Real-time obstacle avoidance navigation strategy in unknown environments

International Symposium on
Ambient Intelligence and Embedded Systems

Despina E. Athanasaki, Spyros Panagiotakis, Nikos Pinikas, Zacharias G. Sifakis, Andreas Vlisidis

22 - 24 September, 2016
Heraklion, Crete, Greece

Outlines

- Introduction (Title)
- Why autonomous?
- Material & energy cost
- The proposed materials
- The proposed algorithm
- Hardware Comparison
- Future Plans

Obstacle Avoidance Autonomous Mobile Robot (mobot)

Autonomous (adj) :

- (creature) having the power to make its own decisions and act independently
- (vehicle) is capable of sensing its environment and navigating without a need for human control.

Why autonomous?

That’s why autonomous

There are several places that can’t be accessed and observed by humans.

- Earthquake wreckage
- Tiny and long caves
- Radioactive areas
- Polluted tanks
- Outer space

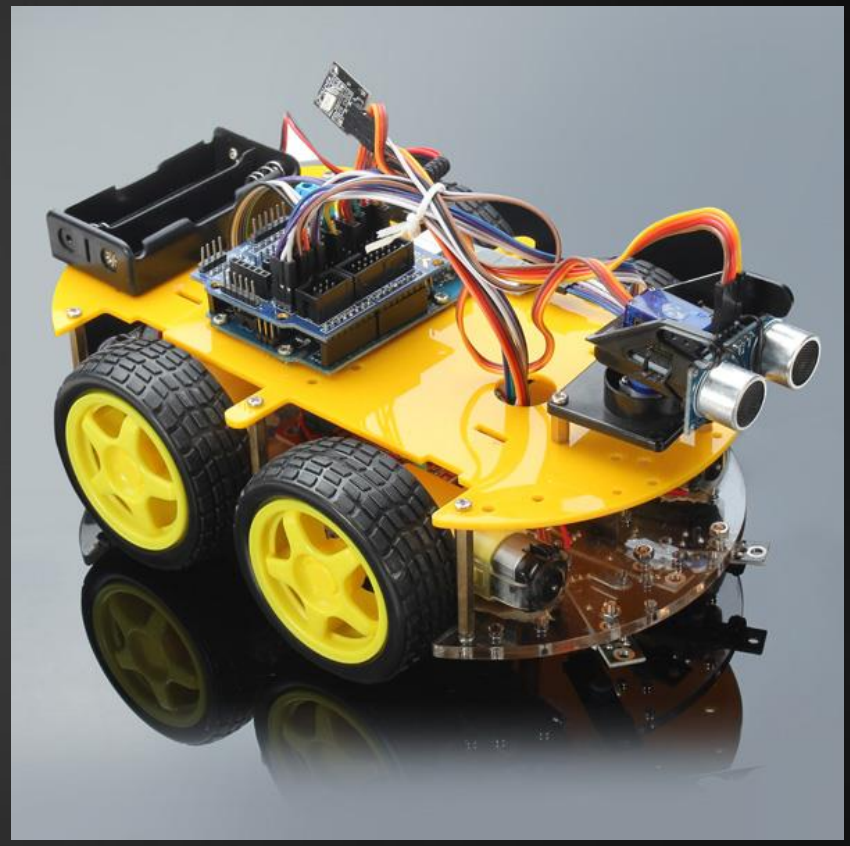
Material & Energy Cost

A major issue, for the implementation of an autonomous mobile robot are both material and energy cost.

