

TXVThrustcurve

User Manual and Technical Documentation

Directory

Introduction and System Description

Input Variables

Operation Procedure

Engine Model

Altitude Compensation

Tank Modeling and CG- Calculation

Output File

Introduction and System Description

OpenRocket is an open source 6 degrees of freedom simulation tool initially developed for high power rocketry flight simulations. Due to import regulations of commercial rocketry simulation programs (RocksimPro), OpenRocket is extensively used in the realms of european experimental- and high power rocketry. While the program itself excels in terms of intuitive handling and reasonable simulation data, there is no tool for generating custom rocket engines included in the programme. Also, due to usual short burntime and low level flights, Engine Altitude Compensation is not implemented in OpenRocket. Especially for teams, that are designing their own liquid- or hybrid bipropellant rocket engines, this is a huge disadvantage.

TXVThrustcurve is a programme designed for generating Liquid/ Hybrid Rocket engine files for the use in OpenRocket and other commercial Simulation Software. In its current state, the python based programme is capable of calculating engine- values for given nozzle performance input parameters(Manual input or from CEA File), fitting tank size and arrangement to the required amount of propellant, determining shift in cg over the course of motor operation and even accounting for change in rocket performance due to shift in ambient pressure. Due to hardcoded input constraints in terms of OpenRocket demanding a fixed Thrust Curve with respect to time, fitting of the Altitude Compensation has to be done by an iterative process.

Input Variables

The following chapter gives a short overview of the parameter and variables, that can be changed by the user in order to generate the desired rocket motor.

datainput: ('F' for direct input from CEA File, 'M' for manual Input)

sw (dictionary): This Dictionary contains the exhaust gas composition of the rocket engine ('Name': Mole/Mass-fraction). Supports Mole as well as Mass Fractions and every substance, that is contained in CoolProp.

ratioflag: Can be set to either mass or mole and has to be set corresponding to the Fraction Type of the sw- Dictionary.

ofr: The Ratio of Oxidizer mass flow to Fuel mass flow.

Oxidizer: Has to be set to the used Oxidizer Type ('Name'). Supports all substances of CoolProp.

Fuel: Has to be set to the used Fuel Type ('Name'). Supports all substances of CoolProp.

Tch: Adiabatic Flame Temperature at given O/F- Ratio

Pch: Pressure in Combustion Chamber

Ps: Reference Pressure of Engine Sea Level Thrust Data (Default Setting to standard atmospheric pressure at sea level)

overexp: Defines the fraction of burnout altitude, to which the Nozzle expansion ratio is optimized (hopt/hmax)

Thrust: Thrust of the Rocket Engine at Reference Pressure (Default Setting to standard atmospheric pressure at sea level)

burntime: Total engine burn time

vch: expected flow velocity inside the combustion chamber (Default setting to 40 m/s)

ispkor: Corrects isp for friction- and heat losses, can be set between 1(100% of theoretical performance) and 0 (0% of theoretical performance)(Default setting to 1).

cells: Amount of Nodes for discretizing along the nozzle pressure drop

inputfile: Name of OpenRocket simulation output data (.csv)

stripfactor: Reduces Input Data Points (datapointsnew=datapointsold/stripfactor)

dt: Diameter of Rocket tanks (cylinder assumed)

mtl: mass per unit length of tank structure

deadff: percentage of dead weight fuel

deadof: percentage of dead weight oxidizer

mar(list): List of mass arrangement. Component masses are added to list in the order, they appear in the rocket (from bottom to tip). At Fuel tank position, write 'F', at Oxidizer Tank Position, write 'O'.

lar(list): List of length arrangement. Component lengths are added to list the order, they appear in the rocket (from bottom to tip) At Fuel Tank position, write 'F', at Oxidizer Tank Position, write 'O'.

Ttankf: Temperature of Fuel at storage condition

Ptankf: Pressure of Fuel at storage condition

Ttanko: Temperature of Oxidizer at storing condition

Qtanko: gas phase fraction of Oxidizer at initial storing condition

cox: Defines, if Dead Oxidizer Mass is treated as liquid or as gas ('g' for gaseous, 'l' for liquid)

cf: Defines, if Dead Fuel Mass is treated as liquid or as gas ('g' for gaseous, 'l' for liquid)

outputfile: Defines name of thrustcurve output sheet ('Name.rse')

enginename: Name of the rocket engine ('Name')

Prod: Name of engine manufacturer

Ddis: Diameter, hat is displayed at the simulation program

am: Switch for mass auto calculation, '1', for mass auto calculation, '0' for using mass data points of file (Default setting to '0').

