results.

To standardize the interface for user input and to facilitate data exchange between model developers and end-users, we present a simple front-end template for user input of information or data pertaining to the analysis to be performed. We present two templates – one for batch simulations involving parameter sweeps and the other for DoE and Monte Carlo studies. The proposed workflow is demonstrated in this paper with the example of a Co-Simulation FMU of a automotive thermal management system proposed in [4].

3.1 Batch Simulations

Batch simulations include parameter sweeps and experiments where one or more variables are changed individually or simultaneously and studying the effects of these on certain output variables. As a result, the batch simulation sheet needs to include sections for including the names of the input, parameter and output variables along with a description of each variable. Columns for entering user data are then classified according to the names of the experiments as shown in Figure 3.

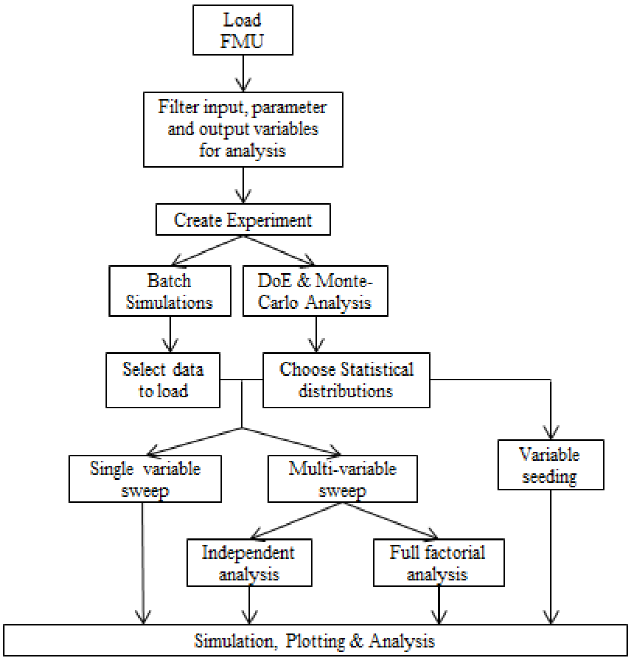


Figure 2: A typical workflow in the set-up, simulation and analysis of an FMU.

3.2 Design of Experiments and Monte-Carlo Analysis

Similar to the batch simulation template, a template for Monte-Carlo analysis is also set-up as shown in Figure 4. This template includes fields to enter the type of statistical distribution and arguments for each statistical distribution. Figure 4 shows an example where the efficiency multiplier of the heat exchanger stack is varied according to a Gaussian distribution with mean 1 and standard deviation 0.1. The current implementation of the workflow supports Gaussian, uniform and triangular distributions and can be easily extended to other Excel-supported statistical distribution functions.

Both templates provide a standardized structure for user-input of data for the workflow and enable convenient exchange of information between model developers and end users of the FMU.

Additional fields like the base class have been included in the template for model developers to provide a short description or a snapshot of the base class in which the variable is used, so as to give the end-users some insight into the variable.

