International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 6
June 2017

Hardware Implementation of Autonomous Surface Vehicle using Arduino Mega

Anil Kumar D [1], Dr. Hariprasad S A [2], Prathiksh S P [3], Vishal H P [3], Payal P Joshi [3], Vidyashree S V [3]

1. Assistant professor, Electronics and Communication, BMSIT&M, Bangalore, India.
2. Director, School of Engineering and Technology, Jain University, Bangalore, India
3. Electronics and Communication, BMSIT&M, Bangalore, India

Abstract—our aim is to construct a working prototype of anautonomous and unmanned water surface vehicle (ASV) that is multi-modular in nature, i.e. makes use of multiple modules and sensors in order to sense and actuate the required functions so as to suit our desired method of travel, also keeping in mind a path-saving algorithm, which takes into account real-time GPS co-ordinate system to find out heading and course change angles. Surveillance is carried out through a camera module.

Keywords—Autonomous surface vehicle; modules; real-time; surveillance; course; algorithm.

I. INTRODUCTION

An autonomous surface vehicle can be defined as any water surface vessel that can function without a crew to direct it. The destination co-ordinates are fed into the vessel initially.

After that, the vessel's GPS module takes care of the entire path planning. An infrared or ultrasonic sensor can be used to evade obstacles.

The ultimate goal of mankind coming into the second decade of the 21st century has been complete automation. When it comes to automation, transportation has made many-a-leap. Land transport automation is fairly simpler when we compare it to water transport automation. Unlike land, a lot of factors have to be paid attention to, like water pressure, tidal forces, and winds and so on. The COLREGs are a set of rules and regulations laid down in order to be followed by all marine vehicles in transit, irrespective of their nationality and command. These rules must be kept in mind while designing any type of marine path algorithm.

II. PROBLEM STATEMENT

Build an efficient autonomous surface vehicle prototype and a homegrown travel algorithm with modular integration of an independent GPS system assisted with an ultrasonic sensor.

Test possible travel algorithms and choose the best one. Power the dynamics of the vehicle using propellers and test this setup in a shallow water body against desired requirements [2]. Equip the setup with a camera module as a measure for surveillance by taking pictures of obstacles.

III. GENERAL LAYOUT

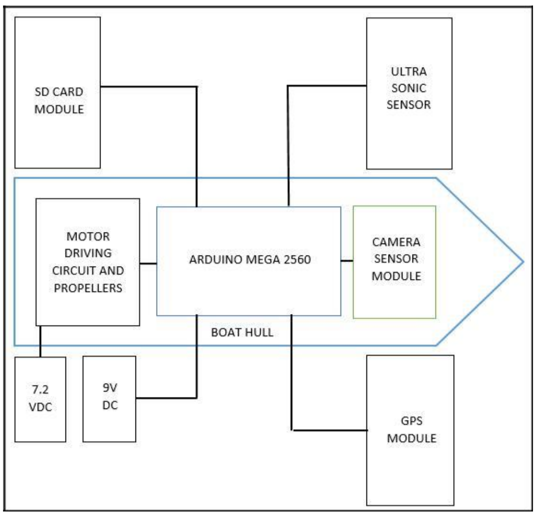


Fig1.

IV. HARDWARE SPECIFICATIONS

1. Arduino Mega 2560
2. 4-Pin Ultrasonic Module Hc-Sr04
3. IC L293D motor driver
4. DC Motors x 2

5. GPS Ublox 6M Module
6. RC Boat hull
7. OV7670 CMOS Camera
8. SD Card Module
9. 7.2v and 9V DC Power supplies

V. PROPOSED METHODOLOGY

The USV prototype will use an RC boat chassis as the main hull. Prototype construction will happen in three phases. Phase I will include dismantling an RC boat and embedding an external development board to check directions. Phase II will include autonomous navigation testing using various travel algorithms.

A. Phase I

In this phase, the RC components and internal modules of an RC boat are removed. We place an Arduino board assisted with a L293D motor driver to control both DC motor propeller motors to check for left and right boat movements.

Since there is no rudder and we don't depend on point-accurate angles, we use different delays in motor movements to change the direction of the ship heading by changing motor speeds.

B. Phase II

Now that the boat is ready with a custom driving system and functioning motors, the next step is to run it autonomously. We come up with 2 obstacle avoidance algorithms. We test both of these algorithms in a water body.

