ExcelInterface – A Tool for Interfacing Dymola through Excel

Kristian Tuszynski,
Modelon AB, Ideon Science Park, SE-22370 Lund, Sweden

kristian.tuszynski@modelon.se

Abstract

This paper presents a tool created in Excel which enables interfacing with Dymola. The tool was created to simplify batch simulations and allow easy post processing of a large number of simulations. The interface handles both steady state sweeps of a model as well as continuing from a previous simulation. Support for calibration using linear regression is also implemented which allows calibration of simpler models.

Keywords: Excel; Simulation; DDE; Scripting; Dymola; Batch simulation; Steady State; Interface

1 Introduction

When simulating a large number of cases, either to validate a model against measurement data or when acquiring experimental results, there is strong need to be able to organize and get a good overview of both the experiment setup and the result of the simulations. The ExcelInterface greatly improves and simplifies both the post processing and setup involved when running a batch of simulations with changing boundary conditions between the cases. The tool allows the user to define a number of cases to run and then get the result from the simulations presented in Excel for easy comparison. This gives a good overview of what has been set in the model, without actually changing the model code allowing the model stored in Dymola to be generic and instead all different simulation cases are defined in the Excel sheet.

Doing the same thing using Dymola directly would force the user to make model changes for each parameter set, create multiple models where each uses a different parameter set or make a custom made script file where the simulation cases are defined. All these options are quite time consuming and do not provide a good overview.

Having the result in Excel also enables the use of the tools included in the program. Excel and its tools have the advantage that the knowledge and use of them are wide spread which means that it is not necessary for a person with Modelica or Dymola knowledge to analyze and make further post processing of the result. This simplifies the result exchange when working with someone without any prior Modelica knowledge.

2 Overview

The tool is built using VBA (Visual Basic for Applications) which comes with Excel. The communication between Excel and Dymola is performed both using files and a DDE connection established between Dymola and Excel.

When simulating, the interface works by creating a Dymola script based on chosen settings in the ExcelInterface. This script is executed in Dymola, using DDE commands sent from Excel. For each simulation, specified output values are saved in temporary files which are read by Excel after all simulations have completed. In Excel the result is presented at position and with appearance defined by the user through the interface. The communication is illustrated in Figure 1.

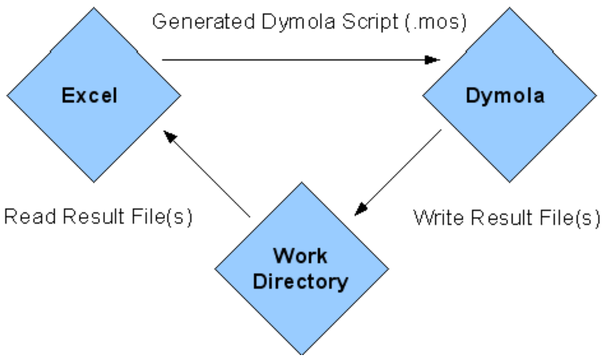
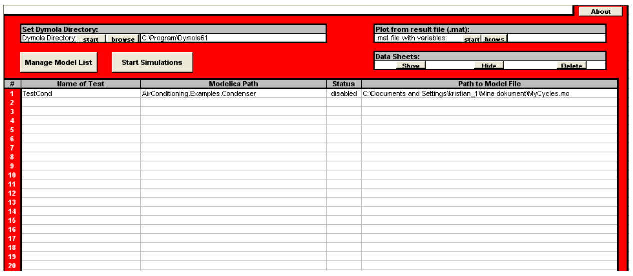


Figure 1 Communication between Excel and Dymola

3 Setup

The ExcelInterface contains a setup sheet, seen in Figure 2, where all cases to be simulated are specified. A case is defined by a unique name used in the interface, a path to a model file and the Modelica path to the model to be simulated within the file. Each case can be enabled and disabled deciding if they are run or not when starting the simulations.



