PyFME Documentation

Release 0.1.dev0

AeroPython Team

January 17, 2016

Contents

PyFME :rocket: :airplane: stands for Python Flight Mechanics Engine.

The main idea behind a Flight Dynamics Engine is to model the physics involved in aircraft's flight. Models for the aircraft, atmosphere, dynamics, aerodynamics... implemented here allow us to calculate the aircraft response to the pilot's inputs.

Contents

1.1 API Reference

1.1.1 *pyfme.environment* package

ISA functions. Implementation based on:

`pyfme.environment.isa.`**`atm`**`(`*h*`)`

ISA 1976 Standard atmosphere temperature, pressure and density.

h [float] Geopotential altitude (m). h values must range from 0 to 84500 m.

T [float] Temperature (K).

p [float] Pressure (Pa).

rho [float] Density (kg/m³)

ValueError If the value of the altitude is outside the defined layers.

Check layers and reference values in [2].

1.1.2 *pyfme.models* package

Flight Dynamic Equations of Motion

These are the equations to be integrated, thus they have the following order for the arguments: func(time, y, ...) where dy/dt = func(y, ...)

Assumptions:

- ...

`pyfme.models.euler_flat_earth.`**`jac_kinematic_angular_eqs`**`(`*time*, *euler_angles*, *angular_vel*`)`

Jacobian of kinematic angular equations

time [float] Current time (s)

euler_angles [array_like] Current value of euler angles following z-y'-x" convention (theta, phi, psi) or (pitch, roll, yaw) in (rad, rad, rad).

angular_vel [array_like] Current value of absolute angular velocity, expressed in body axes (p, q, r) in (rad/s, rad/s rad/s).

jac [array_like] Current value of jacobian of kinematic angular equationes. It is a 3x3 matrix.

kinematic_angular_eqs

`pyfme.models.euler_flat_earth.`**`jac_linear_and_angular_momentum_eqs`**(*time*, *vel*, *mass*, *inertia*)

Jacobian of linear and angular momentum equations

time [float] Current time (s)

vel [array_like] Current value of absolute velocity and angular velocity, both expressed in body axes (u, v, w, p, q, r) in (m/s, m/s, m/s, rad/s, rad/s rad/s).

mass [float] Current mass of the aircraft (kg).

inertia [array_like] 3x3 tensor of inertia of the aircraft (kg · m²) Current equations assume that the aircraft has a symmetry plane (x_b - z_b), thus J_xy and J_yz must be null.

jac [array_like] Current value of jacobian of linear and angular momentum equationes. It is a 6x6 matrix.

linear_and_angular_momentum_eqs

`pyfme.models.euler_flat_earth.`**`kinematic_angular_eqs`**(*time*, *euler_angles*, *angular_vel*)

Kinematic angular equations

time [float] Current time (s)

euler_angles [array_like] Current value of euler angles following z-y'-x" convention (theta, phi, psi) or (pitch, roll, yaw) in (rad, rad, rad).

angular_vel [array_like] Current value of absolute angular velocity, expressed in body axes (p, q, r) in (rad/s, rad/s rad/s).

deuler_angles_dt [array_like] Current value of euler angle time derivative (dtheta_dt, dphi_dt, dpsi_dt) in (rad/s, rad/s, rad/s).

linear_and_angular_momentum_eqs, navigation_eqs

`pyfme.models.euler_flat_earth.`**`linear_and_angular_momentum_eqs`**(*time*, *vel*, *mass*, *inertia*, *forces*, *moments*)

Linear and angular momentum equations

time [float] Current time (s).

vel [array_like] Current value of absolute velocity and angular velocity, both expressed in body axes (u, v, w, p, q, r) in (m/s, m/s, m/s, rad/s, rad/s rad/s).

mass [float] Current mass of the aircraft (kg).

inertia [array_like] 3x3 tensor of inertia of the aircraft (kg · m²) Current equations assume that the aircraft has a symmetry plane (x_b - z_b), thus J_xy and J_yz must be null.

forces [array_like] 3 dimensional vector containing the total forces (including gravity) in x_b, y_b, z_b axes (N).

moments [array_like] 3 dimensional vector containing the total moments in x_b, y_b, z_b axes (N·m).

accel [array_like] Current value of absolute acceleration and angular acceleration, both expressed in body axes (du_dt, dv_dt, dw_dt, dp_dt, dq_dt, dr_dt) in (m/s**2 , m/s**2, m/s**2, rad/s**2, rad/s**2, rad/s**2).

navigation_eqs, kinematic_angular_eqs

`pyfme.models.euler_flat_earth.`**`navigation_eqs`**(*time*, *linear_vel*, *euler_angles*)

Kinematic linear equations

time [float] Current time (s)

euler_angles [array_like] Current value of euler angles following z-y'-x" convention (theta, phi, psi) or (pitch, roll, yaw) in (rad, rad, rad).

linear_vel [array_like] Current value of absolute linear velocity, expressed in body axes (u, v, w) in (m/s, m/s m/s).

dpos_dt [array_like] Current value of absolute linear velocity, expressed in Earth axes (dx_dt, dy_dt, dz_dt) in (m/s, m/s, m/s).

kinematic_angular_eqs, linear_and_angular_momentum_eqs

1.1.3 *pyfme.utils* package

Frames of Reference orientation functions

`pyfme.utils.coordinates.`**`body2hor`**(*body_coords*, *theta*, *phi*, *psi*)

Transforms the vector coordinates in body frame of reference to local horizon frame of reference.

body_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in body axes.

theta [float] Pitch (or elevation) angle (rad).

phi [float] Bank angle (rad).

psi [float] Yaw (or azimuth) angle (rad)

hor_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in local horizon axes.

