Introduction to Robotics

Lecture 8: Following the Line

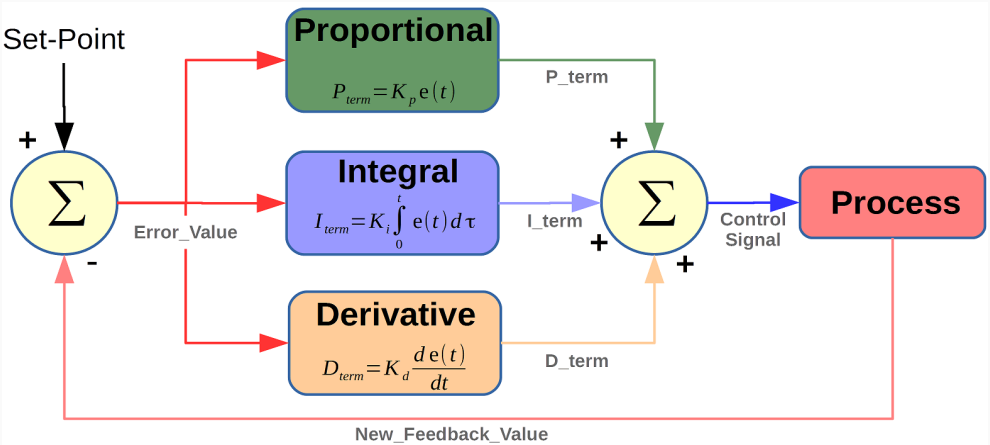
5. 11. 2018

ParaDiSe

Today Goals

- learn to follow the line
- build a blocking interface of atomic steps
  - one step forward
  - turn left/right
  - detect field type
  - check if there is a wall

Feedback Loop – PID Controller



Source: https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/58257/versions/2/screenshot.png

Tips for Implementing PID Controller

- you have to either
  - call the loop at constant rate (tip: use timer 2)
  - compensate for
- PWM frequency vs. control loop frequency
- clamp the output
- find the constants by trial & error

Line Detection

Goal: express position of line as a number such that:

- 0 represents line directly in the center
- when line moves to the right, value increases
- when line moves to the left, value decreases

Function: `int line_position()`

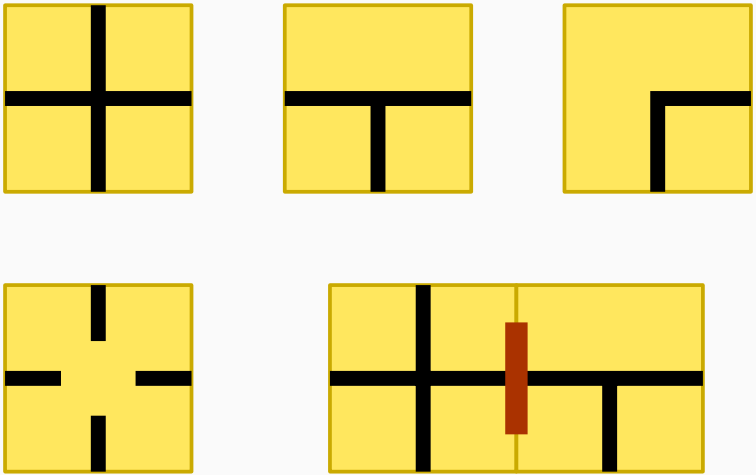
How to implement:

- linear combination of sensor values
- detect peak (and possible interpolate between the sensors)
- sophisticated graphic algorithm?
- machine learning?

Traditional Algorithm for Line Following

- start moving at constant speed (tip: add acceleration phase)
- let PID do its work and based on its output slow down/speed up left & right wheel
- tune the constants
- watch & enjoy

Type of Fields in Our Game Plan



Tasks

- implement line detection
- implement line following

Design blocking API of atomic steps; e.g.:

- `void step()`
- `void rotate_left()`, `void rotate_right()`
- `bool wall_{front, left, right}()`
- `Directions allowed_directions()`
- `bool is_final()`