PID Library

1.0.0

Generated by Doxygen 1.9.5

Chapter 1

Namespace Index

1.1 Package List

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Namespace Documentation

3.1 PIDLibrary Namespace Reference

3.2 PIDLibrary.Control Namespace Reference

Classes

- class PIDController

  The base type for any PID controllers.
- class PIDRotationController

  Provides functionality for computing the required acceleration to orient a Rigidbody.

Chapter 4

Class Documentation

4.1 PIDLibrary.Control.PIDController Class Reference

The base type for any PID controllers.

Public Member Functions

- **PIDController** (float pKp, float pKi, float pKd)
- float Update (float pError, float pDeltaTime)

  Updates the PID controller and returns the output.
- void SetIntegral (float pIntegral)

  Sets the undetlying 'integral' component of the PID controller.
- void SetDerivative (float pDerivative)

  Sets the underlying 'derivative' component of the PID controller.
- void SetPreviousError (float pPreviousError)

  Sets the underlying 'previous error' component of the PID controller.

Public Attributes

- float **Kp**

  The proportion tuning value.
- float **Ki**

  The integral tuning value.
- float **Kd**

  The derivative tuning value.

4.1.1 Detailed Description

The base type for any PID controllers.

Author: Mathew Aloisio

4.1.2 Member Function Documentation

4.1.2.1 SetDerivative()

```
void PIDLibrary.Control.PIDController.SetDerivative (
            float pDerivative )
```

Sets the underlying 'derivative' component of the PID controller.

Parameters

pDerivative	

4.1.2.2 SetIntegral()

```
void PIDLibrary.Control.PIDController.SetIntegral (
            float pIntegral )
```

Sets the undetlying 'integral' component of the PID controller.

Parameters

pIntegral	

4.1.2.3 SetPreviousError()

```
void PIDLibrary.Control.PIDController.SetPreviousError (
            float pPreviousError )
```

Sets the underlying 'previous error' component of the PID controller.

Parameters

pPreviousError	

4.1.2.4 Update()

```
float PIDLibrary.Control.PIDController.Update (
            float pError,
            float pDeltaTime )
```

Updates the PID controller and returns the output.

Parameters

pError	
pDeltaTime	

Returns

the output of the updated PID controller.

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

- PIDController.cs

4.2 PIDLibrary.Control.PIDRotationController Class Reference

Provides functionality for computing the required acceleration to orient a Rigidbody.

Public Member Functions

- **PIDRotationController** (float pKp, float pKi, float pKd)
- void **SyncPIDs** ()

  Syncs all PID tuning parameters.
- void **SyncPIDsIfDirty** ()

  Syncs all PIDs only if the tuning values are out of sync.
- Vector3 ComputeRequiredAngularAcceleration (Quaternion pFromRotation, Quaternion pToRotation, Vector3 pCurrentAngularVelocity, float pDeltaTime)

  Computes the required angular acceleration to rotate the object associated with the PID loops from the rotation, pFromRotation, to the target rotation, pToRotation.
- Quaternion Update (Quaternion pError, Quaternion pDelta, float pDeltaTime)

  Updates the PID controllers and returns the result.

Public Attributes

- float **Kp**

  The proportion tuning value.
- float **Ki**

  The integral tuning value.
- float **Kd**

  The derivative tuning value.

4.2.1 Detailed Description

Provides functionality for computing the required acceleration to orient a Rigidbody.

Author: Mathew Aloisio

4.2.2 Member Function Documentation

4.2.2.1 ComputeRequiredAngularAcceleration()

```
Vector3 PIDLibrary.Control.PIDRotationController.ComputeRequiredAngularAcceleration (
            Quaternion pFromRotation,
            Quaternion pToRotation,
            Vector3 pCurrentAngularVelocity,
            float pDeltaTime )
```

Computes the required angular acceleration to rotate the object associated with the PID loops from the rotation, pFromRotation, to the target rotation, pToRotation.

Updates the underlying PIDs.

Parameters

pFromRotation	
pToRotation	
pCurrentAngularVelocity	
pDeltaTime	

Returns

the angular acceleration required to rotate the relevant object from the specified rotation to the desired rotation under the current angular velocity condition and relevant time delta between PID updates.

4.2.2.2 Update()

```
Quaternion PIDLibrary.Control.PIDRotationController.Update (
            Quaternion pError,
            Quaternion pDelta,
            float pDeltaTime )
```

Updates the PID controllers and returns the result.

Parameters

pError	
pDelta	
pDeltaTime	

Returns

the result of the latest PID update.

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

- PIDRotationController.cs

Index