IMAV2018-43
http://www.imavs.org/pdf/imav.2018.43

Autonomous landing algorithm using a sun position predicting model for extended use of solar powered UAVs

B.P. Duisterhof and G.C.H.E. de Croon

ABSTRACT

In the field of robotics, a major challenge is extending the flight range of micro aerial vehicles. One way to extend the range is by charging batteries with solar arrays on the ground, while resting on intermediate landing positions. The solution we propose in this study differentiates itself from other solutions as it does not focus on improving UAV efficiency but rather on finding the most efficient landing position. In particular, an algorithm is developed to show the usefulness of the approach. This algorithm makes uses of the sonar sensor on board of the Parrot Bebop 1 drone in combination with an OptiTrack system to scan the environment for potential landing opportunities. After these measurements are discretized on a 2D grid, analysis is carried out with a sun position predicting model. Finally, a landing position is chosen within the scanned area and the drone will land accordingly.
Little is known on whether a solar powered charge on the ground could be effective in a limited period of time. We present a coarse analysis, showing that the DelftaCopter with solar arrays on its wings charges its batteries in 1.3 days with relatively cheap solar cells in Africa or Australia. Future work includes the use of computer vision instead of sonar as well as the ensurance of a safe landing position using vision.

1 INTRODUCTION

Future UAV operations will become more and more autonomous as human interaction is expensive and inefficient. This means there is a need for increasing UAV range in an autonomous fashion, limiting human interaction. Solar energy has a pivotal role in extending UAV range and has been used extensively to increase in-flight efficiency [1, 2, 3]. Other methods using piezoelectricity have also been considered in previous studies [4].
The main new concept introduced in this study is to use this technology with a new strategy. Very little is known about the use of solar arrays to charge UAVs while being at rest on the ground. This study was set out to develop an algorithm to charge a UAV at intermediate landing positions, using sonar and position data. With this the usefulness and feasibility is investigated.
A variety of applications exist, from delivering packages in remote areas (as the company 'Zipline' does [1]) to investigating large areas in nature [5, 6, 7]. UAVs capable of performing these type of tasks have been developed, but there might be cheaper options in the future using the new strategy. Mission time might be extended due to charging time, yet this could be compensated for by operating more UAVs in swarm formation.
In this study we will first evaluate previous work whereafter a sun model is selected for the purpose. Then, sonar and landing strategy are considered which is applied in the flight experiments section. Finally, sections on feasibility, conclusion and future work follow.

2 PREVIOUS WORK

Extending the range of UAVs has been subject of many studies in the field of robotics, mainly focusing on improving overall efficiency of aerial vehicles. Different energy harvesting methods have been considered in the past, such as the use of solar panels [1] or piezoelectricity (e.g. vibrations or rigid body movements)[4]. The AtlantikSolar from ETH [2] performed an uninterrupted flight of 81 hours using these cells. Also, vision based autonomous landing has been investigated before [8, 9].
To our knowledge, this is the first time that these two different topics are combined by selecting a landing spot not just based on safety, but also maximal exposure to the sun for fast recharging.

3 SUN MODEL

First the sun model will be discussed after which the scanning and environment analysis will be treated.
The sun position predicting model necessary for this application is elaborated on now. With this model, the UAV will finally be able to model the solar intensity on its surroundings and choose an appropriate landing position. To find the optimal landing position, it will calculate a 'solar-score', which is a measure for how illuminated a certain position is over a day of sun.

3.1 Sun model deliverables

It is of great use to describe the desired outcomes of the sun position predicting model, being: (i) Azimuth angle , a horizontal angle measured clockwise from a north base line

[1] `http://www.flyzipline.com/`, August 27 2018

or meridian, (ii) Elevation (altitude) angle, a vertical angle measured positive above the horizon. The physical meaning of these angles is depicted in Figure 1.

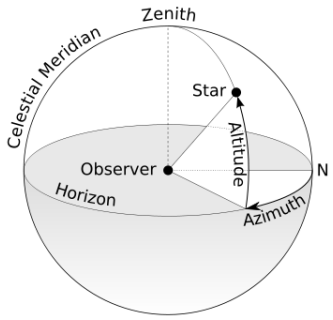

Figure 1: Elevation and Azimuth angles depicted on earth surface

Besides these deliverables, more requirements are present. Calculations should be computationally cheap enough to run on board a MAV and the calculated angles should be accurate enough to finally come up with the true optimal landing position.