About

Set Dymola Directory:		
Dymola Directory:	start / browse	C:\Program\Dymola61

Plot from result file (.mat):	
.mat file with variables:	start / brows

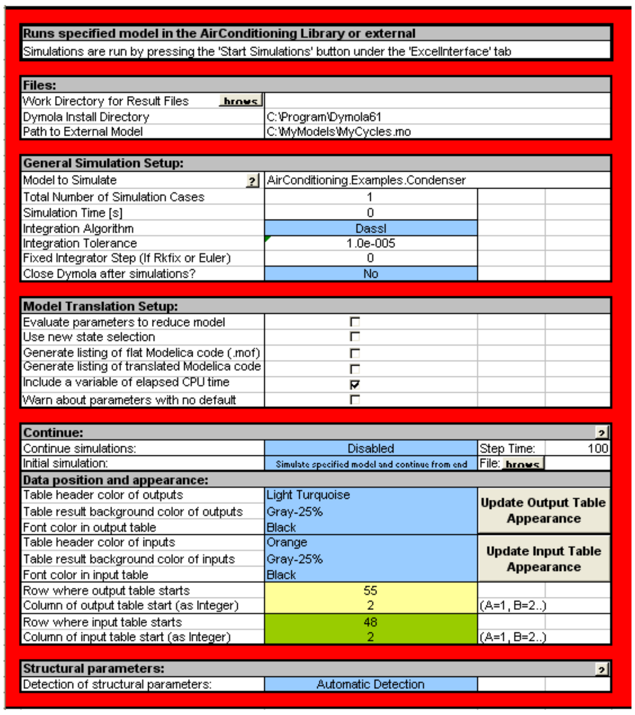
Manage Model List | Start Simulations

Data Sheets: Show | Hide | Delete

#	Name of Test	Modelica Path	Status	Path to Model File
1	TestCond	AirConditioning.Examples.Condenser	disabled	C:\Documents and Settings\kristian_1\Mina dokument\MyCycles.mo
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

Figure 2 Setup sheet of the ExcelInterface

For each case a new excel sheet is created where the user has to specify a number of parameters including work directory, integrator, tolerance, number of simulations cases and simulation time as seen in Figure 3.



Runs specified model in the AirConditioning Library or external
Simulations are run by pressing the 'Start Simulations' button under the 'ExcelInterface' tab

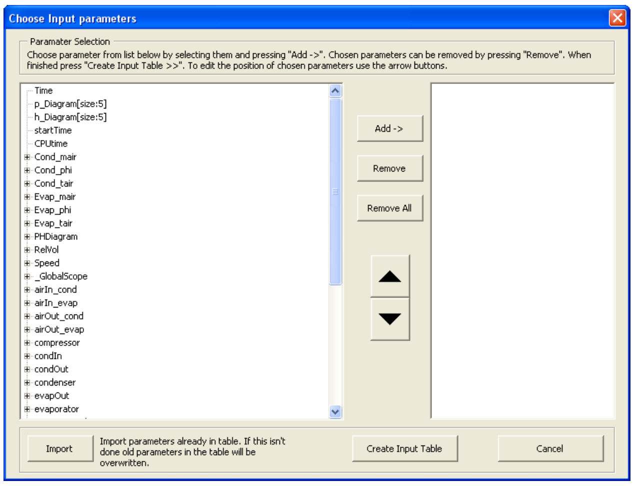
Files:			
Work Directory for Result Files (brows)			
Dymola Install Directory	C:\Program\Dymola61		
Path to External Model	C:\MyModels\MyCycles.mo		
General Simulation Setup:			
Model to Simulate (?)	AirConditioning.Examples.Condenser		
Total Number of Simulation Cases	1		
Simulation Time [s]	0		
Integration Algorithm	Dassl		
Integration Tolerance	1.0e-005		
Fixed Integrator Step (If Rkfix or Euler)	0		
Close Dymola after simulations?	No		
Model Translation Setup:			
Evaluate parameters to reduce model	☐		
Use new state selection	☐		
Generate listing of flat Modelica code (.mof)	☐		
Generate listing of translated Modelica code	☐		
Include a variable of elapsed CPU time	☑		
Warn about parameters with no default	☐		
Continue:			
Continue simulations:	Disabled	Step Time:	100
Initial simulation:	Simulate specified model and continue from end	File: (brows)	
Data position and appearance:			
Table header color of outputs	Light Turquoise	Update Output Table Appearance	
Table result background color of outputs	Gray-25%		
Font color in output table	Black		
Table header color of inputs	Orange	Update Input Table Appearance	
Table result background color of inputs	Gray-25%		
Font color in input table	Black		
Row where output table starts	55		
Column of output table start (as Integer)	2	(A=1, B=2..)	
Row where input table starts	48		
Column of input table start (as Integer)	2	(A=1, B=2..)	
Structural parameters:			
Detection of structural parameters:	Automatic Detection		

