

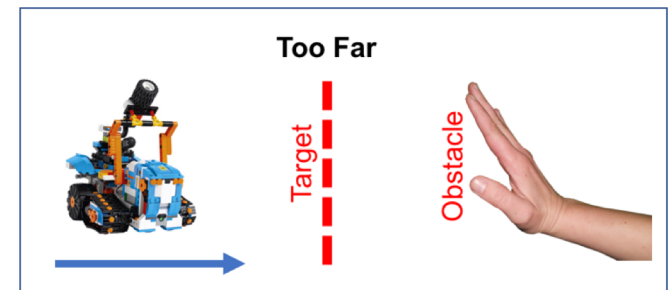
Use the Force: Proportional Control



By Sanjay and
Arvind Seshan

Objectives

- You will learn the concept of proportional control
- You will program your robot to stay at a target distance from an obstacle
- The robot will back away from the obstacle if it is too close and move forward if the obstacle is further away
- In addition, Vernie will say out loud the color he detected
- You will be programming using the Creative Canvas area of the App



What is Proportional Control?

- We will be implementing this movement with proportional control, which makes the robot move slower or faster based on how close it is to the target distance. 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
- For this task, *error* will be how far the robot is from the target distance and *correction* will be moving the robot forward or backward at a particular speed.

What do you
need to build
for this
project?

- Build any design you like with two wheels or treads such that the robot can move forward and backwards
- Be sure to include the Color & Distance Sensor somewhere on the front of your design



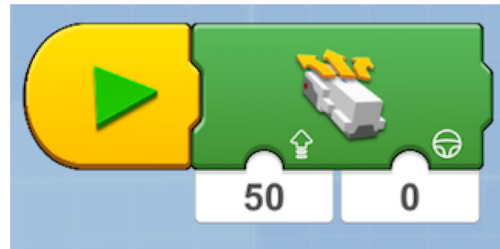
Creative Canvas

- The Creative Canvas area of the App allows you to use the App in a free play mode
- You will have access to all the programming blocks needed to complete this project



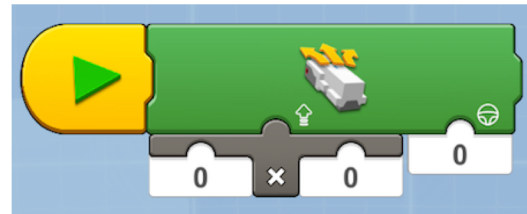
Getting Started

- **Step 1:** Drag in a *Start Sequence Block* from the Yellow Palette
- **Step 2:** Drag in a *Drivebase Move Steering Block* from the Green Palette and place near *Start Sequence Block*

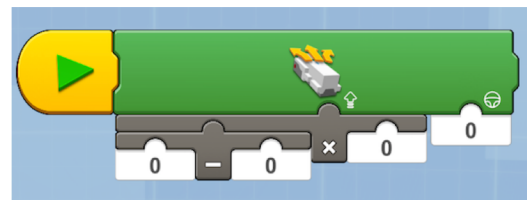


Adding Math Blocks

- Drag in the *Multiply Operator Block* from the White Palette and place as indicated. This will be the part of the code that computes the proportional response to the distance from the target (i.e. the *correction*).

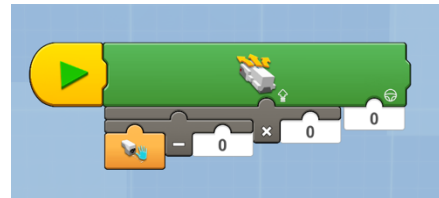


- **Step 4:** Drag in the *Subtraction Operator Block* from the White Palette and place as indicated on the left parameter of the Motor Block. This part of the code will compute the distance from the target (i.e. the *error*)

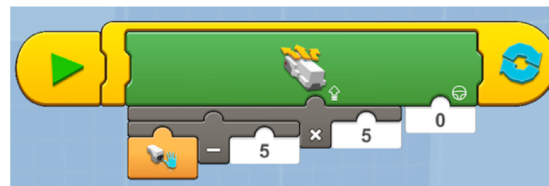


Adding Sensors

- **Step 5:** Drag in the *Sensor Distance Reporter Block* from the Orange Palette and place as indicated in the left parameter of the subtraction block.



- **Step 6:** Change all the parameters to 5. The 5 in the subtraction block sets the target because the range of the sensor is 0-10 and 5 is halfway in between. The multiplication block scales the power so that instead of going 0-5 power, the robot will go 0-25 power. When you take the output from the sensor and subtract 5, the result may be a negative number. Since negative power makes the motor move backwards, the robot will move away from the obstacle.
- **Step 7:** Drag in a *Loop Forever Block* from the Yellow Palette so that the code is repeated again and again forever.



Ready to
play

- Place your hand in front of the robot (really close or far) and see what the robot does
- Surprise your friends with you new Force!



Credits

- Lesson by Sanjay and Arvind Seshan
- More lessons available at BoostLessons.com
- MINDSTORMS lessons available at EV3Lessons.com
- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.