

In order to pass the whole set of 19 args to the script one must group in two main categories: 1) 'fixed' values, i.e., not changing from one pass to another, for a given engine, and 2) values that change with each pass according to actual engine-powertrain conditions. In order to make the things easier, i.e., to decrease the call effort the first category is to be stored within a list, built once and passed with every call.

The 'fixed' values are as following (passed within a list exactly in this order):

- 1) the transmission final ratio 'xi_f',
- 2) the slipping factor 's_f',
- 3) the wheel rolling radius 'r_d',
- 4) the engine maximum speed 'n_max',
- 5) the engine maximum output, 'P_max',
- 6) the engine type, 'type',
- 7) the transmission overall efficiency, 'eta_t',
- 8) the engine maximum efficiency, 'eta_max',
- 9) the vehicle mass, 'm_a',
- 10) the rolling drag coefficient, 'c_r',
- 11) the wind drag coefficient, 'C_d',
- 12) the vehicle frontal area, 'A_f',
- 13) the air density, 'ro_a',
- 14) the fuel burning value, 'Q_f' (in Joule/liter),
- 15) the fuel density, 'ro_f'.

The second category ('volatile' values, in this order) is:

- 1) the vehicle initial speed, 'v_init',
- 2) the gearbox ratio, 'xi_g',
- 3) the acceleration, 'a',
- 4) the time, 't'.

At RL agent level the specific fuel consumption script is to be invoked in an API-style, presented in the 'sfc_call.py' file.

If, by passed args, vehicle speed leads to an engine speed greater than maximum possible the program displays a warning and exists. Same is true for power/output.

I recommend to read every function and method *docstring* carefully. Also every comment is meant for further clarification.