Figure 3 Sheet specific for a case

The input and output variables to/from the model are selected from menus in Excel. The first time a model is to be simulated through the ExcelInterface, the model has to be analyzed to find all parameters and variables contained in the model. A DDE connection is used between Excel and Dymola to execute the commands necessary to perform these operations which include:

- translation and simulation of the model
- parameter and variable names extraction from the generated result file
- saving extracted parameter and variable names in a user specified file

This procedure only has to be performed once for every model as any following need to add parameters/variables the saved file is used.



Choose Input parameters

Paramater Selection
Choose parameter from list below by selecting them and pressing "Add ->". Chosen parameters can be removed by pressing "Remove". When finished press "Create Input Table >>". To edit the position of chosen parameters use the arrow buttons.

Time
p_Diagram[size:5]
h_Diagram[size:5]
startTime
CPUtime
Cond_mair
Cond_phi
Cond_tair
Evap_mair
Evap_phi
Evap_tair
PHDiagram
RelVol
Speed
_GlobalScope
airIn_cond
airIn_evap
airOut_cond
airOut_evap
compressor
condIn
condOut
condenser
evapOut
evaporator

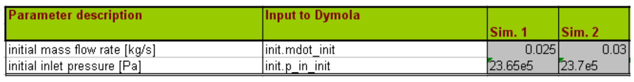
Add -> | Remove | Remove All

Import — Import parameters already in table. If this isn't done old parameters in the table will be overwritten.

Create Input Table | Cancel

Figure 4 Tree view menu with parameters and variables

Input and output variables are then selected from the generated tree view menu, seen in Figure 4, and finally values are set in the generated input table such as the one shown in Figure 5.



Parameter description	Input to Dymola	Sim. 1	Sim. 2
initial mass flow rate [kg/s]	init.mdot_init	0.025	0.03
initial inlet pressure [Pa]	init.p_in_init	23.65e5	23.7e5

Figure 5 Set boundary conditions

4 Running Simulations

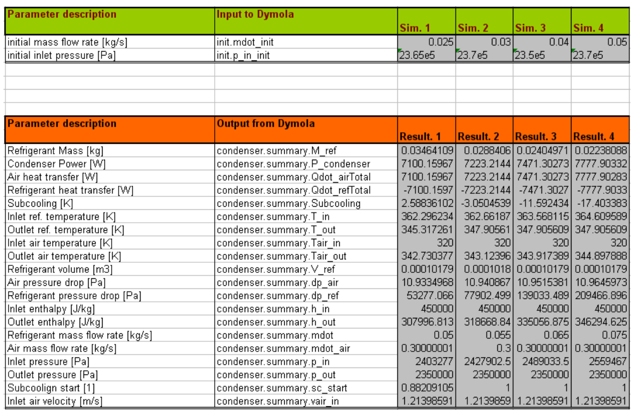
There are two ways to run multiple simulations using the ExcelInterface:

- Steady State Simulations
- Continue Simulation
  - Continue from First
  - Continue from Previous

4.1 Steady State Simulations

This option simulates the specified cases one after another and the results at the specified end time are returned. In case of a model that initializes in steady state, the simulation time should be set to zero for faster execution, for all other cases the user has to determine a simulation time that is long enough for the simulation to reach steady state.

