MAVS Python

Generated by Doxygen 1.8.14

Contents

Chapter 1

Python interface to the MSU Autonomous Vehicle Simulator (MAVS).

API documentation for the MAVS python package. For access to the source code, see https://gitlab.com/cgoodin/msu-autonomous-vehicle-simulator.

For installation and user guide, see the wiki at https://gitlab.com/cgoodin/msu-autonomous-vehicle-simulato

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

Namespace Documentation

5.1 mavs_interface Namespace Reference

This module provides classes and functions for interfacing with the MAVS library.

5.1.1 Detailed Description

This module provides classes and functions for interfacing with the MAVS library.

Include it in your code like this: import mavs_interfaces as mavs

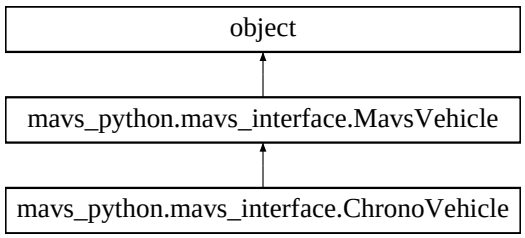
MAVS is natively in C++, with C interfaces written to make features accessible from python.

Chapter 6

Class Documentation

6.1 mavs_python.mavs_interface.ChronoVehicle Class Reference

Inheritance diagram for mavs_python.mavs_interface.ChronoVehicle:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def Load (self, fname)

Public Attributes

- vehicle

  vehicle (void): Pointer to a Chrono vehicle.

6.1.1 Detailed Description

```
ChronoVehicle class.

Inherits from MavsVehicle base class.

Attributes:
vehicle (void): Pointer to a Chrono vehicle.
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def mavs_python.mavs_interface.ChronoVehicle.__init__ (
              self )
```

ChronoVehicle constructor.

6.1.2.2 __del__()

```
def mavs_python.mavs_interface.ChronoVehicle.__del__ (
              self )
```

Chrono vehicle destructor.

6.1.3 Member Function Documentation

6.1.3.1 Load()

```
def mavs_python.mavs_interface.ChronoVehicle.Load (
              self,
              fname )
```

Load an Chrono vehicle file.

Examples are in mavs/data/vehicles/chrono_inputs.

Parameters:
fname (string): Full path to the vehicle input file to load.

6.1.4 Member Data Documentation

6.1.4.1 vehicle

```
mavs_python.mavs_interface.ChronoVehicle.vehicle
```

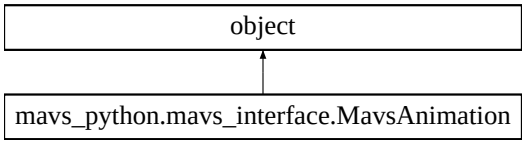
vehicle (void): Pointer to a Chrono vehicle.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.2 mavs_python.mavs_interface.MavsAnimation Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsAnimation:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def Load (self, path, file)
- def SetScale (self, scale)
- def MoveToWaypoint (self, dt, x, y)
- def LoadPathFile (self, path_file)
- def SetSpeed (self, speed)
- def SetBehavior (self, behavior)
- def SetPosition (self, x, y)
- def SetHeading (self, heading)
- def SetRotations (self, y_to_x, y_to_z)

Public Attributes

- object

  object (void): Pointer to a MAVS animation.

6.2.1 Detailed Description

```
MavsAnimation class.

A MavsAnimation has a sequence of keyframes and behavior associated with it,
as well as scaling parameters for the keyframes.

Attributes:
object (void): Pointer to a MAVS animation.
```

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def mavs_python.mavs_interface.MavsAnimation.__init__ (
              self )
```

Constructor for a MavsAnimation.

6.2.2.2 __del__()

```
def mavs_python.mavs_interface.MavsAnimation.__del__ (
              self )
```

Destructor for a MavsAnimation.

6.2.3 Member Function Documentation

6.2.3.1 Load()

```
def mavs_python.mavs_interface.MavsAnimation.Load (
              self,
              path,
              file )
```

Load an animation from an input file.

The animation input file lists the mesh scaling and keyframes.
Example input files can be found in mavs/data/actors.

Parameters:
path (string): The path to the animation file.
file (string): The animation file name.

6.2.3.2 LoadPathFile()

```
def mavs_python.mavs_interface.MavsAnimation.LoadPathFile (
              self,
              path_file )
```

Load a path file for the animation to follow.

Path files are in the .vprp format. Examples can be found in mavs/data/waypoints.

Parameters:
path_file (string): Full path to the file to load.

6.2.3.3 MoveToWaypoint()

```
def mavs_python.mavs_interface.MavsAnimation.MoveToWaypoint (
              self,
              dt,
              x,
              y )
```

Move the animation to a specified waypoint.

Moves the animation to an x-y point and automatically places it on the ground.
The dt parameter is the length of time it takes to move the animation, which
will be used to calculate its velocity.

Parameters:
dt (float): The duration in seconds to move the animation.
x (float): The global x ENU coordinate to move to.
y (float): The global y ENU coordinate to move to.

6.2.3.4 SetBehavior()

```
def mavs_python.mavs_interface.MavsAnimation.SetBehavior (
              self,
              behavior )
```

Set the motion behavior of the animation.

Options are 'wander', 'straight', or 'circle'.

Parameters:
behavior (string): The desired behavior.

6.2.3.5 SetHeading()

```
def mavs_python.mavs_interface.MavsAnimation.SetHeading (
              self,
              heading )
```

Set the heading of the animation in radians relative to global ENU.

East is 0, North is pi/2, West is pi, south is -pi/2 or 3pi/2.

Parameters:
heading (float): Desired heading.

6.2.3.6 SetPosition()

```
def mavs_python.mavs_interface.MavsAnimation.SetPosition (
              self,
              x,
              y )
```

Set the position without updating the velocity.

Move the animation to a specified position.

Parameters:
x (float): The global x ENU coordinate to move to.
y (float): The global y ENU coordinate to move to.

6.2.3.7 SetRotations()

```
def mavs_python.mavs_interface.MavsAnimation.SetRotations (
              self,
              y_to_x,
              y_to_z )
```

Set the rotations to be applied to the mesh.

Use these if the mesh was created in a coordinate system that doesn't match MAVS.
For example, if the mesh was created in a "y-up" coordinate system, set y_to_z to True.

Parameters:
y_to_x (bool): Rotate y to x.
y_to_z (bool): Rotate y to z.

6.2.3.8 SetScale()

```
def mavs_python.mavs_interface.MavsAnimation.SetScale (
              self,
              scale )

Set the scale of the animation.

Scale the animation equally in the x-y-z dimensions.

Parameters:
scale (float): Factor to scale the animation by.
```

6.2.3.9 SetSpeed()

```
def mavs_python.mavs_interface.MavsAnimation.SetSpeed (
              self,
              speed )

Set the speed of the animation in m/s.

The animation will move following a prescribed behavior.
Call this to set the speed of the linear motion in m/s.

Parameters:
speed (float): Desired speed.
```

6.2.4 Member Data Documentation

6.2.4.1 object

```
mavs_python.mavs_interface.MavsAnimation.object
```

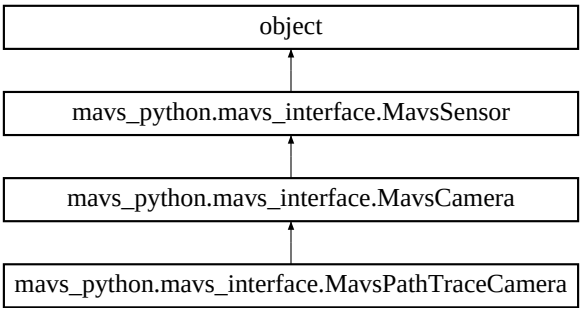
object (void): Pointer to a MAVS animation.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.3 mavs_python.mavs_interface.MavsCamera Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsCamera:



Public Member Functions

- def __init__ (self)
- def Model (self, model)
- def SaveCameraImage (self, fname)
- def Initialize (self, nx, ny, faw, fah, fl)
- def FreePose (self)
- def GetBuffer (self)
- def RenderShadows (self, shadows)
- def SetAntiAliasingFactor (self, numsamples)
- def SetEnvironmentProperties (self, env)
- def SetDropsOnLens (self, onlens)
- def SetGammaAndGain (self, gamma, gain)
- def GetDrivingCommand (self)
- def GetDimensions (self)
- def ConvertToRccb (self)

Public Attributes

- type

  type (string): Must be set to 'camera'.
- sensor

  sensor (void): Pointer to a MAVS sensor.
- gamma

  gamma (float): Camera compression factor.
- gain

  gain (float): Camera amplification factor.
- aa_fac

  aa_fac (int): Anti-aliasing factor - pixel oversampling rate.
- render_shadows

  render_shadows (bool): Set to True to render shadows.
- raindrop_lens

  raindrop_lens (bool): If raining, set to true to add raindrops to the camera lens.

6.3.1 Detailed Description

```
MavsCamera class.

Base class for several types of cameras.

Attributes:
type (string): Must be set to 'camera'.
sensor (void): Pointer to a MAVS sensor.
gamma (float): Camera compression factor.
gain (float): Camera amplification factor.
aa_fac (int): Anti-aliasing factor - pixel oversampling rate. Default is 1.
render_shadows (bool): Set to True to render shadows.
raindrop_lens (bool): If raining, set to true to add raindrops to the camera lens.
```

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def mavs_python.mavs_interface.MavsCamera.__init__ (
              self )
```

Constructor for a MavsCamera.

6.3.3 Member Function Documentation

6.3.3.1 ConvertToRccb()

```
def mavs_python.mavs_interface.MavsCamera.ConvertToRccb (
              self )
```

Convert an image generated with an RGB colormask to RCCB.

The green channel is modified by the equation.
green = 0.3*red + 0.59*green + 0.11*blue

6.3.3.2 FreePose()

```
def mavs_python.mavs_interface.MavsCamera.FreePose (
              self )
```

Make the camera moveable through the display window.

When this is called, the camera position and orientation can
be moved with the W-A-S-D keys, arrow keys, and Page Up / Page Down keys.

6.3.3.3 GetBuffer()

```
def mavs_python.mavs_interface.MavsCamera.GetBuffer (
              self )
```

Return a python list with the rgb values of the camera frame.

The buffer will read out rows in the sequentially, listing each pixel's r-g-b
as three floats.

Returns:
buffer (list of floats): The camera buffer.

6.3.3.4 GetDimensions()

```
def mavs_python.mavs_interface.MavsCamera.GetDimensions (
              self )
```

Get the dimensions of the current camera frame.

Returns:
width (int): Width of the image in pixels.
height (int): Height of the image in pixels.
depth (int): Depth of the image, usually 3.

6.3.3.5 GetDrivingCommand()

```
def mavs_python.mavs_interface.MavsCamera.GetDrivingCommand (
              self )
```

Get driving command as user input through the camera display window.

When the camera window is highlighted, user can issue driving commands
with the W-A-S-D keys.

Returns:
dc (MavsDrivingCommand): Returns a MavsDrivingCommand.

6.3.3.6 Initialize()

```
def mavs_python.mavs_interface.MavsCamera.Initialize (
              self,
              nx,
              ny,
              faw,
              fah,
              fl )
```

Initialize the camera system to a particular geometry.

Parameters:
nx (int): The number of pixels in the horizontal direction.
ny (int): The number of pixels in the vertical direction.
faw (float): The width of the focal plane in meters.
fah (float): The height of the focal plane in meters.
fl (float): The focal length in meters.

6.3.3.7 Model()

```
def mavs_python.mavs_interface.MavsCamera.Model (
              self,
              model )
```

Initialize a camera to a certain model.

Available models are :
'XCD-V60', 'Flea', 'HD1080', 'MachineVision',
'MachineVisionPathTraced', 'HDPathTraced',
'HalfHDPathTraced', 'Sf3325', 'Sf3325PathTraced',
'UavCamera', 'UavCameraPathTraced', and 'UavCameraPathTracedLow'

Parameters:
model (string): The camera model string.

6.3.3.8 RenderShadows()

```
def mavs_python.mavs_interface.MavsCamera.RenderShadows (
              self,
              shadows )
```

Turn shadow rendering on/off.

Rendering is faster with shadows turned off.

Parameters:
shadows (bool): Turn shadows on (True) or off (False).

6.3.3.9 SaveCameraImage()

```
def mavs_python.mavs_interface.MavsCamera.SaveCameraImage (
              self,
              fname )
```

Save the current camera frame to a file.

Must specify the extension. Currently .bmp and .png are supported.

Parameters:
fname (string): The file save name, including path and extension

6.3.3.10 SetAntiAliasingFactor()

```
def mavs_python.mavs_interface.MavsCamera.SetAntiAliasingFactor (
              self,
              numsamples )
```

Set the camera anti-aliasing factor.

Each pixel will be oversampled by a factor of numsamples
Default is 1, increasing numsamples also increases the rendering time

Parameters:
numsamples (int): The number of samples at each pixel

6.3.3.11 SetDropsOnLens()

```
def mavs_python.mavs_interface.MavsCamera.SetDropsOnLens (
              self,
              onlens )
```

If raining, turn raindrops on the camera lens on/off.

Parameters:
onlens (bool): Set to true for raindrops on lens, false for none.

6.3.3.12 SetEnvironmentProperties()

```
def mavs_python.mavs_interface.MavsCamera.SetEnvironmentProperties (
              self,
              env )
```

Set the environmental properties for the camera.

Give the camera a pointer to the current environment to set
properties like the sun color and position.

Parameters:
env (void): Pointer to a MavsEnvironment object.

6.3.3.13 SetGammaAndGain()

```
def mavs_python.mavs_interface.MavsCamera.SetGammaAndGain (
              self,
              gamma,
              gain )
```

Set the camera compression and gain.

Pixels are modified by is given by I = gain*I_0^gamma.

Parameters:
gamma (float): Compression value.
gain (float): Gain value.

6.3.4 Member Data Documentation

6.3.4.1 aa_fac

`mavs_python.mavs_interface.MavsCamera.aa_fac`

aa_fac (int): Anti-aliasing factor - pixel oversampling rate.

Default is 1.

6.3.4.2 gain

`mavs_python.mavs_interface.MavsCamera.gain`

gain (float): Camera amplification factor.

6.3.4.3 gamma

`mavs_python.mavs_interface.MavsCamera.gamma`

gamma (float): Camera compression factor.

6.3.4.4 raindrop_lens

`mavs_python.mavs_interface.MavsCamera.raindrop_lens`

raindrop_lens (bool): If raining, set to true to add raindrops to the camera lens.

6.3.4.5 render_shadows

`mavs_python.mavs_interface.MavsCamera.render_shadows`

render_shadows (bool): Set to True to render shadows.

6.3.4.6 sensor

`mavs_python.mavs_interface.MavsCamera.sensor`

sensor (void): Pointer to a MAVS sensor.

