

Automated Deployment of Modelica Models in Excel via Functional Mockup Interface and Integration with modeFRONTIER

John Batteh¹ Jesse Gohl¹ Anand Pitchaikani¹
Alexander Duggan² Nader Fateh²

¹Modelon Inc., USA, {john.batteh, jesse.gohl, anand.pitchaikani}@modelon.com

²ESTECO North America Inc., USA, {duggan, fateh}@esteco.com

Abstract

This paper describes a method for automated deployment of Modelica models as simulators in Microsoft Excel using Functional Mockup Interface (FMI) and FMI Add-in for Excel. Using existing interfaces, integration with modeFRONTIER is demonstrated and illustrated with several different example models in different physical domains to highlight the range of applications and types of analyses that can be covered with the automated toolchain. This toolchain can be applied to any FMU and streamlined with automation enabled by the supporting annotations.

Keywords: Modelica, Functional Mockup Interface, automation, simulator, optimization, robust design, Microsoft Excel

1 Introduction

Model-based methods for development of physical and control systems have been applied across engineering domains to streamline development, reduce time to market, and manage cost and innovation. As integrated systems become increasingly complex with multi-domain interactions spanning a range of disciplines, the role of virtual models and analysis techniques in the product development process continues to grow in importance.

To meet the demand for increased model-based engineering, the ability to efficiently develop and deploy models across an enterprise is a key enabler. Models are no longer handled only by domain experts in CAE departments but are being deployed to engineers who may not have intimate knowledge of the underlying models but still are required to use models effectively to support engineering processes. With the proliferation of models throughout the enterprise, the desire for simulators outside of the original model development environment is natural and a key enabler for increased acceptance and usage of models. While it is clearly in the best interest of model users to receive models in a format of their choosing, this desire requires careful balancing of the time and effort

required to deploy the simulators, typically time spent by highly-skilled and resource-constrained model developers. Automated simulator deployment can certainly help bridge the gap between the model development and deployed simulator environments.

Open standards such as the Modelica modeling language and Functional Mockup Interface (FMI) for model exchange and co-simulation can streamline the modeling and deployment process by providing standard, non-proprietary interfaces between tools. In addition to the ability to share and integrate models from a variety of tools as FMUs, the FMI co-simulation standard provides a convenient way to deploy models outside of the original development environment as simulators. FMI-based simulators are increasingly common and rapidly gaining acceptance across industries due to the flexibility offered in simulation platforms, IP protection, and also due to the potential for flexible licensing of the deployed simulators. While FMI capability exists in nearly every Modelica-based modeling platform, the rapid adoption of FMI continues and also allows for FMI-based simulators even outside of traditional CAE tools. Common platforms for FMI-based simulators include both open source and commercial offerings in a range of environments including Python (PyFMI) and MATLAB/Simulink (FMI Toolbox for MATLAB/Simulink).

Another key aspect for the efficient utilization of deployed models is the ease with which different engineering analyses can be created and executed. modeFRONTIER (ESTECO SpA, 2015) is a process integration and design optimization tool widely used in industry. The process integration platform allows multiple third party CAE tools to be coupled together to create an automated chain. With state of the art analyses capabilities and algorithms for multi-objective and multi-disciplinary optimization, robust design, sensitivity, and statistical engineering methods, modeFRONTIER offers sophisticated features to automate the design simulation process and facilitate analytic decision making. The software's advanced post-processing modules include sophisticated data

visualization and statistical tools to facilitate understanding and gain deep insights into study results.

This paper outlines a toolchain for automated deployment of models as FMUs as simulators in Microsoft Excel. The automation relies on a set of annotations in the FMU, and these annotations are fully described. By including the annotations in the Modelica code such that they are present in the generated FMUs, an automated, streamlined path from a Modelica model to a simulator in Excel is demonstrated. Integration of the automated simulators in Excel with modeFRONTIER brings a powerful suite of analysis and optimization capabilities to the simulator toolchain. Following a description of the toolchain and automation enablers, several different examples demonstrate the entire toolchain from Modelica model to deployed simulator in Excel using FMI Add-in for Excel and integrated with modeFRONTIER. These applications also highlight a range of different analysis and optimization capabilities provided by modeFRONTIER, including parameter estimation, multi-objective optimization, and robust design.

2 Toolchain Overview

This section provides a description of the entire automation toolchain. The toolchain supports any FMU annotated as described. This section describes the annotation requirements and demonstrates the inclusion of the annotations in the Modelica code to provide an automated path from Modelica model to deployed simulator in Excel with FMI Add-in for Excel (Modelon AB, 2015) and additional integration with modeFRONTIER for analysis and optimization. The entire workflow is shown in Figure 1 and described in detail in the following sections.

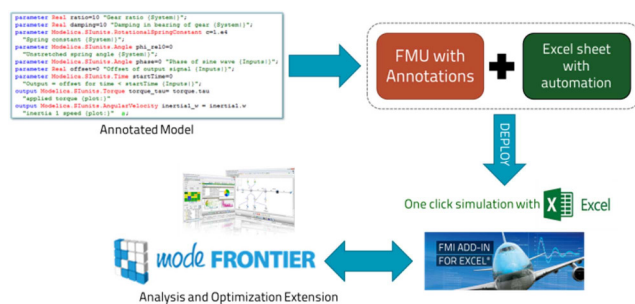


Figure 1. Workflow overview

2.1 Annotations

The toolchain automation is based on a set of annotations to identify parameters and outputs in the FMU for use in the simulator and subsequent analyses. These annotations can be provided in the Modelica code to provide a direct path for automated deployment of Modelica models as FMI-based simulators in Excel.

