Autonomous Mobile Robots

Progress Report #2

Vacuum Cleaning Robot

Nuno Gomes
Joaquim Ferreira

Instructors:
Professor Keith L. Doty
Professor Scott Jantz

4 July 1995

Dept. of Electronics and Telecommunications
University of Aveiro

Index

Abstract

This progress report describes the early stages of the development of a vacuum cleaner robot prototype, both on hardware and software point of view. At this stage of the project the hardware installed is only the one needed for obstacle avoidance, the mechanical platform on which the hardware lies is a circle of wood with a diameter of 25 cm, with a hole in the middle where the vacuum cleaner fan is attached. This initial hardware is successfully running simple algorithms for obstacle and stairs avoidance.
The future capabilities of the robot are also discussed, namely: more elaborated algorithms for obstacle and stairs avoidance, collision detection mechanism and some other behaviors such as fan motor overload (equal to bag full of trash) and low batteries (search for charging station).

1 - Introduction

The research and development of an autonomous mobile robot prototype able to vacuum cleaning a room or even an entire house is not a trivial challenge. In order to tackle such a task, so that it could be completed in six weeks (the duration of the course), some simplifications and assumptions were made to the designers initial idea of an “ideal” autonomous vacuum cleaner. In this way, some functional requirements that would improve the robot performance were not taking into account due either to their inherent complexity or to their mechanical implications. Probably the decision that the most affects the robot complexity is the ability of mapping the environment so that it would exhibit a much better efficiency when compared with the minimalist approach as the one followed (random navigation).

With the aim of keeping our robot as simple as possible, while able to perform the initial goals, i.e. an autonomous vacuum cleaner robot able to randomly navigate through a room or a house with the minimum human assistance, the following specifications were found:

- Obstacle avoidance.
- Floor detection.
- Collision detection.
- Battery monitoring.
- Autonomous battery charging.
- Fan motor current monitoring.
- Autonomous dust bag dump.

These specifications correspond to some of the expected behaviors that will be programmed into the robot. Other behaviors that will increase the overall performance of the robot, such as self calibration of the sensors and navigation with some memory (not completely random) were also considered.

2 - Current Status of the Project

So far we have built the robot platform and assembled the driving wheels, the IR sensors (5) the fan and the wire wrapping of the basic circuitry. A continuity check of all connections was carried out to ensure the correctness of the wrapping.

The IR sensors are distributed along the front of the robot as shown in figure 1.

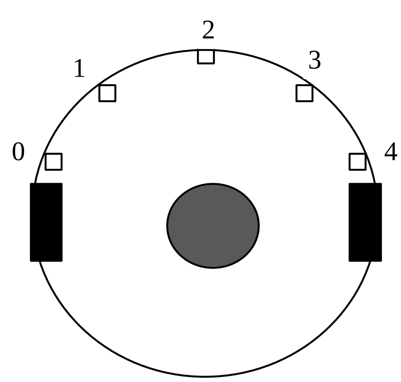

Figure 1 - Top view of the robot and IR sensor distribution.

In the previous figure, IR sensor 1 is facing front, IR sensor 1 and 3 are also facing front but they are rotated by 45 degrees. IR sensors 0 and 4 are aimed at the floor near the wheels, to prevent the robot to fall in stairs (or even a single wheel of it), independently of the angle on which the robot approximates the stairs.

Our robot has already two primitive behaviors: it avoids obstacles such as walls and well-contoured objects in the floor and it also avoids falling on stairs.

A continuous self calibration mechanism for the IR sensors is currently being studied, so that the robot should be able to adapt itself to the light intensity changes of the environment.

3 - Future Work

In the future we expect to improve the obstacle avoidance by including bump sensing all around the circular body of the robot. This capability is needed because the present sensors (IR LED's/receivers) are quite directive and, thus, they cannot detect sharp obstacles as chair legs, for example. The next step will be to include some mechanism to sense the fan motor current to detect if the dust entrance is blocked or the dust bag is full, in both cases the fan motor current will increase. The robot reaction to this input will be another behavior, on which the fan motor will be stooped and the robot will move for a while, trying to remove a possible blocking material. Then, the fan motor will be turn on again and if the its current keeps the same value as before the robot knows that the dust bag is full or the fan motor is being blocked. In this case the fan motor should be stooped and the robot should look for a maintenance station. If the fan motor current decreases to its normal value, it means that an object was blocking the fan motor and it dropped when the fan motor was stopped. In this situation, the robot should proceed its normal activity.

