"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XIX – 2016 – Issue 2
The journal is indexed in: PROQUEST / DOAJ / Crossref / EBSCOhost / INDEX COPERNICUS / DRJI / OAJI / JOURNAL INDEX / I2OR / SCIENCE LIBRARY INDEX / Google Scholar / Academic Keys/ ROAD Open Access / Academic Resources / Scientific Indexing Services / SCIPIO / JIFACTOR

IMPLEMENTATION OF A MACHINE LEARNING ALGORITHM IN AN AUTONOMOUS SAILBOAT

Pedro Castro FERNANDES[1]
Mario Monteiro MARQUES[1]
Victor LOBO[1]

[1]CINAV – Escola Naval, pedro.castro.fernandes@marinha.pt

***Abstract**: The sea always pumped up human's curiosity. We have been exploring it since the beginning of time. It had always an important role in the society since we make use of it in several activities like collecting resources, dispatching merchandise or just for recreating activities. The number of ships crossing the oceans is incredible high, and there are a lot of illegal activities.*
A robotic sailing boat is a complex system. It has several parts that come together in a specific order to achieve the goal of sailing. The important sensors are the wind meter, the compass and the GPS sensor, but, this project, as also other sensors like a SD card reader sensor and a 433MHz trans-receiver. This sensors describes the environment. This paper presents an online machine learning agent developed to control a small scale sail autonomous sail boat. Implement this is a big challenge, as we are running our project into an Arduino mega, clocking at the speed of 16 Mhz. Design this agent is complex, because, sailing depends on a lot of variables, and we have restricted processing capacity. In this paper we describe the variables that we used to construct the different matrixand how they become a usable information to the boat successfully sail.

***Keywords**: Sailboat, autonomous, Machine Learning.*

Introduction

The ocean not only provides a way of communication and transportation, it also is used to collect important resources with major impact in economies. Nowadays, countries are interested in maintain a constant surveillance on the oceans in order to guaranty the sovereignty and the control of that resources. The idea of using autonomous sailing boats isn't new, since there are other people developing similar devices[1], however it is always good to have different approaches to the same problem, because each solution cuts both ways as anything in engineering, having strong as weak points, being better in some aspects, and worst in another's.

A boat to stay long periods of time in the sea as to be self-sustainable in energy, and being a sailboat is a good solution, although, being environmental dependent can be bad, as the wind can stop, and we lose the propulsion, however, most of the times we got enough wind to sail.

People see an autonomous device as a physical platform that receive inputs and perform a given task, in function of internal instructions implemented on the device[2]. Our goal is to save human labour, so, our device, as to be independent as possible. In general people see as a good thing the use of these devices to do this kind of jobs [3], although, at this moment, we are only using the sail boat in competitions. We expect that, in the future, it can be used in real missions of surveillance, patrol or search and rescue.

This paper focus on the development of an agent. This agent tries to learn the best way to sail the boat in which is installed. Our main goal was to develop an agent that can be used in different types of sailboats and take advantage on the particularities of the platform in which is performing.

This paper is divided as follows: in the section II is described the architectures used in the system, the sensors, actuators and processor unit used. On the section III, we talk about the Arduino shield and how it was programmed to be able to read all sensors data and communicate. In this section is also explained the shield developed in this project that groups all sensors signals and output them in the Arduino. In section IV we talk about the agent developed. The agent is an online Machine Leaning algorithm that performs hill climbing process to converge to the best sail configuration. In the section V, we talk about future work that is going to be implemented on the work develop. The section VI is the conclusion of the paper.

System Architecture

Sailing is a complex job. We got a lot of things to take care before we successfully sail. For a ship to sail, there are a lot of different jobs to be profited. In a really ship, we got different persons taking care of the different jobs, in order to everything run smoothly. In our boat, we have

DOI: 10.21279/1454-864X-16-I2-032

only one processor, so, metaphorical we got only one person, although, that person think fast, and can make a lot of jobs quickly. The processor unit we are using is an Arduino Mega 2560[4]. This is a microcontroller board based on the microprocessor ATmega2560. It has 54 digital input/output pins, 16 analog inputs and 4 universal asynchronous receiver/transmitter (UART), which can be used as hardware serial ports. We use three UART ports to sensors: Compass; GPS and wireless communications. The anemometer works with one Hall Effect sensor, connected to an analogic port. Different directions or speed of wind are characterized for different voltages outputs. Those voltages are read in the Arduino and converted to numerical values.

