ISBN: 978-628-95207-4-3, ISSN: 2414-6390, Digital Object Identifier: 10.18687/LACCEI2023.1.1.821

A route creator algorithm for Saildrones by comparing a straight-line pseudo route and wind direction

Diego Córdova, Br[1], Leonardo Vinces, Ing[2], José Oliden and, Ing[3]
[1,2]Peruvian University of Applied Sciences (UPC), Peru, u201714237@upc.edu.pe, leonardo.vinces@upc.pe
[3]Peruvian University of Applied Sciences (UPC), Peru, pceljoli@upc.edu.pe

Abstract*—Control algorithms for drones are commonly focused on transporting the robot from point A to point B, this type of straight path becomes sufficient in the case of aerial drones, but since it is a Saildrone (marine drone propelled in a partially or totally by a sail, commonly rigid), straight navigation between 2 points is less than optimal. That is why navigation strategies are used to control vehicles with sails, consisting of the variation of the angle of attack of the sail and the route that the vehicle will follow. This last parameter of the strategy is the one that is sought to be obtained using an algorithm based on obtaining the coordinates in real time of the drone and the destination point, together with the direction of the wind and the inclination of the Saildrone concerning magnetic North, uniting all this data through analytical geometry equations; to be implemented in an autonomous drone controlled by a Raspberry Pi 4 SBC. In this way, it is sought that the generated routes are similar to those commonly used in sailboat navigation, with the ability to update the route in case it varies from unexpectedly the initial parameters, whether they are the location of the boat or the direction of the wind, to obtain an energy saving greater than 50% to following a route in a straight line. The purpose of this project is to build a route generator with low processing costs for low-resource drones. In this way, it is sought that the generated routes are similar to those commonly used in sailboat navigation, with the ability to update the route in case the initial parameters change unexpectedly, whether they are the location of the vessel or the direction of the wind. , to obtain energy saving greater than 50% by following a route in a straight line. The purpose of this project is to build a route generator with low processing costs for low-resource drones. In this way, it is sought that the generated routes are similar to those commonly used in sailboat navigation, with the ability to update the route in case the initial parameters change unexpectedly, whether they are the location of the vessel or the direction of the wind. , to obtain energy saving greater than 50% concerning following a route in a straight line. The purpose of this project is to build a route generator with low processing costs for low-resource drones.

***Keywords*—sail drone, algorithm, navigation, route generator, autonomous, Raspberry**

I. Introduction

Drones can be used in different areas of work, whether for infrastructure monitoring and inspection, emergencies such as rescue or transport of medicines, defense, and security, fieldwork, or ecosystem study [1].

In this last area, the Saildrone is used a lot, specifically for the investigation of the marine environment. This is due to its low cost of production, maintenance, and high autonomy time at sea, with the potential to remain active for 365 days [2].

Evidence of this is one of the latest research trips carried out by the Saildrone® company together with the National Oceanic and Atmospheric Administration (NOAA), sending a Saildrone Explorer SD 1045 model drone to investigate how the increase in the intensity of hurricanes, this occurred on 10/01/2022 [3]. As previously demonstrated, the use of Saildrones is becoming essential for the advancement of oceanographic research, thanks in large part to its 24/7 autonomy.

This autonomy is achieved thanks to the use of renewable energies, whether solar panels or wind generators. The use of a sail also helps propulsion a lot, since it allows the transformation of wind energy into kinetic energy, which reduces the power used in engines or just replaces them. Because the power generated by the sail depends to a large extent on variables beyond human control, such as wind speed and direction, one way to optimize this energy is by varying the direction of the boat, following route strategies, which will vary the angle between the sail and the direction of the wind and in this way obtain the greatest possible advance energy. That is why a variety of algorithms capable of generating these routes have been developed.

An example of this can be found in the work of Sun, K et al. [4], in which an optimal route planner based on fuzzy neural networks is developed, collecting information through wireless sensors and using a route classifier for its optimization, achieving a performance increase of 12.5% and an increase in the efficiency of the route. 21.9% However, a connection to a large statistical database of the routes is necessary for their classification and optimization, preventing the planner from being located in the drone, which would limit its autonomous action in the event of a communication cut with the control inland.

Saoud, H et al [5], develop a global route planner through the use of a layered scheme, in the highest level layer there is a PRM-Dijkstra algorithm in charge of planning the route, through the use of meteorological data. , such as the direction of the wind, and the kinematics of the drone; For low-level control, there is a non-linear algorithm based on velocity vectors, in charge of controlling the course of the vessel. However, this division of controls leads to a similar problem to the previous one, the inability to have a controller capable of locating 100% on the boat and having to have wind vector information globally.

Dos Santos, D et al [6] developed a layered controller, the low-level layer consists of a PI controller, while the high-level layer seeks to generate a route based on intermediate points between the

Digital Object Identifier:(only for full papers, inserted by LACCEI).
ISSN, ISBN:(to be inserted by LACCEI).
DO NOT REMOVE

destination point location and the location ships. However, this method can become slow when changing the initial conditions, because the intermediate points would have to be deleted and generated again for each change in wind direction.

Silva Junior, A et al [7], developed a route generator based on Q-Learning, with the ability to avoid obstacles. However, to achieve such navigation it is necessary to load a map on which the vessel will navigate.