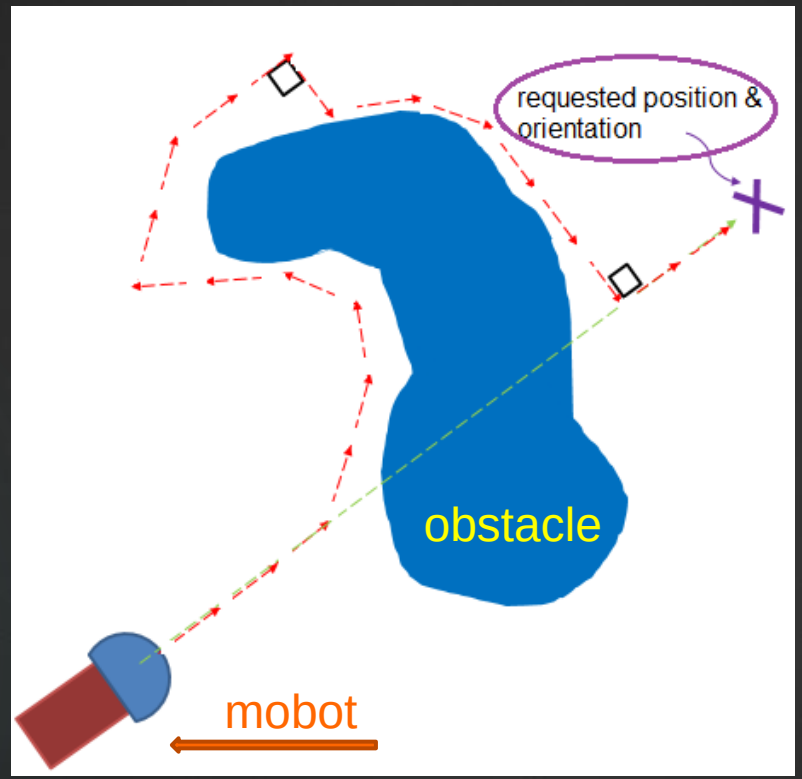
- Creation of an algorithm which needs as less processing power as possible
  - based on simple mathematical equations
- This algorithm will be used in platforms which are as less energy-consuming as possible.
  - Using fewest possible modules and sensors

The proposed materials for implementation (hardware)

- Two DC motors
  - for the wheel motion
- An H-Bridge circuit
  - drives the motors forward and reversed
- A sonar ultrasound sensor
  - with a measuring angle more than 45^o
  - for obstacle avoidance and shifting measurement
- A servo motor
  - (turns the sonar)



The proposed algorithm

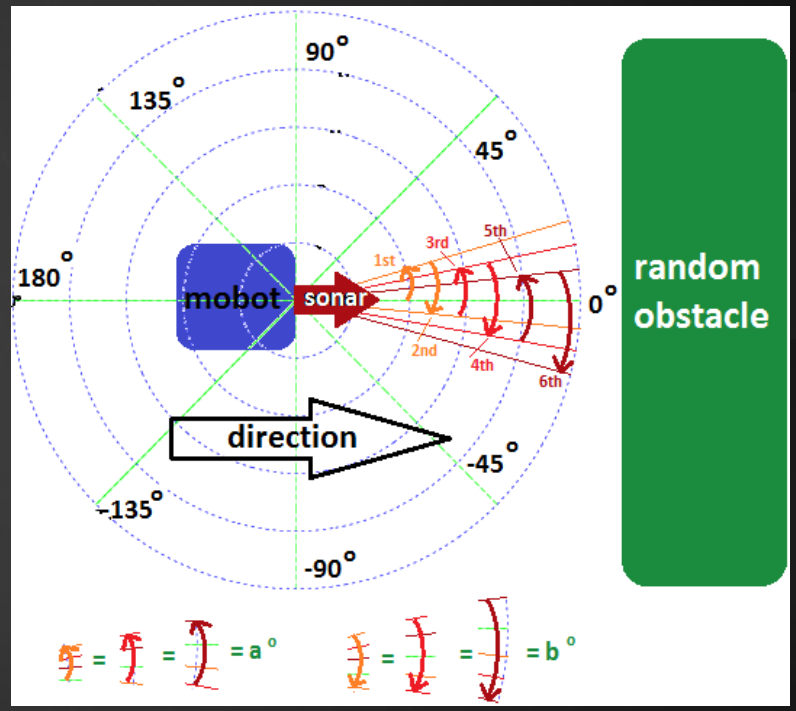


Estimated path

The proposed algorithm
#1 Part

- Originally the mobot is traveling in a predetermined direction.
- It is the first approach to the obstacle and it must decide if it will overtake it from the right or from the left.

    - The initial values of both **a** and **b** angles must be odd numbers, so that the angle calculation algorithm not to *get into an infinite loop. The even difference value of the angles can return the vehicle at 0 degrees.*