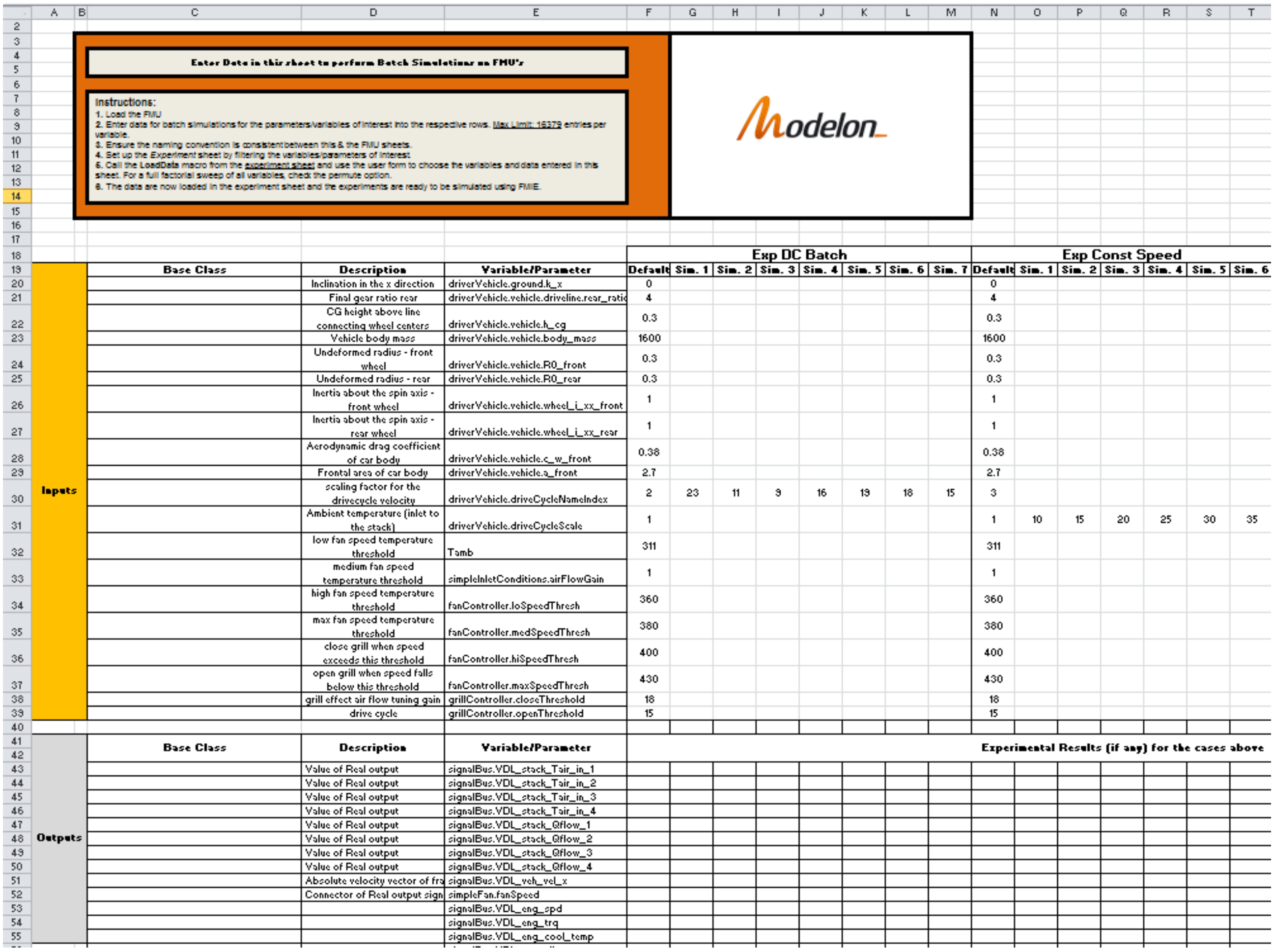
3.3 Load FMU and Filter Variables

With the batch simulation or DoE template filled up with a set of variables and their corresponding input data like that shown in Figure 3-4, the next step in the workflow is to filter these variables and create an experiment sheet. A sample (partial) Visual Basic code shown in Table 2 is an example that loads an fmu and prompts the user to choose variables for filtering and creates an experiment with 7 cases with these variables.

```
Public Sub FMIE_Filter_vars()
Dim fmuFile As String
On Error GoTo ErrFileNotFound
' Path to FMU
    fmuFile  =  Application.ActiveWorkbook.Path  &
"\DriveCycleVTM.fmu"
 ' Load FMU into Excel
    FMILoadFMU fmuFile, True, vbNullString
' Select & Filter variables to create experiment
    Dim data As Range
    Set data = Application.InputBox(Prompt:=
    "Please select range", Title:="Range Select",
    Type:=8)
    <<Add VB code to set the Include attribute of
    selected data to True>>
' Create Experiment sheet with 7 simulation cases
  FMICreateExperiment ActiveSheet,vbNullString ,7
Exit Sub
ErrFileNotFound:
    MsgBox "Error: FMU not found!"
```

Table 2: Sample VB code for loading FMU and filtering variables

The above code loads the fmu named *'DriveCycleVTM.fmu'*, and then prompts the user to select the names of the variables from the batch simulation or the DoE template sheet to be added to an experiment.



