COEX-1

Cooperative Explorer

Aurélien Werenne
Advisor: Prof. B. Boigelot

University of Liège

March 10, 2019

Overview

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
- ...

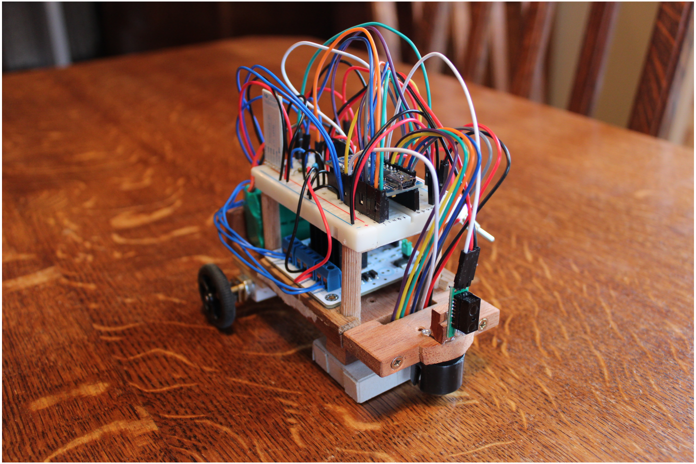


Figure: Front view of COEX-1

Structure (2/3)

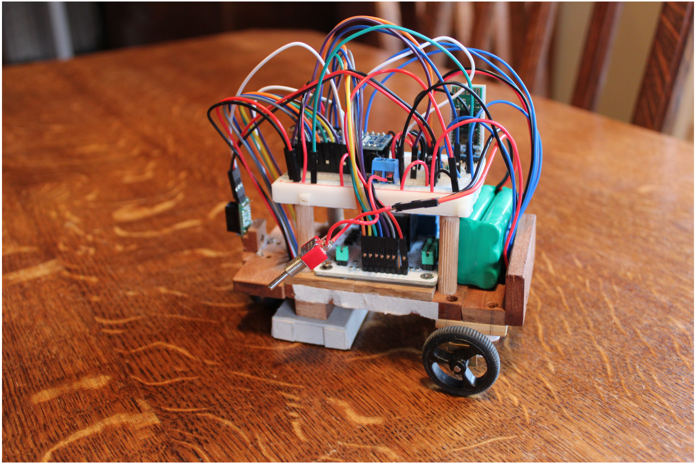


Figure: Side view of COEX-1

General

Structure (3/3)

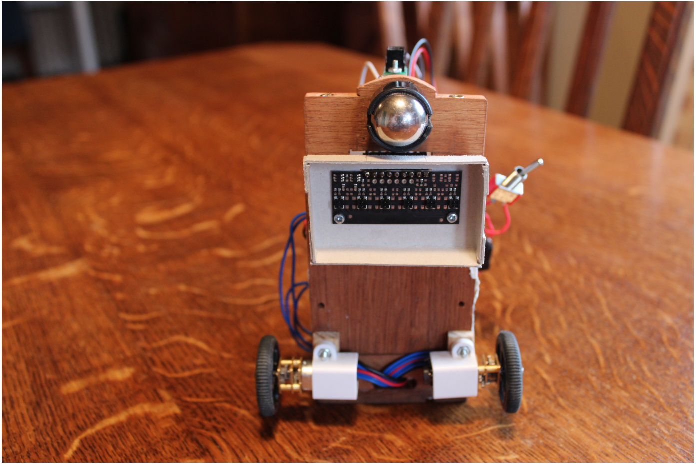


Figure: Bottom view of COEX-1

- List of variables used in equations with small explanations

Components

Sharp sensor

Explain flow test. Conclusion.

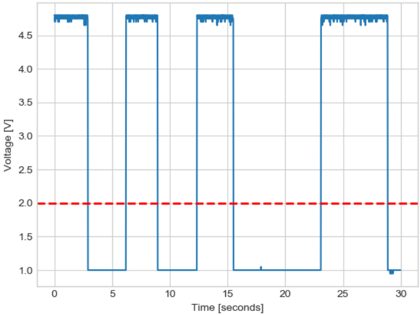


Figure: Obstacle movement (alternating between in and out of reach of sensor)

Explain flow test. Explain err line.Conclusion.

