WUDBF km_sim Documentation (2024)

Summary of what to do:

1. Calculate/obtain necessary single variables for each mission and put them in a single CSV file
2. Obtain CD_vs_AOA and CL_vs_AOA data from Aero (probably the same for each mission)
3. Obtain Thrust_vs_Velocity data from Propulsion (probably the same for each mission)
4. **Right click on "km_sim_2024 (or whatever the top km_sim file is)" -> Add to Path -> Selected Folders and Subfolders***
5. Use createPlaneStruct.m to create a new plane for each mission
6. Create mission files (using previous years' missions as a basis – ex: 'Mission1.m')
7. Run km sim for each mission using the mission's plane struct and mission file

Creating a new plane using createPlaneStruct.m:

- The command takes in 5 parameters in this order:
  - Name of the new plane struct (String)
  - Angle of attack vs coefficient of lift (CSV)
  - Angle of attack vs coefficient of drag (CSV)
  - All single variables (described below) (CSV)
  - Thrust vs velocity data from propulsion (CSV)
- This command, when run successfully, will create a plane struct which can be used to run a mission and get information about its performance.

Running a mission using the run_km_sim command:

- The command requires 2 inputs, but a third can be specified
  - Name of the plane struct
  - Name of a mission file
  - Name of file to write plotting and performance data to (optional)

run_km_sim is the top-level function for executing km_sim, which stands for "kinematic mission simulation".

SYNTAX:

run_km_sim(vehicle_file, mission_file)

run_km_sim(vehicle_file, mission_file, post_pro_file)

INPUTS:

vehicle_file - this is a .mat file that contains a struct variable named "plane" which specifies the aircraft tables and data

mission_file - this is a function that defines the mission plan and air-vehicle initial conditions, and calls "execute_mission"

post_pro_file - this file will be executed after the mission is run, used for plotting, scoring, etc.

EXAMPLE:

- run_km_sim('V9_M1.mat', 'Mission1.m') -> runs km_sim for mission 1 and makes 8 plots
- run_km_sim('V9_M1.mat', 'Mission1.m', 'plot3D.m') -> runs km_sim for mission 1 and makes 3D plot of flight path

Aero and Alt Aero

- Request CL vs Alpha and CD vs AOA plots from Aero team
  - These plots are usually obtained from XFLR
  - They must include CL/CD vs AOA
  - aoa_vs_cl_cd (CSV): CL and CD plotted against the angle of attack (AOA) when flaps are up (cruise – alt_aero data) and when flaps are down (landing – aero data)
    - CD_vs_AOA should originally be without parasitic drag – parasitic drag is added later on as a single variable when running createPlaneStruct.m
- createAero.m and createAltAero.m are scripts called by createPlaneStruct.m
  - They use the XFLR data to create tables of CL/CD vs AOA data when flaps are up (cruise – alt_aero) and when flaps are down (landing – aero)
  - You must know which columns in the CL/CD vs AOA tables align with alt_aero (flaps up) and aero (flaps down)
    - Change the column numbers in createAero.m or createAltAero.m to reflect this

Calculating Single Variables for Plane Struct

Amps (aka Current)

- Description: The sea-level static energy_dot of the power system in units consistent with the energy units of the project (for mAh energy units, take **Amps***1000/3600 to get mAh per second). USER ONLY ENTERS THE AMPS COMPONENT, createPlaneStruct.m calculates the edot using the amps value
- Ask propulsion for this
- 2023 Calculation: 41 Amps
- 2024 Calculation M2/M3: 70 Amps
- (Laps = e_cap / (edot*average lap time)) average lap time ~ 40 sec

maxAlt

- Description: Around 100 meters above sea level altitude
- 2023 Calculation: 769 meters
- 2024 Calculation M2/M3: 497.2 meters

g_limit

- Description: The g-force limit when turning the aircraft. The default limit should be 10, however this may change based on rules of competition. (View 2021-22 rules with g-limit sensors for packages as an example)
- 2023 Calculation: 10
- 2024 Calculation: 10

empty_W

- Description: All-up weight (no fuel)
- 2024 M1 Calculation: 2.242 kg
- 2024 M2 Calculation: 4.207 kg
- 2024 M3 Calculation: 3.535 kg
- 2023 M2 Calculation: 4.84 kg
- 2023 M3 Calculation: 3.4762 kg

spec_fuel_W

- Description: If not using fuel, this is 0
- 2024 Calculation: 0

e_cap

- Description: Battery capacity
- 2023 Calculation: 4500 mAh
- 2024 Calculation: 3250 mAh

weight

- Description: All-up weight (including fuel)
- 2024 M1 Calculation: 2.242 kg
- 2024 M2 Calculation: 4.207 kg
- 2024 M3 Calculation: 3.535 kg
- 2023 M2 Calculation: 4.84 kg
- 2023 M3 Calculation: 3.4762 kg

S_ref

- Description: Wing area
- 2024 Calculation: 0.36 m^2

Parasitic Drag

- Description: Parasitic drag, is added to coefficient of drag in aero and alt_aero (it's higher in M3 because of the antenna)
- 2024 M1/M2/M3 Calculation: 0.0817
- 2023 M3 Calculation: 0.12725

Generating Prop Performance

- Request Thrust vs Velocity data from Propulsion team
  - velocityThrustValues (CSV) – Contains velocity values in first column and corresponding thrust values in second column
    - Must get rid of rows with NaN values
    - velocityThrustValues is used in createPlaneStruct.m
- gen_linear_prop_table.m is called by createPlaneStruct to generate prop performance
  - gen_linear_prop_table.m uses 3 parameters
    - sl_static_edot – calculated using Current
    - max_alt – maximum altitude
    - thrust_vs_velo: Thrust_vs_Velocity CSV, described above

What Needs to Be Added/Improved in km_sim

- Create scoring for mission files and documentation for that process
- Other things(?)