Dong, Y et al [8], developed a sail drone predictive and guided course controller, whose navigation consists of identifying 2 types of navigation, "downwind" and "upwind", taking as reference the polar graph of the speed of the sailboat, with a fixed angle of turn. However, although this logic is simple and fast to process, it can become too rigid, leading to problems with the accuracy of the destination point and not being able to take full advantage of the energy that the wind could provide in the cases of "failure". of the wind".

Analyzing the previous proposals, it was observed that the most complex route generators require a large database and/or a control center with constant communication with the sail dron; these requirements can mean a problem for the construction of low-cost drones. Therefore, the central objective of this work is to build a route generator capable of processing within the same controller of the sail drone, without needing a database or a high-end computer for low-cost sail drones.

The characteristics that this project must meet are: Being able to process from the same computer as the drone, 50% motor energy savings compared to following a straight line route, and the ability to quickly adapt to unforeseen events, such as the involuntary movement of the drone. drone due to swell

To meet this last requirement, the algorithm sends instructions to the drone, to generate a route depending on its current state, by comparing a straight route, between its location and the destination point, and a route that benefits from the increase in energy. of thrust provided by the rigid sail.

II. Methodology

Next, the steps to be developed for the completion of the algorithm will be shown, through a block diagram shown in Fig. 1.

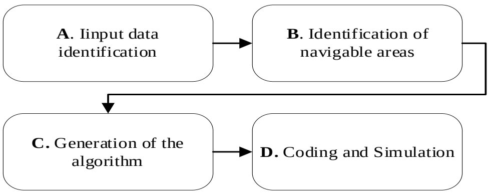

A. Iinput data identification
B. Identification of navigable areas
C. Generation of the algorithm
D. Coding and Simulation

Fig. 1 Block diagram of the development of the algorithm, development of own methodology

A. input data identification

Before developing the logic, it is necessary to take into account what data would be available, for the constant updating of the algorithm. To do this, an operating context is determined, and with it possible hardware to be used. Regarding the computation of the algorithm, a Raspberry Pi was selected, due to its computing capacity. As for the sensors, there are 2 types on the market, the IMU and INS. The IMU (Inertial Measurement Unit) is a sensor that has a gyroscope, accelerometer, and magnetometer, while the INS (Inertial Navigation System) has the same parts as an IMU, with the addition of a GPS and greater precision in measures. The selection of the sensor depends on the use and accuracy that is required for navigation, but as for the data obtained, those granted by an INS will be used, This is because it is important to know the location of the drone at all times. This does not mean that the use of an INS is essential to use this algorithm, since it can be replaced by a 9 DOF IMU by adding an external GPS, taking into account a loss of precision in the data. For the development of this work, the last option was selected, because the 9 DOF IMU has the sensors of an IMU, along with a built-in magnetometer; adding an external GPS, an easy-to-find, low-cost set whose precision (+/- 1 meter) is the minimum possible for correct navigation between 2 points. since it can be replaced by a 9 DOF IMU by adding an external GPS, taking into account a loss of precision in the data. For the development of this work, the last option was selected, because the 9 DOF IMU has the sensors of an IMU, along with a built-in magnetometer; adding an external GPS, an easy-to-find, low-cost set whose precision (+/- 1 meter) is the minimum possible for correct navigation between 2 points. since it can be replaced by a 9 DOF IMU by adding an external GPS, taking into account a loss of precision in the data. For the development of this work, the last option was selected, because the 9 DOF IMU has the sensors of an IMU, along with a built-in magnetometer; adding an external GPS, an easy-to-find, low-cost set whose precision (+/- 1 meter) is the minimum possible for correct navigation between 2 points.

After the establishment of the hardware context for the processing of the information and the location of the drone, the sensors that determine the state of the weather would be missing. Since it is sought that the algorithm can work with the least number of inputs, it was decided to focus on obtaining the parameters that directly affect the navigation of the boat, that variable is the direction of the wind, for this reason, a direction sensor would be added. of wind to the whole.

Because this algorithm generates an optimal route between 2 points, A - B, it takes into account how to provide the location of point B, this is achieved through "Ground Control", which provides useful information to the user, such as the drone location, speed, charge level, etc. As a minimum, this control must have the ability to provide the user with the location of the sailboat and locate the destination point, to send this information to the Raspberry Pi, either through telemetry, satellite communication, etc. This "ground control" was developed in Python language using the Google Earth API, (Fig. 2).

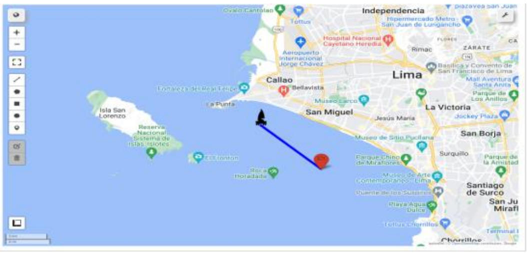


Fig. 2 Ground control map generated with Python and the Geemap library, with the location of the Start point (black figure of the catamaran), Destination (redpoint), and straight route (blue line).

Gathering all the sensors, we proceed to select those used for the route creation algorithm, these are: Current coordinates of the vessel (INS or GPS), Destination points coordinates (Provided by Ground Control), Orientation angle between the Magnetic north and the direction of the drone (Provided by INS or IMU – Gyroscope and

Magnetometer) and the angle of the wind direction to the direction of the ship (Provided by the wind direction sensor).

