Model Documentation: MOCAT-MC

MIT Orbital Capacity Assessment Toolbox - Monte Carlo

# Introduction:

MOCAT-MC is an open-source tool designed for simulating space objects in the circumterrestrial space. The model utilizes Monte Carlo methods to assess the diverging scenarios due to uncertainties in collision and cascade effects following.

# Input:

## Initial Population:

The initial set of space objects with known orbital elements and physical parameters.

Parameters:

* Semi-major axis (idx\_a) [Re]
* Eccentricity (idx\_ecco) [dimensionless]
* Inclination (idx\_inclo) [degrees]
* Longitude of the ascending node (idx\_nodeo) [degrees]
* Argument of periapsis (idx\_argpo) [degrees]
* Mean anomaly (idx\_mo) [degrees]
* B\* coefficient (idx\_bstar) [1/Re]
* Mass (idx\_mass) [kg]
* Radius (idx\_radius) [m]
* Error (idx\_error) [dimensionless]
* Controlled status (idx\_controlled) [0/1]
* Desired semi-major axis (idx\_a\_desired) [km]
* Mission lifetime (idx\_missionlife) [years]
* Constellation (idx\_constel) [dimensionless]
* Date created (idx\_date\_created) [julian day]
* Launch date (idx\_launch\_date) [julian day]
* Position (idx\_r\_x, idx\_r\_y, idx\_r\_z) [km]
* Velocity (idx\_v\_x, idx\_v\_y, idx\_v\_z) [km/s]
* Object class (idx\_objectclass) [dimensionless]
* Identifier (idx\_ID) [dimensionless]

Object Categorization:

* Satellite: idx\_objectclass=1, idx\_controlled=1.
* Derelict: idx\_objectclass=1, idx\_controlled=0.
* Debris: idx\_objectcalss=3,4,6,7,8,9.
* Rocket Body: idx\_objectclass=5.

An example of transposed format is shown here.:

| **Column** | **Description** | **Object 1** | **Object 2** |
| --- | --- | --- | --- |
| **1** | **(idx\_a) [Re, Earth Radius]** | **1.353546341** | **1.280815856** |
| **2** | **(idx\_ecco) [dimensionless]** | **0.1859627** | **0.1510842** |
| **3** | **(idx\_inclo) [degrees]** | **0.59791664** | **0.573834587** |
| **4** | **(idx\_nodeo) [degrees]** | **1.388250595** | **5.819561281** |
| **5** | **(idx\_argpo) [degrees]** | **4.523017266** | **0.626273005** |
| **6** | **(idx\_mo) [degrees]** | **1.388250595** | **5.819561281** |
| **7** | **(idx\_bstar) [1/Re, 1/Earth Radius]** | **0.00033729** | **0.0011816** |
| **8** | **(idx\_mass) [kg]** | **1.46** | **9.71** |
| **9** | **(idx\_radius) [m]** | **0.08** | **0.255011195** |
| **10** | **(idx\_error) [dimensionless]** | **0** | **0** |
| **11** | **(idx\_controlled) [0/1]** | **0** | **0** |
| **12** | **(idx\_a\_desired) [km]** | **0** | **0** |
| **13** | **(idx\_missionlife) [years]** | **0** | **0** |
| **14** | **(idx\_constel) [dimensionless]** | **0** | **0** |
| **15** | **(idx\_date\_created) [julian day]** | **NaN** | **NaN** |
| **16** | **(idx\_launch\_date) [julian day]** | **2436279.5** | **2436616.5** |
| **17** | **(idx\_r\_x) [km]** | **0** | **0** |
| **18** | **(idx\_r\_y) [km]** | **0** | **0** |
| **19** | **(idx\_r\_z) [km]** | **0** | **0** |
| **20** | **(idx\_v\_x) [km/s]** | **0** | **0** |
| **21** | **(idx\_v\_y) [km/s]** | **0** | **0** |
| **22** | **(idx\_v\_z) [km/s]** | **0** | **0** |
| **23** | **(idx\_objectclass) [dimensionless]** | **1** | **1** |
| **24** | **(idx\_ID) [dimensionless]** | **5** | **11** |

## Launch Profile:

* Launch profile follows the same format as the initial population.

