IoT Replan

January 2019

1 Introduction

This is a first attempt to describe the Theoretical Background of IoT Replan Project. As you may and most probably will notice there are many English mistakes here, but let's say it's just a draft, a version 0.1.

2 Theory

2.1 Measurements and Basics

Using the light sensors we manage to calculate the angular velocities of the wheels. The light sensor is an infrared transmitter - receiver between a smaller (than the Robot wheel) wooden wheel with 20 gaps. When a gaps is found the receiver (thereafter reffered as pulse) sends 0, otherwise it sends 1 at the Raspberry output. As the wooden wheel turns by the motor, you can calculate the angular velocity:

- The angular velocity can be found like that:

$$w_r(t) = \frac{\#pulses_r * df_r}{dt_r}$$

  where $\#pulses$ referers to the number of pulses made in dt , and dS refers to the angle of one pulse.

- When we count 20 pulses a full circle of the wooden wheel is made (2π). So, using the rule of three we can have:

$$\frac{20pulses}{\#pulses} = \frac{2\pi}{f}$$

  so finally,

$$f = \frac{\#pulses * \pi}{10dt}$$

  As a result,

$$w_r(t) = \frac{\#pulses_r * \pi}{10dt_r}$$

We can calculate w_r, w_l by calculating the number of pulses and the dt of each measurement.

The next thing is how the Robot moves. To move the robot there is a AlphaBot library function called setMotor(right_voltage , left_votlage). So consider the robot's position as in Figure 1(b). So

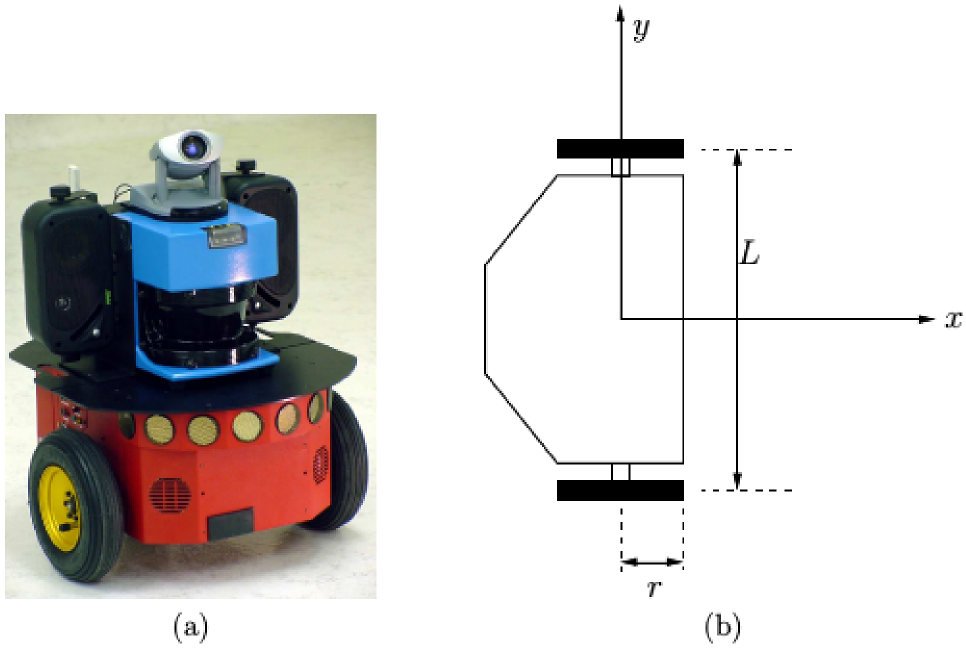


Figure 13.2: (a) The Pioneer 3-DX8 (courtesy of ActivMedia Robotics: MobileRobots.com), and many other mobile robots use a differential drive. In addition to the two drive wheels, a caster wheel (as on the bottom of an office chair) is placed in the rear center to prevent the robot from toppling over. (b) The parameters of a generic differential-drive robot.

