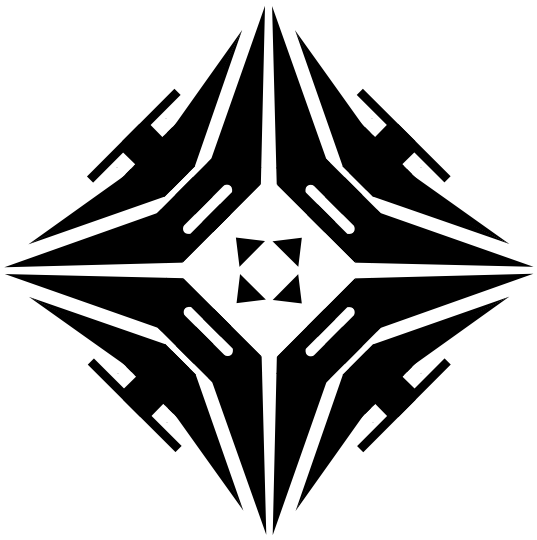


DOXYGEN DOCUMENTATION

FEBRUARY 15, 2019

Potential Field Avoidance

ANDREW TORGESEN

andrew.torgesen@gmail.com

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 pf_avoidance Namespace Reference

Classes

- class PotentialField

Functions

- def obstacleFunction (x, y, z, x_o, y_o, z_o, r_o, h_o)

4.1.1 Detailed Description

```
@package PF Avoidance
Functions for avoiding obstacles dynamically in a path-following setting.

...
```

4.1.2 Function Documentation

4.1.2.1 obstacleFunction()

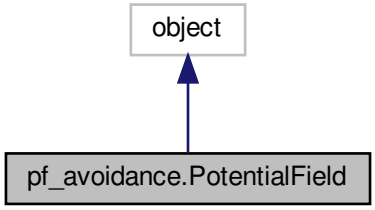
```
def pf_avoidance.obstacleFunction (
            x,
            y,
            z,
            x_o,
            y_o,
            z_o,
            r_o,
            h_o )


Cost function for a single obstacle.

@param x scalar position.
@param y scalar position.
@param z scalar position.
```

Chapter 5

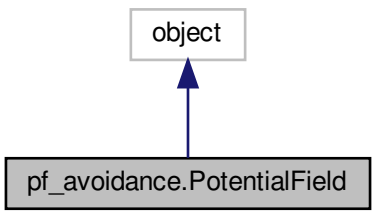
Class Documentation

5.1 pf_avoidance.PotentialField Class Reference

Inheritance diagram for pf_avoidance.PotentialField:



Collaboration diagram for pf_avoidance.PotentialField:



Static Public Member Functions

- def addObstacle (pn, pe, pd, r, h)
- def addBoundaries (filename_or_data)
- def Potential (x)
- def Gradient (x)
- def Hessian (x)
- def directionalDerivative (x, s)
- def secondDirectionalDerivative (x, s)

5.1.1 Detailed Description

```
Class for calculating potential function values and derivatives given
obstacle and boundary positions.

TODO: ADD VISUALIZATION TOOLS
TODO: LOAD BOUNDARIES FROM FILE AND CALCULATE FORCES
```

5.1.2 Member Function Documentation

5.1.2.1 addBoundaries()

```
def pf_avoidance.PotentialField.addBoundaries (
            filename_or_data ) [static]
```

```
FUNCTIONALITY PENDING
```

5.1.2.2 addObstacle()

```
def pf_avoidance.PotentialField.addObstacle (
            pn,
            pe,
            pd,
            r,
            h ) [static]
```

```
Add an obstacle to the potential function.

@param pn Obstacle NORTH position (m)
@param pe Obstacle EAST position (m)
@param pd Obstacle DOWN position (m)
@param r Obstacle radius (m)
@param h Obstacle height (m)
```

5.1.2.3 directionalDerivative()

```
def pf_avoidance.PotentialField.directionalDerivative (
            x,
            s )  [static]
```

Calculate (scalar) directional derivative of a function at vector position x and unit vector direction s.

@param x The 3x1 np.array position x
@param s The 3x1 np.array direction s

5.1.2.4 Gradient()

```
def pf_avoidance.PotentialField.Gradient (
            x )  [static]
```

Calculate gradient of the potential at vector position x.

@param x The 3x1 np.array position x

5.1.2.5 Hessian()

```
def pf_avoidance.PotentialField.Hessian (
            x )  [static]
```

Calculate Hessian of the potential at vector position x.

@param x The 3x1 np.array position x

5.1.2.6 Potential()

```
def pf_avoidance.PotentialField.Potential (
            x ) [static]
```

Calculate potential at vector position x.

@param x The 3x1 np.array position x

5.1.2.7 secondDirectionalDerivative()

```
def pf_avoidance.PotentialField.secondDirectionalDerivative (
            x,
            s ) [static]
```

Calculate (scalar) second directional derivative of a function at vector position x and unit vector direction s.

@param x The 3x1 np.array position x
@param s The 3x1 np.array direction s

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

- pf_avoidance.py

Index