To identify relevant variables for the automation, the approach is to add a special substring to the variable descriptions per the markup specification in the XenGen package from Xogeny (2015). The general syntax for the markup syntax is shown below:

```
"Description
{[GroupName|][Style:][LabelString|]}"
```

Figure 2 shows sample annotations as implemented in Modelica code (as described in the markup specification, items in [] are optional) to identify an output variable and also a parameter. The overall steps are as follows for Modelica models:

- Annotate Modelica model to identify parameters and outputs per markup syntax
- Create FMU from Modelica model (if required, ensure export license usage)

When the Modelica code is annotated, the variable description flows directly to the FMU and is available in the variable description XML file. Thus, there is a direct path to support downstream automation that is implemented and maintained directly in the source before FMU generation. Alternatively, the FMU XML could be edited to add the annotations in cases where the original code is not accessible for markup (or even when the FMU generator is not Modelica-based). The downstream processes in the toolchain leverage only the FMU with annotations.

```
Real V(start=V0) "number of free virions {plot:}";
parameter Real lambda=80 "cell creation rate {Basic}";
```

Figure 2. Sample annotations in Modelica code

2.2 Automated Simulator in Excel

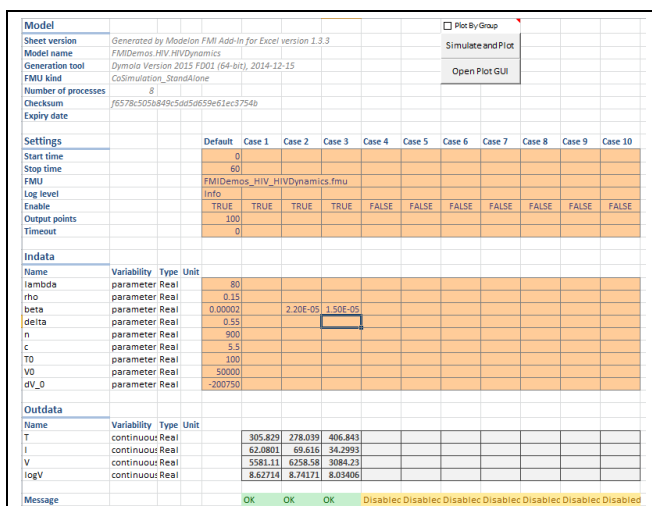
FMI Add-in for Excel (Modelon AB, 2015) provides the ability to load and simulate FMUs in Microsoft Excel. The standard workflow involves choosing the parameters and outputs to be used for experimentation via the experiment sheet which is populated with the chosen variables and ready for batch simulation. Both the final values and dynamic traces are available for post-processing. FMI Add-in also provides scripting capability for controlling the tool from macros.

Leveraging the scripting capability, automation has been added to provide automated deployment of FMUs as simulators in Excel. The automation is implemented in a workbook and provides “one click” simulation capability in Excel. This capability was first introduced to provide a dynamic simulator for small modular reactors (Hale 2014). From the main page in the workbook shown in Figure 3, the user simply points to the FMU, and the automation loads the FMU and creates an experiment sheet that includes the annotated parameter and output variables (Figure 4a). On initial load of the FMU, the workbook also runs the default simulation and plots all outputs (Figure 4b). Subsequent simulations as either single runs or batch

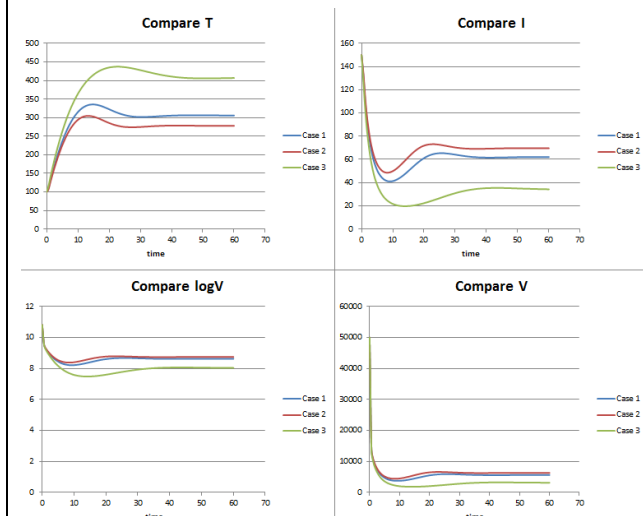
simulations are controlled by the user from the experiment sheet. Automated plotting including the ability to compare results across cases is provided in the automation worksheet. FMI Add-in for Excel offers a convenient platform in Excel for FMI-based deployment and simulation with a flexible, familiar front end for users.



Figure 3. Automated workbook main page



(a) Experiment sheet for batch simulation



(b) Plotting sheet comparing cases

Figure 4. Sample experiment and plot sheets from workbook automation

2.3 Integration with modeFRONTIER

As an analysis and optimization tool, modeFRONTIER integrates with many different CAE tools and modeling formalisms (lumped parameter, CFD, FEA, etc.). Currently modeFRONTIER does not include native FMU capability. Since modeFRONTIER includes a widely-used Excel interface, adding FMU simulation capability easily via FMI Add-in for Excel is a natural extension. Integration between FMI Add-in for Excel and modeFRONTIER leverages the existing Excel interface in modeFRONTIER and does not require any customization thereby maintaining a consistent workflow and user interface. With existing capabilities to interact with Excel sheets, modeFRONTIER leverages the deployed simulator in Excel via FMI Add-in for Excel to set parameters, simulate the FMUs, and extract data from the experiment sheet. The experiment sheet in FMI Add-in for Excel is treated no differently than any other Excel sheet with which modeFRONTIER can interact. A macro to trigger the simulation is provided in the automated worksheet. Both modeFRONTIER and FMI Add-in for Excel can parallelize the simulation runs across available machine cores for maximum utilization of computing resources. Figure 5 shows the Excel node configuration for a sample deployed simulator in FMI Add-in for Excel. The node configuration provides modeFRONTIER with the cell locations (or named ranges) for inputs and outputs along with the workbook location and macro to trigger the simulation. Multiple Excel nodes with FMUs can be coupled in a workflow (note that this coupling does not provide transient coupling between FMUs, but this coupling can be enabled in other FMI co-simulation master tools from which a single FMU can be created for use in an Excel node).