Figure 1: The AlphaBot

imagine that if the robot moves towards $+x$ axis, and it turns towards $+y$ axis we define that angle θ positive, otherwise negative. θ is the same as f before and calculated by the same equation as long as the wooden wheel and the Robot's wheels are on the same cam. The rules the robot is moving are below:

$$setMotor(r,l) = \begin{cases} (-50, 50), \text{moves straight} \\ (50, +50), \text{moves backwards} \\ (50, 0), \text{the right wheel straight} \\ (-50, 0), \text{the right wheel backwards} \\ (0, 50), \text{the left wheel straight} \\ (0, -50), \text{the left wheel backwards} \\ (50, 50), \text{turn right (negative } \theta) \\ (-50, 0), \text{turn left (positive } \theta) \end{cases}$$

The problem is that w_r, w_l is not linear to the voltage you give to the Motors and different for the same votlage. After thorough (sic) experimentation we managed to find a linear relationship using RLS algorithm between w_r for each v_r given.

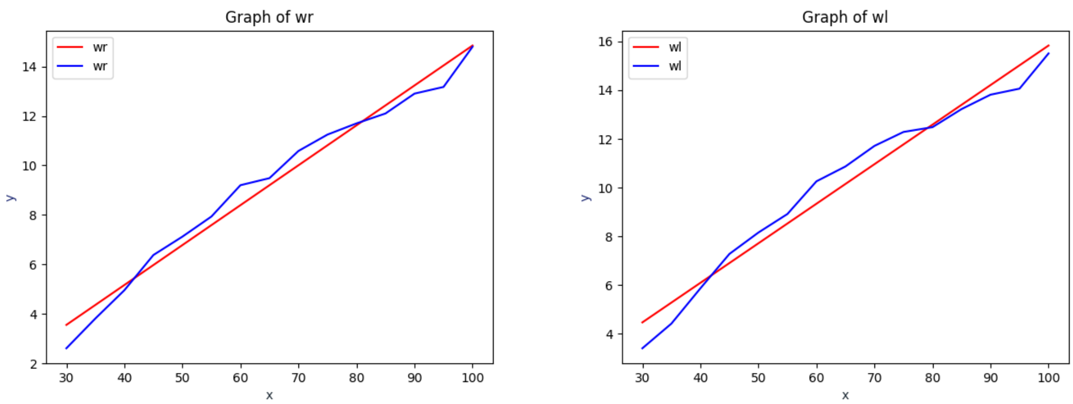
k	wr	wl
30	2.61	3.41
35	3.81	4.42
40	4.95	5.87
45	6.37	7.28
50	7.12	8.15
55	7.93	8.92
60	9.20	10.26
65	9.48	10.87
70	10.58	11.71
75	11.24	12.28
80	11.70	12.48
85	12.10	13.22
90	12.90	13.81
95	13.17	14.06
100	14.80	15.50

After we feeded RLS Algorithm this matrix. we got two linear equations:

$$w_r = 0.16136 * v + 1.2869, \text{where: } v \in [-100, -30] \cap [+30, 100]$$

$$w_l = 0.16225 * v - 0.3943, \text{where: } v \in [-100, -30] \cap [+30, 100]$$

So after that we can give different voltage to right and left Motor but these votlage will have the same angular velocity for the wheels.



(a) wr

(b) wl

Figure 2: RLS on measured w and v

2.2 Model

Based on "A differential drive" (13.1.2.2 pp. 726 Planning Algorithms, Lavalle) The model of x, y, θ is defined by the following equations:

$$\dot{x} = \frac{R}{2}(w_l + w_r)\cos\theta$$

$$\dot{y} = \frac{R}{2}(w_l + w_r)\sin\theta$$

$$\dot{\theta} = \frac{R}{L}(w_l + w_r).$$

Consider the continuous linear system:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) + D(t)u(t).$$

The A matrix of our discrete (but we the same equation as above) system is:

$$A = \begin{bmatrix} \frac{R\cos\theta}{2} & \frac{R\cos\theta}{2} \\ \frac{R\sin\theta}{2} & \frac{R\sin\theta}{2} \end{bmatrix}$$

when $\theta = 0$, meaining the robot is moving straight, then A is:

$$A = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} wq\, 0 & 0 \end{bmatrix}$$

So the system is neither controllable observable. As a result unfortunately you can't have a P controller and find K matrix values to control the output of the system.