3.2 Model decision

To finally compute the azimuth and elevation angle, a trade off between accuracy and computational effort is performed. The model from The Saudi Arabia University [10] is accurate and computationally cheap. It can predict both azimuth and elevation angle with an accepted maximum error of 3 degrees for azimuth angle and 1.4 degrees for elevation angle. To arrive at the deliverables, a number of steps is required which are listed now.

3.2.1 Local Standard Time Meridan (LSTM)

To start with, the LSTM can be calculated using Equation 1.

$$LSTM = 15 \cdot (\mid LT - GMT \mid) \tag{1}$$

Here LT is Local Time and and GMT Greenwich Mean Time. Both have to be inserted as a fraction in military time, s.t. 17:30 is inserted as 17.5.

3.2.2 Equation of Time (EoT)

The final goal of the Equation of Time (EoT) is to compute a standardized time : the Local Solar Time (LST). The equation is computed using curve fitting techniques, arriving at Equations 2 and 3.

$$EOT = 9.87 \cdot \sin(2B) - 7.52 \cdot \cos(B) - 1.5 \cdot sin(B) \tag{2}$$

$$B = \frac{(n-81) \cdot 360}{365} \tag{3}$$

Here n is the day number of the year, such that for January 1, n=1 and for December 31, n=365.

3.2.3 Time Correction Factor

Using the previously computed variables in Subsections 3.2.1 and 3.2.2, the total time correction factor can be computed according to Equation 4.

$$TC = 4 \cdot (Longitude - LSTM) + EOT \tag{4}$$

Here, longitude is included as UAV location is not necessarily synced with LSTM. In other words, LST is dependent on position and not only on the current time zone.

3.2.4 Local Solar Time

Finally, one standard time can be developed using Equation 5 : the Local Solar Time (LST).

$$LST = LT + \frac{TC}{60} \tag{5}$$

3.2.5 Solar Hour Angle (h)

The Solar Hour Angle will be advantageous in the final computation, seeing that it is defined as the angle w.r.t. the sun's orientation at its highest point. It is zero when the local solar time is 12 and 180 when the local solar time is 24.

$$h = 15^\circ \cdot (LST - 12) \tag{6}$$

3.2.6 Solar Declination Angle (δ)

Another angle necessary to finally compute the desired output is the solar declination angle. This angle is defined as the angle the earth's equator makes with the line joining the centers of earth and sun. This angle can be computed using Equation 7.

$$\delta = 23.45 \sin\left[\frac{(n-81)\,360}{365}\right] \tag{7}$$

This solar declination angle is useful for a range of astronomical applications as well as for this problem.

3.2.7 Solar Elevation angle (α_s)

Now, finally the first real output can be generated: the solar elevation angle. This is the angle the sun makes with the horizon and is often referred to as altitude. The solar elevation angle can be computed using Equation 8.

$$\alpha_s = \arcsin\left[\sin\delta \sin\phi + \cos\delta cos\phi \cos h\right] \tag{8}$$

Here ϕ is the latitude on earth where the model is used for.

3.2.8 Solar Azimuth Angle (γ_s)

The second desired output is the solar azimuth angle, which gives the orientation of the sun w.r.t. the north, in a clockwise fashion seen from above. It can be computed using Equation 9.

$$\gamma_s = \arccos\left[\frac{\sin\delta\cos\phi - \cos\delta\sin\phi\cos h}{\cos\alpha_s}\right] \quad (9)$$

This finally gives everything required to predict the sun's position. The curve fitting method assumes equality each year, which in reality is not entirely true. This is one of the sources which contributes to the final error in the model. Alternative methods make use of more dedicated curve fitting methods or even look-up tables. Future work includes a computation of the received energy per m^2 on the grid, including atmospheric conditions.

3.3 Solar Simulator Use

The equations discussed before have to be implemented in a computer program to test the outcome. This will be done in the C language to finally run on board of the Parrot Bebop 1 drone. A stand-alone program is set up first, with three main functions : (i) Manual Mode : compute elevation and azimuth angles for one specific point in time and space, (ii) Graph Mode: Write both angles for an entire day to a .txt file and (iii) Array Mode : Write an entire day into an array which can later be used elsewhere in the UAV. The use of the program is clarified in Figure 2.