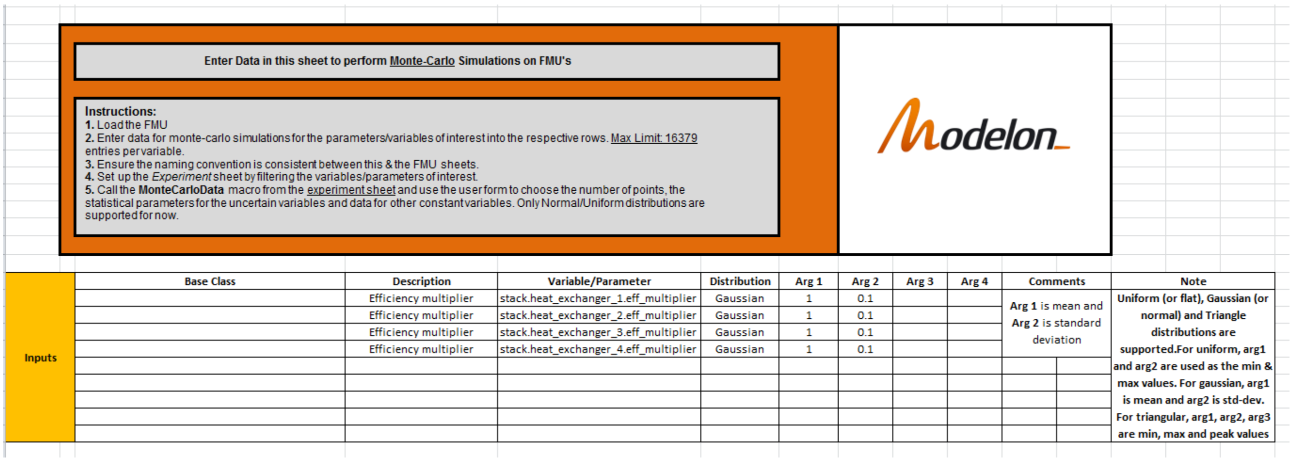
Enter Data in this sheet to perform Batch Simulations on FMU's

Instructions:
1. Load the FMU
2. Enter data for batch simulations for the parameters/variables of interest into the respective rows. Max Limit: 16379 entries per variable.
3. Ensure the naming convention is consistent between this & the FMU sheets.
4. Set up the *Experiment* sheet by filtering the variables/parameters of interest.
5. Call the LoadData macro from the experiment sheet and use the user form to choose the variables and data entered in this sheet. For a full factorial sweep of all variables, check the permute option.
6. The data are now loaded in the experiment sheet and the experiments are ready to be simulated using FMIE.

				Exp DC Batch								Exp Const Speed						
	Base Class	Description	Variable/Parameter	Default	Sim. 1	Sim. 2	Sim. 3	Sim. 4	Sim. 5	Sim. 6	Sim. 7	Default	Sim. 1	Sim. 2	Sim. 3	Sim. 4	Sim. 5	Sim. 6
Inputs		Inclination in the x direction	driverVehicle.ground.k_x	0								0						
		Final gear ratio rear	driverVehicle.vehicle.driveline.rear_ratio	4								4						
		CG height above line connecting wheel centers	driverVehicle.vehicle.h_cg	0.3								0.3						
		Vehicle body mass	driverVehicle.vehicle.body_mass	1600								1600						
		Undeformed radius - front wheel	driverVehicle.vehicle.R0_front	0.3								0.3						
		Undeformed radius - rear	driverVehicle.vehicle.R0_rear	0.3								0.3						
		Inertia about the spin axis - front wheel	driverVehicle.vehicle.wheel_i_xx_front	1								1						
		Inertia about the spin axis - rear wheel	driverVehicle.vehicle.wheel_i_xx_rear	1								1						
		Aerodynamic drag coefficient of car body	driverVehicle.vehicle.c_w_front	0.38								0.38						
		Frontal area of car body	driverVehicle.vehicle.a_front	2.7								2.7						
		scaling factor for the drivecycle velocity	driverVehicle.driveCycleNameIndex	2	23	11	9	16	19	18	15	3						
		Ambient temperature (inlet to the stack)	driverVehicle.driveCycleScale	1								1	10	15	20	25	30	35
		low fan speed temperature threshold	Tamb	311								311						
		medium fan speed temperature threshold	simpleInletConditions.airFlowGain	1								1						
		high fan speed temperature threshold	fanController.loSpeedThresh	360								360						
		max fan speed temperature threshold	fanController.medSpeedThresh	380								380						
		close grill when speed exceeds this threshold	fanController.hiSpeedThresh	400								400						
		open grill when speed falls below this threshold	fanController.maxSpeedThresh	430								430						
		grill effect air flow tuning gain	grillController.closeThreshold	18								18						
		drive cycle	grillController.openThreshold	15								15						