This off-the-shelf integration between FMI Add-in for Excel and modeFRONTIER provides FMU simulation capabilities to support a wide variety and rapidly growing list of third party tools with FMI support.

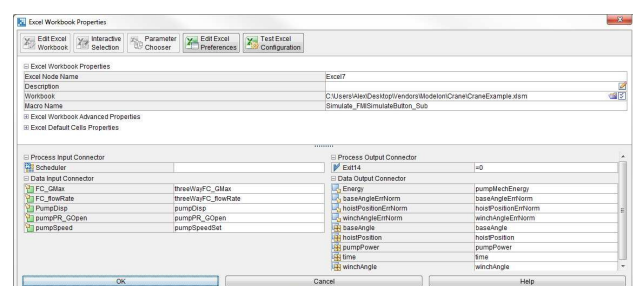


Figure 5. Excel node configuration in modeFRONTIER as applied to a deployed simulator in FMI Add-in for Excel

3 Application Examples

This section of the paper describes several different example models to illustrate the integration and analysis capabilities of modeFRONTIER and Modelica models deployed as FMU-based simulators with FMI Add-in for Excel. Co-simulation FMUs for all models are created using Dymola (Dassault Systemes, 2015). A brief overview of each model will be provided along with a description of the analysis problem, formulation of the workflow in modeFRONTIER, and key results.

3.1 HIV Virus Dynamics

As a simple first example of a dynamic system, a model of the dynamics of the HIV virus in human blood (Soetart and Petzoldt, 2010) has been implemented in Modelica. The model is similar to the standard predator-prey model for the number of uninfected (T) and infected (I) cells and the number of free virions (V). The schematic of the pathways is shown in Figure 6. The model is implemented directly in Modelica as it consists of three differential equations and has a number of parameters which are typically estimated based on clinical data from patients.

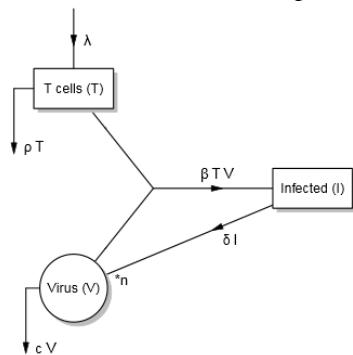


Figure 6. HIV dynamics model showing creation, destruction, and transition paths between cells (Soetart and Petzoldt, 2010)

modeFRONTIER uses a graphical workflow to set up and execute analysis and optimization tasks. These workflows can involve a single simulation node or multiple simulation nodes connected together to construct more complex, multidisciplinary tasks involving multiple tools and modeling formalisms. Figure 7 shows the workflow used to execute the HIV dynamics model correlation application. The Excel node with the FMU simulation is in the middle and labeled “FMIE”. The inputs/parameters are shown at the top of the diagram. The workflow starts by executing a Design of Experiments (DOE) for initial data. The outputs are shown at the bottom of the diagram along with any constraints or post-processing calculations for the algorithms. This application uses ESTECO’s proprietary FAST (Montrone, 2014) strategy to estimate the model parameters to best fit data. The data fitting is applied to the transient T

curve. The FAST strategy applies an algorithm to response surface models (RSM) to accelerate the fitting procedure. In this case, the SIMPLEX (Poles, 2003) algorithm was applied.

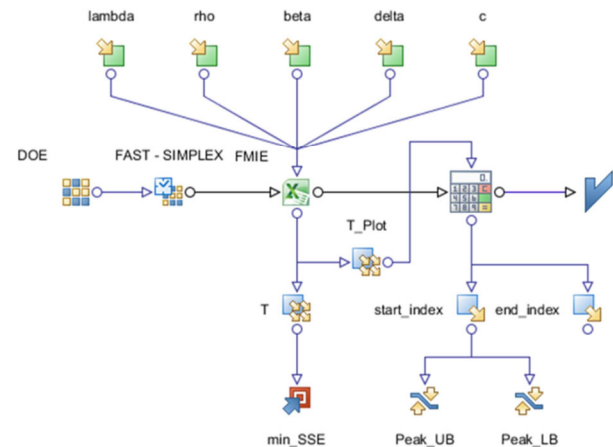


Figure 7. modeFRONTIER workflow for model correlation graphically representing the problem statement with inputs, outputs, target objective, constraints, and process flow

The optimization convergence history and dynamic T curves can be seen in Figure 8 and Figure 9, respectively. The initial T curve and the curve after fitting to the target data are shown in Figure 10.