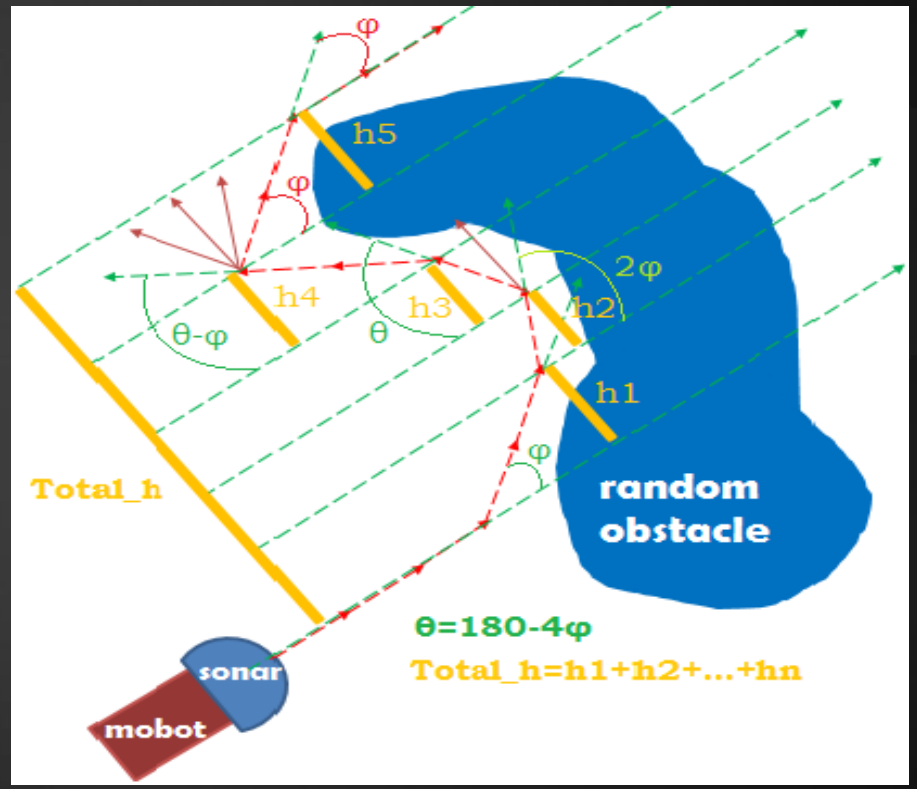


The proposed algorithm
#2 Part

- The mobot has already spotted the obstacle and has decided by which side will overtake it.
- This procedure is accomplished by checking each time the existence of an obstacle in the direction of its movement.
- As there is no obstacle, the function move (d) is called.

```
dist=dist+d;
    if (deg>90)
        h=sin(180-deg)*dist;
    else
        h=sin(deg)*dist;
```

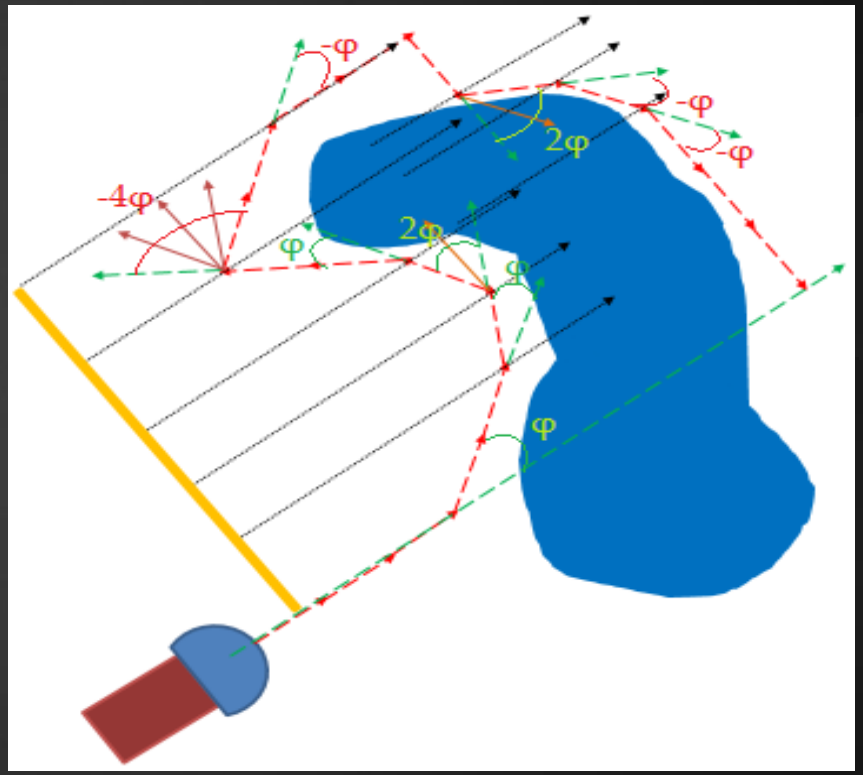
Total_h is the overall shift (height).



Path computation

The proposed algorithm
#3 Part

- This part of the algorithm allows the robot to check if there is free space to access by turning the sonar.
- The verification is done every specific degrees.
- Once the robot finds passage *(obst=0),* the algorithm calls the *turn ()* function, which turns the entire vehicle,
- The sonar returns to 0 degrees that is straight relative to the robot.
- Once this part of the algorithm is finished, the second part is re-executed.



