Wall Following Behavior on E-Puck using Proximity Sensors
Lab Work (P0) For Autonomous Robotics
Master Computer Vision and Robotics
Universite De Bourgogne
Le Creusot Antenna

Pamir Ghimire, GopiKrishna Erabati

April 27, 2017

1 Summary:

Our Objectives in this lab were:
1 . Move the robot forward and stop at some distance from the wall
2. Follow the encountered wall

Our approach was to build small behaviors like 'move forward', 'stop', etc. and organize them into a composite behavior that fulfilled the objectives. To make our robot follow a wall, we developed a 'follow wall' behavior based on PID control.

2 Introduction:

We worked with a robot called e-puck, a differential-drive non-holonomic robot, to develop and deploy simple concepts in autonomous behavior based robotics, namely:

1. Detecting if open space is available in front of the robot and stopping the robot is it is not
2. Making the robot follow a wall (more generally, an obstacle)

Webots, a software for simulating mobile robots, served as our development platform, where we designed and refined controllers that were deployed on the real robot. Along with a simulation of a dimensionally and mechanically accurate model of the e-puck, Webots also provided us several libraries that allowed us to work on an abstract level of robot variables like wheel speeds and distance sensor values without having to bother about the underlying electronics and construction of the robot. We were therefore able to develop and think in terms of discrete behavioral modules to fulfill our tasks.

3 Method:

To accomplish the two objectives of this lab work, we implemented the following set of behaviors:
1. Move forward
2. Turn 90 degrees counter clockwise (CCW)
3. stop
4. Follow wall

We also implemented some functions that returned boolean (yes or no) answers to the following questions:
1. Is there open Space Ahead?
2. Is there a wall to the right of the robot?

To detect whether or not there is open space in front of the robot, we read proximity sensors 0 and 2 and decided that there was a wall in front if

$D_0 > D_{wth}$, or $D_2 > D_{wth}$, D_{wth} being a threshold that we chose [1]. The wall following behavior was such that the wall being followed was always kept on the right side of the robot (facing proximity sensors 0 and 2).

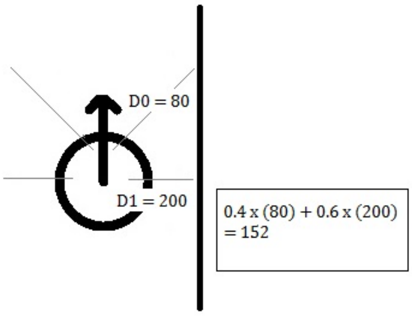

Figure 1: Heuristically choosing values for the PID controlled measure by observing values when robot is oriented along the wall.

For the wall following behavior, we implement the following controller:

$$e(t) = (kD_2(t) + (1-k)D_0(t)) - D_w, \quad k \in \mathcal{R}^1,\ 0 < k < 1 \qquad (1)$$

$$v_{rw} = v_f \qquad (2)$$

$$v_{lw} = v_f + k_p.e(t) + k_d.\frac{\partial e(t)}{\partial t} + k_i.\int_0^t e(t) \qquad (3)$$

In (1),$D_1(t)$ and $D_2(t)$ are measurements returned by proximity sensors 0 and 2, which are in the front and right hand side of an epuck, respectively. Velocities, v_{rw}, v_{lw} and v_f correspond to the left wheel, right wheel and a set 'forward' velocity (just a constant). The derivatives of $e(t)$ were approximated as $e(t) - e(t-1)$ and the integrals as $\sum_{t-1}^{t} e(t)$.

The values of gains k_p, k_d, k_i were picked through trial and error. However, the values of different gains were not the same for the real robot and the simulated one.

A drawback of our method is that because the wall following behavior relies exclusively on infrared-based proximity sensors, the accuracy of the wall-following behavior varies significantly when ambient light conditions change. Further, it also depends on the color of the wall. Concretely, it is hard to follow dark walls since they absorb a big fraction of (infrared) light produced as a probe by the 'active' distance sensor.