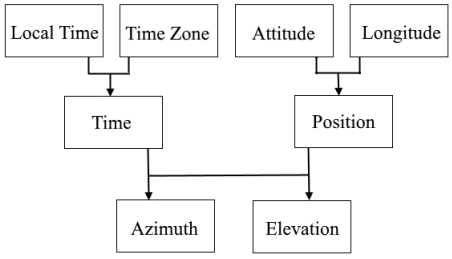

Figure 2: Program flowchart

The Graph Mode is built for verification purposes only, and will not be used in the UAV. Instead, only the array mode is used with location and time as inputs from the Bebop drone.

3.4 Verification

Now the outcomes of this program have to be verified and accepted. It has been shown earlier that the maximum elevation angle error is 1.4 degrees and the maximum azimuth angle error 3 degrees [10]. With this, two thought experiments are carried out: (i) a 2-D object (e.g. a plate), with error modelling of the present shadow over different elevation angles and (ii) a 3-D cylinder, with error modelling of sun position inside the cylinder over an entire day.

3.4.1 2D Case

The 2D case is relatively simple: the length of a shadow behind a wall is modeled with and without error. The three desired variables are: real shadow length, shadow length with error and the difference. General shadow length is defined by:

$$l_{shadow} = h_{obstacle} / \tan(\alpha_s) \quad (10)$$

Here l_{shadow} is the shadow length, $h_{obstacle}$ the obstacle height and α_s the elevation angle of the sun.
To compute the effect of model error, the maximum error will be added to the elevation angle. With this, the error can be calculated with the output as depicted in Figure 3 for a 2 m high object between 20 and 45 degrees elevation. This range has been chosen to demonstrate the behaviour of the error.

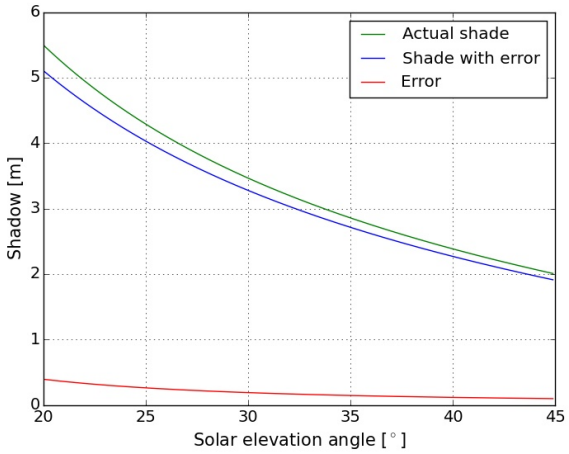

Figure 3: 2D shadow thought experiment for a 2m high object

To understand the behaviour, one can imagine a sunrise on a beautiful island. If the solar rays are coming in exactly horizontal, a palm tree will in theory create a shadow of infinite length. If then an offset of three degrees is added, a finite shadow arises with an infinite offset from the real shadow. In Figure 3 this tendency can be observed: for low elevation angles the offset increases. Anyhow, the error at one point in time is not enough to draw a conclusion for the final landing position. In subsection 3.4.2 the 3D case is analyzed for an entire day.

3.4.2 3D case

Fortunately, the problem is more complex. For the 3D case, a vertically placed cylinder is used in a thought experiment to test if the proposed model is indeed accurate enough. The

goal is to obtain a solar energy map of the ground in the cylinder (which is placed vertical), showing where the most ideal landing position would be.

This thought experiment is carried out by computing the received sun over an entire day. The results are visible in Figure 4, where 100 means the maximum amount of sun is received at this point (100 %). X and Y are zero at the centre of the circle.

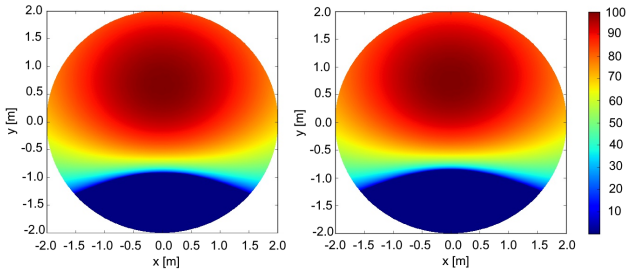

Figure 4: 3D cylinder thought experiment entire day. Left with error, right without error.

3.4.3 Model accuracy discussion

The results in Figures 3 and 4 have been computed numerically. The real important question should be : is the model good enough? Mainly focusing on Figure 4, it can seen that the landing position will not change significantly : the UAV will land in the middle of the dark red dot which is not moved significantly. Also, it is expected that the use of sonar will introduce far more significant errors than the model offset and will result in a relative coarse initial 2D mesh. With this, it is concluded that the sun-position predicting model is sufficiently accurate for the application.