Operation Procedure

Requirements: Java Platform, Anaconda with Python 3.6, Spyder, Coolprop

All required program components, including OpenRocket 15.03 are included within the program folder

Thrust Curve Generation:

Launch Spyder and open up inputfile.py and outputfile.py.

Manual Input: On the input file, all previously mentioned engine parameters are listed and can be altered.

CEA File Input: Program uses CEA.txt outputfile. Only one performance calculation permitted in the input sheet)

In Order to generate the thrust curve, change to the output file and run the script. Notifications in the output console should confirm successful thrust curve generation. The thrust curve file should now be saved in the program folder under the predefined name.

Implementing Engine in OpenRocket:

If the engine file is used the first time, the thrust curve needs to be imported to OpenRocket. This can be done at the setup- menu by entering the file path of the engine file. This is only necessary, if the name of the file changes. Thrust curve updates and changes do not need to be initialized again after altering the data file.

The Engine can now be used and found in the motor database under its manufacturer- and enginename just like a commercially implemented motor. **When the thrust curve file is updated**, in order to note the change, **OpenRocket has to be closed and restarted** again.

Altitude Compensation curve fitting:

In order to fit the engine thrust curve to the present flight path, an iterative system has to be used. After simulating the flight, simulation data has to be exported as 'Name.csv' - file. This file shall consist of time, heighth and ambient pressure and has to be saved under the name, that is stated at the input file. After that, close OpenRocket and update the thrustcurve by running the script. This method usually converges after three iterations.

Engine Model

In order to adapt and change nozzle performance and expansion ratio, a simple nozzle calculator is built into the script. This calculator assumes isentropic flow, therefore neglecting efficiency losses due to friction and heat transfer, but including the effect of change in fluid properties. The expansion pressure domain is determined and discretized by splitting it into several control volumes.

Conservation of Mass, Energy and Entropy are applied to each control volume. While Values for Enthalpy, density and velocity at the boundaries can be calculated exactly, specific heat ratio is considered a cell- centered value. Therefore, a upwind scheme (specific heat ratio is constant over control volume and equals the value at the entry face) is introduced. This results in an explicit formulation, that, considering sufficiently fine mesh, is fairly accurate, fast in calculation and easy to implement. With given boundary condition at the combustion chamber, values are calculated in flow direction of the nozzle. Using the pressure gradient as domain of discretization allows technically the use of only Energy and Entropy equation for performance analysis. Conservation of mass is used to calculate the corresponding area and can be either implemented at certain points, where flow area values are required, or ignored.

Altitude Compensation

Variation in Thrust due to change in ambient pressure during ascent is provided by the basic thrust correction formula provided by Huzel&Huang (Chapter 1, Page 2). The engine is optimized for a certain altitude by the Nozzle calculator and the key performance parameters are fed into the quoted thrust correction formula. Data points for ambient pressure are provided by the OpenRocket simulation file. The fact, that Thrust curves have to be hardcoded into the engine sheet and the lack of internal compensation equations in OpenRocket makes an iterative fitting process for the ambient pressure and respective the engine Thrust over time necessary. This is accomplished by alternating OpenRocket Simulation and Thrust Curve Calculation with the acquired altitude data.

Tank Modeling and CG Calculation

For the sake of simplicity, oxidizer and fuel tanks are modeled as cylinders. Volume and dimensions are automatically calculated and updated according to system operation requirements. Tank dry mass and dead propellant mass are also adapted to tank sizing. Positioning and Mass of other components, such as plumbing, injector and pressure vessels can be arranged and defined, as required for the system. Influence of dead propellant weight on the center of gravity can either be set to gaseous, assuming uniform distribution, and liquid, assuming mass accumulation at the bottom of the tank. Position of Motor CG is calculated with respect to burntime and added to the engine file. The calculation of the center of gravity is one- dimensional and uses standard definition of cg for calculation.

Outputfile

As output formate, the rocksim engine file (.rse) type is used. This file is more complicated, as the standard .eng- file but gives the opportunity to add shift in center of gravity and unequal mass consumption to the engine definition. These features are vital for simulating liquid or hybrid rockets. Drafts for the .rse file system can be found at the following link: http://wiki.openrocket.info/RSE_File

Current Project State (11.09.2018)

Even though initial trials have been successful, the program is still in early development phase. Therefore, calculation errors can not be excluded. Further, the program only contains the model for liquid bipropellant rockets, hybrid modelling is going to be included in the future. For any further questions, please contact **Mr. Alexander Sebo**, Chief Engineer at TXV.