File: define\_rovers.py

Function: define\_rover\_1

1. Dictionary: wheel
   1. 'radius'
      1. Default: 0.30
      2. Unit: [m]
      3. radius of wheel
   2. 'mass':
      1. Default: 1
      2. Unit: [kg]
      3. mass of wheel
2. dictionary: speed\_reducer
   1. 'type'
      1. Default: ‘reverted'
      2. Unit: [string]
      3. input and output gears are co-axial
   2. 'diam\_pinion'
      1. Default :0.04
      2. Unit: [m]
      3. diameter of pinion gear
   3. 'diam\_gear'
      1. Default: 0.07,
      2. Unit: [m]
      3. diameter of main gear
   4. 'mass'
      1. default:1.5
      2. unit: [kg]
      3. mass of speed reducer
3. dictionary: motor
   1. 'torque\_stall'
      1. Default: :170
      2. Unit: [Nm]
      3. torque needed to stall motor
   2. 'torque\_noload'
      1. default: 0
      2. unit: [Nm]
      3. torque applied by motor with zero load
   3. 'speed\_noload'
      1. Default:3.80
      2. Unit: [rad/s]
      3. speed of motor with zero load
   4. 'mass'
      1. Default:5.0
      2. Unit: [kg]
      3. mass of motor
4. dictionary: chassis
   1. 'mass':
      1. Default: 659
      2. Unit: [kg]
      3. mass of chassis
5. dictionary: science\_payload
   1. 'mass'
      1. default:75
      2. unit: [kg]
      3. mass of science payload
6. dictionary: power\_subsys
   1. 'mass'
      1. Default: 90
      2. Unit: [kg]
      3. mass of power sub-system
7. dictionary: wheel\_assembly
   1. 'wheel'
      1. Default : wheel,
      2. Unit: [dictionary]
      3. wheel assembly has wheel
   2. 'speed\_reducer'
      1. Default :speed\_reducer,
      2. Unit: [dictioanry]
      3. wheel assembly has speed reducer
   3. 'motor'
      1. Default:motor}
      2. Unit: [dictionary]
      3. wheel assembly has motor
8. dictionary: rover =
   1. 'wheel\_assembly'
      1. default:wheel\_assembly,
      2. unit: [dictionary]
      3. rover has wheel assembly
   2. 'chassis':
      1. Default: chassis
      2. unit [dictionary]
      3. rover has chassis
   3. 'science\_payload'
      1. default:science\_payload
      2. unit: [dictionary]
      3. rover has science payload
   4. 'power\_subsys'
      1. Default:power\_subsys
      2. Unit: [dictionary]
      3. rover has power sub-system
9. dictionary: planet
   1. 'g'
      1. default:3.72
      2. unit: [m/s^2]
      3. gravity of mars

**Function: define\_rover\_2**

1. Dictionary: wheel
   1. 'radius'
      1. Default: 0.30
      2. Unit: [m]
      3. radius of wheel
   2. 'mass':
      1. Default: 2
      2. Unit: [kg]
      3. mass of wheel
2. dictionary: speed\_reducer
   1. 'type'
      1. Default: ‘reverted'
      2. Unit: [string]
      3. input and output gears are co-axial
   2. 'diam\_pinion'
      1. Default :0.04
      2. Unit: [m]
      3. diameter of pinion gear
   3. 'diam\_gear'
      1. Default: 0.06,
      2. Unit: [m]
      3. diameter of main gear
   4. 'mass'
      1. default:1.5
      2. unit: [kg]
      3. mass of speed reducer
3. dictionary: motor
   1. 'torque\_stall'
      1. Default: :180
      2. Unit: [Nm]
      3. torque needed to stall motor
   2. 'torque\_noload'
      1. default: 0
      2. unit: [Nm]
      3. torque applied by motor with zero load
   3. 'speed\_noload'
      1. Default:3.70
      2. Unit: [rad/s]
      3. speed of motor with zero load
   4. 'mass'
      1. Default:5.0
      2. Unit: [kg]
      3. mass of motor
4. dictionary: chassis
   1. 'mass':
      1. Default: 659
      2. Unit: [kg]
      3. mass of chassis
5. dictionary: science\_payload
   1. 'mass'
      1. default:75
      2. unit: [kg]
      3. mass of science payload
6. dictionary: power\_subsys
   1. 'mass'
      1. Default: 90
      2. Unit: [kg]
      3. mass of power sub-system
7. dictionary: wheel\_assembly
   1. 'wheel'
      1. Default : wheel,
      2. Unit: [dictionary]
      3. wheel assembly has wheel
   2. 'speed\_reducer'
      1. Default :speed\_reducer,
      2. Unit: [dictioanry]
      3. wheel assembly has speed reducer
   3. 'motor'
      1. Default:motor}
      2. Unit: [dictionary]
      3. wheel assembly has motor
8. dictionary: rover =
   1. 'wheel\_assembly'
      1. default:wheel\_assembly,
      2. unit: [dictionary]
      3. rover has wheel assembly
   2. 'chassis':
      1. Default: chassis
      2. unit [dictionary]
      3. rover has chassis
   3. 'science\_payload'
      1. default:science\_payload
      2. unit: [dictionary]
      3. rover has science payload
   4. 'power\_subsys'
      1. Default:power\_subsys
      2. Unit: [dictionary]
      3. rover has power sub-system
9. dictionary: planet
   1. 'g'
      1. default:3.72
      2. unit: [m/s^2]
      3. gravity of mars