3.5 Data acquisition

After the sun position predicting model has been established, a plan is made on how to obtain the required data. Two main data sources can be established : OptiTrack and sonar.

4 Sonar and Landing Strategy

Now that the sun-position predicting model has been approved, the next part of the algorithm can be treated: the construction of an elevation map. To create this map, only 3D coordinates of the UAV are insufficient. In fact, the difference between the UAV z-coordinate and the sonar measurement is the height at a certain point. One of the limitations of sonar is usually noise, which is investigated before continuing with this approach. In Figure 5 the result of flying over a table is depicted.
There is definitely noise, but with a 2D discretized map these outliers could be canceled out. One should be careful in selecting a sonar sensor for this application though. With a damaged or malfunctioning sonar sensor extreme noise can be observed, certainly when operating close to operative engines with major vibrations.

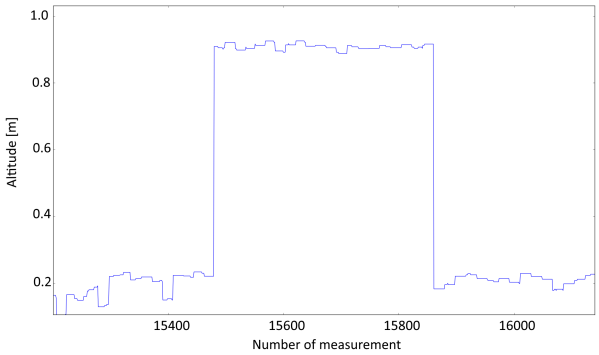

Figure 5: Sonar flight over table

Another aspect to consider is the fact that sonar gives point information. That is, it provides information about the environment at a certain point and not over a certain surface. This causes limitations which could be solved in the future with the use of computer vision.

4.1 UAV 3D coordinates

The other considerable input to compute an elevation map is the 3D coordinates of the UAV. This input is received from the OptiTrack system in TU Delft's Cyberzoo at the faculty of Aerospace Engineering. For outdoor operations, a GPS signal can be used.

4.2 2D discretized map

After sonar and position data has been retrieved, the continuous elevation map has to be discretized. A 2D grid with a known altitude for every cell is desired. In Figure 6 the left picture shows a discretized map and the right picture could be the original data.

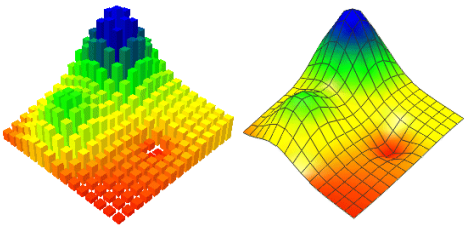

Figure 6: Elevation map discretization [11]

But why would destroying information be favourable? The most important reason is the point-information character of sonar. In reality, the UAV will not be able to cover every single centimeter with little coverage of the total area as a result. Sonar will actually create strings of information in the shape

of the flight path, which is not the desired outcome. Instead, the measurements in each cell are averaged and this averaged value is used to apply the sun model. This process could be seen as converting strings of information to small surfaces of information.

4.3 Sun model application on environment

Once the elevation map has been discretized, the sun position predicting model can be applied. The most illuminated landing position over an entire charge period is to be found. For now charging time is assumed to be 24 hours, but this could easily be changed for different purposes.

4.4 Laser beam approach

The computation is relatively straightforward. The sun model outputs a finite amount of sun positions, for all these positions it is checked whether a virtual "laser beam" cuts through the environment before it leaves the grid. If it does not, for this sun position, this cell "sees" sun. This method is demonstrated in Figure 7.