6.3.4.7 type

`mavs_python.mavs_interface.MavsCamera.type`

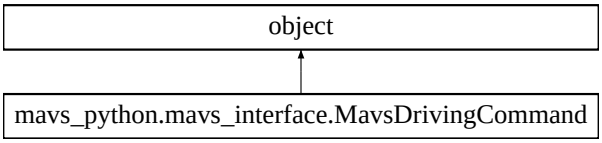
type (string): Must be set to 'camera'.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.4 mavs_python.mavs_interface.MavsDrivingCommand Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsDrivingCommand:



Public Member Functions

- def __init__ (self)

Public Attributes

- throttle

  throttle (float): Throttle value, from 0-1
- steering

  steering (float): Steering value, [-1:1]
- braking

  braking (float): Braking value, from 0-1

6.4.1 Detailed Description

```
Class the specifies the attributes of a MAVS driving command.

Attributes:
throttle (float): Throttle value, from 0-1
steering (float): Steering value, [-1:1]
braking (float): Braking value, from 0-1
```

6.4.2 Constructor & Destructor Documentation

6.4.2.1 __init__()

```
def mavs_python.mavs_interface.MavsDrivingCommand.__init__ (
              self )
```

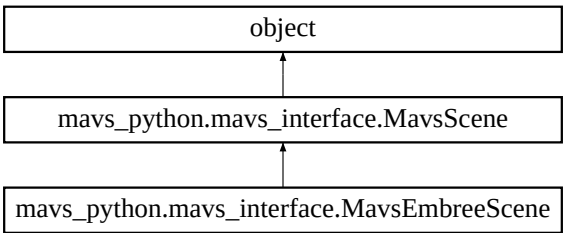
Constructor for a MavsDrivingCommand.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.5 mavs_python.mavs_interface.MavsEmbreeScene Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsEmbreeScene:



Public Member Functions

- def __init__ (self)
- def Load (self, fname)
- def LoadRandom (self, fname)

Public Attributes

- scene

  scene (void): Poiner to a MAVS Embree scene.

6.5.1 Detailed Description

```
MavsEmbreeScene class.

Has methods to load embree scenes.
Inherits from the MavsScene class.

Attributes:
scene (void): Poiner to a MAVS Embree scene.
```

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def mavs_python.mavs_interface.MavsEmbreeScene.__init__ (
              self )
```

Constructor for a MavsEmbreeScene.

6.5.3 Member Function Documentation

6.5.3.1 Load()

```
def mavs_python.mavs_interface.MavsEmbreeScene.Load (
              self,
              fname )
```

Load a json scene file.

Example input files can be found in mavs/data/scenes

Parameters:
fname (string): The scene file name, relative to the MAVS data path.

6.5.3.2 LoadRandom()

```
def mavs_python.mavs_interface.MavsEmbreeScene.LoadRandom (
              self,
              fname )
```

Load a json file with random seed for veg placement

Parameters:
fname (string): The scene file name, relative to the MAVS data path

6.5.4 Member Data Documentation

6.5.4.1 scene

`mavs_python.mavs_interface.MavsEmbreeScene.scene`

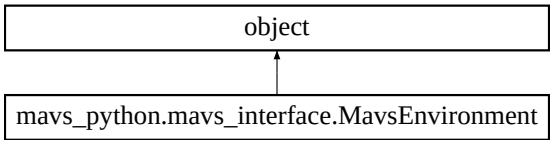
scene (void): Poiner to a MAVS Embree scene.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.6 mavs_python.mavs_interface.MavsEnvironment Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsEnvironment:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def DeleteEnvironment (self)
- def SetTerrainProperties (self, type, strength)
- def SetScene (self, scene)
- def FreeScene (self)
- def AdvanceTime (self, dt)
- def GetAnimationPosition (self, anim_num)
- def SetAnimationPosition (self, anim_id, x, y, heading)
- def AddPointLight (self, color, position)
- def AddSpotLight (self, color, position, direction, angle)
- def AddActor (self, actorfile, auto_update=True)
- def AddDustToActor (self, actor_num)
- def AddDustToLocation (self, position, velocity, dust_size, dust_rate, vel_rand_fac)
- def SetActorPosition (self, actor_id, pos, quat)
- def UpdateParticleSystems (self, dt)
- def SetRainRate (self, r)
- def SetTurbidity (self, turbid)
- def SetAlbedo (self, albedo)
- def SetFog (self, fog)
- def SetTime (self, hour)
- def SetDate (self, year, month, day)
- def SetCloudCover (self, cover)
- def SetSnow (self, snow_rate)
- def SetSnowAccumulation (self, snow_accum)
- def SetWind (self, wind)
- def load_block (self, data)
- def GetNumberOfObjects (self)
- def GetObjectBoundingBox (self, object_id)
- def GetObjectName (self, object_id)

Public Attributes

- obj

  obj (void): Pointer to a MAVS Environment.
- actor_ids

  actor_ids (list of ints): A list of ID numbers for all the actors that have been added.
- rain_rate

  rain_rate (float): Rain rate in mm/h, [0-25].
- turbidity

  turbidity (float): Turbidity (haze) factor, [2-10].
- hour

  hour (int): Time of day from 0-23.
- fog

  fog (float): Fog cover from 0-100.
- year

  year (int): The year in XXXX format.
- snow_rate

  snow_rate (float): The snow rate in mm/h [0-25].
- cloud_cover

  cloud_cover (float): The cloud cover fraction (0-1.0).
- wind

  wind (float): 2D vector specifying the lateral windspeed and direction in m/s.
- albedo

  albedo (float): Global albedo of the local terrain, (0.0-1.0).
- month

  Month of the year, 1-12.
- day

  Day of the month, 1-31.
- minute

  Minute of the hour, 0-59.
- second

  Seconds of the minute, 0-59.

6.6.1 Detailed Description

```
MavsEnvironment class.

A Mavs environment is a description of properties like the
atmosphere, weather, geo-location, and time of day.

The environment must also contain a pointer to a MavsScene, a description of the geometry.

Attributes:
obj (void): Pointer to a MAVS Environment.
actor_ids (list of ints): A list of ID numbers for all the actors that have been added.
rain_rate (float): Rain rate in mm/h, [0-25].
turbidity (float): Turbidity (haze) factor, [2-10].
hour (int): Time of day from 0-23.
fog (float): Fog cover from 0-100.
year (int): The year in XXXX format.
snow_rate (float): The snow rate in mm/h [0-25].
cloud_cover (float): The cloud cover fraction (0-1.0).
wind (float): 2D vector specifying the lateral windspeed and direction in m/s.
albedo (float): Global albedo of the local terrain, (0.0-1.0).
month (int): Month of the year, 1-12
day (int): Day of the month, 1-31
minute (int): Minute of the hour, 0-59
second (int): Seconds of the minute, 0-59
```

6.6.2 Constructor & Destructor Documentation

6.6.2.1 __init__()

```
def mavs_python.mavs_interface.MavsEnvironment.__init__ (
              self )
```

Constructor for a MavsEnvironment.

6.6.2.2 __del__()

```
def mavs_python.mavs_interface.MavsEnvironment.__del__ (
              self )
```

Destructor for a MavsEnvironment.

6.6.3 Member Function Documentation

6.6.3.1 AddActor()

```
def mavs_python.mavs_interface.MavsEnvironment.AddActor (
              self,
              actorfile,
              auto_update = True )
```

Add an actor to the environment.

An actor is a moving object like a car or pedestrian.
Example actor files can be found in mavs/data/actors.

Parameters:
actorfile (string): Name of the json file with the actor inputs.
auto_update (bool): Set to True to have MAVS automatically move the actor using the prescribed behavior.

Returns:
actor_id (int): An ID number for the actor to be used in future modifications.

6.6.3.2 AddDustToActor()

```
def mavs_python.mavs_interface.MavsEnvironment.AddDustToActor (
              self,
              actor_num )
```

Add dust behind and actor.

An actor lica a car can generate dust as it moves.
Call this to add dust to a specific actor.

Parameters:
actor_num (int): The ID of the actor to which dust will be added.

6.6.3.3 AddDustToLocation()

```
def mavs_python.mavs_interface.MavsEnvironment.AddDustToLocation (
              self,
              position,
              velocity,
              dust_size,
              dust_rate,
              vel_rand_fac )
```

Add dust at a given spot and rate.

Dust can be added to the scene at a given position,
independent of any actor.

Parameters:
position ([float, float, float]): x-y-z position of the dust in global ENU.
velocity ([float, float, float]): x-y-z velocity of the dust in global ENU.
dust_size (float): Dust ball radius in meters.
dust_rate (float): Rate that dust is added to the scene in particles/second.
vel_rand_fac (float): The velocity randomization in m/s.

6.6.3.4 AddPointLight()

```
def mavs_python.mavs_interface.MavsEnvironment.AddPointLight (
              self,
              color,
              position )
```

Add a point-light to the scene

A point light has a 1/r^2 falloff in all directions.

Parameters:
color ([float, float, float]): The RGB color of the light in the range [0:255]
position ([float, float, float]): The x-y-z position of the light in global ENU.

6.6.3.5 AddSpotLight()

```
def mavs_python.mavs_interface.MavsEnvironment.AddSpotLight (
              self,
              color,
              position,
              direction,
              angle )
```

Add a spotlight to the scene

A point light has a 1/r falloff in the direction of the light.

Parameters:
color ([float, float, float]): The RGB color of the light in the range [0:255]
position ([float, float, float]): The x-y-z position of the light in global ENU.
direction ([float, float, float]): The normalized x-y-z direction vector of the light in global ENU.
angle (float): The opening angle of the spotlight in radians.

6.6.3.6 AdvanceTime()

```
def mavs_python.mavs_interface.MavsEnvironment.AdvanceTime (
              self,
              dt )
```

Advance environment time by dt seconds.

This will move the snowflakes, actors, and
anything else dynamic in the environment.

Paramters:
dt (float): The length of time to advance in seconds.

6.6.3.7 DeleteEnvironment()

```
def mavs_python.mavs_interface.MavsEnvironment.DeleteEnvironment (
              self )
```

Free the pointer to the environment object.

6.6.3.8 FreeScene()

```
def mavs_python.mavs_interface.MavsEnvironment.FreeScene (
              self )
```

Free the pointer to the MAVS scene.

6.6.3.9 GetAnimationPosition()

```
def mavs_python.mavs_interface.MavsEnvironment.GetAnimationPosition (
              self,
              anim_num )
```

Return the position of an animation in global ENU.

Parameters:
anim_num (int): The animation ID number.

Returns:
position ([float, float, float]): The x-y-z position of the animation in global ENU.

6.6.3.10 GetNumberOfObjects()

```
def mavs_python.mavs_interface.MavsEnvironment.GetNumberOfObjects (
              self )
```

Return the total number of unique objects (meshes) in the scene.

Returns:
num_obj (int): The number of objects in the scene.

6.6.3.11 GetObjectBoundingBox()

```
def mavs_python.mavs_interface.MavsEnvironment.GetObjectBoundingBox (
              self,
              object_id )
```

Get the bound box of a particular object.

Parameters:
object_id (int): The ID number of the object in question.

Returns:
bounding_box ([float, float, float],[float, float, float]): A 2X3 list containing the lower left and upper rig

6.6.3.12 GetObjectName()

```
def mavs_python.mavs_interface.MavsEnvironment.GetObjectName (
              self,
              object_id )
```

Get the name of a particular object.

Parameters:
object_id (int): The ID number of the object in question.

Returns:
ret_name (string): The name of the object.

6.6.3.13 load_block()

```
def mavs_python.mavs_interface.MavsEnvironment.load_block (
              self,
              data )
```

Load environment parameters.

Loads environment parameters from a json dictionary.

Parameters:
data (dictionary): Data block to load.

6.6.3.14 SetActorPosition()

```
def mavs_python.mavs_interface.MavsEnvironment.SetActorPosition (
              self,
              actor_id,
              pos,
              quat )
```

Set the position of the actor in world coordinates.

Parameters:
actor_id (int): The ID number of the actor to be moved.
pos ([float,float,float]): New x-y-z position of the actor in global ENU.
quat ([float,float,float,float]): New w-x-y-z orientation of the actor in global ENU.

6.6.3.15 SetAlbedo()

```
def mavs_python.mavs_interface.MavsEnvironment.SetAlbedo (
              self,
              albedo )
```

Set the local albedo.

Albedo is average surface reflectance.

Parameters:
albedo (float): Albedo from 0-1.

6.6.3.16 SetAnimationPosition()

```
def mavs_python.mavs_interface.MavsEnvironment.SetAnimationPosition (
              self,
              anim_id,
              x,
              y,
              heading )
```

Set the position of the animation in global ENU.

The animation will be automatically locked to the ground.

Parameters:
anim_id (int): The animation ID number.
x (float): The x-positon in global ENU.
y (float): The y-positon in global ENU.
heading (float): Heading relative to East/X in radians.

6.6.3.17 SetCloudCover()

```
def mavs_python.mavs_interface.MavsEnvironment.SetCloudCover (
              self,
              cover )
```

Set the cloud cover fraction.

Set the fraction of the sky that is covered by clouds, from [0,1].

Parameters:
cover (float): The cloud cover fraction, 0.0-1.0.

6.6.3.18 SetDate()

```
def mavs_python.mavs_interface.MavsEnvironment.SetDate (
              self,
              year,
              month,
              day )
```

Set the date of the simulation.

This will influence the location of the sun and stars.

Parameters:
year (int): The year in XXXX format.
month (int): The month in 1-12 format.
day (int): Day of the year in 1-365 format.

6.6.3.19 SetFog()

```
def mavs_python.mavs_interface.MavsEnvironment.SetFog (
              self,
              fog )
```

Set the fogginess.

Parameters:
fog (float): Fog from 0-100. 100 is very foggy.

6.6.3.20 SetRainRate()

```
def mavs_python.mavs_interface.MavsEnvironment.SetRainRate (
              self,
              r )
```

Set the rain rate in the environment in mm/h.

Typical rain rates are 5-10 mm/h (light rain)
to 25 mm/h (heavy rain).

Parameters:
r (float): Rain rate in mm/h.

6.6.3.21 SetScene()

```
def mavs_python.mavs_interface.MavsEnvironment.SetScene (
              self,
              scene )
```

Set the pointer to the MAVS scene.

Parameters:
scene (void): Pointer to the MAVS scene.

6.6.3.22 SetSnow()

```
def mavs_python.mavs_interface.MavsEnvironment.SetSnow (
              self,
              snow_rate )
```

Set the snow rate.

Parameters:
snow_rate (float): Snow rate in mm/h, 0.0-25.0.

6.6.3.23 SetSnowAccumulation()

```
def mavs_python.mavs_interface.MavsEnvironment.SetSnowAccumulation (
              self,
              snow_accum )
```

Set the snow accumalation factor.

Parameters:
snow_accum (float): Snow accumulation factor in mm/hour.

6.6.3.24 SetTerrainProperties()

```
def mavs_python.mavs_interface.MavsEnvironment.SetTerrainProperties (
              self,
              type,
              strength )
```

Set the soil type and strength of the terrain.

Parameters:
type (string): Can be 'dry', 'wet', 'snow', 'clay', or 'sand'.
strength (float): Soil strngth in Cone-Index (PSI).

6.6.3.25 SetTime()

```
def mavs_python.mavs_interface.MavsEnvironment.SetTime (
              self,
              hour )
```

Set the hour in military (0-23) time.

Parameters:
hour (int): The hour from 0 (midnight) to 23 (11 PM).

6.6.3.26 SetTurbidity()

```
def mavs_python.mavs_interface.MavsEnvironment.SetTurbidity (
              self,
              turbid )
```

Set the turbidity of the atmosphere.

Turbidity is a measure of haziness. It should range from
2 (very clear) to 10 (very hazy)

Parameters:
turbid (float): The turbidity index.

6.6.3.27 SetWind()

