MuleSim3.m User Manual

*To use this MATLAB-based model, you need to download the contents of the “MuleSim3” folder in the google drive to a local folder on your computer. This is especially important if you are running any design optimization or sensitivity analysis code, since the MATLAB code is set up to run files that are in the same directory as the source code.*

*The program is set up to accept many different kinds of input, depending on whether it is run standalone or if it is called by an optimization program such as “Surrocket.” This manual covers the use of “MuleSim3” to simulate a hybrid rocket motor as a standalone program.*

**Program Overview**

The main function of “MuleSim3” is to generate performance data and plots for a given hybrid rocket motor design. This design is specified in an excel spreadsheet, which is essentially the program GUI. When used as a standalone motor simulator, the MATLAB function accepts no explicit input and returns no explicit output, and is run by calling the function in the command line or by clicking the “Run” button with the function open.

If the “graph” option is indicated in the input spreadsheet, key results are plotted (typically thrust, combustion chamber pressure, and oxidizer tank pressure). If the “save” option is indicated in the input spreadsheet, key results are saved to a separate excel spreadsheet and any plots are also saved.

**Program Input**

Input to the model is provided by several files:

**CEA lookup table stored as MATLAB data file (default “MuleSimCEA.mat”)**

- Hard-coded as variable “input\_f” in the first input block

- Combustion analysis results obtained by processing a CEA .out file using “CEAtoMATLAB3.m”

- New .mat file needed with any change to oxidizer or fuel composition

**Input excel spreadsheet containing motor data (default “MuleSimINPUT.xlsx”)**

- Hard-coded as variable “CEA\_f” in the first input block

- Excel spreadsheet containing geometry, oxidizer input, and initial conditions for model

- Model generates variables based on the “Symbol” column, so be careful when changing the values

**Nitrous oxide thermodynamic properties at saturation (“N2OSat.m”)**

- Hard-coded as variable “N2OSat” in the first input block, should not be changed

**Benchmarked motor performance curves (“Motor\_Validation.mat”)**

- Hard-coded as variable “valid” in the first input block

- Contains thrust and pressure time curves for benchmarked motors for model validation

- Can be used to calculate error between simulated and actual tests

**Program Output**

All program outputs are hard-coded into the program after the main loop. If you want to make custom graphs for the properties you are interested in, you have to hard-code in plots yourself. The values that are output to the excel spreadsheet are also hard-coded, but this is more difficult to change.

Currently the following data is saved to the output spreadsheet:

- All input data from input excel spreadsheet

- Burn time

- Maximum thrust

- Average thrust

- Total Impulse

- Specific Impulse

- Average O/F ratio

- Average oxidizer mass flow

- Average fuel mass flow

- Total oxidizer mass consumed

- Total fuel mass consumed

- Average regression rate

- Final port diameter

- Maximum combustion chamber temperature

- Maximum combustion chamber pressure

- Tank pressure error (requires hard-coded adjustment)

- Combustion chamber pressure error (requires hard-coded adjustment)

- Thrust error (requires hard-coded adjustment)

Additionally, full time-series data of the following parameters is saved to the output spreadsheet:

- Thrust

- Combustion chamber temperature

- Combustion chamber pressure

- Oxidizer to fuel ratio

- Temperature of oxidizer in tank

- Pressure of oxidizer in tank

- Vapour mass fraction of oxidizer in tank

**Changing Fuel or Oxidizer Composition**

In order to change the fuel or oxidizer composition, the CEA input file “MuleSim3.inp” located in the CEA folder should be edited using a text editor and run in CEA. Be careful to NOT save the file as a text file, but to keep it as an ANSI encoded “.inp” file.

CEA is run by clicking on the “FCEA2” application in the CEA folder, and then entering the input filename without the file extension in the command window that pops up (i.e. type “MuleSim” to run the “MuleSim.inp” file, if “MuleSim.inp” is in the same folder as “FCEA2”).

CEA will generate an output file with the same name as the input file (i.e. “MuleSim.out”), which can then be processed using the MATLAB function “CEAtoMATLAB3.m”. This will generate a .mat file that can be used as input to the “MuleSim.m” model.

Send a message to Benjamin Klammer on slack if you run into any problems