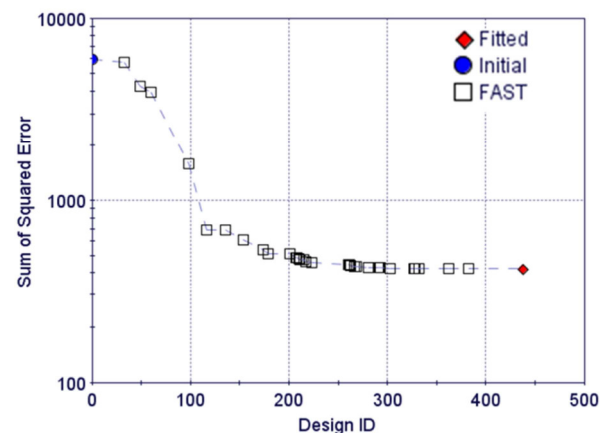


Figure 8. Optimization history showing only improved designs

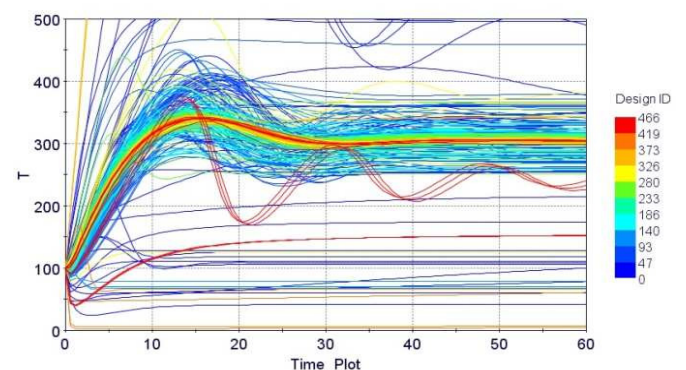


Figure 9. History as the T curve converges to the target data

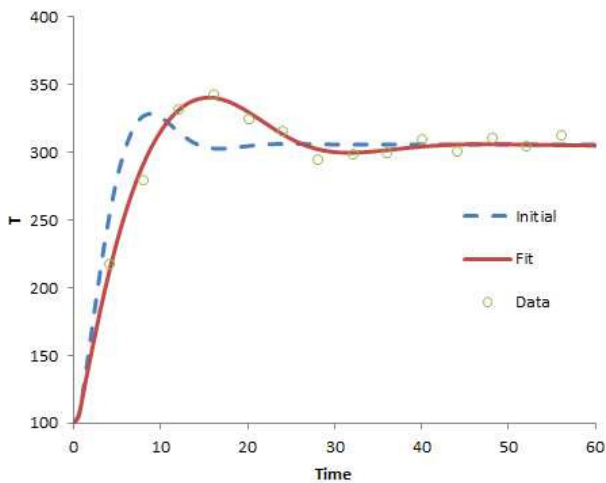


Figure 10. Initial T curve and fit curve to target data

3.2 Hydraulic Crane

Figure 11 shows a model of a crane system with a hydraulic system with motors for the movement of the crane and load. The base motor, winch motor, and hoist jack are all position controlled in closed loop to meet a desired trajectory for the crane and load. A screenshot of the animation of the model from Dymola is shown in Figure 12 with a trace for the movement of the load.

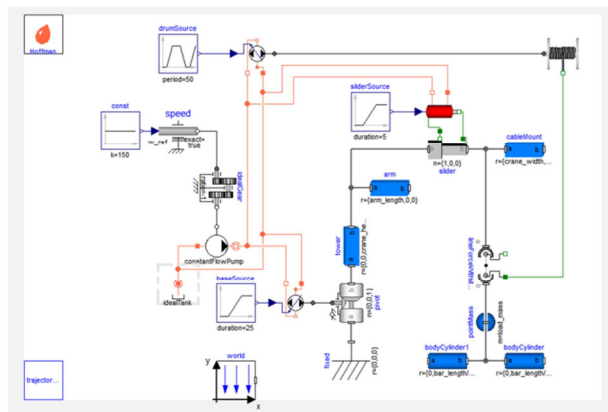


Figure 11. Crane model with hydraulic system

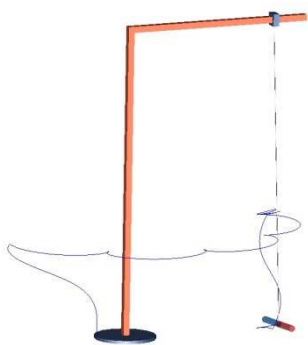


Figure 12. Crane animation

This application is formulated as a multi-objective optimization problem for modeFRONTIER via the workflow shown in Figure 13. The objective is to minimize the total tracking error and pump mechanical energy required to move the load. The total error objective function is the summation of the winch angle, base angle, and hoist position errors. Potential variables for optimization include the pump characteristics, actuator characteristics, and actuator control parameters. The input variable bounds are listed in Table 1.

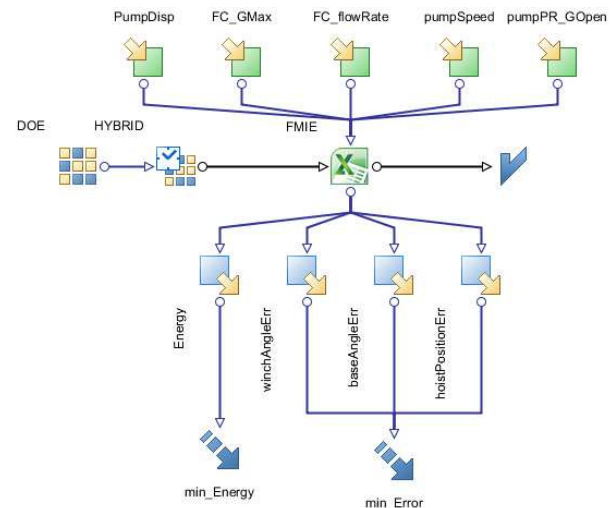


Figure 13. modeFRONTIER workflow for crane optimization showing inputs fed into Excel plugin and outputs with objective functions applied

Table 1. Input variable bounds

Name	Description	Lower Bound	Upper Bound	Units
pumpDisp	Pump Displacement	1.00E-04	0.005	m ³
FC_GMax	threeWay FC_Gmax	1.00E-11	1.00E+08	m ³ /(s.Pa)
FC_flowRate	threeWay FC_flowRate	0.001	0.05	m ³ /s
pumpSpeed	Pump SpeedSet	50	300	rad/s
PR_GOpen	Pump PR_Gopen	1.00E-12	1.00E-05	m ³ /(s.Pa)

ESTECO's proprietary HYBRID (Turco, 2011) algorithm was used along with a Multi-Objective Genetic Algorithm (MOGA-II) (Poles, 2003). The HYBRID algorithm combines the global exploration capabilities of a genetic algorithm with a gradient based method. Using both HYBRID and MOGA-II algorithms gave improved coverage of the pareto front as shown in Figure 14.