```
def mavs_python.mavs_interface.MavsEnvironment.SetWind (
              self,
              wind )
```

Set the wind speed and direction.

Parameters:
wind ([float, float]): The lateral wind speed and direction in m/s.

6.6.3.28 UpdateParticleSystems()

```
def mavs_python.mavs_interface.MavsEnvironment.UpdateParticleSystems (
              self,
              dt )
```

Call this to update particle systems in the environment.

Particle systems include smoke and dust.
This will be called automatically if 'AdvanceTime(dt)'
method is invoked.

Parameters:
dt (float): The time step of the update in seconds.

6.6.4 Member Data Documentation

6.6.4.1 actor_ids

```
mavs_python.mavs_interface.MavsEnvironment.actor_ids
```

actor_ids (list of ints): A list of ID numbers for all the actors that have been added.

6.6.4.2 albedo

```
mavs_python.mavs_interface.MavsEnvironment.albedo
```

albedo (float): Global albedo of the local terrain, (0.0-1.0).

6.6.4.3 cloud_cover

`mavs_python.mavs_interface.MavsEnvironment.cloud_cover`

cloud_cover (float): The cloud cover fraction (0-1.0).

6.6.4.4 fog

`mavs_python.mavs_interface.MavsEnvironment.fog`

fog (float): Fog cover from 0-100.

6.6.4.5 hour

`mavs_python.mavs_interface.MavsEnvironment.hour`

hour (int): Time of day from 0-23.

6.6.4.6 obj

`mavs_python.mavs_interface.MavsEnvironment.obj`

obj (void): Pointer to a MAVS Environment.

6.6.4.7 rain_rate

`mavs_python.mavs_interface.MavsEnvironment.rain_rate`

rain_rate (float): Rain rate in mm/h, [0-25].

6.6.4.8 snow_rate

`mavs_python.mavs_interface.MavsEnvironment.snow_rate`

snow_rate (float): The snow rate in mm/h [0-25].

6.6.4.9 turbidity

`mavs_python.mavs_interface.MavsEnvironment.turbidity`

turbidity (float): Turbidity (haze) factor, [2-10].

6.6.4.10 wind

`mavs_python.mavs_interface.MavsEnvironment.wind`

wind (float): 2D vector specifying the lateral windspeed and direction in m/s.

6.6.4.11 year

`mavs_python.mavs_interface.MavsEnvironment.year`

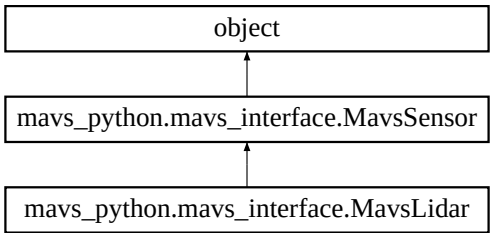
year (int): The year in XXXX format.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.7 mavs_python.mavs_interface.MavsLidar Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsLidar:



Public Member Functions

- def __init__ (self, type)
- def __del__ (self)
- def SetScanPattern (self, horiz_fov_low, horiz_fov_high, horiz_res, vert_fov_low, vert_fov_high, vert_res)
- def SaveLidarImage (self, fname)
- def SaveProjectedLidarImage (self, fname)
- def DisplayPerspective (self, width=768, height=256)
- def SaveColorizedPointCloud (self, fname)
- def SavePcd (self, fname)
- def SetVelocity (self, vx, vy, vz)
- def SaveLabeledPcd (self, fname)
- def SaveLabeledPcdWithNormals (self, fname)
- def SaveLabeledPointCloud (self, fname)
- def GetPoints (self)
- def GetUnRegisteredPointsXYZIL (self)
- def SetDisplayColorType (self, type)
- def AnalyzeCloud (self, fname, frame_num, display)

Public Attributes

- type

  type (string): Must be 'lidar'.
- sensor

  sensor (void): Pointer to a MAVS sensor.

6.7.1 Detailed Description

```
MavsLidar class.

The API allows you to select from a variety of lidar models.

Attributes:
type (string): Must be 'lidar'.
sensor (void): Pointer to a MAVS sensor.
```

6.7.2 Constructor & Destructor Documentation

6.7.2.1 __init__()

```
def mavs_python.mavs_interface.MavsLidar.__init__ (
              self,
              type )
```

```
Construct a MavsLidar.

Available models are 'HDL-32E', 'HDL-64E', 'M8','OS1', 'OS2',
'LMS-291', 'VLP-16' and 'RS32'

Parameters:
type (string): The model of the lidar.
```

6.7.2.2 __del__()

```
def mavs_python.mavs_interface.MavsLidar.__del__ (
              self )
```

```
Destructor for MavsLidar.
```

6.7.3 Member Function Documentation

6.7.3.1 AnalyzeCloud()

```
def mavs_python.mavs_interface.MavsLidar.AnalyzeCloud (
              self,
              fname,
              frame_num,
              display )
```

Automatically annotates the point cloud and saves it to a file.

Parameters:
fname (string): The output file name, without extension.
frame_num (int): The number of the current frame.
display (bool): True to display the result to the screen, False otherwise.

6.7.3.2 DisplayPerspective()

```
def mavs_python.mavs_interface.MavsLidar.DisplayPerspective (
              self,
              width = 768,
              height = 256 )
```

Display a perspective view of the point cloud to the screen.

Parameters:
width (int): The width of the display.
height (int): The height of the display.

6.7.3.3 GetPoints()

```
def mavs_python.mavs_interface.MavsLidar.GetPoints (
              self )
```

Get a list of the x-y-z points in the point cloud.

Returns a Nx3 list of points where N is the number of returns.
Points are 'registered' to world coordinates.

Returns:
points (list of floats): The x-y-z point cloud.

6.7.3.4 GetUnRegisteredPointsXYZIL()

```
def mavs_python.mavs_interface.MavsLidar.GetUnRegisteredPointsXYZIL (
              self )
```

Get a list of the x-y-z-intensity-label points in the point cloud.

Returns a Nx5 list of points where N is the number of returns.
Points are 'unregistered', or in the sensor frame.

Returns:
points (list of floats): The x-y-z-intensity-label point cloud.

6.7.3.5 SaveColorizedPointCloud()

```
def mavs_python.mavs_interface.MavsLidar.SaveColorizedPointCloud (
              self,
              fname )
```

Save the current lidar point cloud to a column file.

Saves x,y,z,intensity,r,g,b to a space delimited text file.

Parameters:
fname (string): The output file name, including path and extension.

6.7.3.6 SaveLabeledPcd()

```
def mavs_python.mavs_interface.MavsLidar.SaveLabeledPcd (
              self,
              fname )
```

Save the current lidar point cloud to a Point Cloud Library pcd file.

Saves a column file with x,y,z,intensity,label
where labelnum is an int defined in the labels.json file.

Parameters:
fname (string): The output file name, including path and extension.

6.7.3.7 SaveLabeledPcdWithNormals()

```
def mavs_python.mavs_interface.MavsLidar.SaveLabeledPcdWithNormals (
              self,
              fname )
```

Save the current lidar point cloud to a Point Cloud Library pcd file.

Saves a column file with x,y,z,intensity,normal_x, normal_y, normal_z,label
where labelnum is an int defined in the labels.json file.

Parameters:
fname (string): The output file name, including path and extension.

6.7.3.8 SaveLabeledPointCloud()

```
def mavs_python.mavs_interface.MavsLidar.SaveLabeledPointCloud (
              self,
              fname )
```

Save the current lidar point cloud to a column text file.

Saves a column file with x,y,z,intensity,labelnum
where labelnum is an int defined in the labels.json file.

Parameters:
fname (string): The output file name, including path and extension.

6.7.3.9 SaveLidarImage()

```
def mavs_python.mavs_interface.MavsLidar.SaveLidarImage (
              self,
              fname )
```

Save the current lidar point cloud to a top-down image.

Parameters:
fname (string): The output file name, including path and extension.

6.7.3.10 SavePcd()

```
def mavs_python.mavs_interface.MavsLidar.SavePcd (
              self,
              fname )
```

Save the current lidar point cloud to a Point Cloud Library pcd file.

Saves a column file with x,y,z,intensity

Parameters:
fname (string): The output file name, including path and extension.

6.7.3.11 SaveProjectedLidarImage()

```
def mavs_python.mavs_interface.MavsLidar.SaveProjectedLidarImage (
              self,
              fname )
```

Save the current lidar point cloud to a projected image.

Saves a "first-person" view of the lidar point cloud.

Parameters:
fname (string): The output file name, including path and extension

6.7.3.12 SetDisplayColorType()

```
def mavs_python.mavs_interface.MavsLidar.SetDisplayColorType (
              self,
              type )
```

Set the colorization of the lidar display.

Options are 'height', 'color', 'range', 'intensity', or 'white'.

Parameters:
type (string): The display type.

6.7.3.13 SetScanPattern()

```
def mavs_python.mavs_interface.MavsLidar.SetScanPattern (
              self,
              horiz_fov_low,
              horiz_fov_high,
              horiz_res,
              vert_fov_low,
              vert_fov_high,
              vert_res )
```

Set the scan patter of the lidar.

This will override the default for the model when the lidar was created.

The angular inputs are in degrees, not radians.

Parameters:
horiz_fov_low (float): The low value (degrees) of the horizontal field-of-view.
horiz_fov_high (float): The high value (degrees) of the horizontal field-of-view.
horiz_res (float): The resolution (degrees) of the horizontal field-of-view.
vert_fov_low (float): The low value (degrees) of the vertical field-of-view.
vert_fov_high (float): The high value (degrees) of the vertical field-of-view.
vert_res (float): The resolution (degrees) of the vertical field-of-view.

6.7.3.14 SetVelocity()

```
def mavs_python.mavs_interface.MavsLidar.SetVelocity (
              self,
              vx,
              vy,
              vz )
```

Set the linear velocity of the lidar sensor.

When this is set, the in-scan motion of the sensor will be calculated,
where a scan is one complete revolution of the sensor.

Parameters:
vx (float) The velocity in the global x direction.
vy (float) The velocity in the global y direction.
vz (float) The velocity in the global z direction.

6.7.4 Member Data Documentation

6.7.4.1 sensor

`mavs_python.mavs_interface.MavsLidar.sensor`

sensor (void): Pointer to a MAVS sensor.

6.7.4.2 type

`mavs_python.mavs_interface.MavsLidar.type`

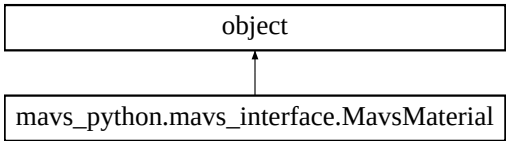
type (string): Must be 'lidar'.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.8 mavs_python.mavs_interface.MavsMaterial Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsMaterial:



Public Member Functions

- def __init__ (self)

Public Attributes

- name

  name (string): The material name.
- ka

  ka ([float, float, float]): The rgb ambient reflectnace.
- kd

  kd ([float, float, float]): The rgb diffuse reflectance.
- ks

  ks ([float, float, float]): The rgb specular reflectance.
- tr

  tr ([float, float, float]): Transmission coefficient.
- ke

  ke ([float, float, float]): The rgb emission.
- ns

  ns (float): Specular exponent.
- ni

  ni (float): Index of refraction.
- dissolve

  dissolve (float): Not used.
- illum

  illum (int): Reflectance model.
- map_kd

  map_kd (string): Diffuse texture map.
- map_ka

  map_ka (string): Ambient texture map.
- map_ks

  map_ks (string): Specular texture map.
- map_ns

  map_ns (string): Phong exponent texture map.
- map_bump

  map_bump (string): Normal map.
- map_d

  map_d (string): Transparency map.
- disp

  disp (string): Height texture map.
- refl

  refl (string): Spectral reflectance file.

6.8.1 Detailed Description

```
Class that defines python materials.

Attributes:
name (string): The material name.
ka ([float, float, float]): The rgb ambient reflectnace.
kd ([float, float, float]): The rgb diffuse reflectance.
ks ([float, float, float]): The rgb specular reflectance.
ke ([float, float, float]): The rgb emission.
tr ([float, float, float]): Transmission coefficient.
ns (float): Specular exponent.
ni (float): Index of refraction.
dissolve (float): Not used.
illum (int): Reflectance model.
```

```
map_kd (string): Diffuse texture map.
map_ka (string): Ambient texture map.
map_ks (string): Specular texture map.
map_ns (string): Phong exponent texture map.
map_bump (string): Normal map.
map_d (string): Transparency map.
disp (string): Height texture map.
refl (string): Spectral reflectance file.
```

6.8.2 Constructor & Destructor Documentation

6.8.2.1 __init__()

```
def mavs_python.mavs_interface.MavsMaterial.__init__ (
              self )
```

MavsMaterial constructor.

6.8.3 Member Data Documentation

6.8.3.1 disp

mavs_python.mavs_interface.MavsMaterial.disp

disp (string): Height texture map.

6.8.3.2 dissolve

mavs_python.mavs_interface.MavsMaterial.dissolve

dissolve (float): Not used.

6.8.3.3 illum

mavs_python.mavs_interface.MavsMaterial.illum

illum (int): Reflectance model.

6.8.3.4 ka

mavs_python.mavs_interface.MavsMaterial.ka

ka ([float, float, float]): The rgb ambient reflectnace.

6.8.3.5 kd

mavs_python.mavs_interface.MavsMaterial.kd

kd ([float, float, float]): The rgb diffuse reflectance.

6.8.3.6 ke

mavs_python.mavs_interface.MavsMaterial.ke

ke ([float, float, float]): The rgb emission.

6.8.3.7 ks

mavs_python.mavs_interface.MavsMaterial.ks

ks ([float, float, float]): The rgb specular reflectance.

6.8.3.8 map_bump

mavs_python.mavs_interface.MavsMaterial.map_bump

map_bump (string): Normal map.

6.8.3.9 map_d

mavs_python.mavs_interface.MavsMaterial.map_d

map_d (string): Transparency map.

6.8.3.10 map_ka

mavs_python.mavs_interface.MavsMaterial.map_ka

map_ka (string): Ambient texture map.

6.8.3.11 map_kd

mavs_python.mavs_interface.MavsMaterial.map_kd

map_kd (string): Diffuse texture map.

6.8.3.12 map_ks

mavs_python.mavs_interface.MavsMaterial.map_ks

map_ks (string): Specular texture map.

6.8.3.13 map_ns

mavs_python.mavs_interface.MavsMaterial.map_ns

map_ns (string): Phong exponent texture map.

6.8.3.14 name

mavs_python.mavs_interface.MavsMaterial.name

name (string): The material name.

6.8.3.15 ni

mavs_python.mavs_interface.MavsMaterial.ni

ni (float): Index of refraction.

6.8.3.16 ns

`mavs_python.mavs_interface.MavsMaterial.ns`

ns (float): Specular exponent.

6.8.3.17 refl

`mavs_python.mavs_interface.MavsMaterial.refl`

refl (string): Spectral reflectance file.

6.8.3.18 tr

`mavs_python.mavs_interface.MavsMaterial.tr`

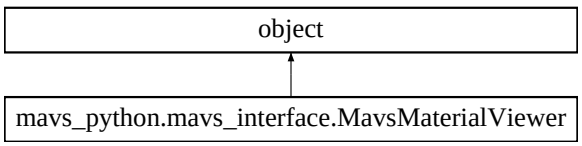
tr ([float, float, float]): Transmission coefficient.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.9 mavs_python.mavs_interface.MavsMaterialViewer Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsMaterialViewer:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def Update (self)
- def SetMaterial (self, matnum)
- def LoadMaterialsFromObj (self, meshfile)

Public Attributes

- viewer

  viewer (void): Pointer to MAVS material viewer.
- mat_name_list

  mat_name_list (list of strings): Names of available materials.
- avail_materials

  avail_materials (list of MavsMaterials): List of loaded materials.
- num_mats

  num_mats (int): Number of loaded materials.

6.9.1 Detailed Description

```
Viewer for MAVS materials.

