File: define_rovers.py

Function: define_rover_1

I. Dictionary: wheel
    a. 'radius'
        i. Default: 0.30
        ii. Unit: [m]
        iii. radius of wheel
    b. 'mass':
        i. Default: 1
        ii. Unit: [kg]
        iii. mass of wheel
II. dictionary: speed_reducer
    a. 'type'
        i. Default: ‘reverted'
        ii. Unit: [string]
        iii. input and output gears are co-axial
    b. 'diam_pinion'
        i. Default :0.04
        ii. Unit: [m]
        iii. diameter of pinion gear
    c. 'diam_gear'
        i. Default: 0.07,
        ii. Unit: [m]
        iii. diameter of main gear
    d. 'mass'
        i. default:1.5
        ii. unit: [kg]
        iii. mass of speed reducer
III. dictionary: motor
    a. 'torque_stall'
        i. Default: :170
        ii. Unit: [Nm]
        iii. torque needed to stall motor
    b. 'torque_noload'
        i. default: 0
        ii. unit: [Nm]
        iii. torque applied by motor with zero load
    c. 'speed_noload'
        i. Default:3.80
        ii. Unit: [rad/s]
        iii. speed of motor with zero load
    d. 'mass'
        i. Default:5.0

        - ii. Unit: [kg]
        - iii. mass of motor
- IV. dictionary: chassis
    - a. 'mass':
        - i. Default: 659
        - ii. Unit: [kg]
        - iii. mass of chassis
- V. dictionary: science_payload
    - a. 'mass'
        - i. default:75
        - ii. unit: [kg]
        - iii. mass of science payload
- VI. dictionary: power_subsys
    - a. 'mass'
        - i. Default: 90
        - ii. Unit: [kg]
        - iii. mass of power sub-system
- VII. dictionary: wheel_assembly
    - a. 'wheel'
        - i. Default : wheel,
        - ii. Unit: [dictionary]
        - iii. wheel assembly has wheel
    - b. 'speed_reducer'
        - i. Default :speed_reducer,
        - ii. Unit: [dictioanry]
        - iii. wheel assembly has speed reducer
    - c. 'motor'
        - i. Default:motor}
        - ii. Unit: [dictionary]
        - iii. wheel assembly has motor
- VIII. dictionary: rover =
    - a. 'wheel_assembly'
        - i. default:wheel_assembly,
        - ii. unit: [dictionary]
        - iii. rover has wheel assembly
    - b. 'chassis':
        - i. Default: chassis
        - ii. unit [dictionary]
        - iii. rover has chassis
    - c. 'science_payload'
        - i. default:science_payload
        - ii. unit: [dictionary]
        - iii. rover has science payload
    - d. 'power_subsys'

        i. Default:power_subsys
        ii. Unit: [dictionary]
        iii. rover has power sub-system
IX. dictionary: planet
    a. 'g'
        i. default:3.72
        ii. unit: [m/s^2]
        iii. gravity of mars

Function: define_rover_2

I. Dictionary: wheel
    a. 'radius'
        i. Default: 0.30
        ii. Unit: [m]
        iii. radius of wheel
    b. 'mass':
        i. Default: 2
        ii. Unit: [kg]
        iii. mass of wheel
II. dictionary: speed_reducer
    a. 'type'
        i. Default: ‘reverted'
        ii. Unit: [string]
        iii. input and output gears are co-axial
    b. 'diam_pinion'
        i. Default :0.04
        ii. Unit: [m]
        iii. diameter of pinion gear
    c. 'diam_gear'
        i. Default: 0.06,
        ii. Unit: [m]
        iii. diameter of main gear
    d. 'mass'
        i. default:1.5
        ii. unit: [kg]
        iii. mass of speed reducer
III. dictionary: motor
    a. 'torque_stall'
        i. Default: :180
        ii. Unit: [Nm]
        iii. torque needed to stall motor
    b. 'torque_noload'
        i. default: 0
        ii. unit: [Nm]