A uniformly distributed Latin Hypercube design of experiments of 10 points was used as the starting population for both algorithms. HYBRID generated a total of 700 designs; MOGA-II generated a total of 1300 designs. The pareto results in Figure 14 illustrate

the trade-off between the error and pump energy objectives.

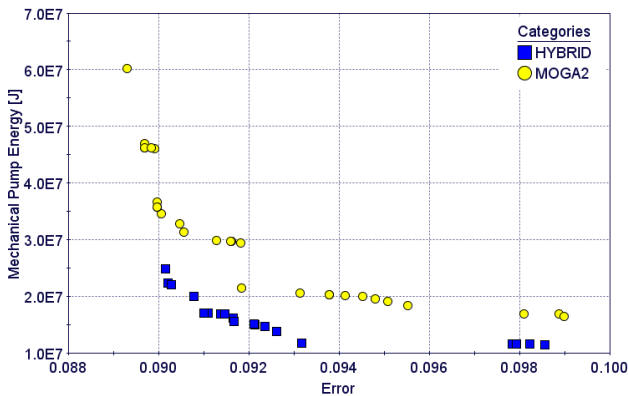


Figure 14. Pareto results from two separate optimizations using HYBRID and MOGA-II algorithms

Further analysis was done using modeFRONTIER's Multivariate tool (Stefano, 2009) to detect clusters in the pareto data. Hierarchical clustering was performed on the pareto points combining the results from both algorithms. Three clusters were identified and are represented as the three bands shown in Figure 15. The width of the band represents a 90% confidence interval. A different local response of the system is expected for each cluster. Rather than choosing a single design point from among the pareto set, a cluster can be chosen. This approach narrows down the number of candidate designs while also offering variability within the confidence interval region. In this case, we have three clusters where the energy and error objectives are distinctly different. Each cluster's corresponding configuration can be seen in Figure 15.

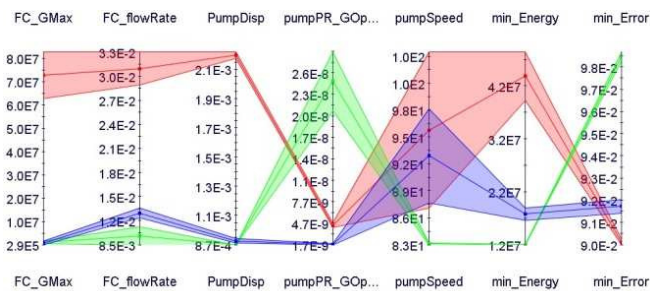


Figure 15. Clustering results on the pareto points showing three distinct pareto solutions (band width represents 90% confidence interval)

3.3 Heat Exchanger Performance with Blockage

Heat exchanger performance under non-uniform boundary conditions is a critical analysis need for vehicle thermal management (Batteh *et al.*, 2014). Blockage due to heat exchanger stacking, geometric interference with the vehicle body, or even fouling can drastically affect heat exchanger performance.

Figure 16 shows a heat exchanger test bench implemented with Modelon Heat Exchanger Library (Modelon AB, 2015). Non-uniform air side boundary conditions are provided across the face of the heat exchanger model using the Modelon DataAccess package. DataAccess provides XML reading capability and preserves dynamic file access even with the model exported as an FMU. Figure 17 shows the simple blockage pattern simulated for the cooler where the first 25% of the cooler is completely blocked.

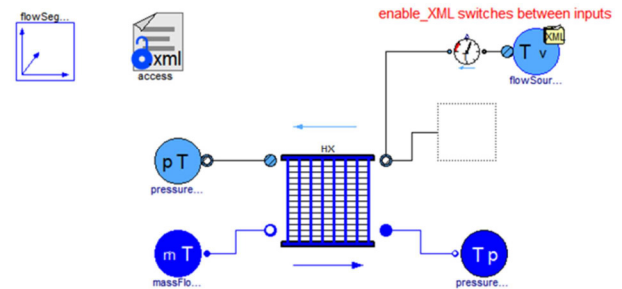


Figure 16. Heat exchanger test bench with non-uniform boundary conditions via XML

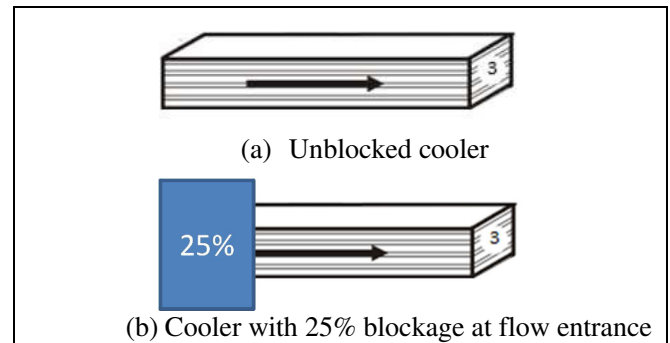


Figure 17. Heat exchanger blockage pattern

The goal of this application is to identify the velocity scale factor required for the blocked cooler such that the heat transfer performance matches that of the unblocked cooler under the same boundary conditions. For this case, the desired heat flow rate $Q_{desired}$ is 38.31kW. In addition, a robustness constraint is applied to ensure that the heat transfer does not drop by more than 1% for a 5% reduction in airflow. The workflow for modeFRONTIER is shown in Figure 18. This problem is executed in modeFRONTIER as a robust design optimization (RDO) using the SIMPLEX algorithm. A schematic showing the heat transfer distribution as a function of airflow distribution is shown in Figure 19 for an unfeasible and feasible solution based on the problem definition to illustrate the approach used for robust design optimization.