Attributes:
viewer (void): Pointer to MAVS material viewer.
mat_name_list (list of strings): Names of available materials.
avail_materials (list of MavsMaterials): List of loaded materials.
num_mats (int): Number of loaded materials.
```

6.9.2 Constructor & Destructor Documentation

6.9.2.1 __init__()

```
def mavs_python.mavs_interface.MavsMaterialViewer.__init__ (
              self )
```

```
Constructor for the MavsMaterialViewer.
```

6.9.2.2 __del__()

```
def mavs_python.mavs_interface.MavsMaterialViewer.__del__ (
              self )
```

```
Destructor for the MavsMaterialViewer.
```

6.9.3 Member Function Documentation

6.9.3.1 LoadMaterialsFromObj()

```
def mavs_python.mavs_interface.MavsMaterialViewer.LoadMaterialsFromObj (
              self,
              meshfile )
```

Load all the materials from a given obj file.

Parameters:
meshfile (string): FUll path to the .obj file to load.

6.9.3.2 SetMaterial()

```
def mavs_python.mavs_interface.MavsMaterialViewer.SetMaterial (
              self,
              matnum )
```

Set the current matieral.

Parameters:
matnum (int): The number of the material to view.

6.9.3.3 Update()

```
def mavs_python.mavs_interface.MavsMaterialViewer.Update (
              self )
```

Update the viewer.

6.9.4 Member Data Documentation

6.9.4.1 avail_materials

```
mavs_python.mavs_interface.MavsMaterialViewer.avail_materials
```

avail_materials (list of MavsMaterials): List of loaded materials.

6.9.4.2 mat_name_list

`mavs_python.mavs_interface.MavsMaterialViewer.mat_name_list`

mat_name_list (list of strings): Names of available materials.

6.9.4.3 num_mats

`mavs_python.mavs_interface.MavsMaterialViewer.num_mats`

num_mats (int): Number of loaded materials.

6.9.4.4 viewer

`mavs_python.mavs_interface.MavsMaterialViewer.viewer`

viewer (void): Pointer to MAVS material viewer.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.10 mavs_python.mavs_interface.MavsMems Class Reference

Public Member Functions

- def __init__ (self, type)
- def SetMeasurementRange (self, range)
- def SetMeasurementResolution (self, resolution)
- def SetConstantBias (self, bias)
- def SetNoiseDensity (self, nd)
- def SetBiasInstability (self, bi)
- def SetAxisMisalignment (self, ma)
- def SetRandomWalk (self, rw)
- def SetTemperatureBias (self, tb)
- def SetTemperatureScaleFactor (self, tsf)
- def SetAccelerationBias (self, ab)
- def Update (self, accel_in, temperature, sample_rate)

Public Attributes

- type

  type (string): Can be 'gyro', 'accelerometer', or 'magnetometer'.
- sensor

  sensor (void): Pointer to MAVS mems sensor.

6.10.1 Detailed Description

```
Mems sensor class.

MEMS sensors include accelerometers, gyroscopes, and magnetometers.

See mavs_imu.pdf for additional documentation of parameters.

Attributes:
type (string): Can be 'gyro', 'accelerometer', or 'magnetometer'.
sensor (void): Pointer to MAVS mems sensor.
```

6.10.2 Constructor & Destructor Documentation

6.10.2.1 __init__()

```
def mavs_python.mavs_interface.MavsMems.__init__ (
              self,
              type )

Constructor for a MavsMems sensor.
```

6.10.3 Member Function Documentation

6.10.3.1 SetAccelerationBias()

```
def mavs_python.mavs_interface.MavsMems.SetAccelerationBias (
              self,
              ab )

Set the acceleration bias error for the sensor.

Parameters:
ab (float): The acceleration bias value.
```

6.10.3.2 SetAxisMisalignment()

```
def mavs_python.mavs_interface.MavsMems.SetAxisMisalignment (
              self,
              ma )

Set the misalignment error for the sensor.

Parameters:
ma (float): The misalignment error value.
```

6.10.3.3 SetBiasInstability()

```
def mavs_python.mavs_interface.MavsMems.SetBiasInstability (
              self,
              bi )
```

Set the bias instability error for the sensor.

Parameters:
bi (float): The bias instability value.

6.10.3.4 SetConstantBias()

```
def mavs_python.mavs_interface.MavsMems.SetConstantBias (
              self,
              bias )
```

Set the constant bias error for the sensor.

Parameters:
bias (float): The constant bias value.

6.10.3.5 SetMeasurementRange()

```
def mavs_python.mavs_interface.MavsMems.SetMeasurementRange (
              self,
              range )
```

Set the measurment range of the sensor.

Parameters:
range (float): The maximum measurement value.

6.10.3.6 SetMeasurementResolution()

```
def mavs_python.mavs_interface.MavsMems.SetMeasurementResolution (
              self,
              resolution )
```

Set the measurement range for the sensor.

Parameters:
resolution (float): The measurement resolution.

6.10.3.7 SetNoiseDensity()

```
def mavs_python.mavs_interface.MavsMems.SetNoiseDensity (
              self,
              nd )
```

Set the noise density error for the sensor.

Parameters:
nd (float): The noise density value.

6.10.3.8 SetRandomWalk()

```
def mavs_python.mavs_interface.MavsMems.SetRandomWalk (
              self,
              rw )
```

Set the random walk error for the sensor.

Parameters:
rw (float): The random walk error value.

6.10.3.9 SetTemperatureBias()

```
def mavs_python.mavs_interface.MavsMems.SetTemperatureBias (
              self,
              tb )
```

Set the temperature bias error for the sensor.

Parameters:
tb (float): The temperature bias value.

6.10.3.10 SetTemperatureScaleFactor()

```
def mavs_python.mavs_interface.MavsMems.SetTemperatureScaleFactor (
              self,
              tsf )
```

Set the temperature scale error for the sensor.

Parameters:
tsf (float): The temperature scale error value.

6.10.3.11 Update()

```
def mavs_python.mavs_interface.MavsMems.Update (
              self,
              accel_in,
              temperature,
              sample_rate )
```

```
Update the sensor and return the 3-axis measurement.

Parameters:
accel_in ([float, float, float]): True acceleration / signal for the sensor.
temperature (float): Ambient temperature in degrees Celsius.
sample_rate (float): Sample rate in Hz.
```

6.10.4 Member Data Documentation

6.10.4.1 sensor

```
mavs_python.mavs_interface.MavsMems.sensor
```

sensor (void): Pointer to MAVS mems sensor.

6.10.4.2 type

```
mavs_python.mavs_interface.MavsMems.type
```

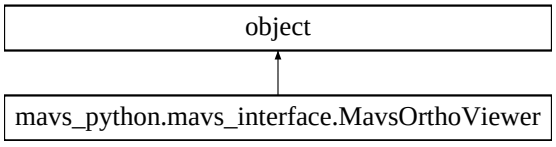
type (string): Can be 'gyro', 'accelerometer', or 'magnetometer'.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.11 mavs_python.mavs_interface.MavsOrthoViewer Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsOrthoViewer:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def SetWaypoints (self, waypoints)
- def Update (self, env, veh_pos)

Public Attributes

- viewer

  viewer (void): Pointer to a MAVS OrthoViewer.

6.11.1 Detailed Description

```
Camera class for a parallel ray camera.

The orth viewer is a top-down, parallel ray renderer with no shadows.

Attributes:
viewer (void): Pointer to a MAVS OrthoViewer.
```

6.11.2 Constructor & Destructor Documentation

6.11.2.1 __init__()

```
def mavs_python.mavs_interface.MavsOrthoViewer.__init__ (
              self )

Constructor for MavsOrthoViewer class.

Attributes:
viewer (void): Pointer to a MAVS camera.
```

6.11.2.2 __del__()

```
def mavs_python.mavs_interface.MavsOrthoViewer.__del__ (
              self )

Destructor for MavsOrthoViewer class.
```

6.11.3 Member Function Documentation

6.11.3.1 SetWaypoints()

```
def mavs_python.mavs_interface.MavsOrthoViewer.SetWaypoints (
              self,
              waypoints )
```

```
Create a set of waypoints for the camera to follow.

A set of "on the ground" waypoints that the camera will follow.
Parameters:
waypoints (MavsWaypoints): The list of waypoints.
```

6.11.3.2 Update()

```
def mavs_python.mavs_interface.MavsOrthoViewer.Update (
              self,
              env,
              veh_pos )
```

```
Update the OrthoViewer
```

6.11.4 Member Data Documentation

6.11.4.1 viewer

```
mavs_python.mavs_interface.MavsOrthoViewer.viewer
```

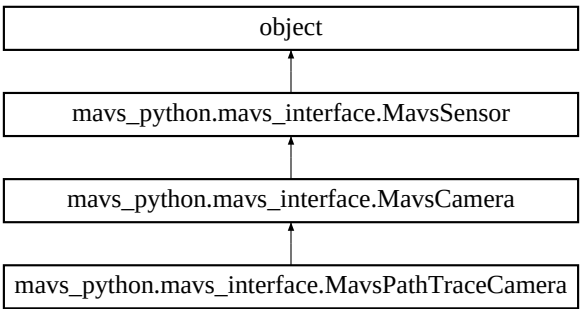
viewer (void): Pointer to a MAVS OrthoViewer.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.12 mavs_python.mavs_interface.MavsPathTraceCamera Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsPathTraceCamera:



Public Member Functions

- def __init__ (self, type, numrays, raydepth, rr_cutoff)
- def SetNormalizationType (self, type)
- def SetFixPixels (self, fix)

Public Attributes

- sensor

  sensor (void): Pointer to a MAVS sensor.

6.12.1 Detailed Description

```
Path tracer camera that inherits from the MavsCamera class.

The path tracer camera uses physics-based path tracing to render
an image. It is much slower than the default camera but makes
nicer images.

Attributes:
sensor (void): Pointer to a MAVS sensor.
```

6.12.2 Constructor & Destructor Documentation

6.12.2.1 __init__()

```
def mavs_python.mavs_interface.MavsPathTraceCamera.__init__ (
              self,
              type,
              numrays,
              raydepth,
              rr_cutoff )
```

```
Constructor for a MavsPathTracerCamera.

Parameters:
numrays (int): The number of rays per pixel.
raydepth (int): The maximum number of reflections for a ray.
rr_cutoff (float): The intensity cutoff factor for a ray [0:1]
```

6.12.3 Member Function Documentation

6.12.3.1 SetFixPixels()

```
def mavs_python.mavs_interface.MavsPathTraceCamera.SetFixPixels (
              self,
              fix )
```

```
Fix bad pixels in the path traced image.

Because path-tracing involves random sampling,
there are occasionally outlier pixels if the sampling factor is too low.
Calling this will instruct the ray-tracer to attempt to identify these
outlier pixels afer the rendering pass and set them to match the local average.

Note that for very low sampling rates where the image is very grainy,
this process may produce undesirable results.

Parameters:
fix (bool): Set to True to fix bad pixels, False to ignore.
```

6.12.3.2 SetNormalizationType()

```
def mavs_python.mavs_interface.MavsPathTraceCamera.SetNormalizationType (
              self,
              type )
```

```
Set the type of image normalization.

Options are 'max' or 'average'.
If 'max' is set, the image will be normalized so that the
maximum pixel intensity = 255.
if 'average' is selected, the image will be normalized so that
the average pixel intensity is 128

Default is 'average'.

Parameters:
type (string): The normalization type.
```

6.12.4 Member Data Documentation

6.12.4.1 sensor

```
mavs_python.mavs_interface.MavsPathTraceCamera.sensor
```

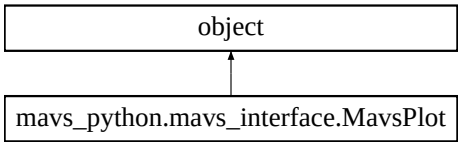
sensor (void): Pointer to a MAVS sensor.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.13 mavs_python.mavs_interface.MavsPlot Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsPlot:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def PlotColorMatrix (self, data)
- def PlotGrayMatrix (self, data)
- def PlotFlatGrayscale (self, width, height, data)
- def PlotTrajectory (self, x, y)
- def AddToTrajectory (self, x, y)

Public Attributes

- plot

  plot (void): Pointer to a MAVS plot.

6.13.1 Detailed Description

```
Class for plotting matrices and lines using CImg through the MAVS interface.

Allows for easier creation of animations / dynamically updated plots
that default python functions.

Attributes:
plot (void): Pointer to a MAVS plot.
```

6.13.2 Constructor & Destructor Documentation

6.13.2.1 __init__()

```
def mavs_python.mavs_interface.MavsPlot.__init__ (
              self )
```

```
Constructor for MavsPlot.
```

6.13.2.2 __del__()

```
def mavs_python.mavs_interface.MavsPlot.__del__ (
              self )
```

Destructor for MavsPlot.

6.13.3 Member Function Documentation

6.13.3.1 AddToTrajectory()

```
def mavs_python.mavs_interface.MavsPlot.AddToTrajectory (
              self,
              x,
              y )
```

Add more points to an existing trajectory plot.

Parameters:
x (float): List of x coordinates to add.
y (float): List of y coordinates to add.

6.13.3.2 PlotColorMatrix()

```
def mavs_python.mavs_interface.MavsPlot.PlotColorMatrix (
              self,
              data )
```

Plot a color matrix to the screen.

Parameters:
data (list): An width x height x 3 array to plot.

6.13.3.3 PlotFlatGrayscale()

```
def mavs_python.mavs_interface.MavsPlot.PlotFlatGrayscale (
              self,
              width,
              height,
              data )
```

Plot a flattened grayscale matrix.

Parameters:
width (int): width of the array
height (int): height of the array
data (list): a flattened width x height array to plot

6.13.3.4 PlotGrayMatrix()

```
def mavs_python.mavs_interface.MavsPlot.PlotGrayMatrix (
              self,
              data )
```

```
Plot a grayscale matrix

Parameters:
data (list): An width x height x 3 array to plot.
```

6.13.3.5 PlotTrajectory()

```
def mavs_python.mavs_interface.MavsPlot.PlotTrajectory (
              self,
              x,
              y )
```

```
Plot a trajectory as a sequence of x-y points.

