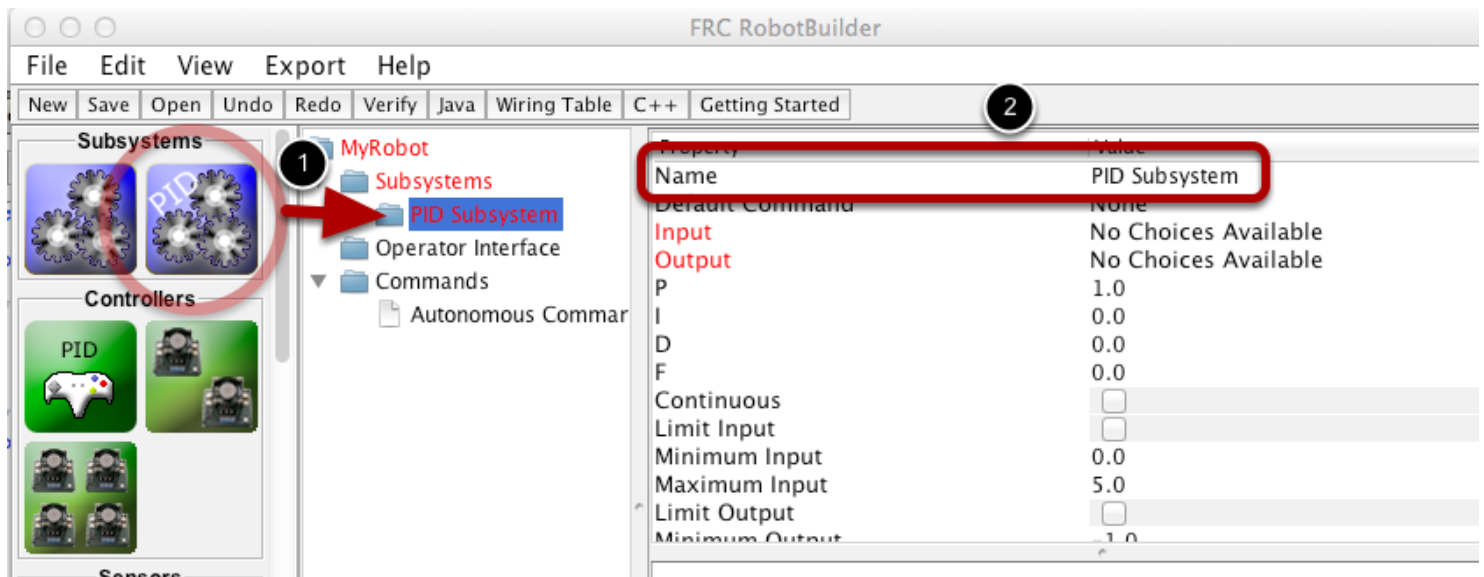


Using PIDSubsystems to control actuators with feedback from sensors

Using PIDSubsystems to control actuators with feedback from sensors

More advanced subsystems will use sensors for feedback to get guaranteed results for operations like setting elevator heights or wrist angles. The PIDSubsystem has a built-in PIDController to automatically set the correct setpoints for these types of mechanisms.

Create a PIDSubsystem



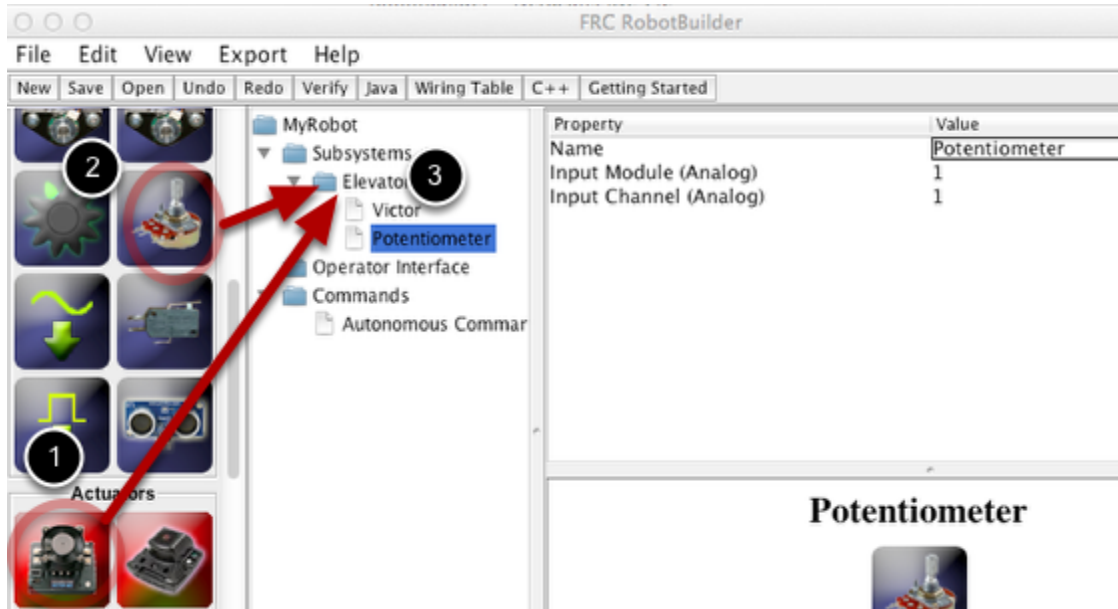
Creating a subsystem that uses feedback to control the position or speed of a mechanism is very easy.