In our sailing boat, we are using the following list of parts, some shown on theFig. 1:

- GPS – Updates positions with a frequency of 1 Hz
- Compass – Besides de head, it also indicates the pitch and roll of the boat
- Rudder – powered by an servo
- Sail – powered by a servo
- Anemometer
- Communications–433MHz (100mW)
- Batteries– 12.1V and 3000 mAh
- Arduino Mega ATM 2560
- SD card reader

Fig. 1 - Top left Sensor: anemometer; top right sensor: SD Card Reader; botom left: Rudder Servo; botomcenter: GPS sensor; botom right: Compass sensor

We implemented this architecture on a lazer sailboat with one meter length, but this solution can be used in other type of boats.

We used two programming languages: C++ and C#. The C++ is the language that was used to program the Arduino that controls the boat. The C# was used in the development of a debugger. We call this application backstation

The boat has three modes of operation. The first one is totally user dependent. On the back station the user define the rudder and sail angle. This mode is used to debug and testing the boat, or to collect information, for instance the coordinates of a certain place. Other mode, the Patrol mode, is used to develop the knowledge of the agent, by testing sail angles and update the matrix. The last mode, the compete mode, is used to the boat follow waypoints using the knowledge stored in the matrix. The modes of function of the boat will be described in detail further in this article.

The backstation Software is a program wrote in C#. The boat can function completely independent from this, although, we have several vantages in having this program running with the boat. It was developed in order to help detect malfunctions, and to manual control the boat, sending commands directly to the rudder or to the sail.

This application displays all data read or generated by the boat. The information gets to the application by a COM port where we attach the communication hardware. The link between the boat and the back station is assured by the trans-receiver sensor, shown in Fig. 2.

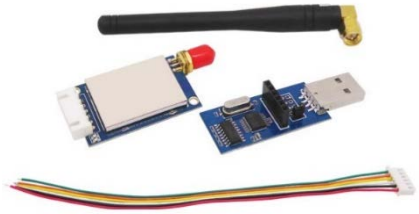

Fig. 2 - Comunication Sensor

This is a 100mW communications sensor, with a range of 1km in open space. If the boat goes out of the connection range it continues doing its job, as it is completely independent from the back station.

The information read by the *backstation* can be saved in log files and is as shown in Fig. 3.

Fig. 3 - Backstationmainpage

Development of the Arduino shield and C++ Code

The first step integrate all sensors in the Arduino. To do that, we develop a PCB board. It was draw using the freeware application *EAGLE schematics*. Using the data sheet of the sensors,

DOI: 10.21279/1454-864X-16-I2-032

we stablish which pins of the Arduino would be used with which sensor. The final board is as represented in figure Fig. 4.

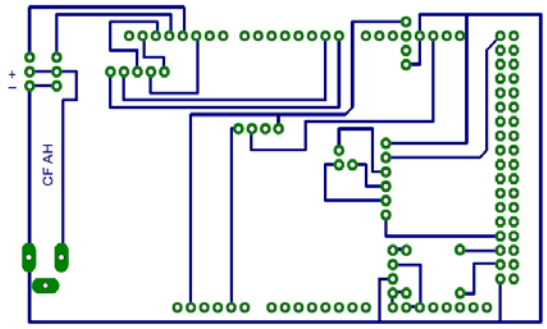

Fig. 4 - Shield used in the arduino

This PCB board, or Arduino Shield, clusters the signals of the different sensors.The next step was to develop the Arduino code, in order to the Arduino interpret the signal of which sensor. The sensors wore connected to the Arduino by UART serial ports. The Arduino Mega has four Serial ports, making the job easy. For the GPS, we used an open library that convert the GPS signal in variables: Angular latitude and longitude, age of the signal, speed in knots and course.The anemometer generates a different voltage in function of the wind direction. It can be measured with an analogic pin in the Arduino. The control of the servos was made using an Arduino open library.