With this, the first fragment of the algorithm flowchart is obtained, shown in Fig. 3.

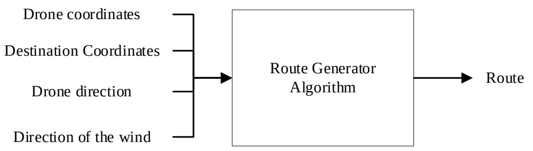


Fig. 3 Input variables of the route generator algorithm

B. *Identification of navigable areas*

To continue with the algorithm, first, the areas in which the sailboat could navigate taking advantage of the wind's energy must be identified. For this, an analysis of a NACA profile is carried out, at different angles of attack concerning the wind, from 0 to 180 degrees, to calculate the different coefficients that favor the advance of the drone.

For this, the work of Luque, C [9], which consists of the design and optimization of a rigid sail, is consulted.

In this thesis, first, a rewrite of the Bernoulli equation is carried out to calculate the lift and drag forces. This is to be able to express the coefficients in terms of "Advance coefficient" and "Rolling coefficient".

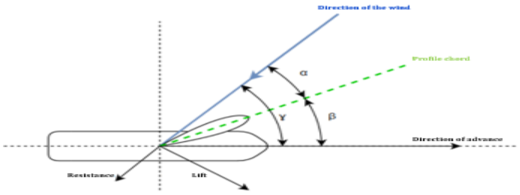


Fig. 4 FBD of the forces generated by a rigid sail [9]

Where α: Angle between the wind direction and the chord of the sail [º]; β: Angle between the chord of the sail and the direction of the boat[º]; γ: Angle between forward direction and wind direction [º].

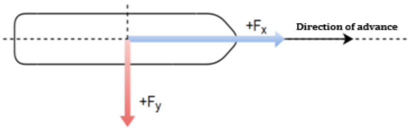


Fig. 5 FBD of the desired forces. [9]
Where Fx: Feed force [N] and Fy: Overturning force [N].

Analyzing Fig. 4 geometrically, the forces of Fig. 5 are obtained, whose equations are shown below.

$$Cx = Cl * sen(\gamma) - Cd * cos\ (\gamma) \qquad (1)$$

$$Cy = Cl * cos(\gamma) + Cd * sen\ (\gamma) \qquad (2)$$

Where: Cx = Advance coefficient; Cy = Rollover coefficient; Cl = Lift coefficient; Cd = Drag coefficient

To continue with the development of the equations, it would be necessary to calculate or identify the values of Cl and Cd, which are independent for each type of NACA profile. For this reason, a NACA 0015 profile [10] was used, and the report by Sheldahl, R. et al. [11] to extract the Cl and Cd data corresponding to the chosen profile and angles of attack between 0 – 180º

Solving the calculations and graphing them, a graph like the one shown in Fig. 6 is obtained.

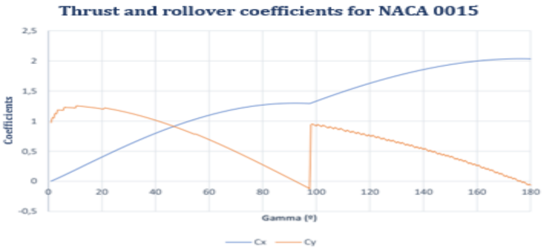


Fig. 6 Coefficients of advance and overturn at different angles Gamma (γ) [9]

It should be noted that the coefficients used for the development of the graph in Fig. 5 were obtained theoretically, these are similar to those found in the report by Sheldahl, R. et al. [10], which were obtained experimentally.

Analyzing Fig. 6, it can be seen that the "Advance Coefficient" is quite small in the first Gamma angles and it would start to be acceptable from 50o, when its value is equal to 1, therefore, that would be the area not navigable, between -50º and 50 º.

C. *Algorithm generation*

After obtaining the input parameters and the limits of the non-navigable zone, we continue with the creation of the algorithm.

The general idea consists in the identification and analysis of a future state where the sail drone will follow a straight path between its location and the destination point, analyzing that its angle between its direction and the direction of the wind is greater than 50 º, if so, the Vessel will continue in a straight line, otherwise, it will turn until the condition is met and move forward until a straight path is efficient.

To achieve this, you must calculate the angle between the direction of the wind and the direction of the boat following a straight route, this angle will be called "α".

To calculate it, we proceed to graph all the direction vectors that would make up the scenario; Direction of the ship, Direction of the wind, Direction of the straight path; and the different angles between the different vectors. Because it will work together with a GPS, the graph to be made will have as reference the magnetic North of the earth, this to resemble the calculations to an environment as real as possible. Something to clarify is the sign of each angle, in this work, all the angles that are to the left of the north will have a negative sign, and those that are on the right side will have a positive sign. This graph is shown below in Fig. 7.

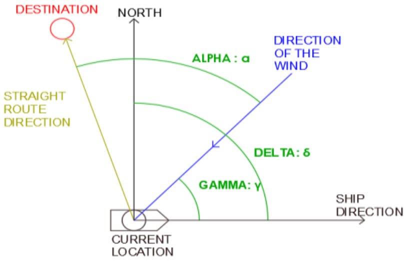


Fig. 7 Geometric graph of the input variables.