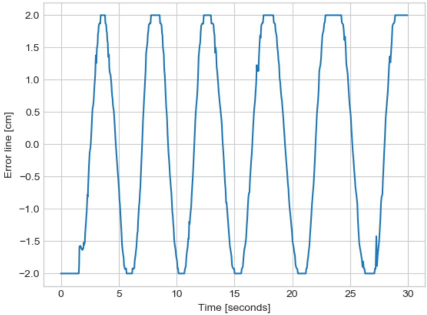


Figure: Obstacle movement (alternating between in and out of reach of sensor)

Components

Shield

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. Explain after wheel.

Explain sender-receiver distortion.

Plots.

Conclusion.

Logic level Arduino is 0 5 and HC 05 module is 0 3.3 V..

$$V_{arduino} = \frac{R_2}{R_1 + R_2} V_{blt}$$

$$= \frac{2.2}{3.2} V_{blt}$$

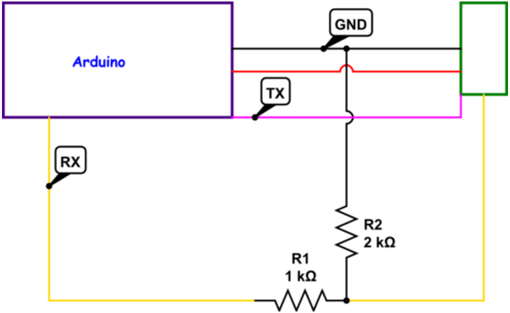


Figure: Voltage divider

Components

Quadrature encoders (1/2)

$$w_L = 2\pi f_L \qquad \text{with} \qquad f_L = \frac{n_L}{N'\,G_b\,\Delta t}$$

$$v = \frac{v_L + v_R}{2} = \frac{R(w_L + w_R)}{2}$$

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

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

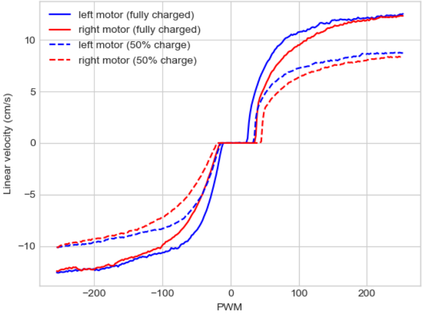


Figure: Relationship between PWM (input) and measured speed (output)

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:

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

Control

Speed & direction

$$\begin{cases} \alpha = o(e_{direction}) \\ \beta = o(e_{speed}) \\ \gamma = o(e_{align}) \end{cases} \Rightarrow \begin{cases} pwm_L = (\beta - \gamma) + \alpha \\ pwm_R = (\beta - \gamma) - \alpha \end{cases}$$

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.

Reason. Target and progress speed.

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

$$\Leftrightarrow \int_0^T a(t) = \int_0^T a'(t)$$
$$\Leftrightarrow A\,T = \underbrace{d\,B}_{triangles} + \underbrace{(T-2d)B}_{rectangle}$$

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

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

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

Diagrams.

Reason.

Target and progress speed.

Equations.

Diagrams.

Explain plot.

PID plot.

Conclusion. Room for fine-tuning.

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

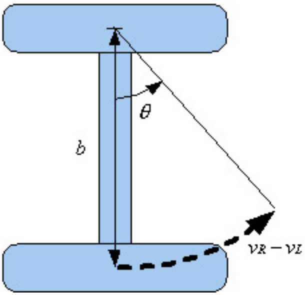
$$e_{direction} = v_R - v_L$$

We verify that if

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

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

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



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

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

First method: Simple equation - explain without mathematics.

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

plot relationship

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

Conclusion.

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

Explain simpler choice is to discretize from second method discretization.

Remard 7.5-12.5 ....

Plot discretization on test set.

Example of resulting big map. Explain annotation

Map.

Explain problem of naive approach.

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

Boolean value for intersection based on decision rule as follows

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

Explain classification.

Figures of 8 types.

Turning & alignment (1/2)

Problem with naive approach.

Solution + plot + equation.

Explain plot.

Plot to verify results.

Conclusion.

Advantage.

Diagram.

Link to code.

Pseudocode.