

Vehicle Class Documentation

Class Definition

`Vehicle (builtins.object)`

Constructor

```
Vehicle(m, I_xx, I_yy, I_zz, I_xy,  
I_yz, I_xz, x_com, y_com, z_com,  
base_aero_coeffs, x_mrc, y_mrc, z_mrc, S_A,  
a_l, b_l, c_l, d_l, input_aero_coeffs=0.0,  
input_force_moment=0.0, input_aero_coeffs_idx=None,  
input_force_moment_idx=None, dim_additional_input=0)
```

Overview

The **Vehicle** class represents a state-less dynamics block that provides common calculations for the dynamics of a flight vehicle. The vehicle model is parameterized based on the following components:

Components

Inertial Properties

- `m`, `I_xx`, `I_yy`, `I_zz`, `I_xy`, `I_yz`, `I_xz`, `x_com`, `y_com`, `z_com`
- Inertia is expressed in body-fixed coordinates about the center of mass position (`x_com`, `y_com`, `z_com`). The center of mass position is relative to an arbitrary reference, such as the CAD origin.

Aerodynamics Model

- `base_aero_coeffs`:
 - A function of angles of attack, sideslip, Mach number, and Reynolds number. It should return an array of coefficients in the following order: drag, sideforce, lift, static roll, pitch, and yaw moments, and dynamic roll, pitch, and yaw moments. Forces are expressed in the wind coordinate system; moments in the body-fixed FRD coordinate system.
 - For transforming aerodynamic forces from body-fixed to wind coordinate system, use the transpose of the direction cosine matrix `dynamics.body_to_wind_dcm(alpha, beta)`.
- `x_mrc`, `y_mrc`, `z_mrc`:
 - Moment reference center position, defined relative to the same arbitrary reference as the center of mass. Moment coefficients are assumed to be about this position.
- `S_A`:
 - Reference area.
- `a_l`, `b_l`, `c_l`:
 - Reference lengths for roll, pitch, and yaw.
- `d_l`:
 - Reference length for Reynolds number calculation.

Note: Damping coefficients are transformed with the static moment coefficients, which does not usually hold. To model damping coefficients, it is recommended to set the center of mass and moment reference center to the same location (i.e., (0, 0, 0)).

Controlled Inputs

Additional, controlled input can be modeled:

- `input_force_moment`, `input_force_moment_idx`:
 - Non-aerodynamic controlled inputs (e.g., propulsion).
- `input_aero_coeffs`, `input_aero_coeffs_idx`:
 - Controlled aerodynamics.

Processing of `input_aero_coeffs` and `input_force_moment` parameterized models follows this logic:

- If `None`: Assume that a child class will overwrite.
- If callable: Use directly (should have the correct signature).
 - `input_aero_coeffs` callback assumes flight condition arguments of the aerodynamics model and returns aerodynamic coefficient increments.
 - `input_force_moment` callback takes only routed control inputs and returns the translational forces and moments being modeled (e.g., propulsion model).
- Otherwise: Attempt to build a function that returns constant value(s).

Attributes `input_aero_coeffs_idx` and `input_force_moment_idx` are used to route extra control inputs to the input aero and force/moment functions, respectively. Use `None` to route all inputs or a list of integers to route particular inputs (including an empty list for no routing). The `dim_additional_input` argument in `__init__` is used to allocate extra control input channels in the block diagram.

Input Components

- [0:13]: `p_x`, `p_y`, `p_z`, `q_0`, `q_1`, `q_2`, `q_3`, `v_x`, `v_y`, `v_z`, `omega_X`, `omega_Y`, `omega_Z` (vehicle state components)
- [13:16]: `lamda_D`, `phi_D`, `h` (translational position in planet-fixed frame: longitude, latitude, altitude)
- [16:19]: `psi`, `theta`, `phi` (Euler angles: yaw, pitch, roll)
- [19:22]: `rho`, `c_s`, `mu` (density, speed of sound, viscosity)
- [22:25]: `V_T`, `alpha`, `beta` (true airspeed, angle of attack, sideslip)
- [25:28]: `p_B`, `q_B`, `r_B` (angular velocity components relative to NED)
- [28:31]: `V_N`, `V_E`, `V_D` (velocity in the planet-fixed frame in NED coordinates)
- [31:34]: `W_N`, `W_E`, `W_D` (winds in NED frame)

Output Components

- [0:3]: `A_X`, `A_Y`, `A_Z` (translational acceleration of the center of mass in FRD coordinates)
- [3:6]: `alpha_X`, `alpha_Y`, `alpha_Z` (angular acceleration in FRD coordinates)

Additional Attributes

- `dim_state`, `dim_output`, `dim_input`, `num_events`: SimuPy interface attributes, must be updated for subclasses.
- `<variable_name>_idx`: Named index for state or output data columns.
- `output_column_names`: List of output names (strings) for constructing data structures.
- `output_column_names_latex`: List of LaTeX output names for labeling plots.

Methods

- `__init__(self, ...)`: Initialize the vehicle object.
- `mrc_to_com_cpm(self)`: Construct a skew symmetric matrix for cross product of the position vector from moment reference center to center of mass.
- `output_equation_function(self, t, u)`: Compute dynamic outputs for simulation.

- `prepare_to_integrate(self, *args, **kwargs)`: Prepares the block for simulation in SimuPy. Returns the first output vector.
- `tot_aero_forces_moments(self, qbar, Ma, Re, V_T, alpha, beta, p_B, q_B, r_B, *args)`: Computes the total aerodynamic forces and moments for a given flight and vehicle condition.