Where: γ = Angle between the direction of the ship and the wind [º]; δ = Angle between the ship's direction and north [º]; α = Angle between the wind direction and a straight path [º].

In Fig. 7 you can better visualize the entry angles (γ, δ) and the desired angle, "α", but as you can see, this angle cannot be calculated directly yet. For this you would need the angle between the direction of the ship and the direction of the straight path, this angle is called "φ".

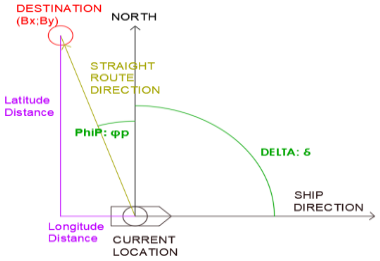


Fig. 8 Geometric graph of the variables necessary to calculate the angle "φ".

To calculate the angle "φ" the coordinates of the destination point and the current location of the vessel will be used, graphing it using the same criteria as in Fig. 7, this graph is shown in Fig. 8.

Where φ_P = Angle between straight path direction and north [º].

Making a geometric analysis of Fig. 8, the following equations can be extracted:
First, the direct way to calculate "φ" is through the sum of the angles"φ_P" and "δ".

$$\varphi = \varphi_P - \delta \tag{3}$$

From the above equation (3), the value of the angle "φ_P", corresponds to the angle between the straight path direction and magnetic north. To calculate it, the available data must be taken into account, as can be seen in Fig. 8, the angle "φ_P" corresponds to the angle opposite the distance of latitude between the destination point and the current location of the drone, this information helps us to relate the value of the angle with the locations of the destination point and the location of the vessel, obtaining the following equations.

$$\varphi_P = \tan^{-1}\frac{D.Longitud}{D.Latitud}e \tag{4}$$

$$D.Longitud = Bx - Ax \tag{5}$$

$$D.Latitud = By - Ay \tag{6}$$

Putting together the previous equations (3, 4, 5, and 6) we would arrive at the general equation for calculating the angle "φ"

$$\varphi = \tan^{-1}\frac{Bx-Ax}{By-Ay} - \delta \tag{7}$$

Where, φ = Angle between the direction of the ship and the direction of the straight path [º]; Bx = Latitude coordinates of the destination point [º]; Ax = Latitude coordinates of the ship's location[º]; By = Longitude coordinates of the destination point[º]; Ay = Longitude coordinates of the ship's location [º].

This equation is valid if the latitude distance is greater than 0, which means that the destination point is north of the start point (Drone location). Otherwise, the similarity between the angle "φ_P" and the angle opposite the longitudinal distance between the start and endpoints. To calculate the new "φ_P" for this particular case, a graph will be made for a better geometric analysis, shown in Fig. 9.

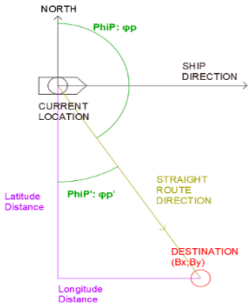


Fig. 9 Geometric graph of the special case for the calculation of φ_P.

Where $\varphi_P{}'$= Angle opposite the longitude distance between the current location and the destination point [º].

As you can see, you can calculate the new "φ_P" using the "$\varphi_P{}'$", leaving the following equation.

$$\varphi_P = 180 + \varphi_P{}' \tag{8}$$

180º is increased, instead of subtracted, because the coordinate system was mentioned from the beginning that it would be altered, being the sign dependent on magnetic North, in this case being "$\varphi_P{}'$" inverse of that used in a coordinate system where the x-axis assigns the sign.

After obtaining the angle "φ" and locating it in Fig. 7, the angle "α" can be calculated, leaving a representation as in Fig. 10 and resulting in (9).

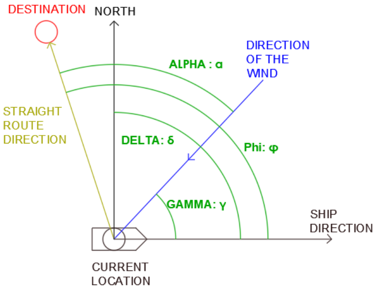


Fig. 10 Geometric graph of the input variables with the addition of the angle "φ".

$$\alpha = \varphi - \gamma \tag{9}$$

Now that we have the angle that determines the direction of the boat according to the direction of the wind and the location of the destination point, it would be necessary to calculate the angle of rotation that the boat would give, this angle will be "θ". To calculate it, the scenario is shown in Fig. 11, using the same criteria as in Fig. 8 and Fig. 7.

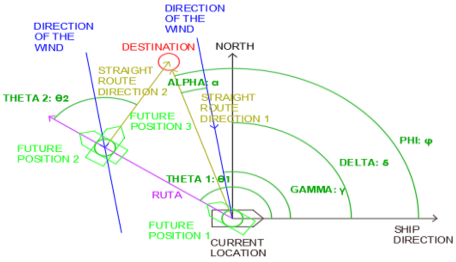


Fig. 11 Geometric graph of all the angles that make up the desired route

Where θ = Angle of twist [º].

