LEGO Mindstorms Line Following Project

Aniruddha Pal, Markus Wiktorin

23rd of September 2016

University of Applied Science, Bonn-Rhein-Sieg

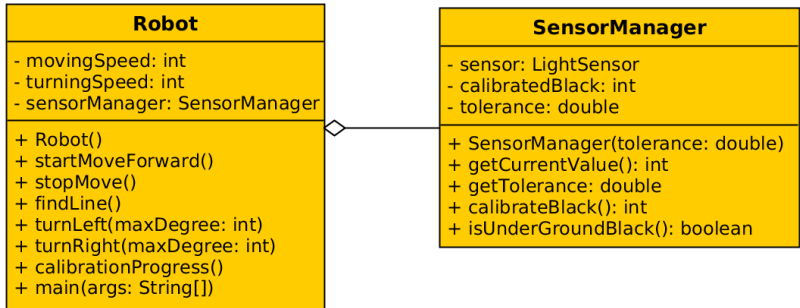Program a LEGO robot which can follow a line.

Light conditions and track can change, the robot should work under different conditions.

Approaches

1. One light sensor in front of the robot
2. Two light sensors in front of the robot
3. Two light sensors in front of the robot using PID
4. Two light sensors, ultrasonic sensor and touch sensor in front of the robot using PID

Robot
- movingSpeed: int
- turningSpeed: int
- sensorManager: SensorManager

+ Robot()
+ startMoveForward()
+ stopMove()
+ findLine()
+ turnLeft(maxDegree: int)
+ turnRight(maxDegree: int)
+ calibrationProgress()
+ main(args: String[])

SensorManager
- sensor: LightSensor
- calibratedBlack: int
- tolerance: double

+ SensorManager(tolerance: double)
+ getCurrentValue(): int
+ getTolerance: double
+ calibrateBlack(): int
+ isUnderGroundBlack(): boolean

Light sensor tries to stay in the middle of the line. If it looses the line it looks left and right in order to find it again.

Bad approach but we tried to use object oriented concepts.

2. Approach (two light sensors)

- We used two light sensors attached infront of the robot side by side.
- The robot calculates the vakue difference of valus between the sensors.
- It takes left or right turn based on the difference of the two sensors.Also adjusts speed to take sharper turns.
- Advantage:
  - Works better than the one sensor approach.
  - Can take sharp turns
  - Faster.
- Disadvantage:
  - Uses more power.
  - Too much wear and tear of the motors. Harmful for it.

3. Approach (two light sensors + PID)

- Same two sensor approach as the last one.
- Calculates the difference between the two motors which works as error value for proportional controller.
- Uses the previous error for better contol of movement. Integral approach.
- Appropiate values of Proportional, Integral and Differential co-efficients used.
- Advantage:
  - Less wear and tear of the motor. Works smoother.
  - Is able to follow the track better.
- Disadvantage:
  - Finding out appropiate co-efficient values.

4. Approach (two light, ultrasonic, touch sensor + PID)

- We found some working values for PID in the third approach
- The robot is also fast on straights
- Next improvement: Make it possible to go on the same track with other robots
  - Use ultrasonic in the front in order to detect other robots
  - This was not working realiable, thats why we added the touch sensor
  - The robot can still crash when it goes to turns, but on straights it is working good