    - iii. torque applied by motor with zero load
  - c. 'speed_noload'
    - i. Default:3.70
    - ii. Unit: [rad/s]
    - iii. speed of motor with zero load
  - d. 'mass'
    - i. Default:5.0
    - ii. Unit: [kg]
    - iii. mass of motor
- IV. dictionary: chassis
  - a. 'mass':
    - i. Default: 659
    - ii. Unit: [kg]
    - iii. mass of chassis
- V. dictionary: science_payload
  - a. 'mass'
    - i. default:75
    - ii. unit: [kg]
    - iii. mass of science payload
- VI. dictionary: power_subsys
  - a. 'mass'
    - i. Default: 90
    - ii. Unit: [kg]
    - iii. mass of power sub-system
- VII. dictionary: wheel_assembly
  - a. 'wheel'
    - i. Default : wheel,
    - ii. Unit: [dictionary]
    - iii. wheel assembly has wheel
  - b. 'speed_reducer'
    - i. Default :speed_reducer,
    - ii. Unit: [dictioanry]
    - iii. wheel assembly has speed reducer
  - c. 'motor'
    - i. Default:motor}
    - ii. Unit: [dictionary]
    - iii. wheel assembly has motor
- VIII. dictionary: rover =
  - a. 'wheel_assembly'
    - i. default:wheel_assembly,
    - ii. unit: [dictionary]
    - iii. rover has wheel assembly
  - b. 'chassis':
    - i. Default: chassis

    - ii. unit [dictionary]
    - iii. rover has chassis
  - c. 'science_payload'
    - i. default:science_payload
    - ii. unit: [dictionary]
    - iii. rover has science payload
  - d. 'power_subsys'
    - i. Default:power_subsys
    - ii. Unit: [dictionary]
    - iii. rover has power sub-system
- IX. dictionary: planet
  - a. 'g'
    - i. default:3.72
    - ii. unit: [m/s^2]
    - iii. gravity of mars

Function: define_rover_3

- I. Dictionary: wheel
  - a. 'radius'
    - i. Default: 0.30
    - ii. Unit: [m]
    - iii. radius of wheel
  - b. 'mass':
    - i. Default: 2
    - ii. Unit: [kg]
    - iii. mass of wheel
- II. dictionary: speed_reducer
  - a. 'type'
    - i. Default: ‘standard
    - ii. Unit: [string]
    - iii. input and output gears are co-axial
  - b. 'diam_pinion'
    - i. Default :0.04
    - ii. Unit: [m]
    - iii. diameter of pinion gear
  - c. 'diam_gear'
    - i. Default: 0.06,
    - ii. Unit: [m]
    - iii. diameter of main gear
  - d. 'mass'
    - i. default:1.5
    - ii. unit: [kg]
    - iii. mass of speed reducer
- III. dictionary: motor

- a. 'torque_stall'
  - i. Default: :180
  - ii. Unit: [Nm]
  - iii. torque needed to stall motor
- b. 'torque_noload'
  - i. default: 0
  - ii. unit: [Nm]
  - iii. torque applied by motor with zero load
- c. 'speed_noload'
  - i. Default:3.70
  - ii. Unit: [rad/s]
  - iii. speed of motor with zero load
- d. 'mass'
  - i. Default:5.0
  - ii. Unit: [kg]
  - iii. mass of motor

- IV. dictionary: chassis
  - a. 'mass':
    - i. Default: 659
    - ii. Unit: [kg]
    - iii. mass of chassis
- V. dictionary: science_payload
  - a. 'mass'
    - i. default:75
    - ii. unit: [kg]
    - iii. mass of science payload
- VI. dictionary: power_subsys
  - a. 'mass'
    - i. Default: 90
    - ii. Unit: [kg]
    - iii. mass of power sub-system
- VII. dictionary: wheel_assembly
  - a. 'wheel'
    - i. Default : wheel,
    - ii. Unit: [dictionary]
    - iii. wheel assembly has wheel
  - b. 'speed_reducer'
    - i. Default :speed_reducer,
    - ii. Unit: [dictioanry]
    - iii. wheel assembly has speed reducer
  - c. 'motor'
    - i. Default:motor}
    - ii. Unit: [dictionary]
    - iii. wheel assembly has motor

