DHTable Class Reference

Classes

struct	**Frame**

Public Member Functions

	DHTable () **DHTable** Constructor with no Parameters. More...
	~DHTable () Destructor for the **DHTable** Class. More...
void	**addFrame** (const **Frame** &) addFrame Method for the user to describe the Robot The **Frame** access is 1 indexed. (Preserving 0 for the base frame) Frames must be added in the order of relation. i.e. frame 1 describes the relationship from frame 0 to frame 1 and frame 2 the relationship from frame 1 to frame 2. More...
Frame	**getFrame** (std::vector< **Frame** >::size_type) getFrame returns a copy of the frame that was stored More...
Eigen::Matrix4d	**getTransform** (std::vector< **Frame** >::size_type, std::vector< **Frame** >::size_type) getTransform This function returns the Transformation Matrix from one frame to the another. Currently it is required that the first frame must precede the second frame. More...

Private Member Functions

Eigen::Matrix4d	**getTransform** (std::vector< **Frame** >::size_type) getTransform This function acts as a helper by taking the index of a frame, retrieving that frame and then using those parameters to build a tranformation matrix. More...

Private Attributes

std::vector< **Frame** >	**frames**

Constructor & Destructor Documentation

◆ DHTable()

DHTable::DHTable ( )

DHTable Constructor with no Parameters.

Parameters

None.

Returns

None.

◆ ~DHTable()

DHTable::~DHTable ( )

Destructor for the **DHTable** Class.

Parameters

None.

Returns

None.

Member Function Documentation

◆ addFrame()

void DHTable::addFrame (const **Frame** & aFrame)

addFrame Method for the user to describe the Robot The **Frame** access is 1 indexed. (Preserving 0 for the base frame) Frames must be added in the order of relation. i.e. frame 1 describes the relationship from frame 0 to frame 1 and frame 2 the relationship from frame 1 to frame 2.

Parameters

const Frame &, The **Frame** that will be added to the DH Table

Returns

None.

◆ getFrame()

DHTable::Frame DHTable::getFrame (std::vector< **Frame** >::size_type aFrameIdx)

getFrame returns a copy of the frame that was stored

Parameters

std::vector<Frame>::size_type 1 Indexed index of desired frame

Returns

A copy of the frame.

◆ getTransform() [1/2]

Eigen::Matrix4d DHTable::getTransform (std::vector< **Frame** >::size_type aFrameIdx) private

getTransform This function acts as a helper by taking the index of a frame, retrieving that frame and then using those parameters to build a tranformation matrix.

Parameters

size_type,The index of the frame being transformed to. This function assumed that the tranform from is from the frame that

Returns

Eigen::Matrix4d The Transformation Matrix to go to frame index from the previous frame.

◆ getTransform() [2/2]

Eigen::Matrix4d DHTable::getTransform (std::vector< **Frame** >::size_type aStartFrame,
std::vector< **Frame** >::size_type aEndFrame
)

getTransform This function returns the Transformation Matrix from one frame to the another. Currently it is required that the first frame must precede the second frame.

Parameters

size_type,The index of the frame being transformed from.

size_type,The index of the frame being transformed to.

Returns

Eigen::Matrix4d The Transformation Matrix to go from frame one to frame two.

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

- C:/Users/ethan/Desktop/Doxygen Environment/ENPM808X_Midterm-master/include/**DHTable.hpp**
- C:/Users/ethan/Desktop/Doxygen Environment/ENPM808X_Midterm-master/app/DHTable.cpp

Generated by doxygen 1.8.16