The code was written in C++ using the Arduino Integrated Development Environment (IDE). We used different header files to different parts of the code, in order to simplify the comprehension of the code. The main function call the other ones when needed. Which sensor has it own header file. If we need to use a different sensor, is only necessary to change this part of the code. The main function as Boolean variables to set the different modes of function the boat. By default, the boat acts in compete mode. The waypoints can be stored in the SD Card, or written in hardcode directly in the microcontroller.

The Agent

Sailing dependson two major components: the ones that are related with the platform and the ones that related with the environment and the surroundings. We can study and deeply understand the way that our platform works. We know its characteristics, we know the code, we know the components and the way they are relate in function of the circuits we created. We can also know the surrounding, but, this is harder to define, the sea is a severe environment and it is constantly changing. It is hard to define it, and, it is even harder to define the relation that the boat will create with it. Most autonomous sail boats use reactive agents[5]. Those algorithms tell the boat how to sail, but, if we want the sailboat to be effective as possible, we have to put itself exploring that interacting and discovering the best sail configuration in each situation.

On-Line Algorithms and Machine Learning are areas that study the problematic of making decisions from limited information [6]. This method is characterized by the data used in the learning process become available in a sequential order, step by step. This area of machine learning has been extensively studied[6].

The ability to adapt to the surroundings as a very important role in autonomous vehicles. This could be particularly important in the situation that something goes wrong with the boat, like for instance, the sail gets damaged and the boat begins to behaviour differently. In this case, if the boat as the ability to test, it could adapt and still be able to get to its destination.

We implemented an online searching agent[5]. The first step of this kind of algorithm is looking for the environment stat, the sensors readings data. This data is saved as a vector of a space in R^n. We attached a SD Card Reader to the Arduino that controls the boat, this way, we can store there all data. The readings that we are saving are: difference between the head and thewind direction, speed of approaching to a waypoint, and sail angle. Those are the ones that most influence the sailing. We could use more variables, for instance: wind speed, pitch and roll, but, the space would become significantly larger. The processor unit wouldn't react fast enough and it would take much time to collect data to fill the space. One set of this readings is a vector, and all te vectors put together are the matrix that we store in the SD Card.

The next step is filling the matrix. The method that we are using to do this is the downhill simplex method[7]. This method iteratively approaches an optimal solution[8]. We set that, for the same conditions of difference of head and wind direction, the boat would try the sail full at starboard, full at portside, and half position. Then, it would choose the best two sail angles, the sail angles that provide more velocity of approaching, and try the sail middle position between and repeat this five times, narrow the sail angle to a range of about 5 degrees. The wind and the head are always changing, so, we set an interval of 10degrees. If the value goes off this range, that data is kept way, and it restarts the algorithm.

It will take much time to save all possible situations, and it all depends on the environmental

DOI: 10.21279/1454-864X-16-I2-032

conditions like the wind. We set that we would fill variable difference between head and wind direction in steps of 5 degrees. There are a range of differences that are impossible to sail. Normally, the sail boat can't sail upwind with less than 30 degrees. In this cases the speed is going to be null or negative, and the agent will steep to the next test frame.
The final step is to organize all data in a way that the sail boat can use the information to sail. It creates a new matrix, and stores only the sail angles that give the best speed, in function of the values of difference between head and wind speed.
The boat will have two function modes:

- Patrol – Is in this mode that the boat can run the algorithms ofmachine learning, and try new things and update the matrix of decisions. The diagram shown in Fig. 5represents the used agent.