VIII. dictionary: rover =
  a. 'wheel_assembly'
    i. default:wheel_assembly,
    ii. unit: [dictionary]
    iii. rover has wheel assembly
  b. 'chassis':
    i. Default: chassis
    ii. unit [dictionary]
    iii. rover has chassis
  c. 'science_payload'
    i. default:science_payload
    ii. unit: [dictionary]
    iii. rover has science payload
  d. 'power_subsys'
    i. Default:power_subsys
    ii. Unit: [dictionary]
    iii. rover has power sub-system
IX. dictionary: planet
  a. 'g'
    i. default:3.72
    ii. unit: [m/s^2]
    iii. gravity of mars

Function: define_rover_4

I. Dictionary: wheel
  a. 'radius'
    i. Default: 0.20
    ii. Unit: [m]
    iii. radius of wheel
  b. 'mass':
    i. Default: 2
    ii. Unit: [kg]
    iii. mass of wheel
II. dictionary: speed_reducer
  a. 'type'
    i. Default: ‘reverted
    ii. Unit: [string]
    iii. input and output gears are co-axial
  b. 'diam_pinion'
    i. Default :0.04
    ii. Unit: [m]
    iii. diameter of pinion gear
  c. 'diam_gear'

    - i. Default: 0.06,
    - ii. Unit: [m]
    - iii. diameter of main gear
  - d. 'mass'
    - i. default:1.5
    - ii. unit: [kg]
    - iii. mass of speed reducer
- III. dictionary: motor
  - a. 'torque_stall'
    - i. Default: :165
    - ii. Unit: [Nm]
    - iii. torque needed to stall motor
  - b. 'torque_noload'
    - i. default: 0
    - ii. unit: [Nm]
    - iii. torque applied by motor with zero load
  - c. 'speed_noload'
    - i. Default:3.85
    - ii. Unit: [rad/s]
    - iii. speed of motor with zero load
  - d. 'mass'
    - i. Default:5.0
    - ii. Unit: [kg]
    - iii. mass of motor
- IV. dictionary: chassis
  - a. 'mass':
    - i. Default: 674
    - ii. Unit: [kg]
    - iii. mass of chassis
- V. dictionary: science_payload
  - a. 'mass'
    - i. default:80
    - ii. unit: [kg]
    - iii. mass of science payload
- VI. dictionary: power_subsys
  - a. 'mass'
    - i. Default: 100
    - ii. Unit: [kg]
    - iii. mass of power sub-system
- VII. dictionary: wheel_assembly
  - a. 'wheel'
    - i. Default : wheel,
    - ii. Unit: [dictionary]
    - iii. wheel assembly has wheel

b. 'speed_reducer'

i. Default :speed_reducer,

ii. Unit: [dictioanry]

iii. wheel assembly has speed reducer

c. 'motor'

i. Default:motor}

ii. Unit: [dictionary]

iii. wheel assembly has motor

VIII. dictionary: rover =

a. 'wheel_assembly'

i. default:wheel_assembly,

ii. unit: [dictionary]

iii. rover has wheel assembly

b. 'chassis':

i. Default: chassis

ii. unit [dictionary]

iii. rover has chassis

c. 'science_payload'

i. default:science_payload

ii. unit: [dictionary]

iii. rover has science payload

d. 'power_subsys'

i. Default:power_subsys

ii. Unit: [dictionary]

iii. rover has power sub-system

IX. dictionary: planet

a. 'g'

i. default:3.72

ii. unit: [m/s^2]

iii. gravity of mars

X. Dictionary: end_event

a. ‘max_distance’

i. Default: 10

ii. Unit: [m]

iii. maximum distance before stopping

b. ‘max_time’

i. Default: 10000

ii. Unit: [s]