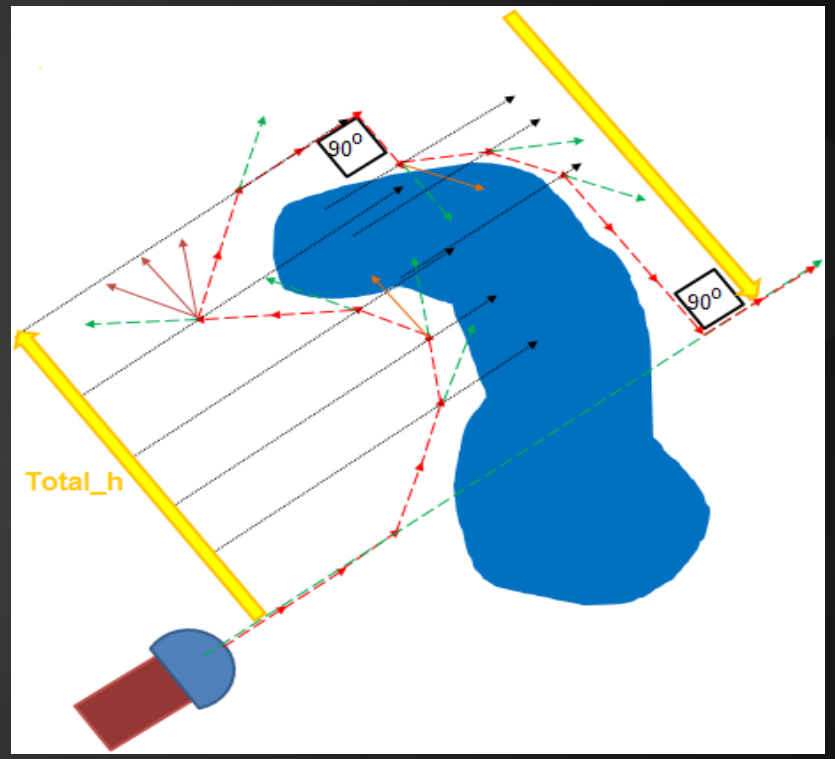
Angles computation

The proposed algorithm
#4 Part

- At regular intervals, this part of the algorithm, which checks whether the robot can turn in its original direction, will be called.
- The total degrees that the robot has turned are kept in *Total_deg* variable.

```
L_deg=Total_deg;
    do
    {
        L_deg=L_deg-sonardeg;
        obst2=sonar(-sonardeg);
        if (obst2=0)
        {
            obst=obst2;
            turn(-sonardeg);
```

- This will occur, until there are no remaining degrees, up to the initial orientation.

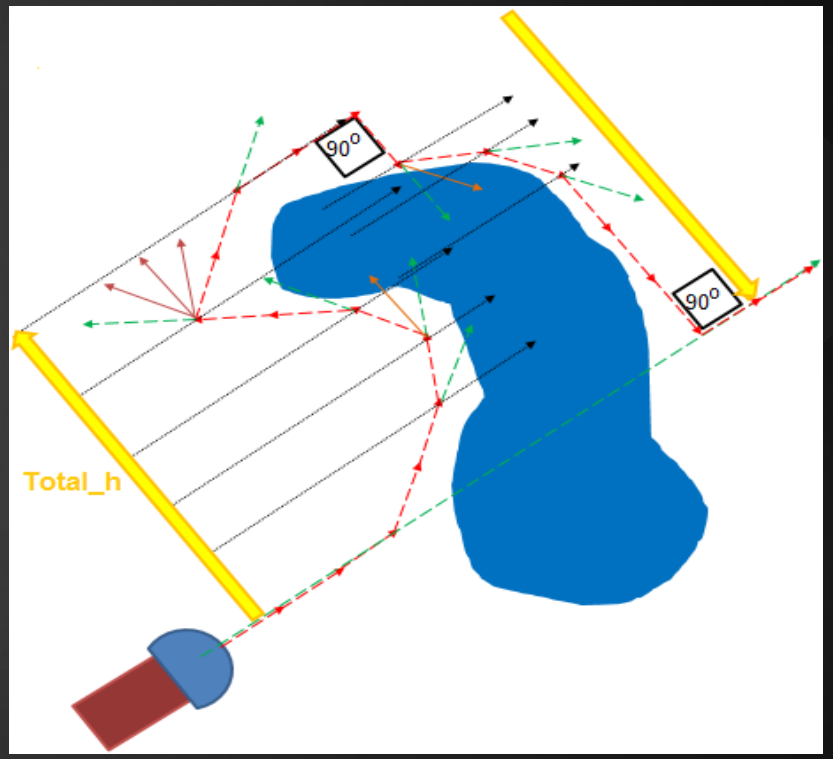


Return to initial direction

The proposed algorithm
#5 Part

- This part of the algorithm is used after the robot has been oriented (*L_deg=0*) and its distance from its original position is equal to *Total_h*.
- So it should come back as much as it was shifted, after turning 90 degrees
- A new variable (L_dist) is used which keeps the distance that is remaining to the final position.