	Base Class	Description	Variable/Parameter	Experimental Results (if any) for the cases above
Outputs		Value of Real output	signalBus.VDL_stack_Tair_in_1	
		Value of Real output	signalBus.VDL_stack_Tair_in_2	
		Value of Real output	signalBus.VDL_stack_Tair_in_3	
		Value of Real output	signalBus.VDL_stack_Tair_in_4	
		Value of Real output	signalBus.VDL_stack_Qflow_1	
		Value of Real output	signalBus.VDL_stack_Qflow_2	
		Value of Real output	signalBus.VDL_stack_Qflow_3	
		Value of Real output	signalBus.VDL_stack_Qflow_4	
		Absolute velocity vector of fra	signalBus.VDL_veh_vel_x	
		Connector of Real output sign	simpleFan.fanSpeed	
			signalBus.VDL_eng_spd	
			signalBus.VDL_eng_trq	
			signalBus.VDL_eng_cool_temp	

Figure 3: Template for user-input for batch simulations in FMIE.



Enter Data in this sheet to perform Monte-Carlo Simulations on FMU's

Instructions:
1. Load the FMU
2. Enter data for monte-carlo simulations for the parameters/variables of interest into the respective rows. Max Limit: 16379 entries per variable.
3. Ensure the naming convention is consistent between this & the FMU sheets.
4. Set up the *Experiment* sheet by filtering the variables/parameters of interest.
5. Call the **MonteCarloData** macro from the experiment sheet and use the user form to choose the number of points, the statistical parameters for the uncertain variables and data for other constant variables. Only Normal/Uniform distributions are supported for now.

	Base Class	Description	Variable/Parameter	Distribution	Arg 1	Arg 2	Arg 3	Arg 4	Comments	Note
Inputs		Efficiency multiplier	stack.heat_exchanger_1.eff_multiplier	Gaussian	1	0.1			Arg 1 is mean and Arg 2 is standard deviation	Uniform (or flat), Gaussian (or normal) and Triangle distributions are supported.For uniform, arg1 and arg2 are used as the min & max values. For gaussian, arg1 is mean and arg2 is std-dev. For triangular, arg1, arg2, arg3 are min, max and peak values
		Efficiency multiplier	stack.heat_exchanger_2.eff_multiplier	Gaussian	1	0.1				
		Efficiency multiplier	stack.heat_exchanger_3.eff_multiplier	Gaussian	1	0.1				
		Efficiency multiplier	stack.heat_exchanger_4.eff_multiplier	Gaussian	1	0.1				

Figure 4: Template for user-input for DoE and Monte Carlo analysis in FMIE.

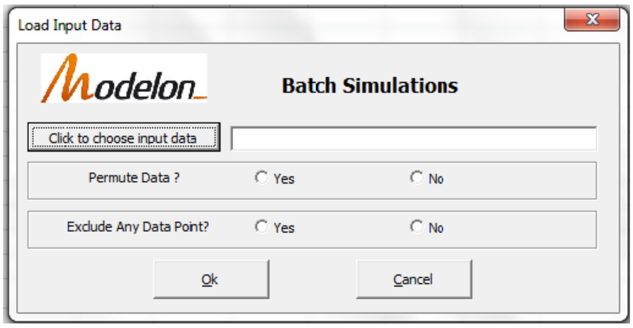
The number of simulation cases to be set-up in the experiment sheet is also stated explicitly as shown in Table 2.