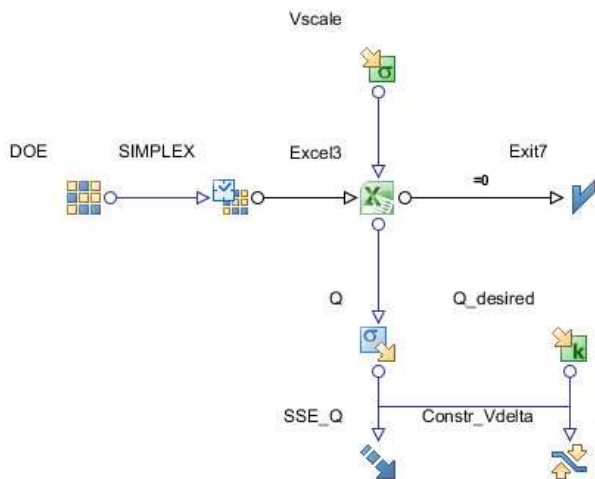


Figure 18. modeFRONTIER workflow for heat exchanger robust design application showing the stochastic input V_{scale} and reliability objective SSE_Q with constraint $Constr_Vdelta$

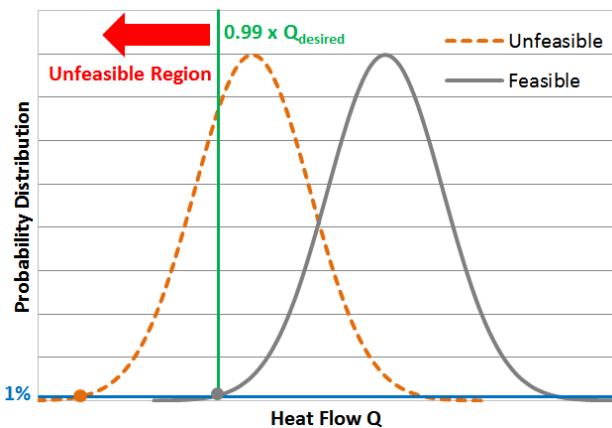


Figure 19. Schematic of feasible and unfeasible run from robust design optimization

The results of the robust design optimization are shown in Figure 20. Note the series of runs around the V_{scale} value of 1.88 which provides the required heat transfer but falls just at the constraint boundary. The value of V_{scale} that meets both the heat transfer requirement and the robustness constraint is slightly higher. While a relatively straight forward application with only a single design variable, this problem becomes significantly more complex with several design constraints and multiple design variables.

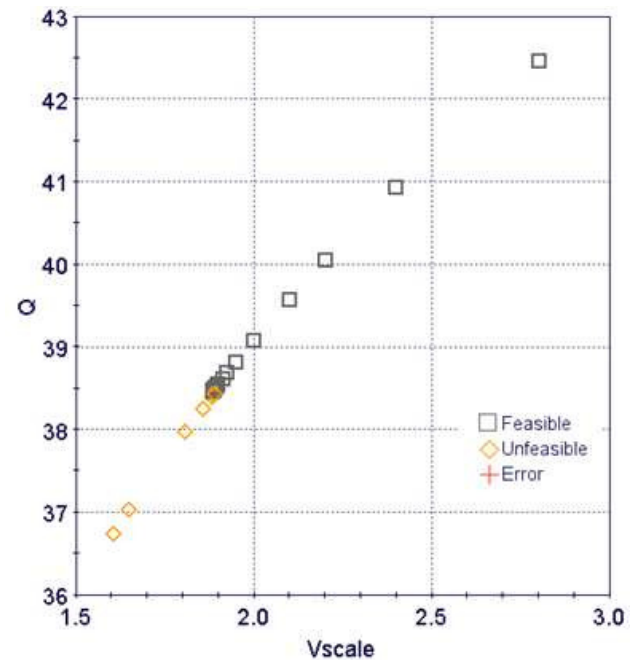


Figure 20. Convergence of the robust design optimization runs for the heat exchanger blocking problem

3.4 Hybrid Vehicle Electric Range

Vehicle range is a key metric for hybrid vehicles with electric-only mode. In addition to key vehicle parameters which affect the loads and losses, critical battery parameters strongly affect vehicle range. Battery performance is affected by both battery temperature and battery age. As batteries age, their capacity decreases while the internal resistance increases and leads to larger heat losses. Particular battery formulations typically have an optimal temperature operating range and performance degrades when the temperature strays above or below the range.

Figure 21 shows a series hybrid truck model implemented with the Modelon Vehicle Powertrain package which uses the PowerTrain Library (DLR, 2015). The battery model is a table-based model that provides open circuit voltage and internal resistance as a function of current and battery State-Of-Charge (SOC). However, the model has been enhanced to include temperature scaling in the battery tables and an approximate aging model that increases internal resistance and decreases capacity based on an aging factor.

Since actual vehicle populations in the field will have some sort of aging distribution based on usage, the resulting vehicle electric range for the fleet will be a distribution. It is important to understand the impact of aging and temperature on vehicle electric range and also analytically determine fleet population distributions.

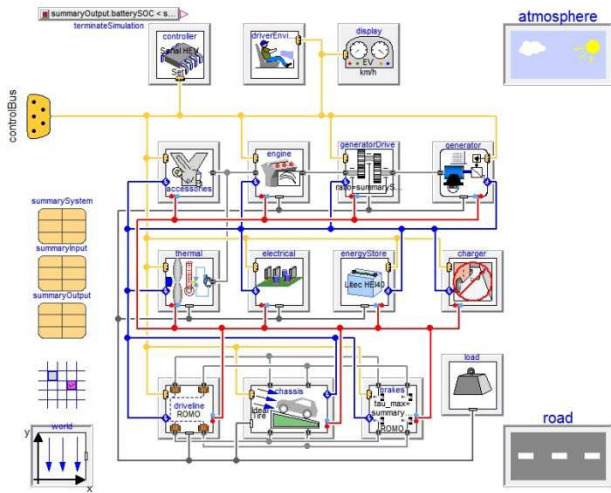


Figure 21. Series hybrid truck model

One approach to analytically determine populations is to assume a distribution and then run Monte Carlo simulations to estimate the fleet output distribution. modeFRONTIER can certainly perform Monte Carlo analyses. However, for this simple example, it is possible to run the simulations over the battery aging factor and then simply construct the distributions offline, thereby saving computational resources. The modeFRONTIER workflow for the electric vehicle range mapping is shown in Figure 22. For a given battery sink temperature, Latin Hypercube sampling is used to span the battery age factor. The simulations are run starting with a battery SOC=1 until the SOC is depleted via repeated execution of the New European Driving Cycle (NEDC) cycles. Due to numerical effects around zero SOC, the simulation is terminated at an epsilon SOC (0.05). In real practice, to prevent damage, a battery is never over charged or discharged. When studying relative effects (e.g. battery age on vehicle range), the absolute minimum SOC is not critical.