As explained above, for the drone to move forward, the angle formed by the wind direction with the direction of the boat (γ) must be greater than 50º, taking this into account, the boat would be turned, such that the "γ" is equal to 0, in this way there is a reference margin to increase the minimum 50o for the advance of the vessel, with this the following equation (10) is obtained.

$$\theta = \gamma \pm 50^{o} \qquad (10)$$

As can be seen in the angle equation "θ", it can be increased or decreased by 50º, this determines the direction of the turn, being positive in case of wanting to turn to starboard or negative in case of turning to port. To determine the sign, the turn will be related to the direction of the wind.

This can be done in 2 ways, the first is to take the value of "γ" and determine if the wind is coming from port or starboard, the second option, and the one used in this work, is to determine it according to the sign of the value of "α”. As previously mentioned, the sign of all the variables in this work depends on whether they are to the left or right of magnetic north. In the case of "α", its sign depends on the direction of the wind, being left-negative and right - Positive, this allows, in a simpler way, to consider a relationship between the direction of the wind and the turn of the boat.

Since the variable that determined the sign of the turn was chosen, we proceeded to establish the criteria, for which 2 routes were "simulated" by graphing the cases, one whose $\theta = \gamma + 50^{o}$ if $\alpha < 0$ and $\theta = \gamma - 50^{o}$ if $\alpha > 0$; and the second case is the inverse, $\theta = \gamma - 50^{o}$ if $\alpha < 0$ and $\theta = \gamma + 50^{o}$ if $\alpha > 0$. These graphs are shown in Fig. 11 a and b.

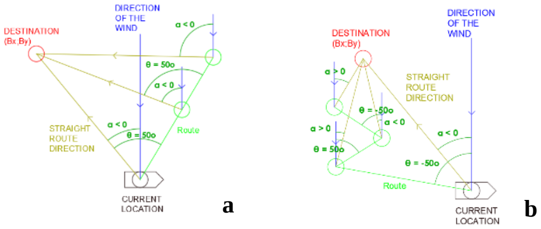


Fig. 12 Route graphs of two cases posed for the selection of the selection criteria of θ. a) $\theta = \gamma + 50^{o}$ if $\alpha < 0$ and $\theta = \gamma - 50^{o}$ if $\alpha > 0$. b) $\theta = \gamma - 50^{o}$ if $\alpha < 0$ and $\theta = \gamma + 50^{o}$ sí $\alpha > 0$.

As can be seen in Fig. 12, in the first case (12. a) a linear path of one turn is traced, in the case of the second graph (12. b) it goes in a straight line until the line marking the direction of the route is straight east to the right of the wind direction, in this case, the boat goes against the wind in a "zig-zag" fashion. Having seen the behavior of the drone in both cases, we proceed to select one, in this case, the criterion corresponding to Fig. 12.a, this is because it is the most stable route, in the case of Fig. 12.b, at the moment of the "zig-zag", the route is quite rough, with multiple sharp turns. Later on, some routes will also be simulated with the criteria proposed in Fig. 12.b to confirm this selection.

D. *Coding and simulation*

For the development of the simulation coding, MATLAB software was used, and some equations that simulate the operation of some sensors had to be added.

One of the sensors to be simulated is the wind direction sensor, since the angle "γ", a variable delivered by the sensor to be simulated, is the angle between the direction of the wind and the direction of the ship, some relationship must be found. between a static variable and the direction of the ship. For this, one more variable is added, which corresponds to the angle between the direction of the wind and the magnetic north. It should be noted that this would simulate an environment where the direction of the wind does not vary during the route. One way to exemplify this relationship is through a graph showing the different variables to participate. This graph is shown in Fig. 13.

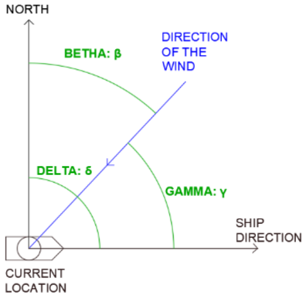


Fig. 13 Geometric graph for the analysis of the relationship between γ and β.

Where β = Angle between the wind direction and north [o]

As can be seen in Fig. 13, there is a relationship between “β” and “δ”, at first glance the value of “γ” would be "$\delta - \beta$", but because the reference frame is altered, as mentioned before, the final equation is:

$$\gamma = \beta - \delta \qquad (11)$$

The variation of the gyroscope was also simulated, this was achieved by the relationship between “δ” and “θ” due to what “δ” corresponds to the angle between the direction of the boat and magnetic north, this angle would only vary depending on the turn of the boat, which is represented by the variable “θ”, this relationship is best appreciated in equation 12.

$$\delta F = \theta + \delta \qquad (12)$$

Where: δF= New angle between the ship's direction and magnetic North after making a turn equal to θ degrees [º].

A peculiarity that was implemented in the simulation is the impression of the nose of the boat, to better show how the turns would be made. This is important as it would determine how the drone advances in the simulation.

To calculate the point where the nose of the boat is located, a translation of the current location is made to a certain distance in the direction according to the angle “δ", this relationship is shown in equations 13 and 14.

$$CX = Ax + L * sin(\delta F) \qquad (13)$$
$$CY = Ay + L * cos(\delta F) \qquad (14)$$

Where CX and CY = Coordinates of the nose of the boat [º]; L = Length of the vessel [m].