Parameters:
x (float): List of x coordinates.
y (float): List of y coordinates.
```

6.13.4 Member Data Documentation

6.13.4.1 plot

```
mavs_python.mavs_interface.MavsPlot.plot
```

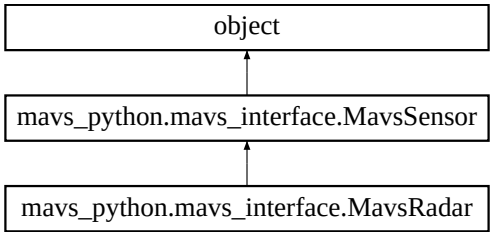
plot (void): Pointer to a MAVS plot.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.14 mavs_python.mavs_interface.MavsRadar Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsRadar:



Public Member Functions

- def __init__ (self)
- def SetMaxRange (self, mr)
- def SetFieldOfView (self, fov)
- def SaveImage (self, fname)
- def GetTargets (self)

Public Attributes

- type

  type (string): Must be set to 'radar'.
- sensor

  sensor (void): Pointer to a MAVS sensor.

6.14.1 Detailed Description

```
MavsRadar model.

Uses ray-tracing and target cross-sections based solely on size.
Inherits from MavsSensor.

Attributes:
type (string): Must be set to 'radar'.
sensor (void): Pointer to a MAVS sensor.
```

6.14.2 Constructor & Destructor Documentation

6.14.2.1 __init__()

```
def mavs_python.mavs_interface.MavsRadar.__init__ (
              self )

MavsRadar constructor.
```

6.14.3 Member Function Documentation

6.14.3.1 GetTargets()

```
def mavs_python.mavs_interface.MavsRadar.GetTargets (
              self )

Get a list containing the x-y position of all the returned targets.

The target locations are in the sensor frame.
Returns a list of Nx2 points where N is the number of targets.

Returns:
xy_targ (list of floats): x-y positions of each target.
```

6.14.3.2 SaveImage()

```
def mavs_python.mavs_interface.MavsRadar.SaveImage (
              self,
              fname )
```

Save the current radar scan to a top-down image.

Parameters:
fname (string): The output file name, including path and extension.

6.14.3.3 SetFieldOfView()

```
def mavs_python.mavs_interface.MavsRadar.SetFieldOfView (
              self,
              fov )
```

Set the horizontal field of view of the radar, in degrees.

Parameters:
fov (float): The horizontal field of view in degrees.

6.14.3.4 SetMaxRange()

```
def mavs_python.mavs_interface.MavsRadar.SetMaxRange (
              self,
              mr )
```

Set the maximum range of the radar, in meters.

Parameters:
mr (float): The maximum range in meters.

6.14.4 Member Data Documentation

6.14.4.1 sensor

```
mavs_python.mavs_interface.MavsRadar.sensor
```

sensor (void): Pointer to a MAVS sensor.

6.14.4.2 type

`mavs_python.mavs_interface.MavsRadar.type`

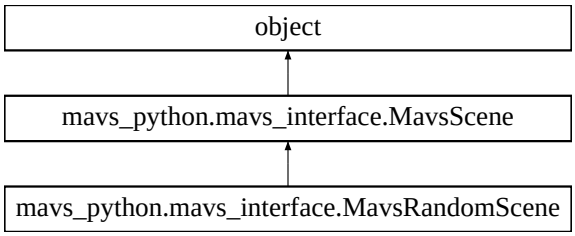
type (string): Must be set to 'radar'.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.15 mavs_python.mavs_interface.MavsRandomScene Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsRandomScene:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def AddPotholeAt (self, x, y, depth, diameter)
- def CreateScene (self)

Public Attributes

- scene

  scene (void): Pointer to a MAVS scene.
- terrain_width

  terrain_width (float): Width (x-dimension) of the terrain in meters.
- terrain_length

  terrain_length (float): Length (y-dimension) of the terrain in meters.
- lo_mag

  lo_mag (float): Magnitude in meters of the low-frequency terrain roughness.
- hi_mag

  hi_mag (float): Magnitude in meters of the high-frequency terrain roughness.
- mesh_resolution

  mesh_resolution (float): Resolution of the mesh in meters.
- trail_width

  trail_width (float): Width of the automatically generated trail in meters.
- track_width

  track_width (float): Width of the tire tracks on the trail in meters.
- wheelbase

wheelbase (float): Distance between tire tracks on the trail in meters.

- pothole_depth

  pothole_depth (float): Depth of potholes in the trail in meters.
- pothole_diameter

  pothole_diameter (float): Diameter of the potholes in the trail in meters.
- num_potholes

  num_potholes (int): Total number of potholes in the scene.
- path_type

  path_type (string): Type of automatically generated trail.
- eco_file

  eco_file (string): The ecosystem file to generate vegetation distribution.
- output_directory

  output_directory (string): Directory to save the generated scene.
- basename

  basename (string): Naming to use for all the generated output files.
- plant_density

  plant_density (float): Density of vegetation from [0:1]
- pothole_locations

  pothole_locations (list of floats): Nx2 list where N is the number of potholes and x-y position is specified.

6.15.1 Detailed Description

```
MavsRandomScene class.

A random scene is created with specific terrain and vegetation properties.
Inherits from the MavsScene class.

Attributes:
scene (void): Pointer to a MAVS scene.
terrain_width (float): Width (x-dimension) of the terrain in meters.
terrain_length (float): Length (y-dimension) of the terrain in meters.
lo_mag (float): Magnitude in meters of the low-frequency terrain roughness.
hi_mag (float): Magnitude in meters of the high-frequency terrain roughness.
mesh_resolution (float): Resolution of the mesh in meters.
trail_width (float): Width of the automatically generated trail in meters.
track_width (float): Width of the tire tracks on the trail in meters.
wheelbase (float): Distance between tire tracks on the trail in meters.
pothole_depth (float): Depth of potholes in the trail in meters.
pothole_diameter (float): Diameter of the potholes in the trail in meters.
num_potholes (int): Total number of potholes in the scene.
path_type (string): Type of automatically generated trail. Options are 'Loop', 'Ridges', or 'Valleys'.
eco_file (string): The ecosystem file to generate vegetation distribution. Examples in mavs/data/ecosystem_fil
output_directory (string): Directory to save the generated scene. cwd is the default.
basename (string): Naming to use for all the generated output files.
plant_density (float): Density of vegetation from [0:1]
pothole_locations (list of floats): Nx2 list where N is the number of potholes and x-y position is specified.
```

6.15.2 Constructor & Destructor Documentation

6.15.2.1 __init__()

```
def mavs_python.mavs_interface.MavsRandomScene.__init__ (
              self )

MavsRandomScene constructor.
```

6.15.2.2 __del__()

```
def mavs_python.mavs_interface.MavsRandomScene.__del__ (
              self )
```

MavsRandomScene destructor.

6.15.3 Member Function Documentation

6.15.3.1 AddPotholeAt()

```
def mavs_python.mavs_interface.MavsRandomScene.AddPotholeAt (
              self,
              x,
              y,
              depth,
              diameter )
```

Add pothole of a given size and location.

Parameters:
x (float): x-coordinate of the pothole in global ENU.
y (float): y-coordinate of the pothole in global ENU.
depth (float): Depth of the pothole in meters.
diameter (float): Diameter of the pothole in meters.

6.15.3.2 CreateScene()

```
def mavs_python.mavs_interface.MavsRandomScene.CreateScene (
              self )
```

Generate the scene from the given parameters.

Once all the parameters have been set, call this to generate the scene.
Will perform ecosystem simulation and save files to the output directory.

6.15.4 Member Data Documentation

6.15.4.1 basename

```
mavs_python.mavs_interface.MavsRandomScene.basename
```

basename (string): Naming to use for all the generated output files.

6.15.4.2 eco_file

`mavs_python.mavs_interface.MavsRandomScene.eco_file`

eco_file (string): The ecosystem file to generate vegetation distribution.

Examples in mavs/data/ecosystem_files.

6.15.4.3 hi_mag

`mavs_python.mavs_interface.MavsRandomScene.hi_mag`

hi_mag (float): Magnitude in meters of the high-frequency terrain roughness.

6.15.4.4 lo_mag

`mavs_python.mavs_interface.MavsRandomScene.lo_mag`

lo_mag (float): Magnitude in meters of the low-frequency terrain roughness.

6.15.4.5 mesh_resolution

`mavs_python.mavs_interface.MavsRandomScene.mesh_resolution`

mesh_resolution (float): Resolution of the mesh in meters.

6.15.4.6 num_potholes

`mavs_python.mavs_interface.MavsRandomScene.num_potholes`

num_potholes (int): Total number of potholes in the scene.

6.15.4.7 output_directory

`mavs_python.mavs_interface.MavsRandomScene.output_directory`

output_directory (string): Directory to save the generated scene.

cwd is the default.

6.15.4.8 path_type

mavs_python.mavs_interface.MavsRandomScene.path_type

path_type (string): Type of automatically generated trail.

Options are 'Loop', 'Ridges', or 'Valleys'.

6.15.4.9 pothole_depth

mavs_python.mavs_interface.MavsRandomScene.pothole_depth

pothole_depth (float): Depth of potholes in the trail in meters.

6.15.4.10 pothole_diameter

mavs_python.mavs_interface.MavsRandomScene.pothole_diameter

pothole_diameter (float): Diameter of the potholes in the trail in meters.

6.15.4.11 pothole_locations

mavs_python.mavs_interface.MavsRandomScene.pothole_locations

pothole_locations (list of floats): Nx2 list where N is the number of potholes and x-y position is specified.

6.15.4.12 scene

mavs_python.mavs_interface.MavsRandomScene.scene

scene (void): Pointer to a MAVS scene.

6.15.4.13 terrain_length

mavs_python.mavs_interface.MavsRandomScene.terrain_length

terrain_length (float): Length (y-dimension) of the terrain in meters.

6.15.4.14 terrain_width

`mavs_python.mavs_interface.MavsRandomScene.terrain_width`

terrain_width (float): Width (x-dimension) of the terrain in meters.

6.15.4.15 track_width

`mavs_python.mavs_interface.MavsRandomScene.track_width`

track_width (float): Width of the tire tracks on the trail in meters.

6.15.4.16 trail_width

`mavs_python.mavs_interface.MavsRandomScene.trail_width`

trail_width (float): Width of the automatically generated trail in meters.

6.15.4.17 wheelbase

`mavs_python.mavs_interface.MavsRandomScene.wheelbase`

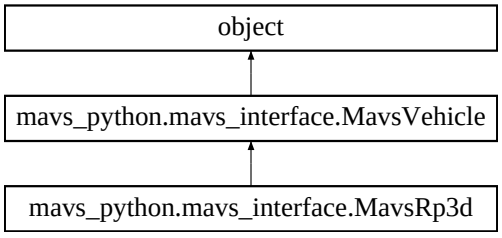
wheelbase (float): Distance between tire tracks on the trail in meters.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.16 mavs_python.mavs_interface.MavsRp3d Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsRp3d:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def Load (self, fname)
- def SetGravity (self, gx, gy, gz)
- def GetTireDeflection (self, i)
- def SetTerrainProperties (self, terrain_type='flat', terrain_param1=0.0, terrain_param2=0.0, soil_type='paved', soil_strength=100.0)

Public Attributes

- vehicle

  vehicle (void): Pointer to a MavsRp3dVehicle.

6.16.1 Detailed Description

```
MavsRp3d class.

Inherits from the MavsVehicle base class.

Attributes:
vehicle (void): Pointer to a MavsRp3dVehicle.
```

6.16.2 Constructor & Destructor Documentation

6.16.2.1 __init__()

```
def mavs_python.mavs_interface.MavsRp3d.__init__ (
              self )
```

```
MavsRp3dVehicle constructor.
```

6.16.2.2 __del__()

```
def mavs_python.mavs_interface.MavsRp3d.__del__ (
              self )
```

```
MavsRp3dVehicle destructor.
```

6.16.3 Member Function Documentation

6.16.3.1 GetTireDeflection()

```
def mavs_python.mavs_interface.MavsRp3d.GetTireDeflection (
              self,
              i )
```

Tire deflection as a fraction of section height.

Returns:
d (float): Tire deflection as a fraction of section height (0-1).

6.16.3.2 Load()

```
def mavs_python.mavs_interface.MavsRp3d.Load (
              self,
              fname )
```

Load an RP3D vehicle file.

Examples are in mavs/data/vehicles/rp3d_vehicles.

Parameters:
fname (string): Full path to the vehicle input file to load.

6.16.3.3 SetGravity()

```
def mavs_python.mavs_interface.MavsRp3d.SetGravity (
              self,
              gx,
              gy,
              gz )
```

Set the gravity constant in m/s^2 in local ENU.

Default is [0.0, 0.0, -9.806].

Parameters:
gx (float): Gravity constant in the x-direction (m/s^2).
gy (float): Gravity constant in the y-direction (m/s^2).
gz (float): Gravity constant in the z-direction (m/s^2).

6.16.3.4 SetTerrainProperties()

```
def mavs_python.mavs_interface.MavsRp3d.SetTerrainProperties (
              self,
              terrain_type = 'flat',
              terrain_param1 = 0.0,
              terrain_param2 = 0.0,
              soil_type = 'paved',
              soil_strength = 100.0 )
```

```
Set the properties of the terrain.

Calling this will dis-associate the loaded mesh file and use an analytical surface instead.
Set the soil type and strength. Available soil types are
'snow', 'ice', 'wet', 'sand', 'clay', 'paved'
The soil strength param is in PSI and is only used when the type is 'clay' or 'sand'
AND
Set the terrain height function. The available terrain types are
'flat', 'slope', 'sine', and 'rough'. The second argument is a list of
parameters for the height model.

flat: terrain_param1 = terrain height, terrain_param2 = not used

sloped: terrain_param1 = fractional slope (1 = 45 degrees), terrain_param2 = not used

sine: terrain_param1 = wavelength in meters, terrain_param2 = magnitude of oscillation

rough: terrain_param1 = wavelength of roughness in meters, terrain_param2 = magnitude of roughness, in meters

Parameters:
terrain_type (string): The type of terrain.
terrain_param1 (float): Parameter 1, see above.
terrain_param2 (float): Parameter 2, see above.
soil_type (string): Soil type, see above.
soil_strngth (float): RCI of soil, see above.
```

6.16.4 Member Data Documentation

6.16.4.1 vehicle

```
mavs_python.mavs_interface.MavsRp3d.vehicle
```

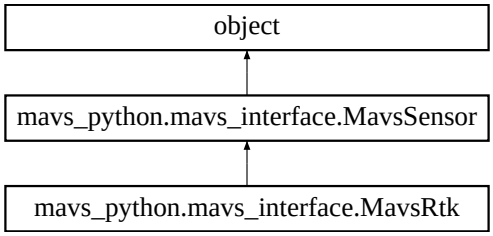
vehicle (void): Pointer to a MavsRp3dVehicle.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.17 mavs_python.mavs_interface.MavsRtk Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsRtk:



Public Member Functions

- def __init__ (self)
- def SetError (self, error)
- def SetDropoutRate (self, dropout_rate)
- def SetWarmupTime (self, warmup_time)
- def GetPosition (self)
- def GetOrientation (self)

Public Attributes

- type

  type (string): Must be 'rtk'.
- sensor

  sensor (void): Pointer to a MAVS sensor.

6.17.1 Detailed Description

```
MavsRtk is an empirical model of a real-time-kinematics positioning sensor.

Inherits from MavsSensor base class.