iii. Maximum time before stopping

c. ‘min_velocity’

i. Default: 0.01

ii. Unit: [m/s]

iii. Minimum velocity rover moves

File: define_edl_system.py

function: define_edl_system_1

I. Dictionary: parachute
   a. 'deployed'
      i. Default: True,
      ii. Type: boolean
      iii. true means it has been deployed but not ejected
   b. 'ejected'
      i. Default: False
      ii. Type: boolean
      iii. true means parachute no longer is attached to system
   c. 'diameter'
      i. Default: 16.25
      ii. Unit: [m]
      iii. Diameter of parachute
   d. 'Cd'
      i. default : 0.615,
      ii. unit [-]
      iii. non-dimensional drag coefficient of heat shield
   e. 'mass'
      i. default: 185.0
      ii. unit [kg]
      iii. mass of parachute
II. dictionary: rocket
   a. 'on'
      i. default: False
      ii. unit: Boolean
      iii. rocket is by default off
   b. 'structure_mass'
      i. default : 8.0,
      ii. unit: [kg]
      iii. mass of everything not fuel
   c. 'initial_fuel_mass'
      i. default: 230.0,
      ii. unit: [kg]
      iii. intial mass of fuel
   d. 'fuel_mass'

        i. default: 230.0,
        ii. unit: [kg]
        iii. current fuel mass (<= initial)
    e. 'effective_exhaust_velocity'
        i. default : 4500.0
        ii. unit: [m/s]
        iii. velocity of exhaust
    f. 'max_thrust'
        i. default: 3100.0,
        ii. unit: [N]
        iii. maximum thrust force of rocket
    g. 'min_thrust'
        i. default : 40.0}
        ii. unit: [N]
        iii. minimum thurst force of rocket

III. dictionary: speed_control =
    a. 'on'
        i. Default: False
        ii. Unit: boolean
        iii. indicates whether control mode is activated
    b. 'Kp'
        i. default: 2000,
        ii. unit: constant
        iii. proportional gain term
    c. 'Kd'
        i. Default : 20
        ii. Unit: constant
        iii. derivative gain term
    d. 'Ki'
        i. Default : 50
        ii. Unit: constant
        iii. integral gain term
    e. 'target_velocity'
        i. default: -3.0
        ii. unit : [m/s]
        iii. desired descent speed

IV. dictionary: position_control
    a. 'on'
        i. default: False,
        ii. unit: boolean
        iii. indicates whether control mode is activated
    b. 'Kp'
        i. default : 2000,
        ii. unit: constant

iii. proportional gain term

c. 'Kd'

i. default: 1000,

ii. unit: constant

iii. derivative gain term

d. 'Ki'

i. Default: 50,

ii. Unit: constant

iii. integral gain term

e. 'target_altitude'

i. defaut: 7.6

ii. unit: [m]

iii. needs to reflect the sky crane cable length

V. dictionary: sky_crane

a. 'on'

i. default :False

ii. unit: Boolean

iii. true means lowering rover mode

b. 'danger_altitude'

i. default : 4.5,

ii. unit: [m]

iii. altitude at which considered too low for safe rover touch down

c. 'danger_speed'

i. default: -1.0,

ii. unit: [m/s]

iii. speed at which rover would impact to hard on surface

d. 'mass'

i. default: 35.0,

ii. unit: [kg]

e. 'area'

i. default: 16.0,

ii. unit: [m^2]

iii. frontal area for drag calculations

f. 'Cd'

i. default: 0.9

ii. unit: constant

iii. [-] coefficient of drag

g. 'max_cable'

i. default : 7.6,

ii. unit: [m]

iii. max length of cable for lowering rover

h. 'velocity'

i. default: -0.1

ii. [m]

