controller.Controller Class Reference

A controller object. More...

Public Member Functions

def **__init__** (self, **K_p**=1, **setpoint**=0)
Constructor for the controller. More...

def **clear_data** (self)
Clears the data in instance variables time and vals. More...

def **get_data** (self)
Prints time and value data to the console. More...

def **run** (self, measurement)
Returns output value to actuator. More...

def **set_Kp** (self, **K_p**)
The "set_Kp()" function sets the proportional gain value, K_p.

def **set_setpoint** (self, **setpoint**)
The "set_setpoint()" function sets the setpoint value.

Public Attributes

K_p
Creates instance variables for K_p, setpoint, time and vals. More...

setpoint
Sets the instance variable of setpoint to the function input.

time
Offsets time data by the initial value for time, such that the time data begins at 0.

vals

Detailed Description

A controller object.

The controller class is a general proportional controller. The initialization consists of setting a proportional gain and setpoint. The controller must then be repeatedly called and given a measurement of the value being controlled. The controller will then return a value to be output to the controller. As an added functionality, the controller records its time and output.

Author

Rahul Goyal, Cameron Kao, and Harry Whinnery

Copyright

License Info

Date

January 31, 2019

Constructor & Destructor Documentation

◆ __init__()

```
def controller.Controller.__init__ ( self,
                                     K_p = 1,
                                     setpoint = 0
                                   )
```

Constructor for the controller.

The constructor is called by passing K_p and the setpoint. The constructor then saves these values as variables. It also creates empty arrays for the controller’s time and values.

Parameters

K_p	Proportional gain constant for controller.
setpoint	Initial sepoint for controller. Determines where the controller will try to go when the run function is called.

Member Function Documentation

◆ clear_data()

```
def controller.Controller.clear_data ( self )
```

Clears the data in instance variables time and vals.

The "clear_data" function is used whenever the user would like to clear the existing time and vals variables

◆ get_data()

def controller.Controller.get_data (self)

Prints time and value data to the console.

When the "get_data()" function is called, it offsets the time data by the inital value for time, such that the time data begins at 0. It then prints the time and value data to the console.

◆ run()

```
def controller.Controller.run ( self,
                                measurement
                              )
```

Returns output value to actuator.

The "run()" function passes the sensor measurement and outputs an actuation value based on the K_p and setpoint. The avtuation value saturates at -100 and 100.

Member Data Documentation

◆ K_p

controller.Controller.K_p

Creates instance variables for K_p, setpoint, time and vals.

Sets the instance variable of K_p to the function input.

Data will be later put into time and vals, where time is the time passed and vals is the input measurement

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

- **controller.py**