ValueError If the values of the euler angles are outside the proper ranges.

hor2body function.

See [1] or [2] for frame of reference definition. Note that in order to avoid ambiguities ranges in angles are limited to:

- -pi/2 <= theta <= pi/2
- -pi <= phi <= pi
- 0 <= psi <= 2*pi

`pyfme.utils.coordinates.`**`body2wind`**(*body_coords*, *alpha*, *beta*)

Transforms the vector coordinates in body frame of reference to wind frame of reference. Parameters ——— body_coords : array_like

3 dimensional vector with (x,y,z) coordinates in body axes.

alpha [float] Angle of attack (rad).

beta [float] Sideslip angle (rad).

wind_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in wind axes.

ValueError If the values of the wind-body angles are outside the proper ranges.

wind2body function. Notes —– See [1] or [2] for frame of reference definition. Note that in order to avoid ambiguities ranges in angles are limited to: * -pi/2 <= alpha <= pi/2 * -pi <= beta <= pi References ——— .. [1] B. Etkin, "Dynamics of Atmospheric Flight," Courier Corporation,

pp. 104-120, 2012.

`pyfme.utils.coordinates.`**`check_alpha_beta_range`**(*alpha*, *beta*)

Check alpha, beta values are inside the defined range. This comprobation can also detect if the value of the angle is in degrees in some cases.

`pyfme.utils.coordinates.`**`check_gamma_mu_chi_range`**(*gamma*, *mu*, *chi*)

Check gamma, mu, chi values are inside the defined range. This comprobation can also detect if the value of the angle is in degrees in some cases.

`pyfme.utils.coordinates.`**`check_theta_phi_psi_range`**(*theta*, *phi*, *psi*)

Check theta, phi, psi values are inside the defined range. This comprobation can also detect if the value of the angle is in degrees in some cases.

`pyfme.utils.coordinates.`**`hor2body`**(*hor_coords*, *theta*, *phi*, *psi*)

Transforms the vector coordinates in local horizon frame of reference to body frame of reference.

hor_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in local horizon axes.

theta [float] Pitch (or elevation) angle (rad).

phi [float] Bank angle (rad).

psi [float] Yaw (or azimuth) angle (rad)

body_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in body axes.

ValueError If the values of the euler angles are outside the proper ranges.

body2hor function.

See [1] or [2] for frame of reference definition. Note that in order to avoid ambiguities ranges in angles are limited to:

- -pi/2 <= theta <= pi/2
- -pi <= phi <= pi
- 0 <= psi <= 2*pi

`pyfme.utils.coordinates.`**`hor2wind`**(*hor_coords*, *gamma*, *mu*, *chi*)

Transforms the vector coordinates in local horizon frame of reference to wind frame of reference. Parameters ——— hor_coords : array_like

3 dimensional vector with (x,y,z) coordinates in local horizon axes.

gamma [float] Velocity pitch (or elevation) angle (rad).

mu [float] Velocity bank angle (rad).

chi [float] Velocity yaw (or azimuth) angle (rad)

wind_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in wind axes.

ValueError If the values of the wind-horizon angles are outside the proper ranges.

wind2hor function. Notes —– See [1] for frame of reference definition. Note that in order to avoid ambiguities ranges in angles are limited to: * -pi/2 <= gamma <= pi/2 * -pi <= mu <= pi * 0 <= chi <= 2*pi References ——— .. [1] Gómez Tierno, M.A. et al, "Mecánica del Vuelo," Garceta, pp. 1-12,

2012

`pyfme.utils.coordinates.`**`wind2body`**(*wind_coords*, *alpha*, *beta*)

Transforms the vector coordinates in wind frame of reference to body frame of reference. Parameters ——— wind_coords : array_like

3 dimensional vector with (x,y,z) coordinates in body axes.

alpha [float] Angle of attack (rad).

beta [float] Sideslip angle (rad).

body_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in wind axes.

ValueError If the values of the wind-body angles are outside the proper ranges.

body2wind function. Notes —– See [1] or [2] for frame of reference definition. Note that in order to avoid ambiguities ranges in angles are limited to: * -pi/2 <= alpha <= pi/2 * -pi <= beta <= pi References ——— .. [1] B. Etkin, "Dynamics of Atmospheric Flight," Courier Corporation,

pp. 104-120, 2012.

`pyfme.utils.coordinates.`**`wind2hor`**(*wind_coords*, *gamma*, *mu*, *chi*)

Transforms the vector coordinates in wind frame of reference to local horizon frame of reference. Parameters ——— wind_coords : array_like

3 dimensional vector with (x,y,z) coordinates in wind axes.

gamma [float] Velocity pitch (or elevation) angle (rad).

mu [float] Velocity bank angle (rad).

chi [float] Velocity yaw (or azimuth) angle (rad)

hor_coords [array_like] 3 dimensional vector with (x,y,z) coordinates in local horizon axes.

ValueError If the values of the wind-horizon angles are outside the proper ranges.

hor2wind function. Notes —– See [1] for frame of reference definition. Note that in order to avoid ambiguities ranges in angles are limited to: * -pi/2 <= gamma <= pi/2 * -pi <= mu <= pi * 0 <= chi <= 2*pi References ——— .. [1] Gómez Tierno, M.A. et al, "Mecánica del Vuelo," Garceta, pp. 1-12,

2012

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

A

B

C

H

J

K

L

N

P

W