**Function: define\_rover\_3**

1. Dictionary: wheel
   1. 'radius'
      1. Default: 0.30
      2. Unit: [m]
      3. radius of wheel
   2. 'mass':
      1. Default: 2
      2. Unit: [kg]
      3. mass of wheel
2. dictionary: speed\_reducer
   1. 'type'
      1. Default: ‘standard
      2. Unit: [string]
      3. input and output gears are co-axial
   2. 'diam\_pinion'
      1. Default :0.04
      2. Unit: [m]
      3. diameter of pinion gear
   3. 'diam\_gear'
      1. Default: 0.06,
      2. Unit: [m]
      3. diameter of main gear
   4. 'mass'
      1. default:1.5
      2. unit: [kg]
      3. mass of speed reducer
3. dictionary: motor
   1. 'torque\_stall'
      1. Default: :180
      2. Unit: [Nm]
      3. torque needed to stall motor
   2. 'torque\_noload'
      1. default: 0
      2. unit: [Nm]
      3. torque applied by motor with zero load
   3. 'speed\_noload'
      1. Default:3.70
      2. Unit: [rad/s]
      3. speed of motor with zero load
   4. 'mass'
      1. Default:5.0
      2. Unit: [kg]
      3. mass of motor
4. dictionary: chassis
   1. 'mass':
      1. Default: 659
      2. Unit: [kg]
      3. mass of chassis
5. dictionary: science\_payload
   1. 'mass'
      1. default:75
      2. unit: [kg]
      3. mass of science payload
6. dictionary: power\_subsys
   1. 'mass'
      1. Default: 90
      2. Unit: [kg]
      3. mass of power sub-system
7. dictionary: wheel\_assembly
   1. 'wheel'
      1. Default : wheel,
      2. Unit: [dictionary]
      3. wheel assembly has wheel
   2. 'speed\_reducer'
      1. Default :speed\_reducer,
      2. Unit: [dictioanry]
      3. wheel assembly has speed reducer
   3. 'motor'
      1. Default:motor}
      2. Unit: [dictionary]
      3. wheel assembly has motor
8. dictionary: rover =
   1. 'wheel\_assembly'
      1. default:wheel\_assembly,
      2. unit: [dictionary]
      3. rover has wheel assembly
   2. 'chassis':
      1. Default: chassis
      2. unit [dictionary]
      3. rover has chassis
   3. 'science\_payload'
      1. default:science\_payload
      2. unit: [dictionary]
      3. rover has science payload
   4. 'power\_subsys'
      1. Default:power\_subsys
      2. Unit: [dictionary]
      3. rover has power sub-system
9. dictionary: planet
   1. 'g'
      1. default:3.72
      2. unit: [m/s^2]
      3. gravity of mars