# Simulation Parameters:

| **Parameter** | **Description** | **Units** |
| --- | --- | --- |
| **cfgMC.CUBE\_RES** | **Cube Size** | **km** |
| **cfgMC.PMD** | **Probability of Collision** | **dimensionless** |
| **cfgMC.alph** | **Probability of Collision Avoidance failure with one satellite active** | **dimensionless** |
| **cfgMC.alph\_a** | **Probability of Collision Avoidance failure with both satellites active** | **dimensionless** |
| **cfgMC.orbtol** | **Orbit Control Tolerance for controlled satellites** | **km** |
| **cfgMC.step\_control** | **Step Control for orbit control tolerance checking** | **dimensionless** |
| **cfgMC.P\_frag** | **Explosion Probability per day of Rocket Body Fragmentation**  (when P\_frag=0, explosions are not considered) | **dimensionless** |
| **cfgMC.altitude\_limit\_low** | **Lower Limit of Altitude** | **km** |
| **cfgMC.altitude\_limit\_up** | **Upper Limit of Altitude** | **km** |
| **cfgMC.missionlifetime** | **Operational Life of Payloads** | **years** |
| **t0\_prop** | **Initial Propagation Time** | **minutes** |
| **nyears** | **Length of Propagation** | **years** |
| **tf\_prop** | **Length of Propagation** | **minutes** |
| **cfgMC.dt\_days** | **Sampling Time for Cube Method and Propagation** | **days** |
| **DeltaT** | **Sampling Time for Cube Method and Propagation** | **minutes** |
| **cfgMC.launchRepeatYrs** | **Year Range for Repeated Launches** | **years** |
| **cfgMC.launchRepeatSmooth** | **Smoothing for Yearly Launch Rate** (0/1) | **dimensionless** |
| **cfgMC.collision\_alt\_limit** | **Collision Altitude Limit** | **km** |
| **cfgMC.animation** | **Animation Setting** | **dimensionless** |
| **cfgMC.save\_diaryName** | **Output File Saving Name** | **dimensionless** |
| **cfgMC.save\_output\_file** | **Output File Saving Option** | **dimensionless** |
| **cfgMC.saveMSnTimesteps** | **Timesteps for Output File Saving** | **dimensionless** |
| **filename\_save** | **Filename for Saving Information** | **file name** |
| **cfgMC.fillMassRadius** | **Fill in Missing Satellite Physical Parameters**  (0: don't fill in missing DISCOS data where many objects with 0 radius and/or mass.  1: ESA's method -- assume spherical aluminum depending on RCS size (S/M/L).  2: resampling method.) | **dimensionless** |
| **cfgMC.initpopMultiplier** | **Multiplier for Initial Population** | **dimensionless** |
| **cfgMC.physicalBstar** | **Recalculate B\* Coefficient**  (0/1) | **dimensionless** |

# Output:

The model provides users with various options to save. The control option is specified by the parameter save\_output\_file.

| **Save Option** | **Description** |
| --- | --- |
| **0** | **Don't save** |
| **1** | **Entire workspace** |
| **2** | **Satellite information, configuration, and parameters**  **('sats\_info','MCconfig','param','paramSSEM')** |
| **3** | **Summary statistics and configuration**  **('S\_MC','D\_MC','N\_MC','MCconfig','param','paramSSEM','frag\_info5')** |
| **4** | **Summary statistics and collision statistics (frag\_info)**  **('S\_MC','D\_MC','N\_MC','MCconfig','param','paramSSEM','frag\_info','frag\_info5')** |
| **5** | **Just collision statistics (frag\_info)**  **('MCconfig','param','paramSSEM','frag\_info','frag\_info5')** |
| **6** | **Summary statistics and mean/variance/median of physical parameters**  **('S\_MC','D\_MC','N\_MC','MCconfig','param','paramSSEM','param\_mean','param\_var','param\_median')** |
| **10** | **Save mat\_sats every 'saveMSnTimesteps' timesteps**  **('MCconfig','param','paramSSEM','matsatsperN')** |
| **11** | **Save configuration and object numbers of different categories**  **('MCconfig','param','paramSSEM','mat\_sats','nS','nD','nN','nB')** |