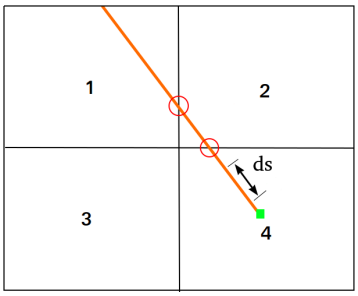

Figure 7: Laser beam approach

This is a test for cell four, where it starts in the middle of the cell at the green dot. The next step is walking along the "laser beam" over a distance ds per step. X, Y and Z are updated according to Equations 11, 12 and 13.

$$x_{i+1} = x_i + ds \cdot cos(\gamma_s) * cos(\alpha_s) \tag{11}$$

$$y_{i+1} = y_i + ds \cdot sin(\gamma_s) * cos(\alpha_s) \tag{12}$$

$$z_{i+1} = z_i + ds \cdot sin(\alpha_s) \tag{13}$$

Here i denotes the i-th timestep, α_s is solar elevation angle and γ_s is the solar azimuth angle.
Limitation of this model is that atmospheric conditions and solar cell orientation are not taken into account. Basically, the most illuminated landing position is calculated rather than the position with most solar energy.

4.5 UAV implementation

An efficient way to implement this algorithm into the Paparazzi autopilot software [12] has to be established. Two modules are added : one to collect data into arrays and one to analyze the data.

4.5.1 Data acquisition module

This module is fairly straightforward : it keeps loading sonar and position data into the desired arrays as long as the scanning pattern is not completed. These arrays are included in the header file of this module to later pass them to the analysis module.

4.5.2 Data analysis module

This module is somewhat more complex and consists of three parts, being : (i) Discretize obtained data, (ii) Compute finite amount of solar position and (iii) Apply model and choose best landing position.

The discretization is nothing more than taking the average of all measurements within one cell. After this has been done, the sun positions will be computed for a finite amount of points in time, depending on the computational power available.

Finally, the sun model needs to be applied. The general approach here is to leave the mesh relatively coarse at first to damp outliers as sonar has the tendency to show peaky outliers. Out of all cells, the mesh with highest solar intensity and with the highest altitude is chosen. Most important reason for this is that a high position is generally favourable in terms of sun, as there are less obstacles preventing sun from reaching that position. It could very well be that a large obstacle falls out of the scope of the scanning pattern and goes unnoticed. Therefore, it is useful to select the highest and stable landing position out of all positions with optimum illumination.
Then, after this cell has been chosen, it is decided on which place in that cell the UAV will land. This is done by looking at the original data and "walking" over this data in time. The landing position is chosen at the point in time where the area under the altitude graph between t-x and t+x is largest, that is, for Equation 14 :

$$P(t) = \int_{t-x}^{t+x} (z(t) - s(t))dt \tag{14}$$

where the highest value of P is obtained. Here x is in seconds and determines the time window considered, z(t) is the z GPS (OptiTrack) coordinate and s(t) is the sonar value. The advantage of this approach is that relatively easily the middle of a flat object can be found, as for example a table.
More in general, this two-step approach to first use a coarse mesh and than analyze data within that mesh makes for precise landing while still limiting noise. Only disadvantage is that a UAV must fly over the final landing position to finally land there, as no interpolation is done. Thisis a future step if sonar continues to be used, but computer vision is preferred.

As an example, imagine flying over a table. As shown in Figure 8 and 9, the middle of the table can be identified using Equation 14. The UAV will then land exactly there where it measured the maximum value of $P(t)$.

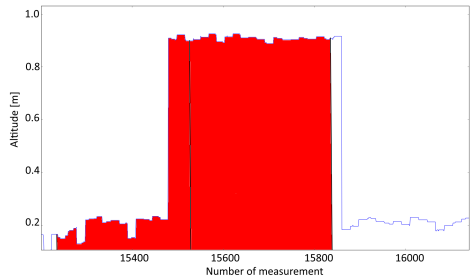

Figure 8: Table not centered

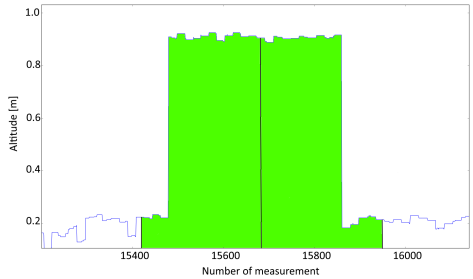

Figure 9: Table centered : max area

Using a denser mesh and analyzing multiple cells has also been considered. Anyhow, this approach did not damp out outliers enough. More importantly, scanning coverage was too limited for this approach. Only a limited amount of cells would be filled with data and the analysis becomes unreliable. Without interpolation, a cell without data would have an altitude of zero which would make the analysis a lot less reliable. This is why, at this stage, it is chosen to analyze a coarse mesh and in future work use computer vision with optic flow.