We would also like to have the robot sensing the battery charge. If the charge goes below a certain level, the fan motor should be stopped and the robot should look for a charging station.

We do not know yet if the detection of the fan motor current and the battery charge level will be done by specialized electronics or if we will just artificially make the robot detect that conditions by some binary inputs (switches).

Some extra circuitry will be needed so that the robot can heading to the maintenance station or to the charging station. For this electronics we are thinking in using 3 extra IR sensors, 2 for the heading and the other one to choose between both stations. This last IR sensor will work with polarized infrared light (we hope ...).

In fifteen days time we will have the air flow aperture just a few millimeters above the floor, the dust bag, the bump sensors and all additional circuitry working. In the mean time, we will continue to discuss some constructive aspects as experimental results became available.

Appendix A - Program code

```
void wait(int milisec)
{
 long timer_a;

 timer_a = mseconds()+ (long) milisec;

 while(timer_a > mseconds())
       {
         defer();
       }
}

    int     Left, Right,
                Center, FloorRight, FloorLeft, Floor0,Floor1,
                FloorMissing=0,
                ObjectRight,ObjectLeft, ObjectFront,NoObject=1,
                SUM_M=500,     /* sum of measures         */
                SUM_T=500,     /* sum of thresholds       */
                T0=95,
                T1=105,
                T2=95,
                T3=115,
                T4=105,
                ops=1;              /* in which direction have I turn last time ? */


void get_thresholds()
{
if ( !(FloorMissing) && (NoObject) )
    if ( (SUM_M > SUM_T+25) || (SUM_M < SUM_T-25) ) /* I don't see anything but everything
becomes brighter or darker */
    {
            T0=analog(0);
            T1=analog(1);
            T2=analog(2);
            T3=analog(3);
            T4=analog(4);
            SUM_T=T0+T1+T2+T3+T4; /* that's my new perception of environment brightness */
    }
}

void sensor_module()
{
 while(1)
       {
         poke(0x4000,0xf8);
```

```
          wait(150);

          Floor0=analog(0);
          Right=analog(1);
          Center=analog(2);
          Left=analog(3);
          Floor1=analog(4);
          SUM_M=Floor0+Floor1+Right+Left+Center;

          poke(0x4000,0x00);
          wait(100);

        ObjectRight=(Right > T1);
        ObjectLeft=(Left > T3);
        ObjectFront=(Center > T2);
        FloorRight=(Floor0 > T0);
        FloorLeft=(Floor1 > T4);

        FloorMissing=( !(FloorRight) || !(FloorLeft) );
        NoObject=( !(ObjectRight) && !(ObjectLeft) && !(ObjectFront) );

        }
}


void avoid_objects()
{
 while(1)
        {
        if ( (NoObject) && !(FloorMissing) ) /* go straight ahead */
              {
                  motor(0,50.0);
                  motor(1,50.0);
                }

          else if ( (ObjectFront) && !(FloorMissing) ) /* ops! maybe some corner */
              { if (ops) /* turn right*/
                  {
                    motor(1,100.0);
                    motor(0,-100.0);
                    wait(250);
                    ops=0; /* next time turn left */
                  }
                  else /* turn left*/
                  {
                    motor(0,100.0);
                    motor(1,-100.0);
                    wait(750);
                    ops=1; /* next time turn right */
                  }
                }
```

```
        else if ((ObjectRight) && !(ObjectLeft || ObjectFront) && !(FloorMissing))
                { /*turn left */
                 motor(0,100.0);
                 motor(1,0.0);
                }

        else if ((ObjectLeft) && (!(ObjectRight || ObjectFront)) && !(FloorMissing))
                { /*turn right */
                 motor(0,0.0);
                 motor(1,100.0);
                }
        else if ( !(FloorRight) && (FloorLeft) )
        { /*go back to right */
        motor(1,-70.0);
        motor(0,0.0);
        }
        else if ( (FloorRight) && !(FloorLeft) )
        { /*go back to left */
                 motor(0,-70.0);
                 motor(1,0.0);
        }
        else if ( !(FloorRight) && !(FloorLeft) )
        { /*go back to right and have a new heading */
                 motor(1,-100.0);
                 motor(0,-50.0);
        }
}
}


void main()
{
      start_process(get_thresholds());
      start_process(sensor_module());
      start_process(avoid_objects(),500);
}
```