3.4 Loading Data into an Experiment

As noted in 3.1 above, the batch simulation template sheet includes fields to include the data for inputs and parameters to be set in experiments. Figure 3 includes data for two experiments labeled ‘Exp DC Batch’ and ‘Exp Const Speed’ where the model is simulated with different drive cycles in the former and a sweep over different speeds with a constant speed drive cycle in the latter.

To allow a more general way for users to load input data into the experiment sheets involving multiple parameter sweeps in the same experiment either as a full factorial or independent study, the user form shown in Figure 5 is used. A similar form for data input for Monte-Carlo Analysis is shown in Figure 6.

These user forms contain private sub-routines for processing the selected input and based on the options chosen, load the input data into the experiment sheet.



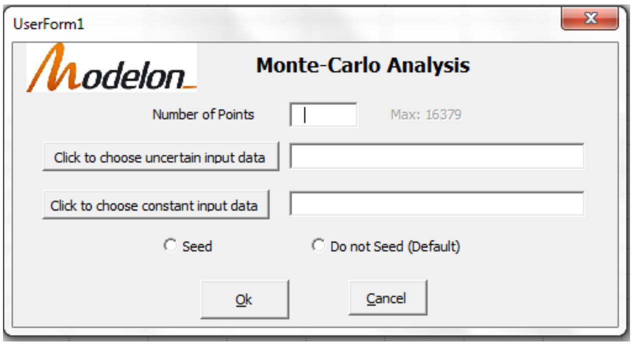
Load Input Data

Batch Simulations

Permute Data ? ○ Yes ○ No

Exclude Any Data Point? ○ Yes ○ No

Figure 5: User form for data input for batch simulations



UserForm1

Monte-Carlo Analysis

Number of Points | Max: 16379

○ Seed ○ Do not Seed (Default)

Modelon
Click to choose uncertain input data
Click to choose constant input data
Ok
Cancel

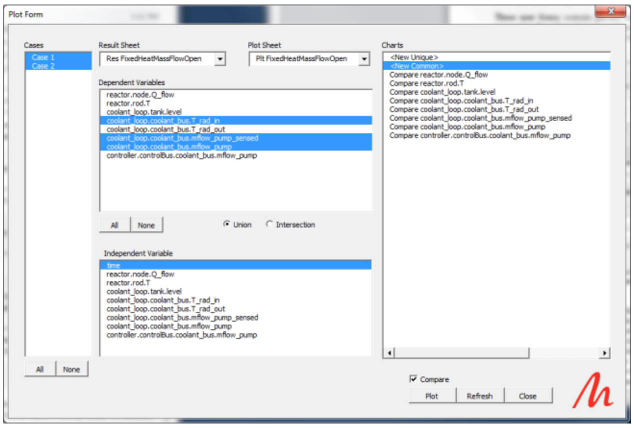
Figure 6: User form for data input for Monte Carlo analysis.

The user form for batch simulation for example lets the user choose a range of values for each variable and gives the option to run a full factorial sweep of all values for all the parameters or input variables and also the option to choose a particular set of data points for exclusion from simulations. The user form for Monte-Carlo analysis has fields to enter the number of simulations cases to be performed in the study and options to choose the uncertain parameters and their corresponding attributes (statistical distributions and the corresponding arguments) and also the option to set input data for variables that remain constant throughout the simulation study. The option to seed gives the user the ability to repeat the simulation by generating the same set of randomly generated points as from a prior experiment.

The userforms described in this section along with the private subroutines that process the user input and load the data into the FMIE experiment sheet can be combined with the VB code shown in Table 2 to seamlessly integrate the entire FMU and experiment set-up procedure into a single macro for execution. And when combined with the analysis tools for plotting that are described in the next section, provide the user with an efficient and easy-to-use work flow in working with FMUs.