    - iii. speed at which sky crane lowers rover
- VI. dictionary: heat_shield
  - a. 'ejected'
    - i. Default: False
    - ii. Unit: boolean
    - iii. true means heat shield has been ejected from system
  - b. 'mass'
    - i. default : 225.0
    - ii. unit: [kg]
    - iii. mass of heat shield
  - c. 'diameter'
    - i. default : 4.5,
    - ii. unit: [m]
    - iii. diameter of heat shield
  - d. 'Cd'
    - i. default : 0.35
    - ii. [-] non-dimensional drag coefficient of heat shield
- VII. Dictionary: edl_system
  - a. 'altitude'
    - i. Default : np.NaN,
    - ii. Unit: NaN
    - iii. Altitude of EDL; system state variable that is updated throughout simulation
  - b. 'velocity'
    - i. default: np.NaN,
    - ii. unit: NaN
    - iii. velocity of EDL; state variable that is updated throughout simulation
  - c. 'num_rockets'
    - i. default: 8,
    - ii. unit: integer
    - iii. system level parameter
  - d. 'volume'
    - i. default :150,
    - ii. unit: [m^3]
    - iii. volume of EDL; system level parameter
  - e. 'parachute'
    - i. default : parachute,
    - ii. unit: dict
    - iii. variable assigned to parachute dictionary
  - f. 'heat_shield'
    - i. default : heat_shield,
    - ii. unit: dict
    - iii. variable assigned to heat_shield dictionary
  - g. 'rocket'
    - i. default : rocket

ii. unit: dict
iii. variable assigned to rocket dictionary

h. 'speed_control'
   i. default : speed_control
   ii. unit: dict
   iii. variable assigned to speed_control dictionary

i. 'position_control'
   i. default : position_control
   ii. unit: dict
   iii. variable assigned to position_control dictionary

j. 'sky_crane'
   i. default : sky_crane
   ii. unit: dict
   iii. variable assigned to sky_crane dictionary

k. 'rover'
   i. default : rover
   ii. unit: dict
   iii. variable assigned to rover dictionary

File: define_mission_events.py

Function: define_mission_events

I. dictionary: mission_events
    a. 'alt_heatshield_eject'
        i. Default: 8000
        ii. Unit: [m]
        iii. altitude at which heat shield is ejected
    b. 'alt_parachute_eject'
        i. Default: unit : 900,
        ii. Unit: [m]
        iii. altitude at which parachute is ejected
    c. 'alt_rockets_on'
        i. default : 1800
        ii. unit: [m]
        iii. altitude at which power descent sequence is initiated using variable-thrust solid rockets
    d. 'alt_skycrane_on'
        i. default : 7.6
        ii. unit: [m]
        iii. altitude appropriate for sky crane operation

File: define_planet.py

Function: define_planet

- II. dictionary: high_altitude
  - a. 'temperature'
    - i. Default: lambda altitude: -23.4 - 0.00222*altitude
    - ii. Unit: [C]
    - iii. Temperature at high altitudes
  - b. 'pressure'
    - i. Default: lambda altitude: 0.699*np.exp(-0.00009*altitude)}
    - ii. unit [KPa]
    - iii. pressure at high altitudes
- III. dictionary: low_altitude
  - a. 'temperature'
    - i. Default: lambda altitude: -31 - 0.000998*altitude,
    - ii. unit [C]
    - iii. temperature at low altitudes
  - b. 'pressure'
    - i. default : lambda altitude: 0.699*np.exp(-0.00009*altitude)}
    - ii. unit:[KPa]
    - iii. pressure at low altitude
- IV. dictionary: mars
  - a. 'g'
    - i. default : -3.72,
    - ii. unit: [m/s^2]
    - iii. gravity constant of mars
  - b. 'altitude_threshold'
    - i. default : 7000,
    - ii. unit: [m]
    - iii. maximum altitude of “low altitudes”
  - c. 'low_altitude'
    - i. default: low_altitude
    - ii. unit: dict
    - iii. variable assigned to low_altitude
  - d. 'high_altitude'
    - i. default : high_altitude
    - ii. unit: dict
    - iii. variable assigned to high_altitude
  - e. 'density'
    - i. default : density
    - ii. unit: dict
    - iii. variable assigned to density