# Never-landing drone

## Developing a modular tool to simulate regeneration power potential using orographic wind-hovering UAVs

by

M. Gossye

To obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday September 20, 2022 at 13:00.

| | | |
|---|---|---|
| Student number: | 4460766 | |
| Project duration: | February, 2020 – September, 2022 | |
| Supervisor: | ir. B. D. W. Remes, | TU Delft |

# Preface

This MSc thesis marks the end of being an Aerospace Engineering student in Delft for the last 7 years. It was a time full of challenges, new experiences and acquiring incredible amounts of new knowledge and insights in the fields I am most passionate about, namely aviation, engineering, avionics and programming.

The neverlanding-drone project appealed to me since this project combines novel techniques in the areas of UAVs and electronics. Next to the interesting blend of topics that I was keen on expanding my knowledge in, I was also drawn to this project because I am an avid UAV builder/pilot myself as I was part of the Lambach Aircraft society during my entire study period where I spent many evenings designing, building and flying various UAVs with an amazing team.

I am very grateful to Bart Remes, my project supervisor, for having given me the opportunity to take part in this project, as well as for always encouraging and guiding me to be able to finish the project which was an enormous challenge for me. Special thanks to Sunyou Hwang for the guidance and providing tips when I was writing the journal article, and to Nicholas Dvorsky for the help with setting up the test-setup and letting me borrow some of his equipment to conduct the tests. I would also like to thank all other team members of the NLD project group for the helpful and insightful weekly discussions about our progress. I am very thankful to Prof. dr Guido de Croon for the guidance he provided during the project, as well as being the chair of the graduation committee. I also would like to thank Dr. Ewoud Smeur and Michiel Zaayer for being able to join the graduation commitee on short notice.

Last but not least, I would like to extend my deepest gratitude to some of the most important people in my life. Getting to the point to even start my studies in Delft, let alone finish them successfully would not have been possible without the support, trust and patience of my parents. I cannot even begin to express my gratefulness to them for having given me all of the support they possibly could to allow me to pursue my ambition of studying Aerospace Engineering in Delft. I could always count on them, even when times were tough, and they did everything they could to help me achieve my own goals. I would also like to thank my closest friends for their support and making my time as a student in Delft so enjoyable. I am especially grateful to Josh, without whom I wouldn't have been able to finish my thesis. He was always there for me and encouraged me to successfully finish the project. He was always willing to spend the little precious free time he had to help me with planning and sharing tips he learned during his thesis project .

*M. Gossye*
*Delft, September 2022*

# Contents

# Abbreviations

| | |
|---|---|
| AC | Alternating Current 26, 34 |
| BLDC | Brushless DC electric motor xi, 26, 27, 29, 30, 32–35, 62 |
| CAN | Controller Area Network 29 |
| CFD | Computational Fluid Dynamics v, xi, 37, 40–43, 54, 62 |
| CG | Centre of Gravity 44 |
| DC | Direct Current vii, 26, 27, 34 |
| emf | electromotive force ix, 27–29 |
| ESC | Electronic Speed Controller v, xi, 26, 27, 29, 31–33, 62 |
| FBD | Free Body Diagram xi, 44–47 |
| HAWT | Horizontal Axis Wind Turbine 42 |
| LiPo | Lithium-ion Polymer battery 26, 27, 29, 32, 34 |
| MOSFET | Metal Oxide Semiconductor Field Effect Transistor xi, 28, 29, 31, 33, 35 |
| openFOAM | Open-source Field Operation And Manipulation 41, 42, 62 |
| PPM | Pulse-Position Modulation 31 |
| RC | Radio Control 62 |
| regen | regenerative v, vi, xi, 1, 22, 23, 25–35, 37, 42, 48, 50–54, 62, 63, 66 |
| RPM | Rotations Per Minute ix, xi, 22, 25–27, 32–36, 62, 64, 65 |
| RX | receiver 31 |
| UART | Universal Asynchronous Receiver-Transmitter 29 |
| UAV | Unmanned Aerial Vehicle xi, 1, 19–21, 25, 26, 29, 37, 42, 44, 47, 49–54, 62, 63 |

