Proportional Control

By Sanjay and Arvind Seshan
Lesson Objectives

- Learn what proportional control means and why to use it
- Learn to apply proportional control to different sensors
- Prerequisites: Math Blocks, Color Sensor Calibration, Data Wires
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!
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.
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)

<table>
<thead>
<tr>
<th>Objective</th>
<th>Error</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get to a target distance from human</td>
<td>How many cm from target location (current_distance – target_distance)</td>
<td>Move faster based on distance</td>
</tr>
</tbody>
</table>
# Challenge

## Compute Error

How many cm from target location (current\_distance – target\_distance)

## Compute/Apply Correction

Move faster based on distance

![Diagram showing the process of computing error and applying correction](image)
Putting It All Together: Ultrasonic Robot Follower

We are trying to make a program that stays 7cm away from a moving object. This program uses proportional control.

**Part 1: Compute the Error**
- Error is Current Distance - Target
- We have chosen 15cm as the target.

**Part 2: Computes and Apply the Correction**
- We multiply the Error from Part 1 by 10 to determine the speed
- We picked 5 to create a reasonable range of speeds for our robot

Example:
- Sensor Reading = 10cm
- Error = 3cm
- \( \text{error} \times 5 = (3 \times 5) = 15 \) power
- 15 power is a good speed to be used at 3cm from the target for our robot.
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
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.