

# HillDescent Class Reference

## Public Member Functions

**HillDescent** ()

double **getRandomNumber** (double i, double j)

std::vector< double > **hdNeighbour** (double ang1, double ang2, double ndist)

void **StartHillDescend** ()

## Public Attributes

double **min\_A1**

double **min\_A2**

double **max\_A1**

double **max\_A2**

double **unrchbl\_cost**

int **diff**

double **A1\_init**

double **A2\_init**

## Detailed Description

Author: Poulastya Mukherjee Description: Class implementation for Hill Descent algorithm Inputs: Joint Angle1, Joint Angle2 and Radius Output: Optimized Angle Values

## Constructor & Destructor Documentation

### ◆ HillDescent()

HillDescent::HillDescent ( )

inline

Author: Poulastya Mukherjee Description: Constructor for Hill Descent algorithm Inputs: None Output: Initialized Parameters

## Member Function Documentation

---

### ◆ getRandomNumber()

```
double HillDescent::getRandomNumber ( double i,  
                                     double j  
                                     )
```

[inline](#)

Author: Poulastya Mukherjee Description: Class implementation for Hill Descent algorithms Inputs: 2 integers Output: Random Number between 2 integer inputs

### ◆ hdNeighbour()

```
std::vector<double> HillDescent::hdNeighbour ( double ang1,  
                                              double ang2,  
                                              double ndist  
                                              )
```

[inline](#)

Author: Poulastya Mukherjee Description: Neighbor Selection for Hill Descent algorithm Inputs: Joint Angle1, Joint Angle2 and Radius Output: Randomly Selects a neighbor using Euclidian Distance

### ◆ StartHillDescend()

```
void HillDescent::StartHillDescend ( )
```

[inline](#)

Author: Poulastya Mukherjee Description: Begin Hill Descent algorithm Inputs: Starts the Hill Descent Algorithm using the parameters set Output: Returns the final joint angle values

## Member Data Documentation

---

### ◆ A1\_init

double HillDescent::A1\_init

initial value of joint angle1 of table to be read from Technology Model file

### ◆ A2\_init

double HillDescent::A2\_init

initial value of joint angle2 of table to be read from Technology Model file

### ◆ diff

int HillDescent::diff

Cost Difference between next and current state

### ◆ max\_A1

double HillDescent::max\_A1

Maximum value of joint angle1 of table input from Technology Model File

### ◆ max\_A2

double HillDescent::max\_A2

Maximum value of joint angle2 of table input from Technology Model File

### ◆ min\_A1

double HillDescent::min\_A1

Minimum value of joint angle1 of table input from Technology Model File

#### ◆ min\_A2

double HillDescent::min\_A2

Minimum value of joint angle2 of table input from Technology Model File

#### ◆ unrchbl\_cost

double HillDescent::unrchbl\_cost

Cost Value for states which are unreachable

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

- ServerSensor\_ompl.cpp

Generated by  1.8.14