Now what would be missing would be to update the location data of the drone, and simulate the operation of the GPS. The logic behind this simulation is similar to that proposed for the location of points Cx and Cy, with the difference that, in this simulation, a signal refresh was proposed every second. The equations that represent this sensor are 15 and 16.

$$Avancex = Ax + V * sin(\delta F) \qquad \text{(fifteen)}$$
$$Avancey = Ay + V * cos(\delta F) \qquad (16)$$

Where: “Avancex” and “Avancey” = Coordinates of the new location of the drone after 1 second [o]; V = Vessel speed [m/s].

With these equations, we have what is necessary for the simulation test of the algorithm, shown in Fig. 14.

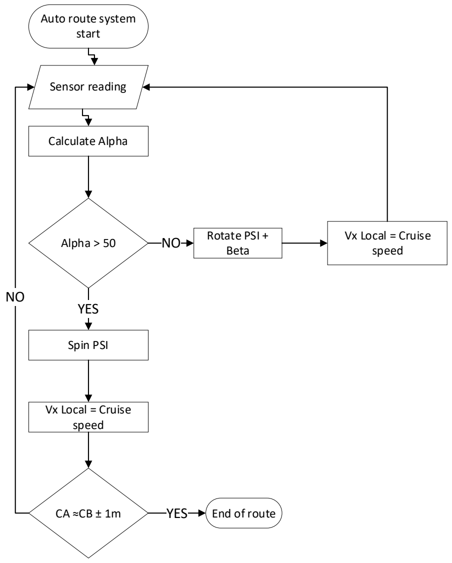


Fig. 14 Flowchart of the proposed algorithm

To check the energy savings, "Forward Energy" calculations were made using a sail drone with a rigid sail and motors, expressed in terms of energy. The dimensions of the rigid glider are enough to propel the drone having a maximum "Cx" for a NACA 0015 profile (Cx = 2), this case is shown in equations 13 - 16.

$$\mathrm{Ea} = \mathrm{Ev} \quad (17)$$

$$Vd * t * Fa = Vd * t * \frac{1}{2} * \rho * S * V^2 * Cx \quad (18)$$

$$Vd * t * Fa = Vd * t * \frac{1}{2} * \rho * S * V^2 * 2 \quad (19)$$

$$Ea = Vd * t * \rho * S * V^2 \quad (20)$$

Where Ea = Energy to advance [J]; Ev = Energy provided by the sail [J]; Vd= Drone speed [m/s]; t = Time interval; Fa = Drag force [N];ρ= Density of the fluid [kg/m^3]; S = Contact surface [m^2]; V = Wind speed [m/s].

It will also be necessary to establish a relationship between the energy necessary to advance and the energy provided by the motors, to calculate the efficiency of the proposed algorithm against a linear route generator, for which general equations are developed for each value of "Cx" (Fig. 6), represented in equations 17 - 20.

$$Ea = Em + Ev \quad (21)$$

$$Ea = Em + Vd * t * \frac{1}{2} * \rho * S * V^2 * Cx \quad (22)$$

$$Ea = Em + Ea * \frac{1}{2} * Cx \quad (23)$$

$$Em = Ea * (1 - \frac{1}{2} * Cx) \quad (24)$$

Where Em = Energy provided by the motor [J]

III. Results

After coding the algorithm in Matlab, 13 simulations were carried out varying the angles "β" and "δ", these angles are typical of the simulation, which describe the sea environment about magnetic North, being "β" the angle between the direction of the wind and the North; and "δ", the angle between the starting ship heading and North.

In this work, the most significant routes of the 13 carried out are shown, since they show all the possible route possibilities with this algorithm.

To better visualize the navigation of the sail drone, 3 routes are shown each with a different color, the light blue line being the direction of the wind; the blue routes are those followed by the center of the boat and the green points correspond to the tip of the boat.

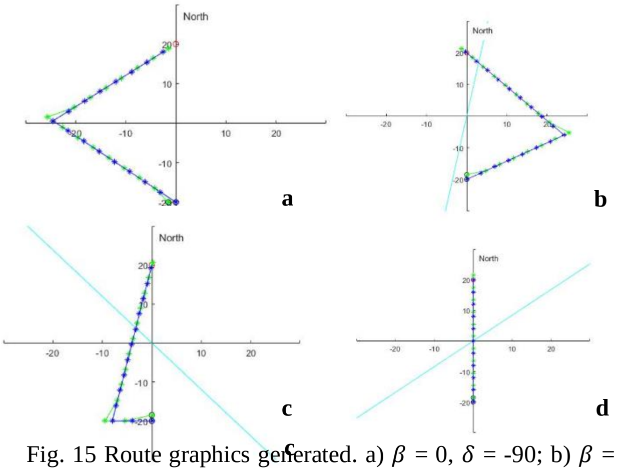


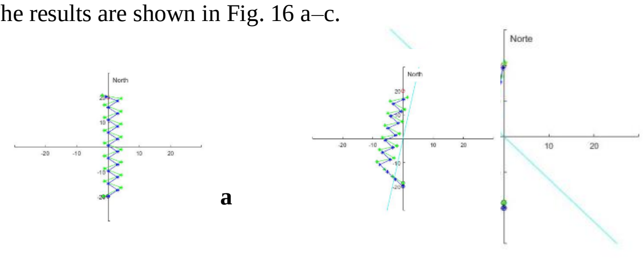
Fig. 15 Route graphics generated. a) β = 0, δ = -90; b) β = 10, δ = 0; c) β = -40, δ = 0; d) β = 50, δ = 0

Once the possible routes have been obtained, we proceed to calculate the percentage of energy that the engine has to provide for the advance of the boat and with it calculate the efficiency of the route. To do this, the percentage of energy delivered by the motor is compared in 2 situations, the first following a straight route and the second following a route similar to the one simulated in Fig. 15,b, both environments have a "β" = 10 and the direction of the boat points to the North.

