Handshaking

Generated by Doxygen 1.8.12

Contents

Chapter 1

README

IMPORTANT:

This code has been tested on Mac OSX and Ubuntu. We do NOT support Windows. If you endeavour to run it on Windows, you're on your own.

REQUIRED DEPENDENCIES:

```
OpenGL https://www.opengl.org

GLFW http://www.glfw.org

GLEW (Linux ONLY) http://glew.sourceforge.net

V-REP http://www.coppeliarobotics.com
```

COMPILING

First, make sure that all the required dependencies (see above) are installed.

For Mac, you can use the .xcodefile and simply run it or follow these Ubuntu instructions:

```
cd wherever_the_code_is/build

cmake ..

make
```

RUNNING

```
./Graphical
```

Before running ./Graphical, open the VREP file mico.ttt in the V-REP file.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 CPG Class Reference

Public Member Functions

- CPG ()
- CPG (const double ∗params, std::vector< double > var)
- ∼CPG ()
- void setInput (const double f)
- double step (const double vmes, const double t, const double dt)
- void init ()
- void finalise ()
- CPG & operator= (const CPG &cpg)
- std::vector< double > getData ()

Friends

- void coupleCPG (CPG ∗cpg1, CPG ∗cpg2)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 CPG::CPG ( )

CPG constructor

See also

CPG(const double∗ params, std::vector<double> var)

3.1.1.2 CPG::CPG (const double ∗ *params*, std::vector< double > *var*)

CPG constructor

See also

CPG()

3.1.1.3 CPG::∼CPG ( )

CPG destructor

3.1.2 Member Function Documentation

3.1.2.1 void CPG::finalise ( )

finalisation function, to be called after init

See also

CPG(const double∗ params, std::vector<double> var)

3.1.2.2 std::vector< double > CPG::getData ( )

data getter function

Returns

a vector

See also

step(const double vmes, const double t, const double dt)

3.1.2.3 void CPG::init ( )

CPG initialisation function

See also

finalize()

3.1.2.4 CPG& CPG::operator= (const CPG & *cpg*) [inline]

CPG equal operator

3.1.2.5 void CPG::setInput (const double *f*)

sets the GPS input

Parameters

f	input force

3.1.2.6 double CPG::step (const double *vmes,* const double *t,* const double *dt*)

step function

Parameters

vmes	current joint velocity
t	current time
dt	timestep

3.1.3 Friends And Related Function Documentation

3.1.3.1 void coupleCPG (CPG ∗ *cpg1,* CPG ∗ *cpg2*) [friend]

CPG coupling function

Parameters

cpg1	first CPG
cpg2	secong CPG

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

- CPG.h
- CPG.cpp

3.2 PIDController Class Reference

Public Member Functions

- PIDController ()
- PIDController (const double kp, const double ki, const double kd, const double dt=0.01)
- ∼PIDController ()
- double update (const double currentPos, const double targetVel)

3.2.1 Constructor & Destructor Documentation

3.2.1.1 PIDController::PIDController ( )

PID Controller Constructor

See also

PIDController(const double kp, const double ki, const double kd, const double dt)

3.2.1.2 PIDController::PIDController (const double *kp*, const double *ki*, const double *kd*, const double *dt* = `0.01`)

PID Controller Constructor

Parameters

kp	P gain (double)
ki	I gain (double)
kd	D gain (double)
dt	timestep (double)

See also

PIDController()

3.2.1.3 PIDController::∼PIDController ( )

PID Controller Destructor

3.2.2 Member Function Documentation

3.2.2.1 double PIDController::update (const double *currentPos*, const double *targetVel*)

updates the controller. Here we transform velocity control into position control to use the traditional PID formulaes.

Parameters

currentPos	current joint angular position (double)
targetVel	target velocity (double)

Returns

velocity to be applied to the joint (double)

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

- PIDController.h
- PIDController.cpp

Index