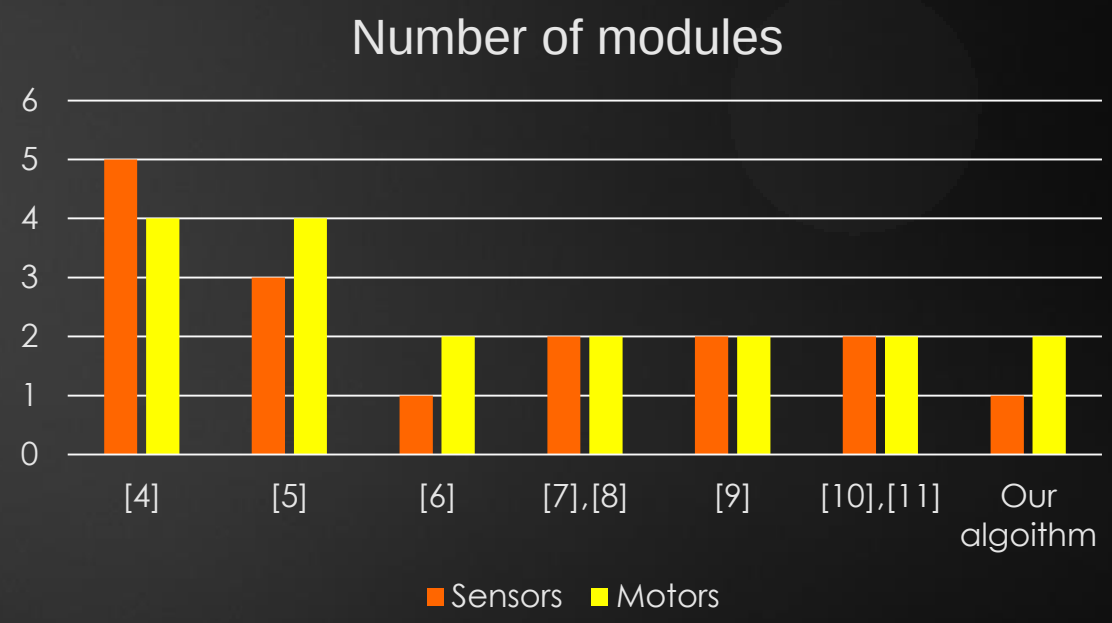
```
L_dist=Total_h;
if (L_deg=0)
{
        ……..
        if (obst=0)
{
                turn(90);
                move (d);

        L_dist=L_dist-d;
            ……..
        }
        turn (-90);
        move(d);
```



Return to initial direction

Hardware comparison between similar algorithms

Algorithm	Sensors	Motors	Blind zone
[4]	5 sonars	4	Very small
[5]	2 sonars & 1 IR	4	small
[6]	1 sonar	2	small
[7][8]	1 laser & 1IR or 1 sonar	2	large
[9]	2 sonars or 1 sonar & 1 IR	2 or 4	small
[10][11]	2 sonars or 1 sonar & 1 IR	2 or 4	small
Our algorithm	1 sonar	2 or 4	small



Future plans

The evolution of this work is:

- an algorithm which takes place in three dimensions and it will be used in flying robots (drones) for easier space exploration.
- This algorithm could be placed in service of the underwater archeology in which the robot will assist archaeologists during possibly dangerous and expensive shipwreck exploration missions.

References

- *[1] Carlos M. Costa, Héber M. Sobreira, Armando J. Sousa and Germano Veiga, "3 DoF/6 DoF Localization System for Low Computing Power Mobile Robot Platforms", in book: Cutting Edge Research in Technologies, Edition: 2015.*
- *[2] Kuo-Lan Su1, *, Yi-Lin Liao2, Shih-Ping Lin3 and Sian-Fu Lin3, "An Interactive Auto-recharging System for Mobile Robots", in International Journal of Automation and Smart Technology 4(1):43-53 · March 2014.*
- *A.J. Hildebrand, "Fundamental Mathematics Math 347", Department of Mathematics University of Illinois, 2015. Available from: http://www.math.uiuc.edu/*