To calculate the percentage of energy delivered by the motor concerning the total energy needed for the sail drone to move forward, first, you must find the value of "Cx" that corresponds to each case; In the first scenario, since the boat is going to the North and the "β" = 10, an angle of 10° is formed between the direction of the boat and the direction of the wind ("γ"), searching in Fig. 6 a value of "Cx" = 0.25, and corresponding to the second scenario, "γ" = 50 and "Cx" = 1. After calculating the value of "Cx" it is replaced in (24), thereby obtaining the "Em", in the first case "Em" = 87.5% "Ea", and in the second case "Em" = 50% "Ea".

When comparing the values of "Em" in both cases, it can be concluded that in the second scenario, the motor needs to deliver a lower amount of energy than the first scenario, in numerical values the second scenario is 57.14% more efficient (%Em2 / %Em1)

A second point to deal with is the variation of the parameters related to the moment to turn the sail drone, for this the same characteristics of Fig. 15 are simulated but with the turning criteria shown in Fig. 12,c. The results are shown in Fig. 16 a–c.



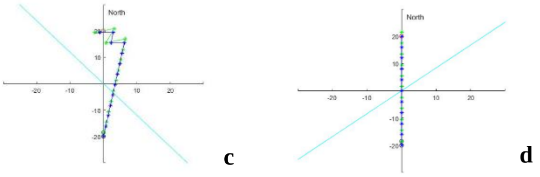


Fig. 16 Path graphs generated using criterion No 2. a) β = 0, δ = -90; b) β = 10, δ = 0; c) β = -40, δ = 0; d) β = 50, δ = 0

As can be seen, the generated route is more erratic when turning and presents a greater number of direction changes.

These changes of direction depend on the speed of the boat ("V") that is programmed in the algorithm, which means that the lower this variable is, the greater the number of turns made during the route.

Making a comparison between both routes shown (Fig.15 ac and Fig. 16 ac) different characteristics contrary to each other can be observed, which can be beneficial or detrimental depending on the terrain to be navigated and the user's objectives.

To better analyze these characteristics, their properties will be classified into 3 categories: travel speed, travel safety, and route width; To qualify these attributes, numbers from 1 to 3 were used, where 1 means that the route is slow, dangerous, or short and 3 means that the route is fast, safe and covers a large space. The results are shown in the following table, being "ROUTE A" the one shown in Fig.15, and "ROUTE B" the one shown in Fig. 16.

Board1
Comparative table between Routes A and B

Routes Analyzed	Travel Speed	Travel Safety	Route Width
Route A	2	3	3
Route B	3	1	1

As can be seen in Table 1, Route A presents a longer travel time, this is because, from the beginning, it is quite far from a straight route, although because it presents few sharp turns it is quite safe, compared to Route B is faster because the route is more like a straight line, but because it has a lot of curves it is quite dangerous. As mentioned above, Route A was selected as the final one for this algorithm, mostly because of the security that the route presents and that it is still more efficient than a straight route. Although using Route B for narrower bodies of water is also not relieved, changing the vessel speed parameter ("V") to a value that is higher than actual, to reduce the number of turns.

IV. CONCLUSIONS

An algorithm has been developed by analyzing the geometry of the location and direction of a drone and a destination point, provided by a 9 DOF IMU sensor, GPS, and a wind direction sensor registered by a Raspberry Pi, to generate an optimal navigation route for sail drone type drones.

b

From this analysis, 2 possible routes to be generated were deduced, a wide one consisting of 1 turn and a narrower one consisting of multiple changes of direction, since both routes present similarities with the standard "Zig Zac" type route and They have an efficiency greater than 50% concerning a straight route, an analysis of characteristics was carried out, the selection of the route that consists of 1 turn is chosen, this due to its safety. The selected route presents an energy saving of 57%, which confirms that the route allows better use of wind energy with the ability to control a sail drone by varying the angle of rotation and speed.

After showing all the results, it is concluded that the objective of the project was met, this is because the shape of the route, energy efficiency, and low level of processing required for its operation are met, and with its power, It can be used in low-cost sail drones and without the need to constantly send data to the drone from a command center.

Regarding the future actions of the project, 2 actions are planned, the first is to modify the equations and criteria for turning selection to generate routes similar to the second route derived to be used by vessels in bodies of water where there is not enough space for routes. wide, but modifying the speed variable of the boat to one greater than that measured by the 9 DOF IMU sensor; The second action refers to the geometric equations used, instead of using Euclidean geometry, use non-Euclidean geometry, because the shape of the earth is similar to a sphere, the geometric shapes that can be generated on the surface have characteristics different from traditional geometric shapes, an example of this can be seen with the triangle; In classical geometry, the sum of the internal angles is equal to 180o, while in non-Euclidean geometry that sum is greater than 180o. The reason for making this change is for better control for very long-distance routes (continental) and to be able to resemble the simulations in an environment that is much more faithful to reality.