**Function: define\_rover\_4**

1. Dictionary: wheel
   1. 'radius'
      1. Default: 0.20
      2. Unit: [m]
      3. radius of wheel
   2. 'mass':
      1. Default: 2
      2. Unit: [kg]
      3. mass of wheel
2. dictionary: speed\_reducer
   1. 'type'
      1. Default: ‘reverted
      2. Unit: [string]
      3. input and output gears are co-axial
   2. 'diam\_pinion'
      1. Default :0.04
      2. Unit: [m]
      3. diameter of pinion gear
   3. 'diam\_gear'
      1. Default: 0.06,
      2. Unit: [m]
      3. diameter of main gear
   4. 'mass'
      1. default:1.5
      2. unit: [kg]
      3. mass of speed reducer
3. dictionary: motor
   1. 'torque\_stall'
      1. Default: :165
      2. Unit: [Nm]
      3. torque needed to stall motor
   2. 'torque\_noload'
      1. default: 0
      2. unit: [Nm]
      3. torque applied by motor with zero load
   3. 'speed\_noload'
      1. Default:3.85
      2. Unit: [rad/s]
      3. speed of motor with zero load
   4. 'mass'
      1. Default:5.0
      2. Unit: [kg]
      3. mass of motor
4. dictionary: chassis
   1. 'mass':
      1. Default: 674
      2. Unit: [kg]
      3. mass of chassis
5. dictionary: science\_payload
   1. 'mass'
      1. default:80
      2. unit: [kg]
      3. mass of science payload
6. dictionary: power\_subsys
   1. 'mass'
      1. Default: 100
      2. Unit: [kg]
      3. mass of power sub-system
7. dictionary: wheel\_assembly
   1. 'wheel'
      1. Default : wheel,
      2. Unit: [dictionary]
      3. wheel assembly has wheel
   2. 'speed\_reducer'
      1. Default :speed\_reducer,
      2. Unit: [dictioanry]
      3. wheel assembly has speed reducer
   3. 'motor'
      1. Default:motor}
      2. Unit: [dictionary]
      3. wheel assembly has motor
8. dictionary: rover =
   1. 'wheel\_assembly'
      1. default:wheel\_assembly,
      2. unit: [dictionary]
      3. rover has wheel assembly
   2. 'chassis':
      1. Default: chassis
      2. unit [dictionary]
      3. rover has chassis
   3. 'science\_payload'
      1. default:science\_payload
      2. unit: [dictionary]
      3. rover has science payload
   4. 'power\_subsys'
      1. Default:power\_subsys
      2. Unit: [dictionary]
      3. rover has power sub-system
9. dictionary: planet
   1. 'g'
      1. default:3.72
      2. unit: [m/s^2]
      3. gravity of mars
10. Dictionary: end\_event
    1. ‘max\_distance’
       1. Default: 10
       2. Unit: [m]
       3. maximum distance before stopping
    2. ‘max\_time’
       1. Default: 10000
       2. Unit: [s]
       3. Maximum time before stopping
    3. ‘min\_velocity’
       1. Default: 0.01
       2. Unit: [m/s]
       3. Minimum velocity rover moves