Structural parameters are parameters which force a re-compilation of the model as they change the generated code structure. A good example of structural parameters is discretization parameters. If all selected input parameters are non-structural the model is only translated and compiled once, enabling fast simulations. As it may not always be trivial to know which parameters in a model that are structural, the ExcelInterface automatically detects if a structural parameter was selected as an input parameter. If one or more structural parameters are detected the model has to be retranslated between each run case.



Parameter description	Input to Dymola	Sim. 1	Sim. 2	Sim. 3	Sim. 4
initial mass flow rate [kg/s]	init.mdot_init	0.025	0.03	0.04	0.05
initial inlet pressure [Pa]	init.p_in_init	23.65e5	23.7e5	23.5e5	23.7e5

Parameter description	Output from Dymola	Result. 1	Result. 2	Result. 3	Result. 4
Refrigerant Mass [kg]	condenser.summary.M_ref	0.03464109	0.0288406	0.02404971	0.02238088
Condenser Power [W]	condenser.summary.P_condenser	7100.15967	7223.2144	7471.30273	7777.90332
Air heat transfer [W]	condenser.summary.Qdot_airTotal	7100.15967	7223.2144	7471.30273	7777.90283
Refrigerant heat transfer [W]	condenser.summary.Qdot_refTotal	-7100.1597	-7223.2144	-7471.3027	-7777.9033
Subcooling [K]	condenser.summary.Subcooling	2.58836102	-3.0504539	-11.592434	-17.403383
Inlet ref. temperature [K]	condenser.summary.T_in	362.296234	362.66187	363.568115	364.609589
Outlet ref. temperature [K]	condenser.summary.T_out	345.317261	347.90561	347.905609	347.905609
Inlet air temperature [K]	condenser.summary.Tair_in	320	320	320	320
Outlet air temperature [K]	condenser.summary.Tair_out	342.730377	343.12396	343.917389	344.897888
Refrigerant volume [m3]	condenser.summary.V_ref	0.00010179	0.0001018	0.00010179	0.00010179
Air pressure drop [Pa]	condenser.summary.dp_air	10.9334968	10.940867	10.9515381	10.9645973
Refrigerant pressure drop [Pa]	condenser.summary.dp_ref	53277.066	77902.499	139033.489	209466.896
Inlet enthalpy [J/kg]	condenser.summary.h_in	450000	450000	450000	450000
Outlet enthalpy [J/kg]	condenser.summary.h_out	307996.813	318668.84	335056.875	346294.625
Refrigerant mass flow rate [kg/s]	condenser.summary.mdot	0.05	0.055	0.065	0.075
Air mass flow rate [kg/s]	condenser.summary.mdot_air	0.30000001	0.3	0.30000001	0.30000001
Inlet pressure [Pa]	condenser.summary.p_in	2403277	2427902.5	2489033.5	2559467
Outlet pressure [Pa]	condenser.summary.p_out	2350000	2350000	2350000	2350000
Subcoolign start [1]	condenser.summary.sc_start	0.88209105	1	1	1
Inlet air velocity [m/s]	condenser.summary.vair_in	1.21398591	1.2139859	1.21398591	1.21398591

Figure 6 Input and output in Excel

Figure 6 shows an example on how the output in Excel can look like after a successful simulation. In the example four different cases were run and two parameters (*init.mdot_init* and *init.p_in_init*) were changed between the simulations.

4.2 Continue Simulation

Besides running each simulation as a separate case, the ExcelInterface offers two other ways for series of steady state calculations. The most common reasons for using these options are that the model can not successfully initialize at every steady state point and/or that the initialization phase of the model is very time consuming making it practical to continue from a initialized model that has reached steady state.

By connecting ramp blocks to the boundary conditions of the simulated model where the start values of the output signals equal the end value from the previous simulation it is possible to start each new simulation from steady state and then change the boundary conditions by setting desired height of the ramp blocks.

Continue from First simulates the model for a specified time and then the remaining simulations continue from this point. This makes it possible to define a number of transients using, for instance, ramp blocks and sweep any number of steady state points.

Figure 7 shows the outlet evaporator temperature in an AC-cycle simulation where the model was first simulated until it was in steady state. Once this point was reached (after 200 seconds) five simulations were executed from the end of the first simulation where the inlet air temperature was changed between the simulations.

