

# COEX-1

## Cooperative Explorer

Aurélien Werenne  
Advisor: Prof. B. Boigelot

University of Liège

March 4, 2019

# Overview

- 1 General
- 2 Components
- 3 Control
- 4 Exploration & Mapping
- 5 Code

The goal of the robot is to map an unknown environment in a centralized multi-agent setting.

The main features of the robots are:

- Following a black line
- Computing the travelled distance
- Detecting, classifying and handling intersections
- Avoiding obstacles
- Communicating with a central unit

- Arduino Nano
- Reflectance sensor array
- Sharp sensor
- Magnetic encoders
- Pololu micro metal gearmotor
- L298 dual H-bridge
- NiMH Battery 7.2V
- HC-06 Bluetooth module
- ...

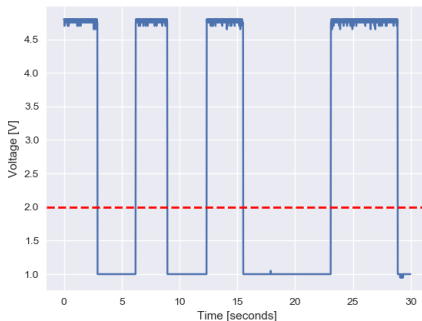
Photos (2 per slides, 4-5 slides)

- List of variables used in equations with small explanations

# Components

## Sharp sensor

Explain flow test. Conclusion.



**Figure:** Obstacle movement (alternating between in and out of reach of sensor)

# Components

Reflectance sensor array (line sensor)

Explain flow test. Explain err line.

Plot.

Conclusion.



The line sensors are fixed vertically well above the given recommendations, such that the robot has place to climb hills. This decision made the line sensor particularly sensitive to ambient light interferences. Thus a shield had to be constructed.

Photo.

# Components

## Bluetooth module (1/2)

Explain sender-receiver distortion.

Plots.

Conclusion.

# Components

## Bluetooth module (2/2)

Logic level Arduino is 0.5 and HC-05 module is 0.3.3 V..

$$\begin{aligned} V_{\text{arduino}} &= \frac{R_2}{R_1 + R_2} V_{\text{blt}} \\ &= \frac{2.2}{3.2} V_{\text{blt}} \end{aligned}$$

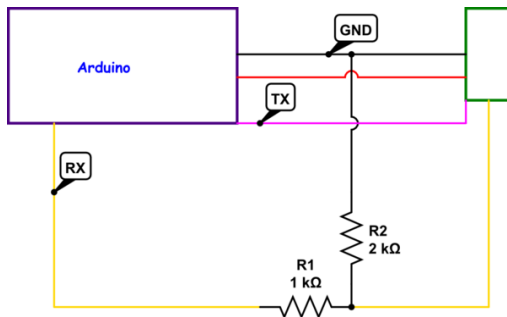


Figure: Voltage divider

# Components

## Quadrature encoders (1/2)

$$w_L = 2\pi f_L \quad \text{with} \quad f_L = \frac{n_L}{N' G_b \Delta t}$$
$$v = \frac{v_L + v_R}{2} = \frac{R(w_L + w_R)}{2}$$

# Components

## Quadrature encoders (2/2)

Remark why divide by 2 for counting. because only 2 pin interrupts.

# Components

## Motors

Conclusion. Need for regulation. Two more curves on same plot but with full/empty battery.

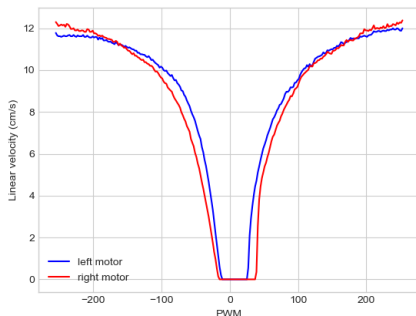


Figure: Relationship between PWM (input) and measured speed (output)

Use of classical PID controller.

$$o_n \leftarrow K_p e_n + K_d \frac{e_n - e_{n-1}}{\Delta t_{n-1:n}} + K_e \sum_{i=0}^n e_i \Delta t_{n-1:n}$$

with the following two precautions called respectively anti-windup and anti-derivative kick:

$$o_n = \begin{cases} \max & \text{if } o_n > \max \\ \min & \text{if } o_n < \min \\ o_n & \text{otherwise.} \end{cases} \quad \text{and} \quad K_d = \begin{cases} 0 & \text{if } n < T \\ K_d & \text{otherwise.} \end{cases}$$

# Control

## Speed & direction

$$\left\{ \begin{array}{l} \alpha = o(e_{direction}) \\ \beta = o(e_{speed}) \\ \gamma = o(e_{align}) \end{array} \right. \Rightarrow \left\{ \begin{array}{l} pwm_L = (\beta - \gamma) + \alpha \\ pwm_R = (\beta - \gamma) - \alpha \end{array} \right.$$



# Control

## Line-following control

Recall plot of sensor.

$$e_{direction} = err_{line} \in [-2500; 2500]$$

Explain perturbation plot.

