# Instructions on how to use the function.

Need Matlab 2020a or newer to work. It may work for older versions, but you would have to change how the function is reading the setup Excel table.

%% README (Instructions)

% GMAT\_LowThrust writes GMAT script files to propagate and optimize

% low-thrust trajectory problems

%

% Low\_Thrust problems may be propagated as is or optimized.

% The bin folder of GMAT must be within your path variable for the API

% commands used in this function to work.

% Optimization is done using SNOPT 7.6 Matlab interface (Other versions

% may work).

% SNOPT Guide Link:

% https://ccom.ucsd.edu/~optimizers/docs/snopt/interfaces.html#matlab

% https://ccom.ucsd.edu/~optimizers/static/pdfs/sndoc7.pdf

% SNOPT Matlab folder must be within the path variable for optimization to work

% The constraint / objective function is called using objFunc\_conFunc

%

% Optimization Notes and Limitations:

%

% 1) Optimization assumes that thrust/acceleration magnitude remains

% constant, and if fuel mass is decremented, it's constant for all time steps.

% 2) Optimization works by updating a GMAT thrust history file and rerunning a

% GMAT script. This process is computationally inefficient as files must be

% saved to the hard drive. -> Long run times

% 3) Optimization performance depends on how the design variable, constraint,

% and objective bounds are defined. You may want to adjust these parameters

% to improve the performance of your problem.

% 4) Problems with more than 200 steps tend to take more than 10 hours to

% optimize for optimal time of flight.

%

% Function inputs and outputs:

%

% [OutPut\_DataStruct]=GMAT\_LowThrust(FileName,varargin)

% 1) OutPut\_DataStruct includes the time step array, ICRF coordinates thrust

% array, Alpha (in-plane), and Beta (out-plane) thrust angles. X is the design

% variable array from the last optimization iteration.

% 2) FileName is the name of the Excel file that includes the set-up

% information about your problem. Example input: "\EarthToMarsProblem.xlsx"

% 3) varargin includes the optional inputs

% i) 'Optimize' option tells the function that you want to optimize your

% low-thrust problem. This will launch the SNOPT optimization

% sequence and run 'objFunc\_conFunc' (objective/constraint function)

% ii) By default, the optimizer will run with default settings. To

% change settings, you need to pass in a struct after 'Optimize' that

% includes your optimization settings. Example:

% opt = struct( ...

% 'TOF\_LowBound',200,'TOF\_UpperBound',1000,'MajorFeasibilityTolerance',1e-6,...

% 'MajorOptimalityTolerance',1e-6,'OptimizationRunTimeLimit',86400,...

% 'MajorIterationLimit',5000,'Obj','Cons');

% a) TOF\_LowBound and TOF\_UpperBound, are the lower and upper bound

% time of flight used in the optimization sequence. Values in days

% b) MajorFeasibilityTolerance and MajorOptimalityTolerance are the feasibility

% and optimality tolerance SNOPT settings

% c) OptimizationRunTimeLimit is how long the SNOPT optimization

% sequence will run. In seconds

% d) MajorIterationLimit is the limit on the number of major

% iterations that will be run during optimization

% e) Obj is the objective function setting option. The two settings are

% 'Cons' and 'TOF'

% 'Cons' is used to minimize the state vector constraints without

% defining the Time of Flight as the objective to minimize

% 'TOF' is used to optimize the Time of Flight as the

% objective function

% 4) The function will also generate the files GMAT\_RunScript\_Plots.script and

% GMAT\_ThrustProfileSolution.thrust in the GMAT\_RunFolder. GMAT\_RunScript\_Plots is a GMAT file

% that can be used to plot the orbital trajectory (for visualization) and

% GMAT\_ThrustProfileSolution is the thrust history file used to propagate

% the trajectory. During optimization, the files GMAT\_RunScript and

% GMAT\_RunThrustProfile will also be created. GMAT\_RunThrustProfile will

% have the most up-to-date iteration in the optimization sequence.

% If you need to end the optimization sequence before GMAT\_RunScript\_Plots

% and GMAT\_ThrustProfileSolution.thrust are generated, you can modify the

% existing GMAT\_RunScript\_Plots to run the last iteration of GMAT\_RunThrustProfile.

% 5) The file SNOPT\_summary.txt will also be created during the

% optimization sequence. This file includes a summary of the SNOPT optimization.

%

% Example Input:

%

% Propgating trajectory without optimization:

% OutPut\_DataStruct=GMAT\_LowThrust("\EarthToMarsProblem.xlsx")

%

% Optimizing using default settings:

% OutPut\_DataStruct=GMAT\_LowThrust("\EarthToMarsProblem.xlsx",'Optimize')

%

% Optimizing using own settings struct:

% OutPut\_DataStruct=GMAT\_LowThrust("\EarthToMarsProblem.xlsx",'Optimize',Opt)

%

% Supporting Functions:

%

% X\_interpreter-> can be used to generate output struct if optimization

% sequence is paused/terminated before completion.

% GMAT\_LowThrustDataInterolator-> can be used to increase or decrease the

% number of time steps of a problem.

% LowThrustOutputStructToExcel-> can be used to regenerate the Excel set-up

% sheet using output struct

# Example problem (Earth to Mars) 20 minutes

1. Use EarthToMars\_10Steps.xlsx low-thrust setup file to propagate an Earth to Mars trajectory.
2. Use this command in this directory:

OutPut=GMAT\_LowThrust(“\EarthToMars\_10Steps.xlsx”)

1. Open GMAT and run “GMAT\_RunScript\_Plots”

It should produce a trajectory that looks like this:

A picture containing circle, screenshot, astronomy

Description automatically generated

1. To improve the initial guess, we can use the command:

OutPut=GMAT\_LowThrust(“\EarthToMars\_10Steps.xlsx”,’Optimize’,OP\_Options)

Where OP\_Options is a struct defined as:

OP\_Options= struct('TOF\_LowBound',200,'TOF\_UpperBound',1000,'MajorFeasibilityTolerance',1e-6,…

'MajorOptimalityTolerance',1e-6,'OptimizationRunTimeLimit',3600,'MajorIterationLimit',…

5000,'Obj','Cons');

Using these commands will launch the SNOPT optimization loop with the objective function of minimizing the constraints (the difference in the final state vector between the satellite and mars). This process should take 10 minutes. Once completed reloading and running “GMAT\_RunScript\_Plots” in GMAT should produce a trajectory that looks like this:

A picture containing circle, screenshot, astronomy

Description automatically generated

1. Although the trajectory reached Mars, the time of flight can be improved. Use this command to save the previous solution to a Excel setup file to rerun the Optimization:

LowThrustOutputStructToExcel(OutPut,“\EarthToMars\_10Steps.xlsx”)

This will generate an excel file called \EarthToMars\_10Steps\_SolutionData.xlsx

1. Now rerun GMAT\_LowThrust with the new excel setup file to optimize for time of flight

OutPut=GMAT\_LowThrust(“\EarthToMars\_10Steps\_SolutionData.xlsx”,’Optimize’,OP\_Options)

Where OP\_Options is a struct defined as:

OP\_Options= struct('TOF\_LowBound',200,'TOF\_UpperBound',1000,'MajorFeasibilityTolerance',1e-6,…

'MajorOptimalityTolerance',1e-6,'OptimizationRunTimeLimit',3600,'MajorIterationLimit',…

5000,'Obj','TOF');

What’s highlighted in yellow is used to specify TOF as the objective function. SNOPT should run for another 10 minutes. The final trajectory should look like this:

A picture containing circle, screenshot, space, astronomy

Description automatically generated