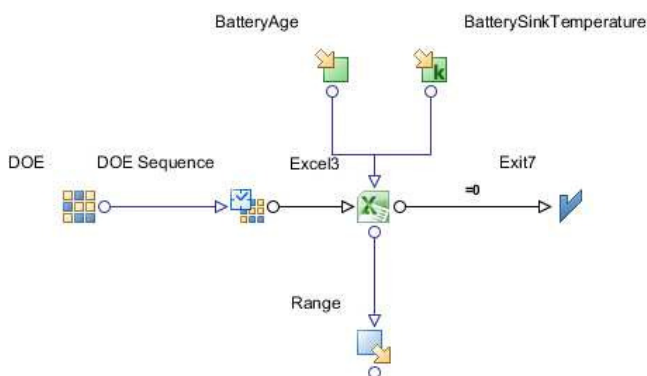
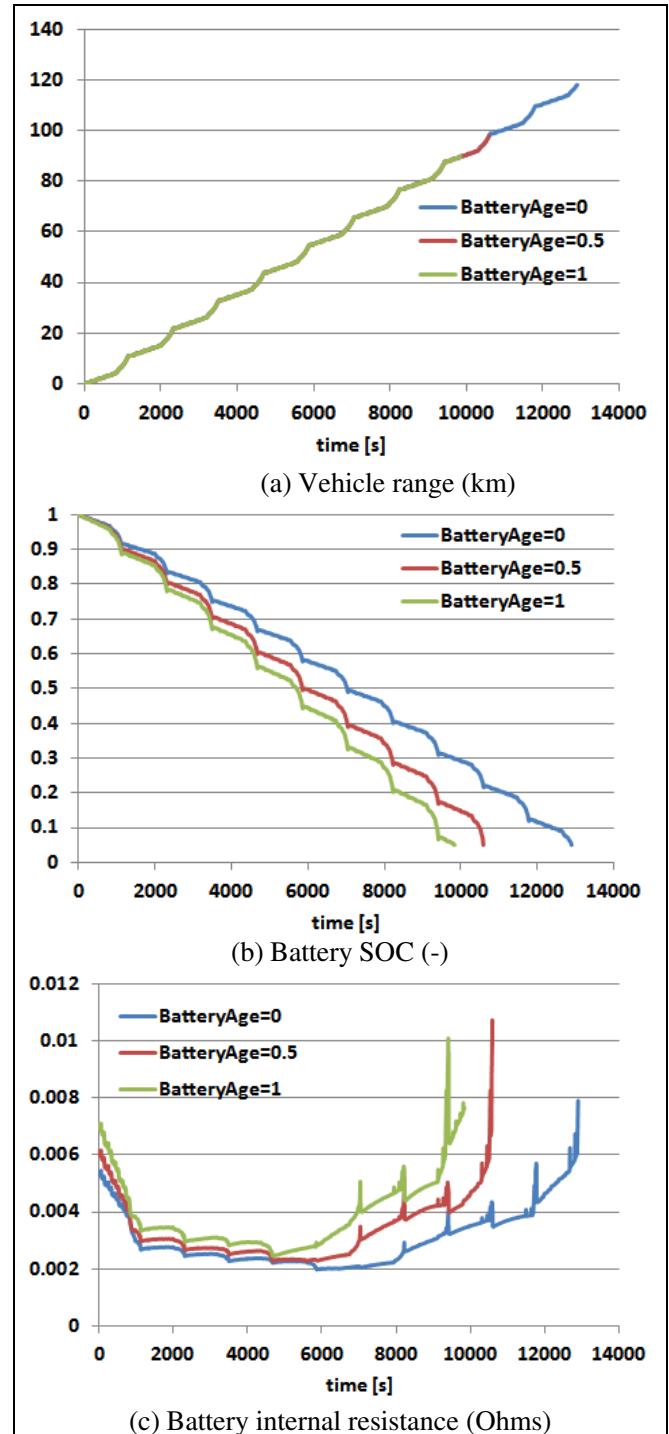


Figure 22. modeFRONTIER workflow for electric vehicle range application

Vehicle simulation results are shown in Figure 23 for a battery with no aging (BatteryAge=0), mid aging (BatteryAge=0.5), and extended aging (BatteryAge=1). As battery aging increases, vehicle range decreases substantially and battery temperatures increase due to higher internal resistance and the passive cooling strategy employed in this model. The modeFRONTIER runs showing vehicle Range over the full BatteryAge distribution are shown in Figure 24.



