

ADVANCED EV3 PROGRAMMING LESSON



EV3 Classroom: Proportional Control

By Sanjay and Arvind Seshan



EV3 CLASSROOM LESSON
BY EV3LESSONS.COM

Lesson Objectives

- Learn what proportional control means and why to use it
- Learn to apply proportional control to the ultrasonic sensors
- Prerequisites: Operator Blocks (Math Blocks), Ultrasonic Sensor

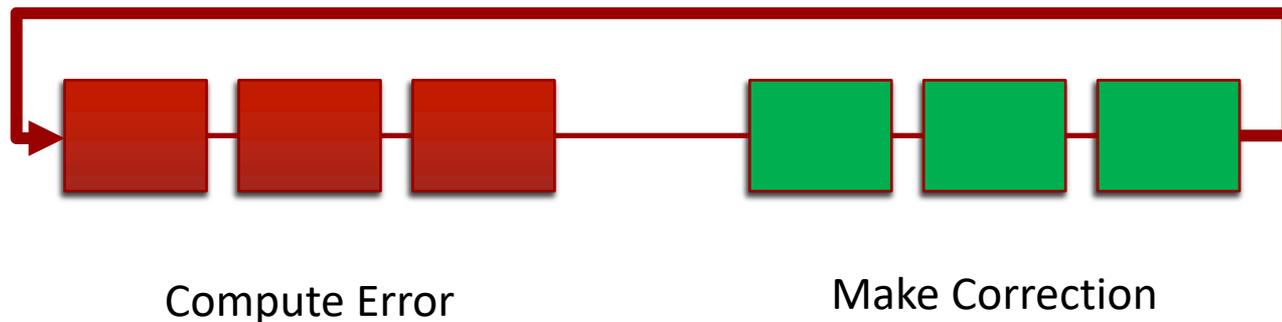
Learn and Discuss Proportional Control

- Let's start with a game
- Imagine that you blindfold one teammate. He or She has to get across the room as quickly as they can and stop exactly on a line drawn on the ground
- The rest of the team has to give the commands.
- When your teammate is far away, the blindfolded person must move fast and take big steps. But as he gets closer to the line, if he keeps running, he will overshoot. So, you have to tell the blindfolded teammate to go slower and take smaller steps.
- You have to program the robot in the same way!



What Proportional Control Looks Like

- The Pseudocode for every proportional control program consists of two stages:
 - Computing an error → how far is the robot from a target
 - Making a correction → make the robot take an action that is proportional to the error (this is why it is called proportional control). You must multiply the error by a scaling factor to determine the correction.

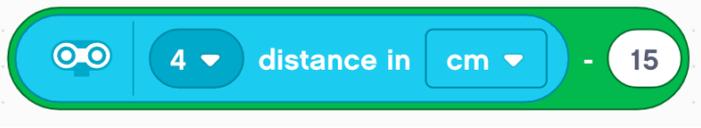


Challenge

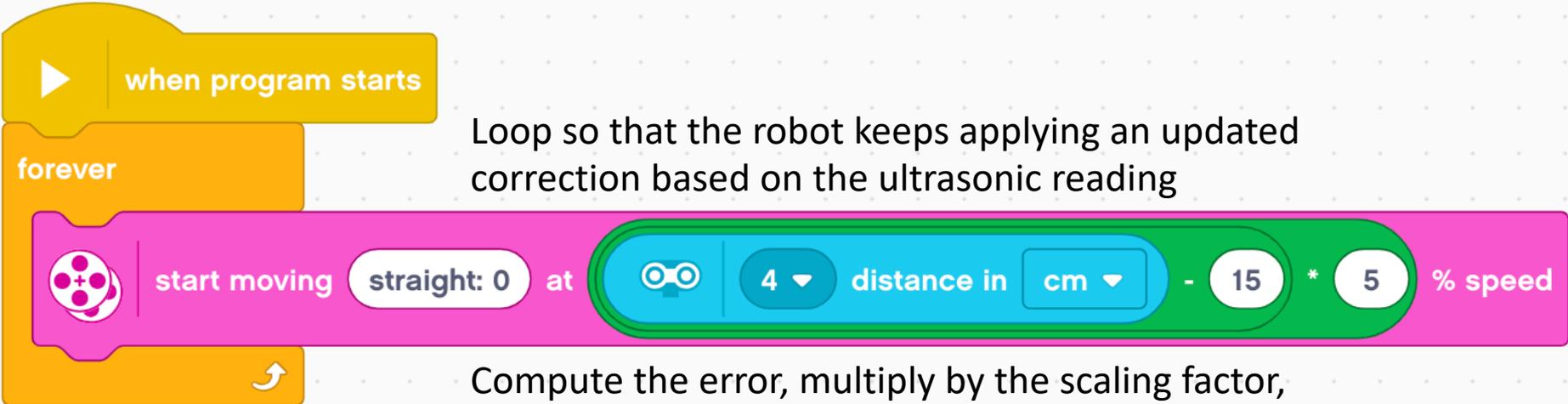
- To learn how to use proportional control, create a Robot Follower program
 - Use proportional control with the ultrasonic sensor to get the robot to stay 15cm away from the human at all times (even when the human moves)

Objective	Error	Correction
Get to a target distance from human	How many cm from target location (current_distance – target_distance)	Move faster based on distance

Challenge

Compute Error	<p>error</p>  <p>A Scratch 'distance in cm' block with a value of 4 and a subtraction sign followed by a value of 15.</p>
How many cm from target location (current_distance – target_distance)	Compute/Apply Correction Multiply by scaling factor and adjust speed based on distance
	 <p>A Scratch 'start moving' block with 'straight: 0' and a scaling factor of 5. A red arrow points to the scaling factor field, which is labeled 'error' below it.</p>

Putting It All Together: Ultrasonic Robot Follower



The image shows a Scratch script on a light gray grid background. It starts with a yellow 'when program starts' block. Below it is an orange 'forever' loop block. Inside the loop is a pink 'start moving' block. The 'start moving' block has a 'straight: 0' field, an 'at' field with a blue ultrasonic sensor icon, a '4' dropdown menu, a 'distance in' field with a 'cm' dropdown menu, a '-' sign, a '15' field, an '*' sign, a '5' field, and a '% speed' field.

when program starts

forever

start moving straight: 0 at 4 distance in cm - 15 * 5 % speed

Loop so that the robot keeps applying an updated correction based on the ultrasonic reading

Compute the error, multiply by the scaling factor, and apply a correction

Discussion Guide

1. **What does proportional control mean?**

Ans. Moving more or less based on how far the robot is from the target distance

2. **What do all proportional control code have in common?**

Ans. Computing an error and making a correction

Credits

- This tutorial was created by Sanjay Seshan and Arvind Seshan
- More lessons at www.ev3lessons.com



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).