V. ACKNOWLEDGMENT

To the Research Directorate of the Peruvian University of Applied Sciences for the nuptial support for the realization of this research work UPC-EXPOST-2022-2

REFERENCES

[1] Mitka, Eleftheria & Mouroutsos, Spyros. (2017). Classification of Drones. American Journal of Engineering Research. 6. 36-41.

[2] Wind & solar powered autonomous vehicles – saildrone. (s/f). saildrone.com. Retrieved on October 29, 2022, from https://www.saildrone.com/technology/vehicles

[3] Inside the eye of the storm: Saildrone collects live video of category 4 hurricane. (s/f). saildrone.com. Retrieved on October 29, 2022, from https://www.saildrone.com/news/saildrone-collects-live-video-inside-hurricane-sam

[4] Sun, K., Wu, J., Sun, Z., & Cao, Z. (2019). Optimal path planning method of marine sailboat based on fuzzy neural network. Journal of coastal research, 93(sp1), 911. https://doi.org/10.2112/si93-131.1

[5] Saud, H., Hua, M.-D., Plumet, F., & Ben Amar, F. (2015). Routing and course control of an autonomous sailboat. 2015 European Conference on Mobile Robots (ECMR).

[6] dos Santos, DH, & Goncalves, LMG (2019). A gain-scheduling control strategy and short-term path optimization with genetic algorithm for autonomous navigation of a sailboat robot. International Journal of Advanced Robotic Systems, 16(1), 172988141882183. https://doi.org/10.1177/1729881418821830

[7] Silva Junior, AG da, Santos, DHD, Negreiros, APF de, Silva, MVP de S., & Gonçalves, LMG (2020). High-Level path planning for an autonomous sailboat robot using Q-learning. Sensors (Basel, Switzerland), 20(6), 1550. https://doi.org/10.3390/s20061550

[8] Dong, Y., Wu, N., Qi, J., Chen, X., & Hua, C. (2021). Predictive course control and guidance of autonomous unmanned sailboat based on efficient sampled Gaussian process. Journal of Marine Science and Engineering, 9(12), 1420. https://doi.org/10.3390/jmse9121420

[9] Luque, CR (2020). Design and optimization of a rigid sail for boat propulsion. University of Almeria.

[10] NACA 0015 (naca0015-il). (s/f). airfoiltools.com. Retrieved on October 31, 2022, from http://airfoiltools.com/airfoil/details?airfoil=naca0015-il

[11] Sheldahl, RE, & Klimas, PC (1981). Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines.

[12] Julio Artica, Marco Klepatzky, Leonardo Vinces, Christian del Carpio, "Development of a positioning system using hybrid control to trace a fixed trajectory applied to a tracked mobile robot," 17th LACCEI International Multi-Conference for Engineering, Education, and Technology: " Industry, Innovation, And Infrastructure for Sustainable Cities and Communities" (LACCEI), 2019, doi: 10.18687/laccei2019.1.1.357.

[13] Milagros Loayza, Juan Alfaro, Leonardo Vinces, and Christian del Carpio, " A Novel Mechanical Design for a mobile robot that carries a load of 30 Kg," 17th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Industry, Innovation, And Infrastructure for Sustainable Cities and Communities" (LACCEI), 2019, doi: /10.18687/LACCEI2019.1.1.124

[14] Mijail Guerrero, Leonardo Vinces, Christian del Carpio, "Development of a Hand Prototype for Gripper Explosive Grenade of Pineapple Type by Applying Soft Robotic Elements and Embedded Sensors," 17th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Industry, Innovation, And Infrastructure for Sustainable Cities and Communities" (LACCEI), 2019, doi: 10.18687/laccei2019.1.1.274

[15] Chavez, L., Cortez, A., Vinces, L. (2022). A Strategy of Potential Fields and Neural Networks in the Control of an Autonomous Vehicle Within Dangerous Environments. In: Iano, Y., Saotome, O., Kemper Vásquez, GL, Cotrim Pezzuto, C., Arthur, R., Gomes de Oliveira, G. (eds) Proceedings of the 7th Brazilian Technology Symposium (BTSym'21). BTSym 2021. Smart Innovation, Systems and Technologies, vol 295. Springer, Cham.https://doi.org/10.1007/978-3-031-08545-1_43

[16] J. Nuñez, N. Rivas and L. Vinces, "A design of an autonomous mobile robot base with omnidirectional wheels and plane-based navigation with Lidar sensor," 2022 International Conference on Innovation and Trends in Engineering (CONIITI), Bogota, Colombia ,2022, p. 1-4. doi: 10.1109/CONIITI57704.2022.9953628.

[17] B. Díaz, N. Pacheco and L. Vinces, "Integration of a robotic arm Lynxmotion to a Robotino Festo through a Raspberry Pi 4," 2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA), Curicó, Chile, 2022, pp. 1-5, doi: 10.1109/ICA-ACCA56767.2022.10005932.