Attributes:
type (string): Must be 'rtk'.
sensor (void): Pointer to a MAVS sensor.
```

6.17.2 Constructor & Destructor Documentation

6.17.2.1 __init__()

```
def mavs_python.mavs_interface.MavsRtk.__init__ (
              self )

Constructor for a MavsRtk.
```

6.17.3 Member Function Documentation

6.17.3.1 GetOrientation()

```
def mavs_python.mavs_interface.MavsRtk.GetOrientation (
              self )

Get the orientation measured by the RTK sensor.

Returns:
orientation ([float, float, float, float]): The measured w-x-y-z orientation in local ENU.
```

6.17.3.2 GetPosition()

```
def mavs_python.mavs_interface.MavsRtk.GetPosition (
              self )
```

Get the position measured by the RTK sensor.

Returns:
position ([float, float, float]): The measured x-y-z position in local ENU.

6.17.3.3 SetDropoutRate()

```
def mavs_python.mavs_interface.MavsRtk.SetDropoutRate (
              self,
              dropout_rate )
```

Set the dropout rate in GPS dropouts/hour.

Parameters:
dropout_rate (float): The number of GPS dropouts/hour.

6.17.3.4 SetError()

```
def mavs_python.mavs_interface.MavsRtk.SetError (
              self,
              error )
```

Set the error in the MAVS Rtk sensor.

Parmeters:
error (float): The error factor in meters.

6.17.3.5 SetWarmupTime()

```
def mavs_python.mavs_interface.MavsRtk.SetWarmupTime (
              self,
              warmup_time )
```

Set the warmup time of the sensor in seconds.

The sensor error exponentially decreases in the minimum error value.
This parameter controls the rate of that decrease.

Parameters:
warmup_time (float): The sensor warmup time in seconds.

6.17.4 Member Data Documentation

6.17.4.1 sensor

`mavs_python.mavs_interface.MavsRtk.sensor`

sensor (void): Pointer to a MAVS sensor.

6.17.4.2 type

`mavs_python.mavs_interface.MavsRtk.type`

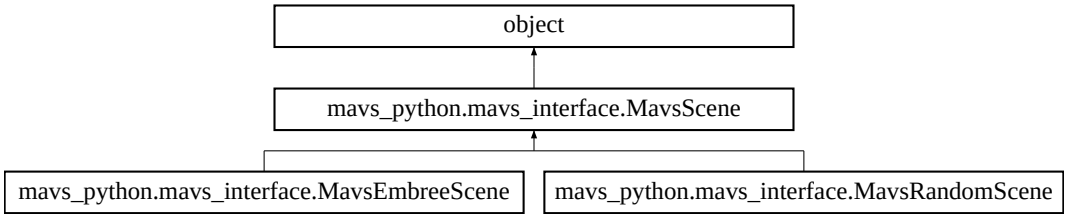
type (string): Must be 'rtk'.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.18 mavs_python.mavs_interface.MavsScene Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsScene:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def DeleteCurrentScene (self)
- def WriteStats (self)
- def DeleteScene (self)
- def TurnOnLabeling (self)
- def TurnOffLabeling (self)
- def AddAnimation (self, animation)
- def GetSurfaceHeight (self, x, y)

Public Attributes

- scene

  scene (void): Pointer to an Embree Raytracer scene.

6.18.1 Detailed Description

```
MavsScene class.

A Mavs Scene is a geometrical description of the environment and the associated raytracer.
While the C++ version of the API can support any type of raytracer, the Python version only
supports the Embree ray-tracer that is the default raytracing kernel in MAVS.

Attributes:
scene (void): Pointer to an Embree Raytracer scene.
```

6.18.2 Constructor & Destructor Documentation

6.18.2.1 __init__()

```
def mavs_python.mavs_interface.MavsScene.__init__ (
              self )
```

```
Constructor for a MavsScene.
```

6.18.2.2 __del__()

```
def mavs_python.mavs_interface.MavsScene.__del__ (
              self )
```

```
Destructor for a MavsScene.
```

6.18.3 Member Function Documentation

6.18.3.1 AddAnimation()

```
def mavs_python.mavs_interface.MavsScene.AddAnimation (
              self,
              animation )
```

Add an animation to a scene.

Parameters:
animation (MavsAnimation): An animation object. It should already have been loaded.

Returns:
anim_num (int): A unique ID for the animation.

6.18.3.2 DeleteCurrentScene()

```
def mavs_python.mavs_interface.MavsScene.DeleteCurrentScene (
              self )
```

Free the pointer to the current scene.

6.18.3.3 DeleteScene()

```
def mavs_python.mavs_interface.MavsScene.DeleteScene (
              self )
```

Free the pointer to the current scene.

Duplicate of 'DeleteCurrentScene'.

6.18.3.4 GetSurfaceHeight()

```
def mavs_python.mavs_interface.MavsScene.GetSurfaceHeight (
              self,
              x,
              y )
```

Get the height of the surface at a given lateral position.

Parameters:
x (float): The x-coordinate in global ENU to get the height.
y (float): The y-coordinate in global ENU to get the height.

Returns:
h (float): The height at (x,y) in global ENU meters.

6.18.3.5 TurnOffLabeling()

```
def mavs_python.mavs_interface.MavsScene.TurnOffLabeling (
              self )
```

Turn off scene labeling.

Call this if you don't plan to use labeled data.
It will speed the simulation upp slightly.

6.18.3.6 TurnOnLabeling()

```
def mavs_python.mavs_interface.MavsScene.TurnOnLabeling (
              self )
```

Turn on scene labeling.

This must be called in order to use a scene to generate annotated data.
Labeling should only be turned on if you plan to use the labeled data
because it slows the simulation down.

6.18.3.7 WriteStats()

```
def mavs_python.mavs_interface.MavsScene.WriteStats (
              self )
```

Write the stats of the scene to scene_stats.txt.

6.18.4 Member Data Documentation

6.18.4.1 scene

```
mavs_python.mavs_interface.MavsScene.scene
```

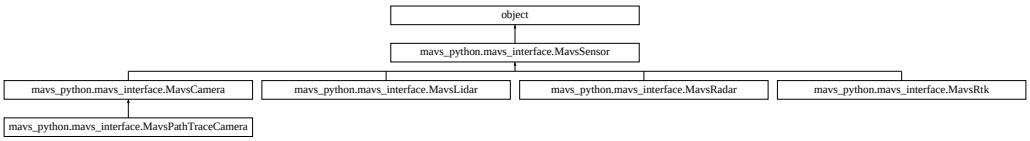
scene (void): Pointer to an Embree Raytracer scene.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.19 mavs_python.mavs_interface.MavsSensor Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsSensor:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def GetPose (self)
- def GetDict (self)
- def Update (self, env, dt)
- def SaveRaw (self)
- def AnnotateFrame (self, env)
- def SetOffset (self, pos, quat)
- def SetPose (self, pos, quat)
- def Display (self)
- def SaveAnnotation (self, env, fname)
- def load_block (self, data)

Public Attributes

- sensor

  sensor (void): Pointer to a MAVS sensor.
- name

  name (string): Name of the sensor.
- model

  model (string): Name of the sensor model (ie 'HDL-64E')
- type

  type (string): Type of the sensor.
- offset

  offset ([float, float, float]): x-y-z offset of the sensor from the vehicle CG.
- rel_or

  rel_or ([float, float, float, float]): Quaternion in q-x-y-z format specifying the relative orientation of the sensor w.r.t the vehicle.
- position

  position ([float, float, float]): Current position of the vehicle the sensor is mounted to in global ENU.
- orientation

  orientation ([float, float, float, float]): Current orientation of the vehicle the sensor is mounted to in global ENU.
- update_rate

  update_rate (float): The update rate of the sensor in Hz.
- elapsed_since_last

  elapsed_since_last (float): The amount of elapsed time since the last sensor update.
- is_active

  is_active (bool): Set to True if the sensor is turned on, False if it is not.

- save_labeled

  save_labeled (bool): Set to True if user wants to save labeled sensor data.
- display

  display (bool): Set to True to display the output of the sensor to the screen during simulation.
- save_raw

  save_raw (bool): Set to True if the user wants to save raw sensor data.

6.19.1 Detailed Description

```
Base class for all types of MAVS sensors.

Attributes:
sensor (void): Pointer to a MAVS sensor.
name (string): Name of the sensor.
type (string): Type of the sensor. Can be 'lidar', 'camera', 'gps', 'compass', 'fisheye', 'radar',  or 'imu'.
offset ([float, float, float]): x-y-z offset of the sensor from the vehicle CG.
rel_or ([float, float, float, float]): Quaternion in q-x-y-z format specifying the relative orientation of the
position ([float, float, float]): Current position of the vehicle the sensor is mounted to in global ENU.
orientation ([float, float, float, float]): Current orientation of the vehicle the sensor is mounted to in glo
update_rate (float): The update rate of the sensor in Hz.
elapsed_since_last (float): The amount of elapsed time since the last sensor update.
is_active (bool): Set to True if the sensor is turned on, False if it is not.
save_labeled (bool): Set to True if user wants to save labeled sensor data.
save_raw (bool): Set to True if the user wants to save raw sensor data.
display (bool): Set to True to display the output of the sensor to the screen during simulation.
```

6.19.2 Constructor & Destructor Documentation

6.19.2.1 __init__()

```
def mavs_python.mavs_interface.MavsSensor.__init__ (
              self )
```

```
MavsSensor constructor.
```

6.19.2.2 __del__()

```
def mavs_python.mavs_interface.MavsSensor.__del__ (
              self )
```

```
MavsSensor destructor.
```

6.19.3 Member Function Documentation

6.19.3.1 AnnotateFrame()

```
def mavs_python.mavs_interface.MavsSensor.AnnotateFrame (
              self,
              env )
```

Calculate semantic labeled data for the sensor.

Use MAVS automated labeling to generate labeled sensor data.

Parameters:
env (MavsEnvironment): The environment object.

6.19.3.2 Display()

```
def mavs_python.mavs_interface.MavsSensor.Display (
              self )
```

Display the output of the sensor in an X-window.

6.19.3.3 GetDict()

```
def mavs_python.mavs_interface.MavsSensor.GetDict (
              self )
```

Return a dictionary with sensor properties.

Used for writing the sensor properties to a json file.

6.19.3.4 GetPose()

```
def mavs_python.mavs_interface.MavsSensor.GetPose (
              self )
```

Get the current sensor pose.

Return the current position and orientation.

Returns:
p ([float, float, float]): The current sensor position, with the offset included.
q ([float, float, float, float): The current sensor orientation, with the offset included.

6.19.3.5 load_block()

```
def mavs_python.mavs_interface.MavsSensor.load_block (
              self,
              data )
```

Load a sensor block from a json file.

Parameters:
data (json): A json dictionary block.

6.19.3.6 SaveAnnotation()

```
def mavs_python.mavs_interface.MavsSensor.SaveAnnotation (
              self,
              env,
              fname )
```

Save an annotated frame.

The 'AnnotateFrame' method should be called first.

Parameters:
env (MavsEnvironment): MavsEnvironment object.
fname (string): The file name of the annotation output.

6.19.3.7 SaveRaw()

```
def mavs_python.mavs_interface.MavsSensor.SaveRaw (
              self )
```

Save raw sensor data (point cloud, image, etc.

Output will be given a generic file name.
To save output with specific file names,
use the save functions that are unique to each sensor.

6.19.3.8 SetOffset()

```
def mavs_python.mavs_interface.MavsSensor.SetOffset (
              self,
              pos,
              quat )
```

Set the offset of the sensor relative to the vehicle CG.

Parameters:
pos ([float,float,float]): x-y-z offset relative to the vehicle CG.
quat ([float,float,float,float]): w-x-y-z quaternion orientation relative to the vehicle.

6.19.3.9 SetPose()

```
def mavs_python.mavs_interface.MavsSensor.SetPose (
              self,
              pos,
              quat )
```

```
Set the position of the vehicle carrying the sensor.

This does not include the offsets,
it is the position of the vehicle that the sensor is attached to.
Offsets will be included automatically if they have been set using the
'SetOffset' method.

Parameters:
pos ([float,float,float]): x-y-z position of the sensor.
quat ([float,float,float,float]): w-x-y-z quaternion orientation of the sensor.
```

6.19.3.10 Update()

```
def mavs_python.mavs_interface.MavsSensor.Update (
              self,
              env,
              dt )
```

```
Update the sensor.

This method calls the internal update function of each sensor.
SetPose should be called beforehand to move the sensor to the proper spot.

Parameters:
env (MavsEnvironment): The environment object.
dt (float): The time step in seconds.
```

6.19.4 Member Data Documentation

6.19.4.1 display

```
mavs_python.mavs_interface.MavsSensor.display
```

display (bool): Set to True to display the output of the sensor to the screen during simulation.

6.19.4.2 elapsed_since_last

```
mavs_python.mavs_interface.MavsSensor.elapsed_since_last
```

elapsed_since_last (float): The amount of elapsed time since the last sensor update.

6.19.4.3 is_active

mavs_python.mavs_interface.MavsSensor.is_active

is_active (bool): Set to True if the sensor is turned on, False if it is not.

6.19.4.4 name

mavs_python.mavs_interface.MavsSensor.name

name (string): Name of the sensor.

6.19.4.5 offset

mavs_python.mavs_interface.MavsSensor.offset

offset ([float, float, float]): x-y-z offset of the sensor from the vehicle CG.

6.19.4.6 orientation

mavs_python.mavs_interface.MavsSensor.orientation

orientation ([float, float, float, float]): Current orientation of the vehicle the sensor is mounted to in global ENU.

6.19.4.7 position

mavs_python.mavs_interface.MavsSensor.position

position ([float, float, float]): Current position of the vehicle the sensor is mounted to in global ENU.

6.19.4.8 rel_or

mavs_python.mavs_interface.MavsSensor.rel_or

rel_or ([float, float, float, float]): Quaternion in q-x-y-z format specifying the relative orientation of the sensor w.r.t the vehicle.

6.19.4.9 save_labeled

mavs_python.mavs_interface.MavsSensor.save_labeled

save_labeled (bool): Set to True if user wants to save labeled sensor data.

6.19.4.10 save_raw

mavs_python.mavs_interface.MavsSensor.save_raw

save_raw (bool): Set to True if the user wants to save raw sensor data.

6.19.4.11 sensor

mavs_python.mavs_interface.MavsSensor.sensor

sensor (void): Pointer to a MAVS sensor.

6.19.4.12 type

mavs_python.mavs_interface.MavsSensor.type

type (string): Type of the sensor.

Can be 'lidar', 'camera', 'gps', 'compass', 'fisheye', 'radar', or 'imu'.

6.19.4.13 update_rate

mavs_python.mavs_interface.MavsSensor.update_rate

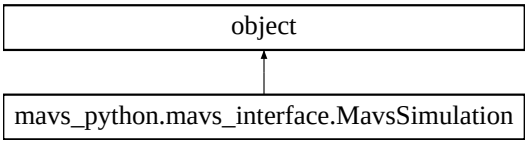
update_rate (float): The update rate of the sensor in Hz.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.20 mavs_python.mavs_interface.MavsSimulation Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsSimulation:



Public Member Functions

- def __init__ (self)
- def LoadScene (self)
- def UnloadScene (self)
- def LoadNewScene (self, scenefile)
- def LoadNewVehicle (self, veh_dyn_file)
- def LoadNewWaypoints (self, wp_file)
- def Load (self, fname)
- def TurnOnSensor (self, sensor_id, display=False, save_raw=False, labeling=False)
- def TurnOffSensor (self, sensor_id)
- def TurnOnSensorLabeling (self, sensor_id)
- def TurnOffSensorLabeling (self, sensor_id)
- def TurnOnSensorDisplay (self, sensor_id)
- def TurnOffSensorDisplay (self, sensor_id)
- def Update (self, dt, throttle=0.0, steering=0.0, braking=1.0, update_actor=False)
- def GetSensorDict (self)
- def WriteToJson (self, fname)

Public Attributes

- env

  env (MavsEnvironment): Environment object.
- scene

  scene (MavsEmbreeScene): Scene object.
- vehicle

  vehicle (MavsRp3d()): Vehicle object.
- waypoints

  waypoints (MavsWaypoints): Waypoints object.
- controller

  controller (MavsVehicleController): Controller object.
- elapsed_time

  elapsed_time (float): Elapsed simulation time in seconds.
- wait_time

  wait_time (float): Bootup time of the simulation in seconds.
- veh_actor_num

  veh_actor_num (int): ID for the vehicle actor.
- origin

  origin ([float, float, float]): Scene origin in local ENU.
- time_zone

  time_zone (int): Time zone offset, CST=6.
- sensors

  sensors ([MavsSensor]): List of sensors on the vehicle.
- scenefile

  scenefile (string): Name of the json scene file.
- vehicle_file

  vehicle_file (string): Name of the json vehicle file.
- posefile

  posefile (string): Name of the waypoints pose file.
- posetype

  posetype (string): Type of pose file, can be 'anvel' or 'json'.

- dusty

  dusty (bool): Is the environment dusty?
- env_block

  env_block (dict): JSON dictionary describing environment.
- free_driving

  free_driving (bool): If True, user drives vehicle with keyboard.
- start_heading

  start_heading (float): Initial vehicle heading in radians.
- start_heading_loaded

  start_heading_loaded (bool): Initial heading supplied in input file?
- env_time

  env_time (float): Wall time taken to simulate environment (seconds).
- veh_time

  veh_time (float): Wall time taken to simulate the vehicle (seconds).
- sensor_times

  sensor_times (list of floats): Wall time taken to simulate each sensor (seconds).
- save_location

  save_location (string): Full path to the default save location for sensor data.
- time_to_update_actor

  time_to_update_actor (bool): Actor is only updated when this is true.
- start_pos

  start_pos ([float, float, float]): Initial position of the vehicle in global ENU.

6.20.1 Detailed Description