File: define\_edl\_system.py

function: define\_edl\_system\_1

1. Dictionary: parachute
   1. 'deployed'
      1. Default: True,
      2. Type: boolean
      3. true means it has been deployed but not ejected
   2. 'ejected'
      1. Default: False
      2. Type: boolean
      3. true means parachute no longer is attached to system
   3. 'diameter'
      1. Default: 16.25
      2. Unit: [m]
      3. Diameter of parachute
   4. 'Cd'
      1. default : 0.615,
      2. unit [-]
      3. non-dimensional drag coefficient of heat shield
   5. 'mass'
      1. default: 185.0
      2. unit [kg]
      3. mass of parachute
2. dictionary: rocket
   1. 'on'
      1. default: False
      2. unit: Boolean
      3. rocket is by default off
   2. 'structure\_mass'
      1. default : 8.0,
      2. unit: [kg]
      3. mass of everything not fuel
   3. 'initial\_fuel\_mass'
      1. default: 230.0,
      2. unit: [kg]
      3. intial mass of fuel
   4. 'fuel\_mass'
      1. default: 230.0,
      2. unit: [kg]
      3. current fuel mass (<= initial)
   5. 'effective\_exhaust\_velocity'
      1. default : 4500.0
      2. unit: [m/s]
      3. velocity of exhaust
   6. 'max\_thrust'
      1. default: 3100.0,
      2. unit: [N]
      3. maximum thrust force of rocket
   7. 'min\_thrust'
      1. default : 40.0}
      2. unit: [N]
      3. minimum thurst force of rocket
3. dictionary: speed\_control =
   1. 'on'
      1. Default: False
      2. Unit: boolean
      3. indicates whether control mode is activated
   2. 'Kp'
      1. default: 2000,
      2. unit: constant
      3. proportional gain term
   3. 'Kd'
      1. Default : 20
      2. Unit: constant
      3. derivative gain term
   4. 'Ki'
      1. Default : 50
      2. Unit: constant
      3. integral gain term
   5. 'target\_velocity'
      1. default: -3.0
      2. unit : [m/s]
      3. desired descent speed
4. dictionary: position\_control
   1. 'on'
      1. default: False,
      2. unit: boolean
      3. indicates whether control mode is activated
   2. 'Kp'
      1. default : 2000,
      2. unit: constant
      3. proportional gain term
   3. 'Kd'
      1. default: 1000,
      2. unit: constant
      3. derivative gain term
   4. 'Ki'
      1. Default: 50,
      2. Unit: constant
      3. integral gain term
   5. 'target\_altitude'
      1. defaut: 7.6
      2. unit: [m]
      3. needs to reflect the sky crane cable length
5. dictionary: sky\_crane
   1. 'on'
      1. default :False
      2. unit: Boolean
      3. true means lowering rover mode
   2. 'danger\_altitude'
      1. default : 4.5,
      2. unit: [m]
      3. altitude at which considered too low for safe rover touch down
   3. 'danger\_speed'
      1. default: -1.0,
      2. unit: [m/s]
      3. speed at which rover would impact to hard on surface
   4. 'mass'
      1. default: 35.0,
      2. unit: [kg]
   5. 'area'
      1. default: 16.0,
      2. unit: [m^2]
      3. frontal area for drag calculations
   6. 'Cd'
      1. default: 0.9
      2. unit: constant
      3. [-] coefficient of drag
   7. 'max\_cable'
      1. default : 7.6,
      2. unit: [m]
      3. max length of cable for lowering rover
   8. 'velocity'
      1. default: -0.1
      2. [m]
      3. speed at which sky crane lowers rover
6. dictionary: heat\_shield
   1. 'ejected'
      1. Default: False
      2. Unit: boolean
      3. true means heat shield has been ejected from system
   2. 'mass'
      1. default : 225.0
      2. unit: [kg]
      3. mass of heat shield
   3. 'diameter'
      1. default : 4.5,
      2. unit: [m]
      3. diameter of heat shield
   4. 'Cd'
      1. default : 0.35
      2. [-] non-dimensional drag coefficient of heat shield
7. Dictionary: edl\_system
   1. 'altitude'
      1. Default : np.NaN,
      2. Unit: NaN
      3. Altitude of EDL; system state variable that is updated throughout simulation
   2. 'velocity'
      1. default: np.NaN,
      2. unit: NaN
      3. velocity of EDL; state variable that is updated throughout simulation
   3. 'num\_rockets'
      1. default: 8,
      2. unit: integer
      3. system level parameter
   4. 'volume'
      1. default :150,
      2. unit: [m^3]
      3. volume of EDL; system level parameter
   5. 'parachute'
      1. default : parachute,
      2. unit: dict
      3. variable assigned to parachute dictionary
   6. 'heat\_shield'
      1. default : heat\_shield,
      2. unit: dict
      3. variable assigned to heat\_shield dictionary
   7. 'rocket'
      1. default : rocket
      2. unit: dict
      3. variable assigned to rocket dictionary
   8. 'speed\_control'
      1. default : speed\_control
      2. unit: dict
      3. variable assigned to speed\_control dictionary
   9. 'position\_control'
      1. default : position\_control
      2. unit: dict
      3. variable assigned to position\_control dictionary
   10. 'sky\_crane'
       1. default : sky\_crane
       2. unit: dict
       3. variable assigned to sky\_crane dictionary
   11. 'rover'
       1. default : rover
       2. unit: dict
       3. variable assigned to rover dictionary

File: define\_mission\_events.py

Function: define\_mission\_events

1. dictionary: mission\_events
   1. 'alt\_heatshield\_eject'
      1. Default: 8000
      2. Unit: [m]
      3. altitude at which heat shield is ejected
   2. 'alt\_parachute\_eject'
      1. Default: unit : 900,
      2. Unit: [m]
      3. altitude at which parachute is ejected
   3. 'alt\_rockets\_on'
      1. default : 1800
      2. unit: [m]
      3. altitude at which power descent sequence is initiated using variable-thrust solid rockets
   4. 'alt\_skycrane\_on'
      1. default : 7.6
      2. unit: [m]
      3. altitude appropriate for sky crane operation

File: define\_planet.py

Function: define\_planet

1. dictionary: high\_altitude
   1. 'temperature'
      1. Default: lambda altitude: -23.4 - 0.00222\*altitude
      2. Unit: [C]
      3. Temperature at high altitudes
   2. 'pressure'
      1. Default: lambda altitude: 0.699\*np.exp(-0.00009\*altitude)}
      2. unit [KPa]

low\_altitude = {'temperature' : lambda altitude: -31 - 0.000998\*altitude, # [C]

'pressure' : lambda altitude: 0.699\*np.exp(-0.00009\*altitude)} # [KPa]

density = lambda temperature, pressure: pressure/(0.1921\*(temperature+273.15)) # [kg/m^3]

mars = {'g' : -3.72, # m/s^2]

'altitude\_threshold' : 7000, # [m]

'low\_altitude' : low\_altitude,

'high\_altitude' : high\_altitude,

'density' : density}