5 Flight Experiments

Once an algorithm has been developed, it can be tested in flight experiments. One component of the algorithm is not yet constructed: the flight plan. The flight plan is crucial for a successful experiment and it is a continuous trade off: by scanning an environment in more detail more battery is sacrificed to recharge efficiently. On a huge flat surface this may be very inefficient, but in more populated areas it could be useful to take more time for scanning.
The general shape can be varied, but the two proposed shapes are zigzag and spirals. Zigzag means the UAV tries to scan a square by flying in U-shapes. That is, it crosses the square perpendicular, moves slightly to the side and crosses it again perpendicular. This is very similar to a creeping line search. Spirals are as the name implies: just spirals within the same square. These shapes could be used for environments where little is known and a more random flight plan is desired.
Both flight plans are programmed into Paparrazi open source autopilot [12].
After carefully constructing the algorithm in the framework of Paparazzi autopilot [12], a real test of this algorithm is to be performed. Main goal is to verify precision as well as plausibility of the outcome. Therefore, different test setups were constructed in TU Delft's Cyberzoo, as depicted in Figures 10 and 11.

Figure 10: Test setup 1 : small table on flat surface

Figure 11: Test setup 2 : random chairs added

The UAV should land on the table as it is the highest, flattest and most illuminated surface in the Cyberzoo. The small table demonstrates precision as it is a relatively small target. Besides, the chairs in Test setup 2 are added to test a more realistic environment.

5.1 Test Setup 1

After having performed the scanning pattern, all data is discretized on-board in a 10x10 mesh. The outcome of this discretization is visible in Figure 12 where the can be distinguished easily.
After the discretization, the local solar intensity was computed as visible in Figure 13. One might notice "gaps" in solar intensity, which is caused by limited coverage by the

IMAV2018-43
http://www.imavs.org/pdf/imav.2018.43

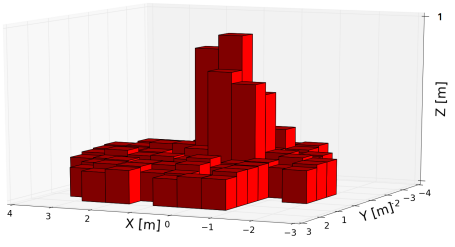

Figure 12: Test Setup 1 discretized

zigzag flightplan. Certain cells don't contain any data and are therefore assigned with a solar intensity score of zero (they are blue).

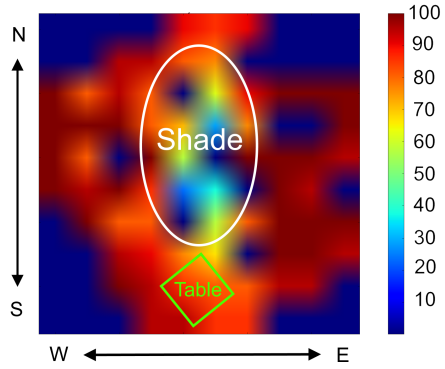

Figure 13: Solar intensity Test Setup 1 over a day, as % of the max

The final outcome in test setup 1 is that it landed on the table multiple times. However, using a well-suited flightplan was critical. It occasionally landed on the edge of the table, as that is where it flew straight over. Once it flew over the table straight, the UAV could land on the table.

5.2 *Test Setup 2*

Once setup 1 was relatively robust, a more complex environment was used for a test, depicted in Figures 14 and 15. Chairs were added randomly in different places.

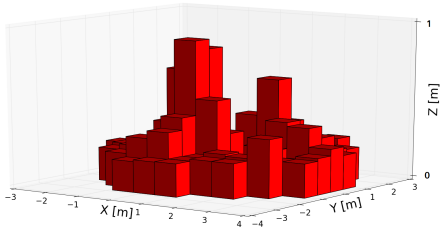

Figure 14: Test Setup 2 discretized

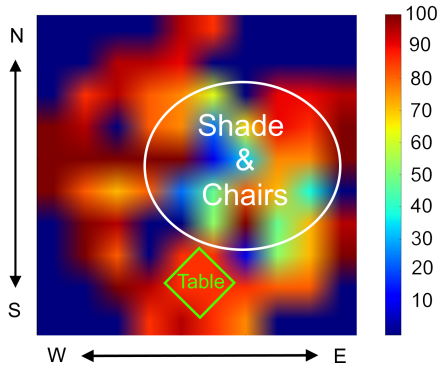

Figure 15: Solar intensity Test Setup 2 over a day, as % of the max

The UAV landed on the table multiple times again correctly but an appropriate flight plan remained important.