```
MavsSimulation class.

Combines vehicle, scene, environment and sensors
into a simulation object that can be automatically
updated.

Attributes:
env (MavsEnvironment): Environment object.
scene (MavsEmbreeScene): Scene object.
vehicle (MavsRp3d()): Vehicle object.
waypoints (MavsWaypoints): Waypoints object.
controller (MavsVehicleController): Controller object.
elapsed_time (float): Elapsed simulation time in seconds.
wait_time (float): Bootup time of the simulation in seconds.
veh_actor_num (int): ID for the vehicle actor.
origin ([float, float, float]): Scene origin in local ENU.
time_zone (int): Time zone offset, CST=6.
sensors ([MavsSensor]): List of sensors on the vehicle.
scenefile (string): Name of the json scene file.
vehicle_file (string): Name of the json vehicle file.
posefile (string): Name of the waypoints pose file.
posetype (string): Type of pose file, can be 'anvel' or 'json'.
dusty (bool): Is the environment dusty?
env_block (dict): JSON dictionary describing environment.
free_driving (bool): If True, user drives vehicle with keyboard.
start_heading (float): Initial vehicle heading in radians.
start_heading_loaded (bool): Initial heading supplied in input file?
env_time (float): Wall time taken to simulate environment (seconds).
veh_time (float): Wall time taken to simulate the vehicle (seconds).
sensor_times (list of floats): Wall time taken to simulate each sensor (seconds).
save_location (string): Full path to the default save location for sensor data.
time_to_update_actor (bool): Actor is only updated when this is true.
start_pos ([float, float, float]): Initial position of the vehicle in global ENU.
```

6.20.2 Constructor & Destructor Documentation

6.20.2.1 __init__()

```
def mavs_python.mavs_interface.MavsSimulation.__init__ (
              self )
```

Construct a simulation.

6.20.3 Member Function Documentation

6.20.3.1 GetSensorDict()

```
def mavs_python.mavs_interface.MavsSimulation.GetSensorDict (
              self )
```

Return a dictionary entry for the sensor block.

This is for saving the current simulation config to a file.

Returns:
outlist (dictionary): Dictionary with all sensor config information.

6.20.3.2 Load()

```
def mavs_python.mavs_interface.MavsSimulation.Load (
              self,
              fname )
```

Load a simulation input file.

Example input files are in mavs/data/sims/sensor_sims

Parameters:
fname (string): Full path the MAVS json input file.

6.20.3.3 LoadNewScene()

```
def mavs_python.mavs_interface.MavsSimulation.LoadNewScene (
              self,
              scenefile )
```

Load a new scene, specified by scenefile.

This will unload the existing scene and load a new scene.

Parameters:
scenefile (string): Full path to the scene file to load.

6.20.3.4 LoadNewVehicle()

```
def mavs_python.mavs_interface.MavsSimulation.LoadNewVehicle (
              self,
              veh_dyn_file )
```

Load a new vehicle, specified by veh_dyn_file.

This will unload the existing vehicle and load a new vehicle.

Parameters:
veh_dyn_file (string): Full path to the rp3d vehicle file to load.

6.20.3.5 LoadNewWaypoints()

```
def mavs_python.mavs_interface.MavsSimulation.LoadNewWaypoints (
              self,
              wp_file )
```

Load a new waypoint file, specified by wp_file.

This will unload the existing waypoints and load a new set of waypoints.

Parameters:
wp_file (string): Full path to the waypoint file to load.

6.20.3.6 LoadScene()

```
def mavs_python.mavs_interface.MavsSimulation.LoadScene (
              self )
```

Load the scene specified by scenefile.

scenefile should already be specified.

6.20.3.7 TurnOffSensor()

```
def mavs_python.mavs_interface.MavsSimulation.TurnOffSensor (
              self,
              sensor_id )
```

Turn off a sensor.

Parameters:
sensor_id (int): The ID number of the sensor to turn off.

6.20.3.8 TurnOffSensorDisplay()

```
def mavs_python.mavs_interface.MavsSimulation.TurnOffSensorDisplay (
              self,
              sensor_id )
```

Turn off display for a sensor.

Parameters:
sensor_id (int): The ID number of the sensor to turn off display.

6.20.3.9 TurnOffSensorLabeling()

```
def mavs_python.mavs_interface.MavsSimulation.TurnOffSensorLabeling (
              self,
              sensor_id )
```

Turn off labeling for a sensor.

Parameters:
sensor_id (int): The ID number of the sensor to turn off labeling.

6.20.3.10 TurnOnSensor()

```
def mavs_python.mavs_interface.MavsSimulation.TurnOnSensor (
              self,
              sensor_id,
              display = False,
              save_raw = False,
              labeling = False )
```

Turn on a particular sensor.

Parameters:
sensor_id (int): The ID number of the sensor to turn on.
display (bool): True to display the sensor output to screen.
save_raw (bool): True to save raw sensor data to disk.
labeling (bool): True to save labeled data to disk.

6.20.3.11 TurnOnSensorDisplay()

```
def mavs_python.mavs_interface.MavsSimulation.TurnOnSensorDisplay (
              self,
              sensor_id )
```

Turn on display for a sensor.

Parameters:
sensor_id (int): The ID number of the sensor to turn on display.

6.20.3.12 TurnOnSensorLabeling()

```
def mavs_python.mavs_interface.MavsSimulation.TurnOnSensorLabeling (
              self,
              sensor_id )
```

Turn on labeling for a sensor.

Parameters:
sensor_id (int): The ID number of the sensor to turn on labeling.

6.20.3.13 UnloadScene()

```
def mavs_python.mavs_interface.MavsSimulation.UnloadScene (
              self )
```

Free the pointer to the Embree scene and clear all memory.

6.20.3.14 Update()

```
def mavs_python.mavs_interface.MavsSimulation.Update (
              self,
              dt,
              throttle = 0.0,
              steering = 0.0,
              braking = 1.0,
              update_actor = False )
```

Update the simulation.

Parameters:
dt (float): The time step in seconds.
throttle (float): Throttle from 0.0-1.0
steering (float): Steering from -1.0 to 1.0.
braking (float): Braking from 0.0 to 1.0
update_actor (bool): If True, update the actor position.

6.20.3.15 WriteToJson()

```
def mavs_python.mavs_interface.MavsSimulation.WriteToJson (
              self,
              fname )
```

```
Write the current simulation configuration to a json file.

Parameters:
fname (string): The output JSON file name.
```

6.20.4 Member Data Documentation

6.20.4.1 controller

```
mavs_python.mavs_interface.MavsSimulation.controller
```

controller (MavsVehicleController): Controller object.

6.20.4.2 elapsed_time

```
mavs_python.mavs_interface.MavsSimulation.elapsed_time
```

elapsed_time (float): Elapsed simulation time in seconds.

6.20.4.3 env

```
mavs_python.mavs_interface.MavsSimulation.env
```

env (MavsEnvironment): Environment object.

6.20.4.4 env_block

```
mavs_python.mavs_interface.MavsSimulation.env_block
```

env_block (dict): JSON dictionary describing environment.

6.20.4.5 env_time

`mavs_python.mavs_interface.MavsSimulation.env_time`

env_time (float): Wall time taken to simulate environment (seconds).

6.20.4.6 free_driving

`mavs_python.mavs_interface.MavsSimulation.free_driving`

free_driving (bool): If True, user drives vehicle with keyboard.

6.20.4.7 origin

`mavs_python.mavs_interface.MavsSimulation.origin`

origin ([float, float, float]): Scene origin in local ENU.

6.20.4.8 posefile

`mavs_python.mavs_interface.MavsSimulation.posefile`

posefile (string): Name of the waypoints pose file.

6.20.4.9 posetype

`mavs_python.mavs_interface.MavsSimulation.posetype`

posetype (string): Type of pose file, can be 'anvel' or 'json'.

6.20.4.10 save_location

`mavs_python.mavs_interface.MavsSimulation.save_location`

save_location (string): Full path to the default save location for sensor data.

6.20.4.11 scene

`mavs_python.mavs_interface.MavsSimulation.scene`

scene (MavsEmbreeScene): Scene object.

6.20.4.12 scenefile

`mavs_python.mavs_interface.MavsSimulation.scenefile`

scenefile (string): Name of the json scene file.

6.20.4.13 sensor_times

`mavs_python.mavs_interface.MavsSimulation.sensor_times`

sensor_times (list of floats): Wall time taken to simulate each sensor (seconds).

6.20.4.14 sensors

`mavs_python.mavs_interface.MavsSimulation.sensors`

sensors ([MavsSensor]): List of sensors on the vehicle.

6.20.4.15 start_heading

`mavs_python.mavs_interface.MavsSimulation.start_heading`

start_heading (float): Initial vehicle heading in radians.

6.20.4.16 start_pos

`mavs_python.mavs_interface.MavsSimulation.start_pos`

start_pos ([float, float, float]): Initial position of the vehicle in global ENU.

6.20.4.17 time_to_update_actor

mavs_python.mavs_interface.MavsSimulation.time_to_update_actor

time_to_update_actor (bool): Actor is only updated when this is true.

6.20.4.18 time_zone

mavs_python.mavs_interface.MavsSimulation.time_zone

time_zone (int): Time zone offset, CST=6.

6.20.4.19 veh_actor_num

mavs_python.mavs_interface.MavsSimulation.veh_actor_num

veh_actor_num (int): ID for the vehicle actor.

6.20.4.20 veh_time

mavs_python.mavs_interface.MavsSimulation.veh_time

veh_time (float): Wall time taken to simulate the vehicle (seconds).

6.20.4.21 vehicle

mavs_python.mavs_interface.MavsSimulation.vehicle

vehicle (MavsRp3d()): Vehicle object.

6.20.4.22 vehicle_file

mavs_python.mavs_interface.MavsSimulation.vehicle_file

vehicle_file (string): Name of the json vehicle file.

6.20.4.23 wait_time

`mavs_python.mavs_interface.MavsSimulation.wait_time`

wait_time (float): Bootup time of the simulation in seconds.

6.20.4.24 waypoints

`mavs_python.mavs_interface.MavsSimulation.waypoints`

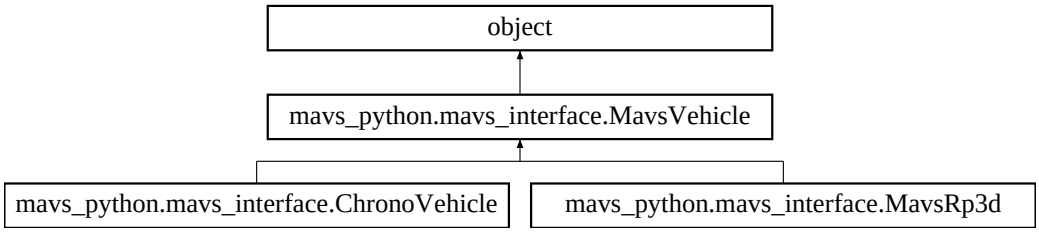
waypoints (MavsWaypoints): Waypoints object.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.21 mavs_python.mavs_interface.MavsVehicle Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsVehicle:



Public Member Functions

- def __init__ (self)
- def Update (self, env, throttle, steering, brake, dt)
- def AddHeadlights (self, env)
- def SetInitialPosition (self, x, y, z)
- def SetInitialHeading (self, theta)
- def GetPosition (self)
- def GetVelocity (self)
- def GetOrientation (self)
- def GetSpeed (self)
- def GetHeading (self)
- def UnloadVehicle (self)

Public Attributes

- vehicle

  vehicle (void): Pointer to a MAVS vehicle.
- position

  position ([float, float, float]): The position of the vehicle in global ENU.
- orientation

  orientation ([float, float, float, float]) The w-x-y-z orientation of the vehicle in global ENU.
- headlight_offset

  headlight_offset (float): How far forward the headlights are from the CG, in meters.
- headlight_width

  headlight_width (float): How far apart he headlights are, in meters.
- headlight_ids

  headlight_ids ([int, int]): ID numbers for the headlights.

6.21.1 Detailed Description

```
MavsVehicle class.

Base class for Mavs Vehicles.
There are two different inherited classes - Rp3d and Chrono.

Attributes:
vehicle (void): Pointer to a MAVS vehicle.
position ([float, float, float]): The position of the vehicle in global ENU.
orientation ([float, float, float, float]) The w-x-y-z orientation of the vehicle in global ENU.
headlight_offset (float): How far forward the headlights are from the CG, in meters.
headlight_width (float): How far apart he headlights are, in meters.
headlight_ids ([int, int]): ID numbers for the headlights.
```

6.21.2 Constructor & Destructor Documentation

6.21.2.1 __init__()

```
def mavs_python.mavs_interface.MavsVehicle.__init__ (
              self )
```

```
Constructor for a MavsVehicle.
```

6.21.3 Member Function Documentation

6.21.3.1 AddHeadlights()

```
def mavs_python.mavs_interface.MavsVehicle.AddHeadlights (
              self,
              env )
```

```
Add headlights to the vehicle.