4 Experiments and Results:

The simulation video of our wall-following behavior can be found at the following hyperlink:

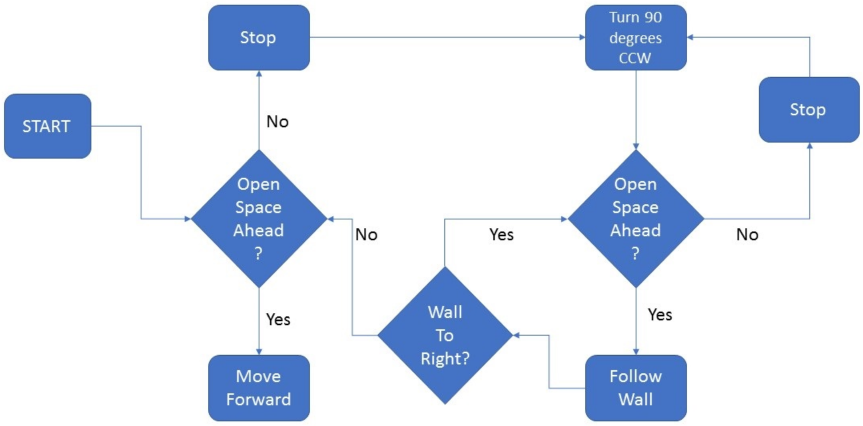

Figure 2: Behavior-transition diagram for a wall follower robot. The 'Follow Wall' behavior is a PID controlled behavior.

Web address: https://www.youtube.com/watch?v=gT462ntTajk&t=3s
Title: Simulation of wall-following epuck _ using Webots

A video of the controller running on the real robot can be found at :
Web address: https://www.youtube.com/watch?v=0B48A3jn3qA
Title: Following wall with a PID_ EPuck + Webots

The PID controller allowed the real e-puck to track walls robustly. We used the following gain values: $k_p = 0.4, k_d = 0.5$ and $k_i = 0.01$. Our v_f was 200 and $k = 0.6$, $D_w = 160$. The robot tracked the walls very accurately when it moved slowly, and sometimes even tracked sharp 'concave' corners. However, moving too slowly could cause the e-puck robot to shake and not move smoothly.

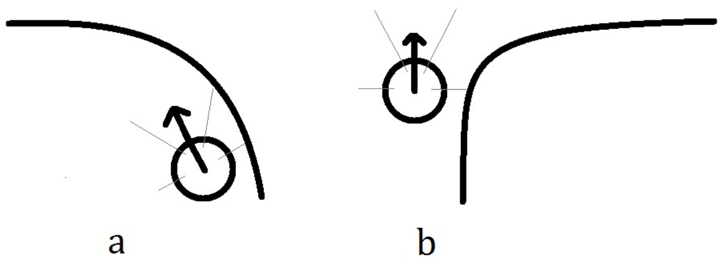

Figure 3: (a) 'Convex' wall vs (b) 'Concave' wall. Our controller tracks convex walls more easily than concave ones.

Although a controller based only on proportional error always worked in simulation, it never worked on the real world. This demonstrated the sharp contrast that can exist between simulation and reality.

What we haven't accounted for : In some instances, when switching between our coded behaviors, the robot exhibited anomalous 'behaviors' that we had neither programmed, nor anticipated (like moving backwards before turning 90 degrees when an obstacle is encountered). We ascribe this to 'careless' switching between behaviors in our hybrid automation.

5 Conclusion:

We conclude that implementing a complex robot behavior by combining smaller behaviors is a effective strategy for inventing and developing complex mobile robot applications.

We also conclude that implementing wall following behavior using only infrared based proximity sensors is not reliable since it is affected significantly by ambient light conditions, color of the wall and well as orientation of the wall.

Finally, although a proportional controller can work fine in simulation as well as a PID, in real world, the PID is far superior to a simple proportional controller (which almost never works). So, the simulation should not be used to make strong predictions about behavior of the mobile robot in the real world.

6 References:

[1] Cyberbotics's documentation on e-puck : https://www.cyberbotics.com/doc/guide/using-the-e-puck-robot

[2] Cyberbotics's tutorial on programming in webots for epuck : https://www.cyberbotics.com/doc

[3] Lecture slides of Dr. Xevi Cufi.