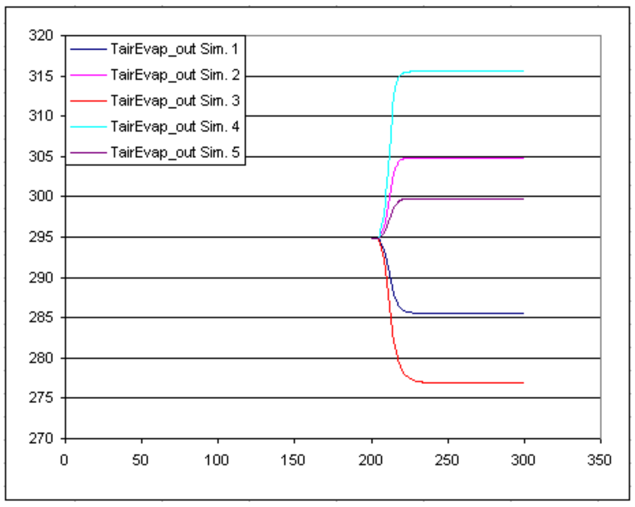


Figure 7 Example result when using "Continue from First"

The second option **Continue from Previous** also simulates the model for a specified time and then each specified case continues from the previous simulation.

A **Continue from Previous** run is illustrated in Figure 8 where the outlet evaporator temperature of an AC-cycle is shown. The initial simulation is continued after 200 seconds. After this time 5 simulations are run where each one is 100 seconds long.

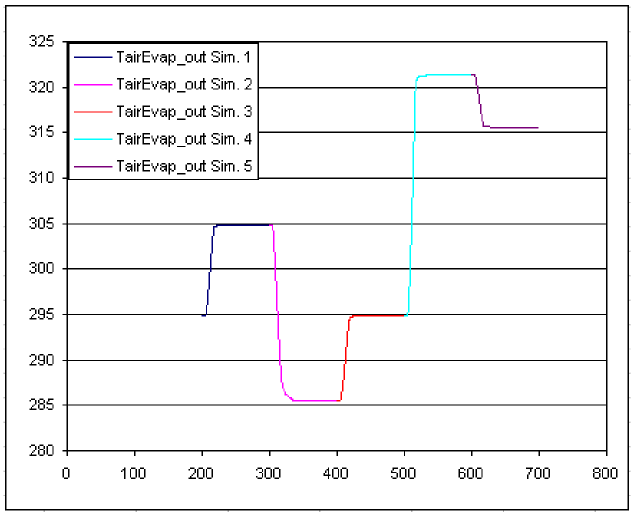


Figure 8 Example result when using "Continue from Previous"

For both these options it is possible to start the simulations from a saved result file. Using this option the initial simulation, which takes the model to steady state, is skipped. Instead all initial values are taken from the result file.

4.3 Plotting and Dynamic Result

For all simulations performed using the interface it is optional to include plots of chosen variables in Excel. Enabling this option is useful to get a quick visual comparison of the different simulation results and when it is necessary to verify that the model really reached steady state after an initial transient.

The plots are created by extracting wanted trajectories from the result files and saving them in sheets within the work book making the trajectories easily accessible.

5 Usage Example – Charge Optimization

To find the optimal charge of an AC-Cycle the cycle is first almost completely drained and then filled in multiple steps until the accumulator of the cycle is over filled. At each step important values such as the power, pressures, subcooling and superheat temperatures are measured.

Simulating this procedure in one continuous simulation might prove difficult as it is often necessary to simulate between 8 and 15 points altogether and there is a risk that the simulation will fail during the transition between, at least, two of the points. If this happens it is quite time consuming to re-run the simulation and there are no guarantees it will work the second time around either.

Using the ExcelInterface the risk is minimized when simulating the charge optimization using the continue feature of Dymola. The experiment is setup by adding a controlled flow source, to the cycle, which fills the accumulator with refrigerant at specified time as shown in .