3.5 Plotting simulation results

When an end user chooses the option “Create Results” from an experiment sheet, FMIE will create a new “Result” worksheet. In general this will be the same name as the experiment sheet with the prefix “Res”. The structure of these result worksheets are column oriented data of the selected output trajectories sequentially ordered by experiment case. From there the user is free to generate charts of this data as they would with other Excel data. To further increase usability for end users, custom GUIs can also be created to manage and generate these charts. An example of this is shown in Figure 7.



Plot Form

Cases
Case 1
Case 2

Result Sheet: Res FixedHeatMassFlowOpen

Plot Sheet: Plt FixedHeatMassFlowOpen

Dependent Variables
reactor.node.Q_flow
reactor.rod.T
coolant_loop.tank.level
coolant_loop.coolant_bus.T_rad_in
coolant_loop.coolant_bus.T_rad_out
coolant_loop.coolant_bus.mflow_pump_sensed
coolant_loop.coolant_bus.mflow_pump
controller.controlBus.coolant_bus.mflow_pump

● Union ○ Intersection

Independent Variable
time
reactor.node.Q_flow
reactor.rod.T
coolant_loop.tank.level
coolant_loop.coolant_bus.T_rad_in
coolant_loop.coolant_bus.T_rad_out
coolant_loop.coolant_bus.mflow_pump_sensed
coolant_loop.coolant_bus.mflow_pump
controller.controlBus.coolant_bus.mflow_pump

Charts
<New Unique>
<New Common>
Compare reactor.node.Q_flow
Compare reactor.rod.T
Compare coolant_loop.tank.level
Compare coolant_loop.coolant_bus.T_rad_in
Compare coolant_loop.coolant_bus.T_rad_out
Compare coolant_loop.coolant_bus.mflow_pump_sensed
Compare coolant_loop.coolant_bus.mflow_pump
Compare controller.controlBus.coolant_bus.mflow_pump

☑ Compare

Figure 7: Illustration of the plotting GUI

The GUI maintains multiple lists relevant to plotable data from experiments. The first is a list of result sheets within the workbook. This allows the user to select which set of results they wish to use. When a particular result sheet is selected, the "Cases" list is updated with all the cases within the result sheet.

The user is then free to choose any combination of cases they wish to work with. As the case list selection changes the lists of dependent and independent variables also changes to match the corresponding variables for the selected cases. The user also has the option of choosing either the union or intersection of the set of variables for each selected case.

The user then chooses any combination of dependent variables and a single independent variable. By default time is selected. Next the user can choose an existing "Plot Sheet" or define a new sheet directly within the drop down list. The Chart list is updated to reflect existing charts on the sheet. This allows the user to add the selected trajectories to existing charts or to create new charts with the selected signals.

When new charts are chosen they are automatically added to the sheet without needing any user intervention on placement or style. The user is however free to manually modify the chart properties after it is created.

4 Conclusions

In this paper, an approach for implementing a workflow to deploy FMUs with the FMI Add-In for Excel was presented. A typical workflow in the set-up and analysis of FMUs was presented along with a general template in Excel of a front-end with which the end-user interacts. The macros created with the user forms for loading data and the FMIE API functions can be used as the back-end which enable the end-user to quickly and efficiently simulate and analyze FMU models with relative ease. They can also be used as the front-end interface for generating charts of simulation data for quick and simple visualization of results.

References

[1] Modelisar ITEA2, "Functional Mock-up Interface for Model Exchange", Accessed from www.fmi-standard.org in Nov 2013.

[2] Modelisar ITEA2, "Functional Mock-up Interface for Co-Simulation", Accessed from www.fmi-standard.org in Nov 2013.

[3] Modelon AB, "FMI Add-In for Excel", Accessed from http://www.modelon.com/products/fmi-add-in-for-excel/ in Nov 2013.

[4] Batteh J., Gohl, J., S Chandrasekar, "Integrated Vehicle Thermal Management in Modelica: Overview and Applications" *Proceedings of the 10th International Modelica Conference*, Lund, 2014.

DOI
10.3384/ECP140961265