1. Drag a PIDSubsystem from the palette to the Subsystems folder in the robot description
2. Rename the PID Subsystem to a more meaningful name for the subsystem

Notice that some of the parts of the robot description have turned red. This indicates that these components (the PIDSubsystem) haven't been completed and need to be filled in. The properties that are either missing or incorrect are shown in red.

Using PIDSubsystems to control actuators with feedback from sensors

Adding sensors and actuators to the PID Subsystem

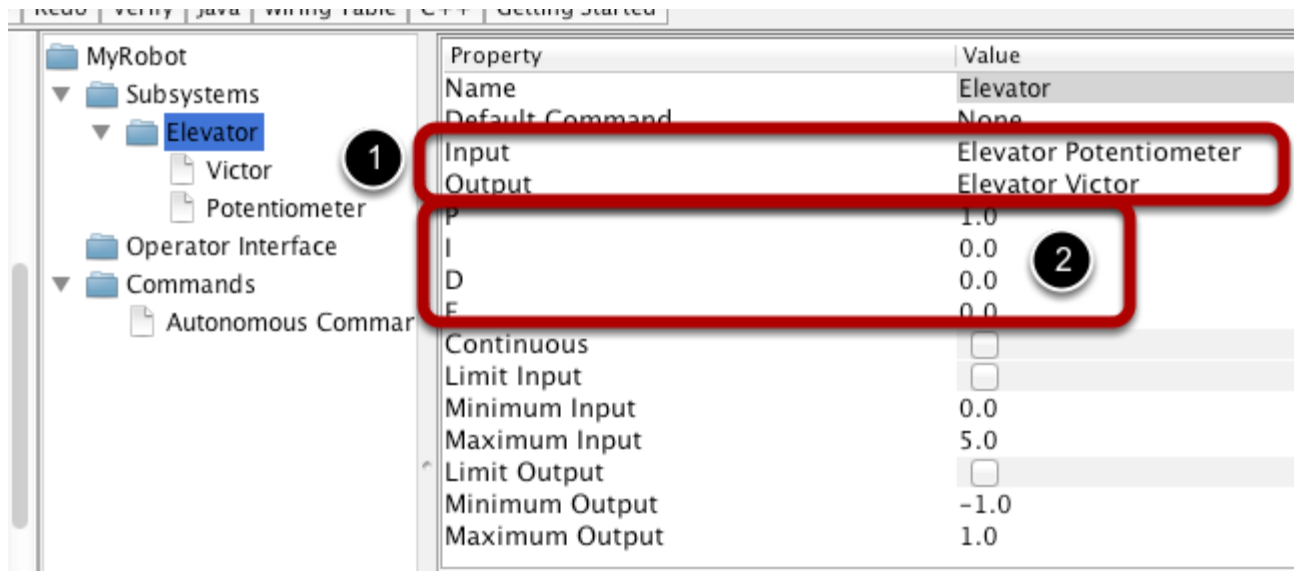


Add the missing components for the PIDSubsystem

1. Drag in the actuator (a speed controller) to the particular subsystem - in this case the Elevator
2. Drag the sensor that will be used for feedback to the subsystem, in this case the sensor is a potentiometer that might give elevator height feedback.

Using PIDSubsystems to control actuators with feedback from sensors

Fill in the PIDSubsystem parameters to get the correct operation of the mechanism



There are a number of parameters for the PIDSubsystem but only a few need to be filled in for most cases

1. The Input and Output components will have been filled in automatically from the previous step when the actuator and sensor were dragged into the PIDSubsystem
2. The P, I, and D values need to be filled in to get the desired sensitivity and stability of the component

See: [Writing the code for a PIDSubsystem in Java](#) and [Writing the code for a PIDSubsystem in C++](#)