

Task 3

The simulation loop runs until `TERMINATE_SIM` becomes “true”, or until an event occurs. An event is if the EDL crashes or lands. Variable ‘fun’ is assigned to the dy/dt vector based on input variables ‘t’ and ‘y’. The dy/dt vector holds information about the EDL acceleration, EDL velocity, change of total mass of rocket due to propellant expelled, error of EDL velocity, error of EDL position, rover acceleration, and rover velocity. From the `scipy.integrate` module, `solve_ivp` is imported to solve the ordinary differential equation. DOP853 is an 8th order explicit RK method; its step size is controlled (to 0.1) to help with accuracy. ‘y0’ contains the initial conditions of the EDL system like the velocity, altitude, and initial fuel mass in all rockets. The ivp solution is stored to variable ‘sol’ which uses initial conditions ‘y0’.

The EDL has 9 phases:

0. Reached altitude to eject heat shield
1. Reached altitude to eject parachute
2. Reached altitude to turn on rockets
3. Reached altitude to turn on crane
4. Out of fuel --> $y(3) \leq 0$. Terminal. Direction: -1.
5. EDL System crashed at zero altitude
6. Reached speed at which controlled descent is required
7. Reached altitude at which position control is required
8. Rover has touched down on surface of Mars

The `update_edl_state` function specifies the initial conditions for each phase of the EDL. The physics simulation is different for each phase so the IVP must iterate through each phase.

The time solutions for the IVP are assigned to ‘t_part’ and the altitude solutions from the IVP are assigned to ‘Y_part’. ‘TE’ is a variable that gets assigned to the array of times that correspond to events (ex. heat shield pops off). ‘YE’ is assigned to an array of altitudes corresponding to events.

The loop then updates the EDL state for that iteration. The time span is also updated based on the last iteration so the time span does not always start from 0s every iteration.

A time value from the IVP solver is added to the time array ‘T’ every iteration. An altitude value from the IVP solver is added to the altitude array ‘Y’ after every iteration.

The loop checks if the maximum time has been exceeded, and if so, the loop will be terminated.