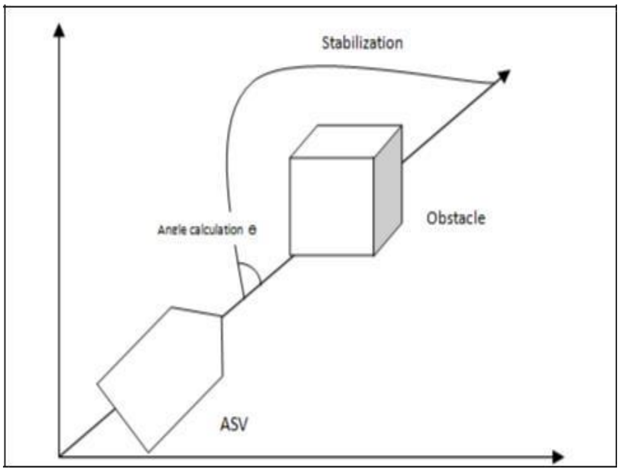


Fig 2. Obstacle Avoidance

First algorithm [1] works by stopping the boat every time an obstacle is encountered, turning left and right and calculating non-avoidable distance. The side with maximum distance is the chosen side to move the vessel. The problem with this algorithm is that stopping momentum is not accounted for, thus leading to inefficient direction change

Fig 3. Extra angle turn due to Stopping momentum

Second algorithm [3] gets over the defects of the first algorithm. Here every time an obstacle is encountered, the course angle heading to the destination is calculated. If the destination is 180<t<360, the boat moves left. If the destination heading is 0<t<180, it moves to the right of the obstacle.

Fig 4. Using stopping momentum to our advantage and avoiding obstacle

C. Phase III

In this phase, both GPS and obstacle avoidance is brought together using the algorithm 2 discussed above. An OV7670 CMOS camera is attached to the setup to act as a surveillance mechanism by periodically clicking photos of obstacles and the destination and saving the same onto an SD card for progressive diagnostic purposes.

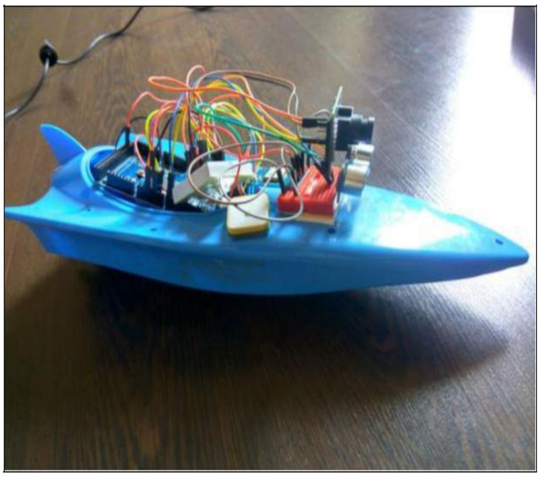
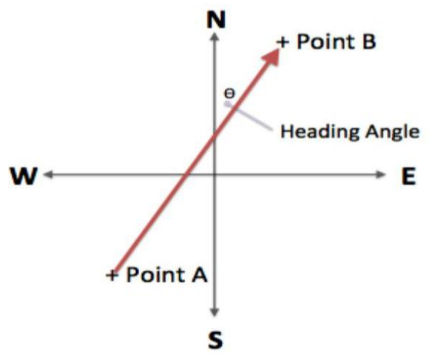


Fig 5. Complete ASV setup

With this final correction, we can compute a value that varies from 0 - 359 degrees in a full clockwise rotation, with North equal to 0 degree, East equal to 90 degrees, etc.



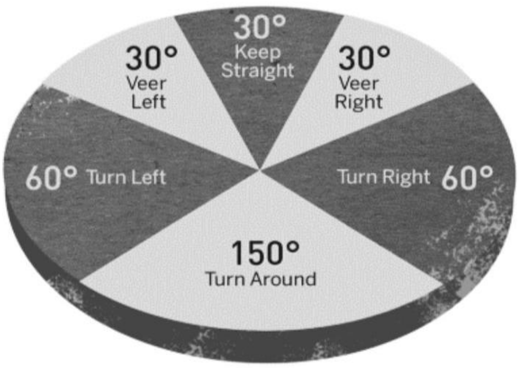


Fig 6. Direction change guide

For the first step, we compute values for DX and DY, like this:

