com._604robotics.utils

# Class VelocityController

java.lang.Object
    com._604robotics.utils.VelocityController

---

```
public class VelocityController
extends Object
```

Class for controlling a motor's velocity, rather than its power directly. Uses a PID loop to scale to said velocity, and a distance-calibrated encoder for feedback.

**Author:**

Michael Smith , Kevin Parker

## Constructor Summary

**Constructors**

| Constructor and Description |
| --- |
| **VelocityController**(double p, double i, double d, **Encoder** encoderLeft, **Encoder** encoderRight, **RobotDrive** robotDrive, **Gyro** gyro)<br>Initializes a new VelocityController. |

## Method Summary

**Methods**

| Modifier and Type | Method and Description |
| --- | --- |
| void | **disable**()<br>Disables the VelocityController. |
| void | **enable**()<br>Enables the VelocityController. |
| double | **getActualVelocity**()<br>Gets the actual, current velocity. |
| double | **getVelocity**()<br>Gets the current target velocity. |
| boolean | **isEnabled**()<br>Is the VelocityController currently enabled? |
| void | **setAngleGains**(double pAngle, double iAngle, double dAngle)<br>Based on gyro angles TODO - javadoc |
| void | **setGains**(double p, double i, double d)<br>Reconfigures the gains on the PIDController. |
| void | **setVelocity**(double velocity)<br>Sets the target velocity. |

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### VelocityController

```
public VelocityController(double p,
                  double i,
                  double d,
                  Encoder encoderLeft,
                  Encoder encoderRight,
                  RobotDrive robotDrive,
                  Gyro gyro)
```

Initializes a new VelocityController.

**Parameters:**

**Parameters:**

`p` - The proportional term for the PIDController.

`i` - The integral term for the PIDController.

`d` - The derivative term for the PIDController.

`encoder` - The encoder to use for feedback.

`output` - The PIDOutput to control. Usually some sort of motor.

## Method Detail

### getVelocity

```
public double getVelocity()
```

Gets the current target velocity.

**Returns:**

The current target velocity.

### getActualVelocity

```
public double getActualVelocity()
```

Gets the actual, current velocity.

**Returns:**

The actual, current velocity.

### setVelocity

```
public void setVelocity(double velocity)
```

Sets the target velocity.

**Parameters:**

`velocity` - The target velocity to set.

### setGains

```
public void setGains(double p,
            double i,
            double d)
```

Reconfigures the gains on the PIDController.

**Parameters:**

`p` - The proportional term for the PIDController.

`i` - The integral term for the PIDController.

`d` - The derivative term for the PIDController.

### setAngleGains

```
public void setAngleGains(double pAngle,
                 double iAngle,
                 double dAngle)
```

Based on gyro angles TODO - javadoc

**Parameters:**

`p` - The

`i` - The

`d` - The

### enable

```
public void enable()
```

Enables the VelocityController.

### disable

```
public void disable()
```

Disables the VelocityController.

### isEnabled

```
public boolean isEnabled()
```

Is the VelocityController currently enabled?

**Returns:**

Whether or not the VelocityController is currently enabled.