

PIDSubsystems for built-in PID control

If a mechanism uses a sensor for feedback then most often a PID controller will be used to control the motor speed or position. Examples of subsystems that might use PID control are: elevators with potentiometers to track the height, shooters with encoders to measure the speed, wrists with potentiometers to measure the joint angle, etc.

There is a PIDController class built into WPILib, but to simplify its use for command based programs there is a PIDSubsystem. A PIDSubsystem is a normal subsystem with the PIDController built in and exposes the required methods for operation.

A PIDSubsystem to control the angle of a wrist joint

```
1 package org.usfirst.frc190.GearsBotME.subsystems;
Source View 2 import edu.wpi.first.wpilibj.*;
3 import edu.wpi.first.wpilibj.command.PIDSubsystem;
4 import org.usfirst.frc190.GearsBotME.RobotMap;
5
6 public class Wrist extends PIDSubsystem { 1
7     Victor motor = RobotMap.wristMotor;
8     AnalogChannel pot = RobotMap.wristPot;
9
10
11     public Wrist() {
12         super("Wrist", 2.0, 0.0, 0.0);
13         setAbsoluteTolerance(0.05);
14         getPIDController().setContinuous(false);
15     }
16
17     public void initDefaultCommand() {
18     }
19
20     protected double returnPIDInput() { 3
21         return pot.getAverageVoltage();
22     }
23
24     protected void usePIDOutput(double output) { 4
25         motor.pidWrite(output);
26     }
27 }
28
```

In this example you can see the basic elements of a PIDSubsystem for the wrist joint:

1. The Wrist subsystem extends PIDSubsystem.
2. The constructor passes a name for the subsystem and the P, I, and D constants that are used when computing the motor output values.

PIDSubsystems for built-in PID control

3. The `returnPIDInput()` method is where you return the sensor value that is providing the feedback for this subsystem. In this case it's a potentiometer connected to an `AnalogChannel`. This method is called about every 20ms and is used for the PID output calculation.
4. The `usePIDOutput` method is where the computed output value from the `PIDController` is applied to your motor. This method is called about every 20 ms to update the motor speed based on the PID parameters from the constructor and the sensor value from the `returnPIDInput()` method.