# Open-Source:

MOCAT-MC is an open-source project. The source code and further information can be found at the project's GitHub repository: https://github.com/ARCLab-MIT/MOCAT-MC.

Parameter Example:

cfgMC.CUBE\_RES = 50 [km]

cfgMC.PMD = 0.95 [dimensionless]

cfgMC.alph = 0.01 [dimensionless]

cfgMC.alph\_a = 0 [dimensionless]

cfgMC.orbtol = 5 [km]

cfgMC.step\_control = 2 [dimensionless]

cfgMC.P\_frag = 0 [dimensionless]

cfgMC.altitude\_limit\_low = 200 [km]

cfgMC.altitude\_limit\_up = 2000 [km]

cfgMC.missionlifetime = 8 [years]

t0\_prop = 0 [minutes]

nyears = 1 [years]

tf\_prop = cfgMC.YEAR2MIN \* nyears [minutes]

cfgMC.dt\_days = 5 [days]

DeltaT = cfgMC.dt\_days\*cfgMC.DAY2MIN [minutes]

cfgMC.launchRepeatYrs = [2018,2022] [years]

cfgMC.launchRepeatSmooth = 0 [dimensionless]

cfgMC.collision\_alt\_limit = 45000 [km]

cfgMC.animation = 'no' [dimensionless]

cfgMC.save\_diaryName = '' [dimensionless]

cfgMC.save\_output\_file = 0 [dimensionless]

cfgMC.saveMSnTimesteps = 146 [dimensionless]

filename\_save = sprintf('TLEIC\_year%i\_rand%i.mat',cfgMC.time0.Year,rngseed) [dimensionless]

cfgMC.filename\_save = filename\_save [dimensionless]

cfgMC.fillMassRadius = 2 [dimensionless]

cfgMC.initpopMultiplier = 1 [dimensionless]

cfgMC.physicalBstar = 1 [dimensionless]

# Output Examples:

switch save\_output\_file:

case 1:

savevars = {'\*'}

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 2:

savevars = {'sats\_info','MCconfig','param','paramSSEM'}

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 3:

savevars = {'S\_MC','D\_MC','N\_MC','MCconfig','param','paramSSEM','frag\_info5'}

MCconfig.mat\_sats = []

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 4:

savevars = {'S\_MC','D\_MC','N\_MC','MCconfig','param','paramSSEM','frag\_info','frag\_info5'}

MCconfig.mat\_sats = []

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 5:

savevars = {'MCconfig','param','paramSSEM','frag\_info','frag\_info5'}

MCconfig.mat\_sats = []

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 6:

savevars = {'S\_MC','D\_MC','N\_MC','MCconfig','param','paramSSEM','param\_mean','param\_var','param\_median'}

MCconfig.mat\_sats = []

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 10:

savevars = {'MCconfig','param','paramSSEM','matsatsperN'}

MCconfig.mat\_sats = []

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

case 11:

savevars = {'MCconfig','param','paramSSEM','mat\_sats','nS','nD','nN','nB'}

MCconfig.mat\_sats = []

save([filename\_save(1:end-4),'\_part\_',num2str(file\_save\_index),'.mat'],'-v7.3',savevars{:})

otherwise:

fprintf('save\_output\_file flag set to unsupported value: %i\n', save\_output\_file)