SPIKE PRIME LESSONS

By the Creators of EV3Lessons

GYRO MOVE STRAIGHT

BY SANJAY AND ARVIND SESHAN

LESSON OBJECTIVES

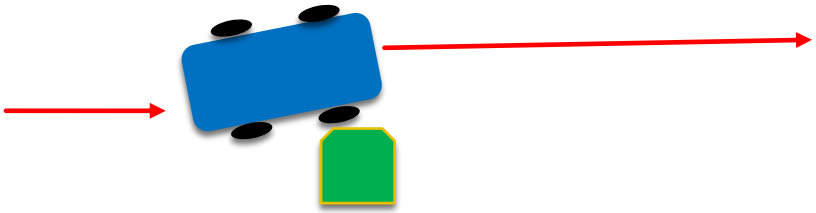
- Learn to apply proportional control to get your robot to move straight
- Learn to apply proportional control to the Gyro sensor move at a particular angle

TIPS FOR SUCCESS

- You must go through the Proportional Line Follower Lesson before you complete this lesson
- You must also complete the Turning With Gyro Lesson

WHAT IS GYRO MOVE STRAIGHT?

- Imagine that you want to drive for 200 cm straight
- As you travel, your robot gets bumped by something
- A gyro move straight program helps the robot correct itself back to straight, but offset by how much it was bumped



HOW IT WORKS

- A proportional line follower and a gyro move straight code share similar properties
- To write a gyro move straight program, you must first think about what the error is and what the correction needs to be

Application	Objective	Error	Correction
Gyro Straight	Make the robot at a constant heading/angle	How far you are from that heading/angle	Turn sharper based on how far you are from that angle
Line Follower	Stay on the edge of the line	How far are our light readings from those at line edge (current_light – target_light)	Turn sharper based on distance from line

PSEUDOCODE

- Set Movement Motors
- Reset your yaw value to 0
- In a loop, compute the error and apply the correction
  - Part 1: Compute Error (How far from target angle)
    - To move straight → Target yaw angle=0 (Note: Assuming a horizontal hub placement, we must look at the yaw direction for the angle offset. This may be different for your setup)
    - Distance from target angle is just current yaw reading
  - Part 2: Compute a Correction that is proportional to the error
    - Multiply the Error from Part 1 by a constant (that you must experiment and discover for your robot)
  - Plug the value from Part 2 into a move block with each motor adjusted proportionally
- Exit loop as required by changing loop block

SOLUTION: GYRO MOVE STRAIGHT

```
from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, Col
orSensor, App, DistanceSensor, Motor, MotorPair
from spike.control import wait_for_seconds, wait_until, Timer
from math import *

hub = PrimeHub()

motor_pair = MotorPair('A', 'E')
hub.motion_sensor.reset_yaw_angle()

while True:
    error = hub.motion_sensor.get_yaw_angle()
    correction = error * -2

    motor_pair.start_tank(int(60+correction), int(60-correction))
    # Note that the int function is used on the inputs to ensure that
    # speed is an integer
```

Reset the yaw angle to set the direction the the robot is trying to stay at.

Calculate the yaw error and correction

Start moving and adjust the steering based on how far off the robot is from its target

Loop so that the robot keeps updating its correction

DISCUSSION GUIDE

1. Compare the proportional line follower code with the proportional move straight code. What similarities and differences do you see?
   Ans. The code is almost the same. The one difference is how the error is calculated. The error is calculated using the gyro sensor. The correction is identical.

2. What if you wanted to travel at a particular angle (not just straight)? How would the code look different?

   Ans. In Part 1 of the solution code, there is no subtraction block because we were just subtracting "0" since our target heading is moving straight. You would have to subtract your current angle from the target angle if you wanted to move at some other angle.

   Target angle = 5 degrees

```
error = hub.motion_sensor.get_yaw_angle() - 5
correction = error * -2
```

CREDITS

- This lesson was created by Sanjay Seshan and Arvind Seshan for SPIKE Prime Lessons
- More lessons are available at www.primelessons.org