6 Feasibility

After the algorithm has been implemented in a Parrot Bebop 1 drone, it is useful to evaluate real world applications. With a preliminary estimation on charging capabilities of the DelftaCopter [13], more insight can be obtained on how feasible the idea is and what a possible recharge time would be. The DelftaCopter [13] is a delta-wing electric long range transitioning autonomous helicopter.

6.1 *Approach*

An accurate estimation of recharging performance is hard, as various factors play a major role. Among others, the landing area influences recharging performance by reflection and possible shades. For this feasibility study, it is estimated what order of magnitude the recharge time would be for a DelftaCopter with solar arrays on its wings. This is done by starting from the average received solar energy per day and with this compute the time required to recharge the DelftaCopter.

6.2 *Solar Energy*

For this, it is of great use to evaluate solar energy on a global level. Figure 16 is a map showing solar energy captured over the scope of a day (averaged) and a year respectively. It can be seen that Africa and Australia receive relatively high amounts of solar energy, while northern Europe is less suitable for the recharging application.

6.3 *Recharging the DelftaCopter*

Now that information on solar energy has been gathered, an estimated recharge time for the DelftaCopter can be computed. First of all, a solar cell is chosen: a Flexible Mono Solar Cell. Main advantages are its price, flexibility and efficiency. Its properties are visible in Table 1.

It has to be estimated now how many panels can be fitted on the DelftaCopter. Via straight forward geometric computations it can be computed that 7 of these cells can be

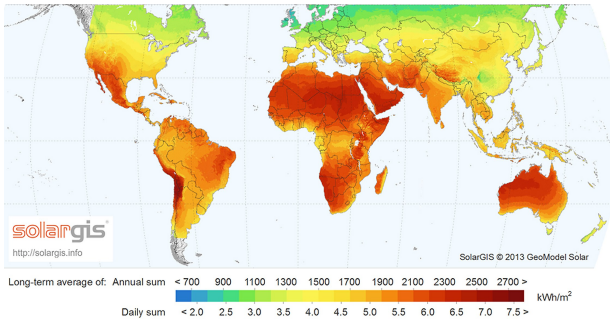

Figure 16: Global Solar Energy, SolarGIS ©2014 GeoModel Solar

Table 1: Flexible Mono Solar Cell 125x125 Monocrystalline, Amazon (25-7-2018)

Spec	Value
Efficiency	22.5 %
Size	125mm x 125mm
Power output	3.2-3.4 watts
Price for 10 pieces	50.5 USD

placed on one side of the wing and thus a stacking factor of 0.44 can be achieved.

It was stated before that this feasibility study will need assumptions and that the final computations will be a rough estimate. It is assumed now that only one side of the DelftaCopter is filled with the solar cells and that they take up 22.5 % of the solar energy they receive. This is similar to laying the drone on the ground horizontally, which would need hardware modifications in the current version of the DelftaCopter. Anyhow, another option would be to stack solar arrays along the other wing too to capture reflected sunlight. The exact layout is beyond the scope of this study and we will for now assume that the DelftaCopter is able to position the wing so that it lies flat on the ground.
To compute the recharge time, Equation 15 is used.

$$T_{charge} = \frac{E_{bat}}{n_{arrays}\, E_{sun}\, A_{array}\, \eta_{array}} \tag{15}$$

Here T_{charge} is the charging time in days, E_{bat} the energy available in the batteries in Wh, n_{arrays} the number of solar arrays used, E_{sun} the solar energy received over one day on 1 m^2 in Wh, A_{array} the surface area of 1 solar cell and finally η_{array} the efficiency of the solar cell.

Knowing that the approximate battery size of the DelftaCopter is 227 Wh and the solar cell properties from before, charging estimates for different positions around the globe can be performed. The results are shown in Table 2.

Again this is a very rough estimate, but it shows that recharging in 'a couple of days' is definitely feasible. Alternative configurations with folding mechanisms for solar arrays can be considered and make the solution more attractive with a lower recharge time.