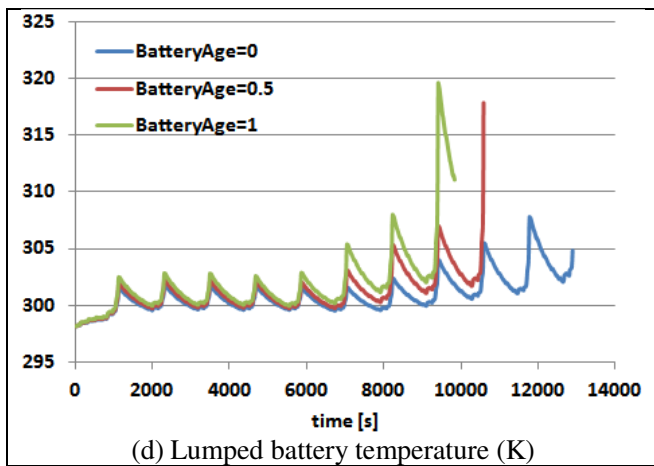


Figure 23. Simulation results for BatteryAge=0 (blue), BatteryAge=0.5 (red), and BatteryAge=1 (green)

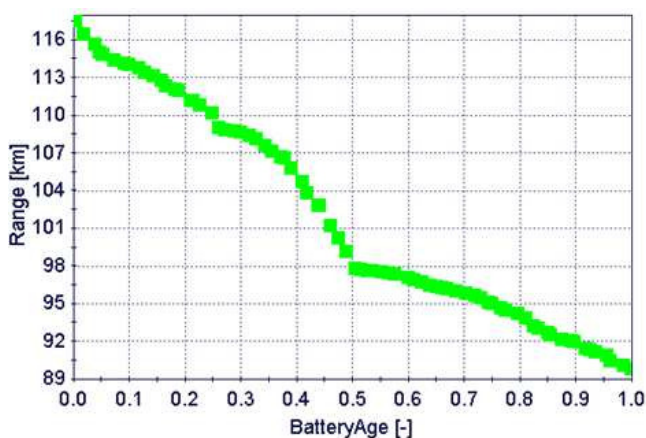


Figure 24. modeFRONTIER runs showing Range over BatteryAge distribution

With the results from modeFRONTIER over the entire age range, any arbitrary vehicle age distribution can be assumed and the fleet range distribution calculated. Figure 25 shows the calculated distribution for a normal BatteryAge distribution with a mean=0.5 and standard deviation=0.05. Similar calculations could be done over a range of different drive cycles, battery ages, and temperatures to estimate more complex fleet populations.

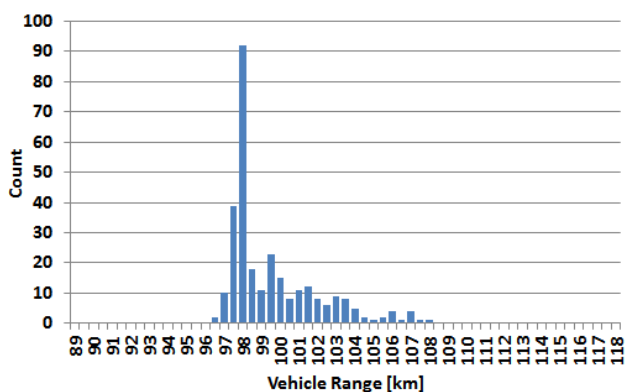


Figure 25. Fleet population assuming normal distribution with BatteryAge=0.5 and standard deviation=0.05

4 Conclusions

This paper demonstrates a method for automated deployment of models as FMU-based simulators in Microsoft Excel using FMI Add-in for Excel. A method for annotating the Modelica code using the XenGen markup syntax supports the automation to provide a streamlined path from a Modelica model to a deployed simulator in Excel. Integration of the automated simulators in Excel with modeFRONTIER brings a powerful suite of analysis and optimization capabilities to the simulator toolchain. Several different examples demonstrate the entire toolchain from Modelica model to deployed simulator in Excel using FMI Add-in for Excel and integrated with modeFRONTIER. These applications also highlight a range of different analysis and optimization capabilities provided by modeFRONTIER, including parameter estimation, multi-objective optimization, and robust design. This toolchain can be applied to any FMU and streamlined with automation enabled by the supporting annotations.

References

- John Batteh, Jesse Gohl, and Chandrasekar Sureshkumar. Integrated Vehicle Thermal Management in Modelica: Overview and Applications, *Proceedings of the 10th International Modelica Conference*, Lund, Sweden, pp. 409-418, March 2014. doi: 10.3384/ecp14096409.
- Dassault Systemes. Dymola 2015 FD01, 2015.
- DLR, PowerTrain Library, v.2.3.0, 2015.
- ESTECO SpA. modeFRONTIER 2014 Update 1, 2015. <http://www.esteco.com/>
- Richard Hale, Sacit Cetiner, David Fugate, Lou Qualls, John Batteh, and Michael Tiller. Dynamic Modeling of Small Modular Nuclear Reactors using MoDSim, *Proceedings of the 10th International Modelica Conference*, Lund, Sweden, pp. 989-998, March 2014. doi: 10.3384/ecp14096989.
- Modelon AB, FMI Add-in for Excel, v1.3.5, 2015. <http://www.modelon.com/products/fmi-add-in-for-excel/>.
- Modelon AB, Heat Exchanger Library, v1.2, 2015. <http://www.modelon.com/products/modelica-libraries/heat-exchanger-library/>.
- T. Montrone, A. Turco, R. Enrico. FAST Optimizers: General Description. *ESTECO Technical Report 2014-001*, Trieste, Italy, 2014.
- S. Poles. The SIMPLEX Method. *ESTECO Technical Report 2003-005*, Trieste, Italy, 2003.
- S. Poles. MOGA-II an Improved Multi-Objective Genetic Algorithm. *ESTECO Technical Report 2003-006*, Trieste, Italy, 2003.
- Karlne Soetaert and Thomas Petzoldt. Inverse Modelling, Sensitivity and Monte Carlo Analysis in R using Package FME. *Journal of Statistical Software*, 33(3), pp. 1-28, February 2010.

- D. Stefano. Multivariate Analysis algorithms in modeFRONTIER v4. *ESTECO Technical Report 2009-001*, Trieste, Italy, 2009.
- A. Turco. Hybrid – description. *ESTECO Technical Report 2011-003*, Trieste, Italy, 2011.
- Xogeny, XenGen package, 2015.
<https://github.com/xogeny/XenGen>