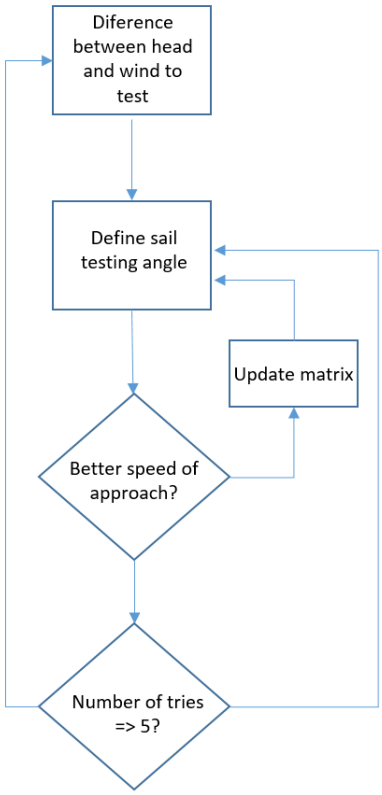

Fig. 5 - AgentFluxogram

- Compete – In this mode the boat don't try to improve the sailing knowledge, it uses the previous values from the matrix. In there, isn't all possible situation of values. When the agent can't find one equal, it uses the nearest neighbour, using Manhattan distance[9] between the situations saved on the matrix and the present situation and applies the one with the less distance.

Future Work

Nowadays we see a big growth in the number of autonomous systems. There are several systems, since another's autonomous boats, UAV's among others, and it will be a good idea to put all this systemsconnected together in somehow. There have been made an effort in order to achieve this goal with the development of Joint Architecture for Unmanned Systems (JAUS), for instance the Standard: AS5684 (SAE). It is a protocol to communicate between autonomous systems. It is important to develop and implement that kind of solution in this project.
It is also necessary to develop a management system to the electrics, at this moment the sail boat is only powered by the on board batteries. It is necessary to think about the implementation of renewable power sources, in order to, give to the boat a better performance in endurance tasks.

CONCLUSIONS

In this article we describe the agent we implemented in our sail boat. We start talking about the concept of autonomous sailing boats and the requirements of such a device. Next, we focused on the Machine Learning concept, and how it should be implement on an autonomous sail boat. Finally we exposed the architecture of our system and the software's developed.
Build an agent to autonomous sail boat is complex. It has to mine a lot of data to successfully sail. We were able to implement algorithms capable of handle such complex task in a microcontroller running with a processor clocking at 16 MHz. Thus, we develop a new approach to the problem of autonomous sailing. Our

DOI: 10.21279/1454-864X-16-I2-032

goal was to make it efficient as possible, making it possible to be used in other project or in real life operations.

BIBLIOGRAPHY

[1] Alves, José C. and Cruz,Nuno A. "FASt - An autonomous sailing platform for oceanographic missions" , Quebec, 2008.

[2] Grollman, D. H. Brown University, Providence, RI Dept. of Comput. Sci., and O. C. Jenkins, "Dogged Learning for Robots" , Roma, 2007.

[3] Ray, C.; Mondada, F. and Siegwart, R. "What do people expect from robots?", Nice, 2008.

[4] ARDUINO. (2016) Arduino MEGA 2560 & Genuino MEGA 2560. [Online]. https://www.arduino.cc/en/Main/ArduinoBoardMega2560

[5] Costa, Ernesto and Simões, Anabela.*Inteligência Artificia - Fundamentos e Aplicações*.: FCA, 2008.

[6] Blum, Avrim.*On-Line Algorithms - The State of the Art*. Pittsburgh: Fiat and Woeginger eds, 1998.

[7] Koshel,R. John. "Enhancement of the downhill simplex method of optimization," , Tucson, 2002.

[8] Gao, Lixing Han Fuchang."Implementing the Nelder-Mead simplex algorithm," vol. Volume 51, no. Issue 1, pp 259-277, January 2012.

[9] Madhulatha, T. Soni. "An Overview On Clustering Methods" vol. Vol. 2(4), no. pp: 719-725, 2012.

[10] Groenen, Patrick J.F.;Kaymaky, Uzay and Rosmalen, Joost van. "Fuzzy clustering with Minkowski distance functions" Rotterdam, EI 2006-24, July 6, 2006.

[11] Blum, Avrim and Mitchell,Tom. "Combining labeled and unlabeled data with co-training" , New York, 1998.

DOI: 10.21279/1454-864X-16-I2-032