Table 2: Recharging parameters around the globe

E_{sun} [Wh/m^2]	T_{charge} [days]	Avg. P_{cell} [W]	Region
7000	1.32	2.05	Australia, Africa
4500	2.05	1.32	USA, Asia
3000	3.07	0.88	Europe

7 Conclusion

Several studies have shown that solar arrays on the wing of an UAV during flight can be effective. Very little was found in literature on the question if using solar arrays to charge while standing on the ground could also be effective. The current study found that this can indeed be an effective way to substantially increase range for UAVs used for long-duration autonomous operations.
The contribution of this study has been to confirm that it is feasible to use this approach for UAVs as the DelftaCopter and make new types of missions possible without a return to home. Similarly, the algorithm developed in this study turned out to be successful and capable of finding the optimal landing position in several situations.
Interesting applications could be payload delivery or surveillance in vast remote areas. Mission duration might be extended, but cost and autonomous capabilities could be enlarged significantly.
UAVs will have to operate for an extended period of time in the future. This study is a step in the right direction on how to recharge effectively and enlarge mission capabilities to put UAVs to good use in a more effective and efficient way.

8 Future Work

The algorithm is limited to sonar and position data and needs more work to make the outcome more reliable and efficient. Vision based elevation mapping could be a major addition to the algorithm, as this allows for scanning of strokes instead of lines. Apart from that, a merge with other landing algorithms could be performed. Also, more work can be done on novel UAV configurations for this purpose. Folding solar arrays could be a solution. Finally, intelligent path planning for this application is another interesting topic. When and where to charge most effectively is an important topic, flying by night might be most efficient for example.

Acknowledgements

This study would not have been possible without the help of our colleagues in the MAVLab, who were extremely helpful. The facilities in the MAVLab, such as the Cyberzoo and the Bebop 1 used for the flight experiments have also been essential.

REFERENCES

[1] Murat Bronz, Jean Marc Moschetta, Pascal Brisset, and Michel Gorraz. Towards a long endurance mav. *International Journal of Micro Air Vehicles*, 1(4):241–254, 2009.

[2] L. Wirth, P. Oettershagen, J. Ambhl, and R. Siegwart. Meteorological path planning using dynamic programming for a solar-powered uav. In *2015 IEEE Aerospace Conference*, pages 1–11, March 2015.

[3] H. Bahrami Torabi, M. Sadi, and A. Yazdian Varjani. Solar power system for experimental unmanned aerial vehicle (uav); design and fabrication. In *2011 2nd Power Electronics, Drive Systems and Technologies Conference*, pages 129–134, Feb 2011.

[4] Steven Anton. Energy harvesting for unmanned aerial vehicles. 01 2008.

[5] Guoqing Zhou and Deyan Zang. Civil uav system for earth observation. In *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 5319–5322, July 2007.

[6] Nicola Casagli, William Frodella, Stefano Morelli, Veronica Tofani, Andrea Ciampalini, Emanuele Intrieri, Federico Raspini, Guglielmo Rossi, Luca Tanteri, and Ping Lu. Spaceborne, uav and ground-based remote sensing techniques for landslide mapping, monitoring and early warning. *Geoenvironmental Disasters*, 4(1):9, Mar 2017.

[7] Mingxing Gao, Xiwei Xu, Yann Klinger, Jerome van der Woerd, and P Tapponnier. High-resolution mapping based on an unmanned aerial vehicle (uav) to capture paleoseismic offsets along the altyn-tagh fault, china. 7, 12 2017.

[8] G.C.H.E. de Croon, H.W. Ho, C. De Wagter, E. van Kampen, B. Remes, and Q.P. Chu. Optic-flow based slope estimation for autonomous landing. *International Journal of Micro Air Vehicles*, 5(4):287–297, 2013.

[9] Omid Shakernia, Yi Ma, T. John Koo, and Shankar Sastry. Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control. *Asian Journal of Control*, 1(3):128–145.

[10] A. A. Rizvi, K. Addoweesh, A. El-Leathy, and H. Al-Ansary. Sun position algorithm for sun tracking applications. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 5595–5598, Oct 2014.

[11] Mark Dorey Derek Smith David Kidner. What's the point? interpolation and extrapolation with a regular grid dem. *GeoComputation 99*, 1999.

[12] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard, and Jeremy Tyler. The Paparazzi Solution. In *MAV 2006, 2nd US-European Competition and Workshop on Micro Air Vehicles*, page pp xxxx, Sandestin, United States, October 2006.

[13] Christophe De Wagter, Rick Ruijsink, Ewoud Smeur, Kevin van Hecke, Freek van Tienen, Erik van der Horst, and Bart Remes. Design, control and visual navigation of the delftacopter. *CoRR*, abs/1701.00860, 2017.