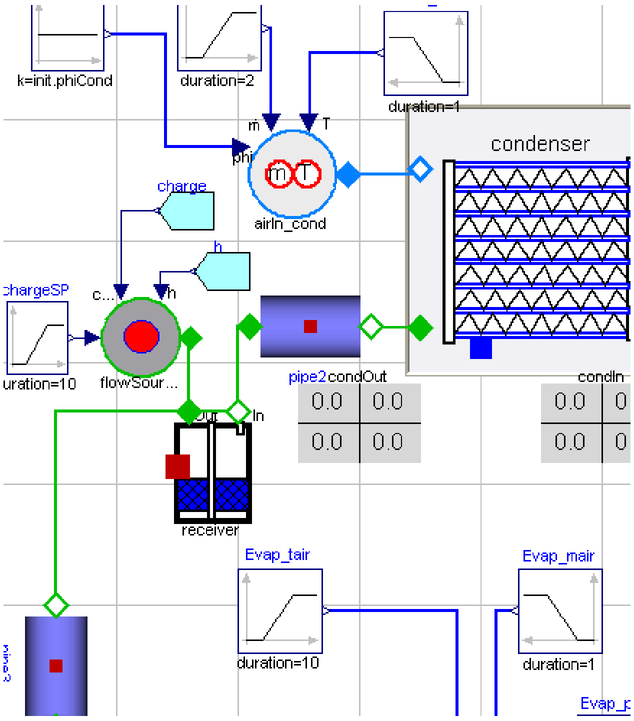
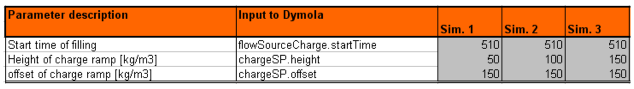


Figure 9 Cycle with controllable flow source

In the ExcelInterface the start time of the filling, height and offset of the set-point ramp block is selected as input parameters and the experiment is run using **Continue from First**. Finally, the cycle model is parameterized to begin the continue simulations having a charge of 150 kg/m^3.



Parameter description	Input to Dymola	Sim. 1	Sim. 2	Sim. 3
Start time of filling	flowSourceCharge.startTime	510	510	510
Height of charge ramp [kg/m3]	chargeSP.height	50	100	150
offset of charge ramp [kg/m3]	chargeSP.offset	150	150	150

Figure 10 Setup in the ExcelInterface

Assuming that the initial simulation, which controls the charge down to 150 kg/m^3 passes the whole experiment will not fail if a single simulation fails. Instead of risking having to redo the whole experiment the worst case scenario is now that some of the simulations have to be redone because they crashed.

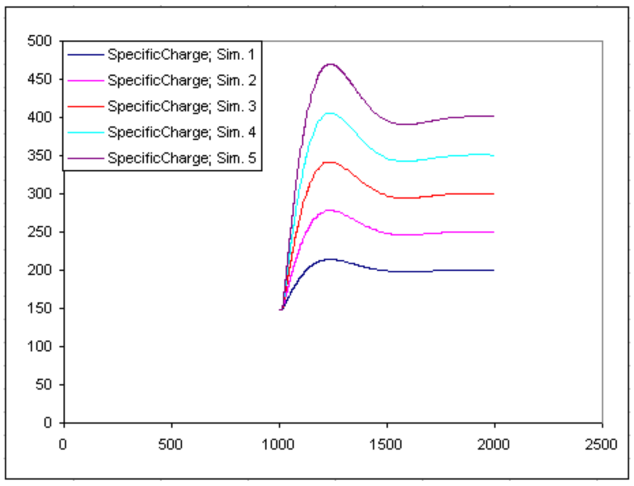


Figure 11 Plot of specific charge. 5 points were simulated from 200-400 kg/m3

6 Summary

The ExcelInterface has proven to be an efficient tool to use when doing batch simulations over a large number of steady state points.

The interface gives the user a good overview of the cases to simulate and simplifies the post processing of the result as well as speeding up the setup of the experiments. This in combination with the fact that Excel is a well known program which many people have experience working with gives the interface great flexibility and a broad user base.

References

[1] Excel VBA Language Reference, www.microsoft.com

[2] Dymola User Manual