The various headlight parameters should be set before calling this.

Parameters:
env (MavsEnvironment): The mavs environment containing the vehicle.
```

6.21.3.2 GetHeading()

```
def mavs_python.mavs_interface.MavsVehicle.GetHeading (
              self )
```

```
Get the current heading of the vehicle.

Relative to the East/X direction.

returns:
heading (float): Current heading in radians.
```

6.21.3.3 GetOrientation()

```
def mavs_python.mavs_interface.MavsVehicle.GetOrientation (
              self )
```

```
Get the current orientation of the vehicle in global ENU.

Returns:
orientation ([float, float, float, float]): w-x-y-z orientation of the vehicle.
```

6.21.3.4 GetPosition()

```
def mavs_python.mavs_interface.MavsVehicle.GetPosition (
              self )
```

```
Get the current position of the vehicle in global ENU.

Returns:
position ([float, float, float]): x-y-z position of the vehicle.
```

6.21.3.5 GetSpeed()

```
def mavs_python.mavs_interface.MavsVehicle.GetSpeed (
              self )
```

Get the current speed of the vehicle in m/s.

Returns:
speed (float): The speed of the vehicle in m/s.

6.21.3.6 GetVelocity()

```
def mavs_python.mavs_interface.MavsVehicle.GetVelocity (
              self )
```

Get the current velocity of the vehicle in global ENU.

Returns:
velocity ([float, float, float]): x-y-z velocity of the vehicle in m/s.

6.21.3.7 SetInitialHeading()

```
def mavs_python.mavs_interface.MavsVehicle.SetInitialHeading (
              self,
              theta )
```

Set the initial heading of the vehicle.

Relative to the East/X direction.

Parameters:
theta (float): Initial heading in radians.

6.21.3.8 SetInitialPosition()

```
def mavs_python.mavs_interface.MavsVehicle.SetInitialPosition (
              self,
              x,
              y,
              z )
```

Set the initial position of the vehicle in global ENU.

Parameters:
x (float): Initial x-coordinate in global ENU.
y (float): Initial y-coordinate in global ENU.
z (float): Initial z-coordinate in global ENU.

6.21.3.9 UnloadVehicle()

```
def mavs_python.mavs_interface.MavsVehicle.UnloadVehicle (
              self )
```

Free the pointer associated with the vehicle.

6.21.3.10 Update()

```
def mavs_python.mavs_interface.MavsVehicle.Update (
              self,
              env,
              throttle,
              steering,
              brake,
              dt )
```

Update the vehicle model.

Apply throttle and steering and move the vehicle.

```
Parameters:
env (MavsEnvironment): The MAVS environment.
throttle (float): Throttle from 0 to 1.
steering (float): Steering from -1 to 1.
dt (float): The time step in seconds.
```

6.21.4 Member Data Documentation

6.21.4.1 headlight_ids

```
mavs_python.mavs_interface.MavsVehicle.headlight_ids
```

headlight_ids ([int, int]): ID numbers for the headlights.

6.21.4.2 headlight_offset

```
mavs_python.mavs_interface.MavsVehicle.headlight_offset
```

headlight_offset (float): How far forward the headlights are from the CG, in meters.

6.21.4.3 headlight_width

mavs_python.mavs_interface.MavsVehicle.headlight_width

headlight_width (float): How far apart he headlights are, in meters.

6.21.4.4 orientation

mavs_python.mavs_interface.MavsVehicle.orientation

orientation ([float, float, float, float]) The w-x-y-z orientation of the vehicle in global ENU.

6.21.4.5 position

mavs_python.mavs_interface.MavsVehicle.position

position ([float, float, float]): The position of the vehicle in global ENU.

6.21.4.6 vehicle

mavs_python.mavs_interface.MavsVehicle.vehicle

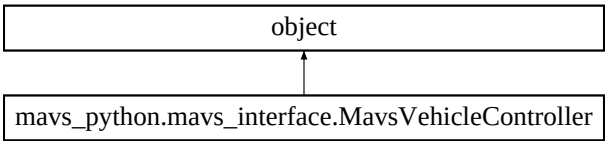
vehicle (void): Pointer to a MAVS vehicle.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.22 mavs_python.mavs_interface.MavsVehicleController Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsVehicleController:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def TurnOnLooping (self)
- def SetDesiredSpeed (self, speed)
- def SetWheelbase (self, wb)
- def SetMaxSteerAngle (self, max_sa)
- def SetMinLookAhead (self, min_la)
- def SetMaxLookAhead (self, max_la)
- def SetSteeringScale (self, steering_k)
- def SetCurrentState (self, px, py, speed, heading)
- def GetDrivingCommand (self, dt)
- def SetDesiredPath (self, path)

Public Attributes

- object

  object (void): Pointer to the MAVS vehicle controller.
- steering_coeff

  steering_coeff (float): The gain in the steering PID controller.
- wheelbase

  wheelbase (float): Wheelbase of the vehicle in meters.
- max_steering_angle

  max_steering_angle (float): Max steering angle of the vehicle in radians.
- desired_speed

  desired_speed (float): Target speed for the throttle controller in m/s.

6.22.1 Detailed Description

```
Class for MavsVehicleController.

The vehicle controller will automatically create driving commands
based on the vehicles current state and a set of waypoints.

Attributes:
object (void): Pointer to the MAVS vehicle controller.
steering_coeff (float): The gain in the steering PID controller.
wheelbase (float): Wheelbase of the vehicle in meters.
max_steering_angle (float): Max steering angle of the vehicle in radians.
desired_speed (float): Target speed for the throttle controller in m/s.
```

6.22.2 Constructor & Destructor Documentation

6.22.2.1 __init__()

```
def mavs_python.mavs_interface.MavsVehicleController.__init__ (
            self )

Constructor for the MavsVehicleController.
```

6.22.2.2 __del__()

```
def mavs_python.mavs_interface.MavsVehicleController.__del__ (
              self )
```

Destructor for the MavsVehiclecontroller.

6.22.3 Member Function Documentation

6.22.3.1 GetDrivingCommand()

```
def mavs_python.mavs_interface.MavsVehicleController.GetDrivingCommand (
              self,
              dt )
```

Get a driving command based on the current vehicle state and waypoints.

This will calculate an updated driving command.
You need to have set the desired path and the current state
for the driving command to refresh.

Parameters:
dt (float): The time step for the update, in seconds.

6.22.3.2 SetCurrentState()

```
def mavs_python.mavs_interface.MavsVehicleController.SetCurrentState (
              self,
              px,
              py,
              speed,
              heading )
```

Set the current vehicle state.

The steering controller uses the current vehicle position,
orientation, and speed to update the driving commands.

Parameters:
px (float): The current vehicle position in the x-coordinate, global ENU.
py (float): The current vehicle position in the y-coordinate, global ENU.
speed (float): The current vehicle speed in m/s.
heading (float): The current heading in radians relative to East / X.

6.22.3.3 SetDesiredPath()

```
def mavs_python.mavs_interface.MavsVehicleController.SetDesiredPath (
              self,
              path )
```

Set the path for the vehicle to follow.

The path is in global ENU coordinates.
It should be a sequence of at least three waypoints.
Paths work better if the waypoints are not spaced out too far.
Optimal spacing is about 1 meter.

Parameters:
path (float): An nx2 list of x-y points.

6.22.3.4 SetDesiredSpeed()

```
def mavs_python.mavs_interface.MavsVehicleController.SetDesiredSpeed (
              self,
              speed )
```

Set the target speed for the vehicle in m/s.

Parameters:
speed (float): The desired speed in m/s.

6.22.3.5 SetMaxLookAhead()

```
def mavs_python.mavs_interface.MavsVehicleController.SetMaxLookAhead (
              self,
              max_la )
```

Set the maximum look-ahead distance in meters.

The look-ahead distance is how far ahead the vehicle looks to plan the path.

Parameters:
max_la (float): The maximum look-ahead distance in meters.

6.22.3.6 SetMaxSteerAngle()

```
def mavs_python.mavs_interface.MavsVehicleController.SetMaxSteerAngle (
              self,
              max_sa )
```

Set the max steering angle for the vehicle in radians.

Parameters:
max_sa (float): The max steering angle in radians.

6.22.3.7 SetMinLookAhead()

```
def mavs_python.mavs_interface.MavsVehicleController.SetMinLookAhead (
              self,
              min_la )
```

Set the minimum look-ahead distance in meters.

The look-ahead distance is how far ahead the vehicle looks to plan the path.

Parameters:
min_la (float): The minimum look-ahead distance in meters.

6.22.3.8 SetSteeringScale()

```
def mavs_python.mavs_interface.MavsVehicleController.SetSteeringScale (
              self,
              steering_k )
```

Set steering scale value.

The controller uses the pure-pursuit algorithm.
This parameters essentially functions as the coefficient
of the 'proportional' term in a steerig PID controller.

Parameters:
steering_k (float): The steering scale factor.

6.22.3.9 SetWheelbase()

```
def mavs_python.mavs_interface.MavsVehicleController.SetWheelbase (
              self,
              wb )
```

Set the wheelbase of the vehicle in meters.

Parameters:
wb (float): The vehicle wheelbase in meters.

6.22.3.10 TurnOnLooping()

```
def mavs_python.mavs_interface.MavsVehicleController.TurnOnLooping (
              self )
```

Set looping to true.

Calling this will make the vehicle loop through the waypoints indefinitely,
automatically returning to the first waypoint when it reaches the last.

6.22.4 Member Data Documentation

6.22.4.1 desired_speed

`mavs_python.mavs_interface.MavsVehicleController.desired_speed`

desired_speed (float): Target speed for the throttle controller in m/s.

6.22.4.2 max_steering_angle

`mavs_python.mavs_interface.MavsVehicleController.max_steering_angle`

max_steering_angle (float): Max steering angle of the vehicle in radians.

6.22.4.3 object

`mavs_python.mavs_interface.MavsVehicleController.object`

object (void): Pointer to the MAVS vehicle controller.

6.22.4.4 steering_coeff

`mavs_python.mavs_interface.MavsVehicleController.steering_coeff`

steering_coeff (float): The gain in the steering PID controller.

6.22.4.5 wheelbase

`mavs_python.mavs_interface.MavsVehicleController.wheelbase`

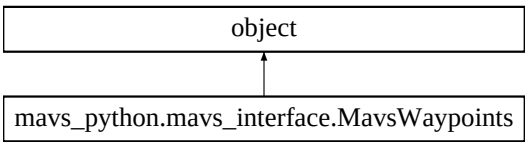
wheelbase (float): Wheelbase of the vehicle in meters.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

6.23 mavs_python.mavs_interface.MavsWaypoints Class Reference

Inheritance diagram for mavs_python.mavs_interface.MavsWaypoints:



Public Member Functions

- def __init__ (self)
- def __del__ (self)
- def UnloadWaypoints (self)
- def Load (self, fname)
- def FillIn (self, spacing)
- def GetNumWaypoints (self)
- def PutWaypointsOnGround (self, scene)
- def GetWaypoint (self, i)
- def GetWaypoints2D (self)
- def GetOrientation (self, i)

Public Attributes

- mavs_waypoints

  mavs_waypoints (void): Pointer to MAVS waypoints object.
- num_waypoints

  num_waypoints (int): Total number of waypoints.
- waypoints

  waypoints (list of floats): Nx2 list of waypoints.

6.23.1 Detailed Description

```
MavsWaypoints class.

Waypoints are a list of x-y locations.

Attributes:
mavs_waypoints (void): Pointer to MAVS waypoints object.
num_waypoints (int): Total number of waypoints.
waypoints (list of floats): Nx2 list of waypoints.
```

6.23.2 Constructor & Destructor Documentation

6.23.2.1 __init__()

```
def mavs_python.mavs_interface.MavsWaypoints.__init__ (
              self )
```

MavsWaypoints constructor.

6.23.2.2 __del__()

```
def mavs_python.mavs_interface.MavsWaypoints.__del__ (
              self )
```

MavsWaypoints destructor.

6.23.3 Member Function Documentation

6.23.3.1 FillIn()

```
def mavs_python.mavs_interface.MavsWaypoints.FillIn (
              self,
              spacing )
```

Fill in gaps between waypoints.

The waypoint follower works best if the waypoints are about 1 meter apart.
Calling this will fill in gaps between waypoints to the specified distance.

Parameters:
spacing (float): Minimum allowed spacing between waypoints.

6.23.3.2 GetNumWaypoints()

```
def mavs_python.mavs_interface.MavsWaypoints.GetNumWaypoints (
              self )
```

Return the number of waypoints.

Returns:
self.num_waypoints (int): The total number of waypoints.

6.23.3.3 GetOrientation()

```
def mavs_python.mavs_interface.MavsWaypoints.GetOrientation (
              self,
              i )
```

Get the path direction at a waypoint.

For a given waypoint, returns the direction to the next waypoint as a quaternion.

Parameters:
i (int): The waypoint number in question.

Returns:
q ([float, float, float, float]): w-x-y-z quaternion specifying the direction of the next waypoint.

6.23.3.4 GetWaypoint()

```
def mavs_python.mavs_interface.MavsWaypoints.GetWaypoint (
              self,
              i )
```

Return a specific waypoint.

Parameters:
i (int): The waypoint number.

Returns:
waypoint ([float, float, float]): The x-y-z position of the waypoint in global ENU.

6.23.3.5 GetWaypoints2D()

```
def mavs_python.mavs_interface.MavsWaypoints.GetWaypoints2D (
              self )
```

Get a 2D list of waypoints, removing the Z coordinate.

Returns:
wp (list of floats): Nx2 list of waypoints in global ENU.

6.23.3.6 Load()

```
def mavs_python.mavs_interface.MavsWaypoints.Load (
              self,
              fname )
```

Load a list of waypoints from an ANVEL .vprp file.

From a .vprp file in ANVEL (in text format), load and generate waypoints.
Example inputs are in the mavs/data/waypoints directory.

Parameters:
fname (string): The name of the .vprp file to load

6.23.3.7 PutWaypointsOnGround()

```
def mavs_python.mavs_interface.MavsWaypoints.PutWaypointsOnGround (
              self,
              scene )
```

Put waypoints on the ground.

Moves the "z" of the waypoints to the ground
Waypoints from a .vprp file may appear to be floating.
This does not affect the MavsVehicleController.

Parameters:
scene (MavsScene): The scene containing the geometry.

6.23.3.8 UnloadWaypoints()

```
def mavs_python.mavs_interface.MavsWaypoints.UnloadWaypoints (
              self )
```

Free pointer to waypoints object and unload waypoints from memory.

6.23.4 Member Data Documentation

6.23.4.1 mavs_waypoints

```
mavs_python.mavs_interface.MavsWaypoints.mavs_waypoints
```

mavs_waypoints (void): Pointer to MAVS waypoints object.

6.23.4.2 num_waypoints

```
mavs_python.mavs_interface.MavsWaypoints.num_waypoints
```

num_waypoints (int): Total number of waypoints.

6.23.4.3 waypoints

```
mavs_python.mavs_interface.MavsWaypoints.waypoints
```

waypoints (list of floats): Nx2 list of waypoints.

The documentation for this class was generated from the following file:

- C:/Users/cgoodin/Desktop/vm_shared/shared_repos/mavs/src/mavs_python/mavs_interface.py

Index