Perturbation plot.

Conclusion. break out angle. Agressive enough such that not seen as intersection.

# Control

## Speed control (1/4)

Reason. Target and progress speed.

$$v_{n+1} \leftarrow v_n + A \Delta t_{n:n+1}$$

$$\Leftrightarrow \int_0^T a(t) = \int_0^T a'(t)$$

$$\Leftrightarrow AT = \underbrace{dB}_{\text{triangles}} + \underbrace{(T-2d)B}_{\text{rectangle}}$$

We introduce the parameter  $\psi = \frac{d}{T}$  such that

$$B = \frac{A}{1 - \psi}$$

$$v_{n+1} \leftarrow v_n + \int_n^{n+1} a'(t) = v_n + \int_0^{n+1} a'(t) - \int_0^n a'(t)$$

Diagrams.

# Control

## Speed control (3/4)

Reason.

Target and progress speed.

Equations.

Diagrams.

# Control

## Speed control (4/4)

Explain plot.

PID plot.

Conclusion. Room for fine-tuning.

Explain forward. Explain improvement would be to consider theta instead.

$$e_{direction} = v_R - v_L$$

We verify that if

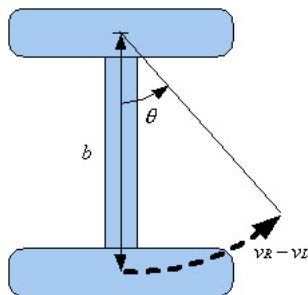
$$v_{R;n} > v_{L;n} \Rightarrow e_n > 0 \Rightarrow \alpha_n > 0 \Rightarrow v_{R;n+1} \uparrow, v_{L;n+1} \downarrow$$

# Control

## Turning (1/2)

From Fig. we can easily deduce the model below by integrating the angular velocity .

$$\theta(t) = \frac{2vt}{b} + \theta_0 \quad \text{because} \quad v_R = -v_L$$



$$\theta_{n+1} \leftarrow \frac{2 v_n \Delta_{n:n+1}}{b}$$

PID.



To explore an unknown environment structured as a maze it needs to be able to:

- Compute the distance travelled since the last intersection
- Detect an intersection
- Turn to the desired intersection

# Exploration & Mapping

Distance (1/5)

First method: Simple equation - explain without mathematics.

Second method: Upperbound of error can be reduced due to relation with distance.

plot relationship

# Exploration & Mapping

Distance (2/5)

Plot comparison with error bars of two methods on test set.

Conclusion.

# Exploration & Mapping

Distance (3/5)

Explain we could try to quantify uncertainty and above certain threshold not accept it.

$$MSE = \sum_{i=0}^N err_{line}^2$$

obtained iteratively by rolling mean method. One plot

# Exploration & Mapping

Distance (4/5)

Explain simpler choice is to discretize from second method discretization.

Remard 7.5-12.5 ....

Plot discretization on test set.

# Exploration & Mapping

Distance (5/5)

Example of resulting big map. Explain annotation

Map.

Explain problem of naive approach.

$$y = \operatorname{argmax}_{x \in \{0,1\}} \operatorname{mode}(x)$$

Boolean value for intersection based on decision rule as follows

$$(y_C = 0) \vee (y_L = 1) \vee (y_R = 1) \vee (y_F = 0)$$

# Exploration & Mapping

## Intersection classification

Explain classification.

Figures of 8 types.



# Exploration & Mapping

## Turning & alignment (1/2)

Problem with naive approach.

Solution + plot + equation.

# Exploration & Mapping

## Turning & alignment (2/2)

Explain plot.

Plot to verify results.

Conclusion.

# Code

## General architecture

Advantage.

Diagram.

Link to code.

# Code

## Pseudo-code exploration

Pseudocode.