However, the fact that AlphaBot does only quantized moves, means that you always move either rotating or forwards-backwards. As a result either x, y or θ are known and constant. So we can calculate the output:

- If the robot moves forward or backwards $\theta = 0$:

$$\dot{x} = \frac{R}{2}(w_l + w_r)$$

$$\dot{y} = \frac{R}{2}(w_l + w_r) * 0 = 0$$

$$\dot{\theta} = 0.$$

- If the robot rotates $\dot{x} = x, \dot{y} = y$

$$\dot{\theta} = \frac{R}{L}(w_l + w_r).$$

As a result if we need to move the robot:

1. Forward by X_{ref} (or Backwards is the same): Then we just need to compute w_r, w_l for setMotor. Consider the two angular velocities are equal $w_r + w_l = 2w$ (because of the same direction) at the beginning so:

$$w = \frac{\dot{x}_{ref} - x_0}{RT}$$

2. Rotate by θ_{ref}: Then we just need to compute w_r, w_l for setMotor. Consider the two angular velocities are equal $w_r + w_l = 2w$ (because of the same direction) at the beginning so:

$$w = \frac{\dot{\theta}_{ref} - \theta_0}{RT}$$

The only difficult part is to fix T at an appropriate value, large enough for the light sensors to calculate with minimum error, while not too large in order to control the Robot's movement accurately.
For forward movement T = 0.5 sec and T = 0.1 for correcting the orientation
For rotating T = 0.6 for first iteration and 0.15 for correcting the orientation error.
The next thing is to calculate voltage_right and voltage_left from the equations found from RLS, with the w_r, w_l given by the model. You know the rad/sec you want to move so find volt for each motor.

MicroController has a while loop running until the Robot is close to the reference position of this step given by Dynamic Programming. At each iteration calculates the needed w and implements the movement. During the T seconds the Robot is moving, we calculate from both light sensors the $\#pulses$ for each wheel. So with the aforementioned equation we can calculate the angular velocities for both wheels and also calculate the position that my measurements indicate.

$$w_r(t) = \frac{\#pulses_r * \pi}{10dt_r}$$

$$w_l(t) = \frac{\#pulses_l * \pi}{10dt_l}$$

And use them in:

$$\dot{x}_{meas} = \frac{R}{2}(w_l + w_r)\cos\theta_{meas}$$

$$\dot{y}_{meas} = \frac{R}{2}(w_l + w_r)\sin\theta_{meas}$$

$$\dot{\theta}_{meas} = \frac{R}{L}(w_l + w_r).$$

In addition, the measured values indicates where the Robot "thinks" it is. So taking into consideration these values it calculates the new w_r, w_l needed to reach the reference position. The first thing the robot tries to accomplish is to have the same orientation as the reference orientation. Afterwards it moves straight and fixes the orientation again.

IMPORTANT we use $\dot{\theta}_{meas}$ to calculate x, y. Meaning that the trajectory of our robots as a vertex with θ orientation gives the correct x, y.

Last but not least,
$R = 0.034$ (m) wheel radius.$6, 6$ cm is the diameter of the wheel.
$r = 0.0165$ (m) wood wheel radius. Actually never used.
$L = 0.132$ (m) distance between wheels.

3 Conclusion

We hope the AlphaBot will make it to the target.

Future work:

1. Take the data for the position from the beacon recognition and fix orientation maintaining Astar's solution
2. Solve Astar for every possible x, y coordinate AND θ orientation for the whole Grid
3. Moving the Grid with the robot's orientation
4. Nikos other crazy ideas