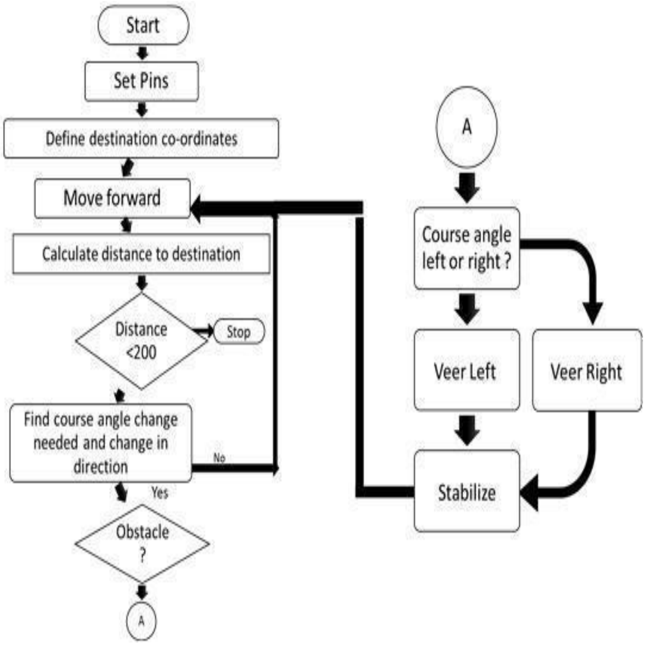
1. **DX = PointB.x - PointA.x**
2. **DY = PointB.y - PointA.y**
3. Where PointA.x and PointB.x are longitude values,
4. and PointA.y and PointB.y are latitude values.
5. **angle in radians = atan2(DY, DX) angle in degrees = radians * (180 / Pi)**
6. **angle = angle < 0? angle + 360: angle**
7. or:
8. **angle = fmod(angle + 360, 360)**

9. **Else**
10. Calculates the course angle required and change in direction
11. **If** (obstacle detected)
12. Click a picture and store in SD Card
13. Calculate difference between current course angle & required course angle.
14. **If** angle is between 180 degrees and 360 degrees
15. Turn to the left of the obstacle
16. **Else**
17. Turn to the right of the obstacle
18. **End If**
19. Stabilize the boat direction
20. **End If**
21. Go to 5
22. **End Program**

VI. FLOW CHART

Algorithm Flow

1. Begin
2. Set all pin numbers and initial starting variables
3. Set serial baud rate
4. Get the GPS latitude and longitude data and define the co-ordinates
5. Adjust the ship heading direction towards the destination/goal
6. Move forward
7. Calculate the distance to destination
8. **If** (distance is lesser than 200metres) then stop



VII. CONCLUSION

The ship was deployed in a short water body i.e. swimming pool for the testing purpose. It was tested for different combinations of speeds for twin propellers. The speed of propeller turned out to be the major issue in the beginning but eventually we came up with the solution for the above mentioned problem. Next we implemented the obstacle detection algorithm in the code and ship seemed to efficiently detect and avoid the obstacle. The ship took the alternative and shortest path available to move towards the GPS point by turning and going slightly off the track to avoid the obstacle and regain the original path after that.

The results verify that ship which when deployed in water follows the designed code by using a shortest path algorithm and loss redundant travel distance by effectively detecting and avoiding collisions. We can also conclude by saying that we have constructed the simplest and efficient autonomous setup for marine travel with minimal resources and pin point GPS coordination. Although some improvements can be made, for an early prototype this is quite an impressive result.

VIII. FUTURE SCOPE

In upcoming variants of a USV, we can introduce a PID controller in order to choose a designated path by giving experimental 'K' values. This is useful particularly when we need more precision. This can be first tested on MATLAB using simulations. More sensors can be installed for efficient obstacle avoidance. IR sensors can be helpful in measuring the lateral or side distance between the obstacle and ship and Sensors like wind sensors to measure the velocity of wind which will be helpful for navigation, tidal sensors to measure the tidal force and disturbances. we can include a wireless Ethernet module to transfer taken images to a user as a better method of surveillance. This is helpful to know what the current scenario on the monitor is. Real world disturbances can be avoided by mounting rain sensors. Solar panels can be used to provide for the power requirements of the boat.

IX. REFERENCES

[1] Tor Arne Johnsen, Tristan Perez, and Andrea Cristofaro,"Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection with Predictive Hazard Assessment", IEEE transactions on intelligent transportation systems, vol. 17, no. 12, December 2016.

[2] Hanlin Niu, Yu Lu, Al Savvaris, Antonios Tsourdos, "Efficient Path Planning Algorithms for Unmanned Surface Vehicle", *IFAC-PapersOnLine*, vol. 49, pp. 121, 2016, ISSN 24058963.

[3] R. Zhang, P. P. Tang, Y. M. Su, "An adaptive obstacle avoidance algorithm for unmanned surface vehicle in complicated marine environments", *IEEE/CAA Journal ofAutomatica Sinica*, vol. 1, no. 4, pp. 385-396, 2014.

[4] M. Caccia, G. Bruzzone, R. Bono, "A practical approach to modeling and identification of small autonomous surface craft", *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 133-145, April 2008.

[5] T Xu, R Sutton, S Sharma, "A multi-sensor data fusion navigation system for an unmanned surface vehicle"**,** Published December 2007. Doi: 10.1243/14750902JEME72