# List of Symbols

| | |
|---|---|
| $\rho$ | Air density 21, 43, 45, 46 |
| $A$ | Aspect ratio 21 |
| $C_D$ | Drag coefficient 21 |
| $C_{D_0}$ | Zero-lift drag coefficient 21 |
| $C_L$ | Lift coefficient 21 |
| $S_{turb}$ | Propeller/wind turbine rotor disc area 43, 45, 46, 49 |
| $D$ | Total drag force of aircraft 44 |
| $\mathscr{E}$ | electromotive force of (brushless) electric motor/generator 26, 27 |
| $\gamma$ | Flight path angle 44, 45, 46, 47 |
| $g$ | Gravitational acceleration 21, 44, 45 |
| $\eta$ | Radial position of aircraft with respect to centre of hill 20 |
| $K_v$ | Back-emf to RPM constant of brushless motor/generator 26, 27 |
| $L$ | Total lift force of aircraft 44, 45, 46 |
| $m$ | Mass of aircraft 21 |
| $\dot{h}_{min}$ | Minimum sink-speed/rate of aircraft 20, 21 |
| $e$ | Oswald efficiency factor 21 |
| $a$ | x-coordinate focal point oval 39 |
| $x_{stag}$ | x-coordinate stagnation point 38, 39 |
| $r$ | Distance between aircraft and centre of hill 20 |
| $R$ | Radius of (circular) hill 20 |
| $z_0$ | Surface roughness 39 |
| $T$ | Total thrust force of aircraft 44, 45 |
| $D_{turb}$ | Drag force of turbine 45, 46 |
| $U_\infty$ | Free-stream velocity potential flow 38, 40 |
| $u_x$ | Horizontal component of local wind velocity 47, 48 |
| $u_z$ | Vertical component of local wind velocity 47, 48, 49 |
| $w_z$ | Local vertical updraft velocity 20 |
| $V_{air}$ | Airspeed 21, 43, 45, 46, 47, 48, 49 |
| $w_\infty$ | Free-stream velocity 20 |
| $W$ | Weight of aircraft 44, 45, 46, 47, 48 |
| $S$ | Wing surface area 21 |

# List of Figures

# 1
# Introduction

Small electric UAVs (Unmanned Aerial Vehicles) are becoming more and more commonplace in today's environment, carrying out missions ranging from surveillance and surveying tasks to film-making, aerial photography, product delivery etc. The technologies employed in these UAVs have been improved considerably over the past few years, including advancements in avionics miniaturisation, propulsion system efficiency, and power storage capacity improvements [21]. However, it has been identified that even with these advancements the effective range and/or endurance of small electric UAVs often falls short of the desired range/endurance for various missions.

It would be beneficial to research existing and new novel methods that can be applied to increase the effective endurance of these unmanned aircraft. One of these methods that shows a lot of potential is to take advantage of wind updrafts generated by hills and other obstacles in a free wind-stream, using static soaring. This technique is also used by some birds such as the sea gull to harvest energy from the environment to increase their range and endurance [5]. However, instead of using this energy to increase the potential energy (height) of the UAV, the energy can also be converted to electrical energy by using a regenerative electric drivetrain that can recharge the battery in-flight. With this type of drivetrain, the propeller and motor on the UAV would essentially function like a wind-turbine to recharge the battery. At a later point in time, the recuperated energy can be used on-demand by the engine and it's controller.

If it would be possible to statically hover in a region with updrafts, the UAV could stay in this region for a longer period of time while using the regenerative drivetrain to recharge the batteries. To date, this concept has not been thoroughly explored yet and it is unknown whether this strategy is feasible to extend the effective endurance of UAVs.

This thesis will focus on researching the feasibility of this static hovering energy harvesting regenerative concept by developing a computationally light-weight model that can predict the regen power generating potential over idealised simple objects such as oval shaped hills. Proof-of-concept hardware for the regenerative drivetrain will also be tested and it's performance measured to get a better grasp on the achievable efficiency of such a system.

The research presented in this thesis was accepted for publication in the journal of Unmanned Systems [15]. The thesis document is structured as follows: Chapter 2 contains the accepted journal paper which accurately summarises the research that was performed.

The following chapters are structured as follows:

In Chapter 3 the research questions and research objectives are described.

Chapter 4 introduces the reader to the different soaring techniques and gives an overview of previous research into regenerative & orographic soaring for UAVs.

Next, Chapter 5 discusses the working principles of a regen drivetrain and gives a detailed explanation of the various components. In the 2nd part of the chapter a test-setup is devised to measure the real-life performance of a representative regen drivetrain that can be used in UAVs, and the results from the measurements are discussed.

The following chapter (Chapter 6) discusses the process on how wind-field estimation and regen hovering flight models were derived and a tool was developed in order to be able to estimate the hovering locations and determine an estimate for the maximum amount of regen power that can be extracted for each point within this region.

In Chapter 7 an overall conclusion of the thesis research is drawn and a recommendations for future work are formulated.

# 2

# Paper

# Unnamed Systems

## Developing a modular tool to simulate regeneration power potential using orographic wind-hovering UAVs

--Manuscript Draft--

| | |
|---|---|
| **Manuscript Number:** | US-D-22-00008R2 |
| **Full Title:** | Developing a modular tool to simulate regeneration power potential using orographic wind-hovering UAVs |
| **Article Type:** | Special Issue Paper |
| **Section/Category:** | Special Issue on Selected Papers from IMAV 2021 |
| **Keywords:** | Wind-hovering; Regenerative soaring; Regenerative drivetrain |
| **Corresponding Author:** | Midas Gossye<br>Delft University of Technology: Technische Universiteit Delft<br>NETHERLANDS |
| **Corresponding Author Secondary Information:** | |
| **Corresponding Author's Institution:** | Delft University of Technology: Technische Universiteit Delft |
| **Corresponding Author's Secondary Institution:** | |
| **First Author:** | Midas Gossye |
| **First Author Secondary Information:** | |
| **Order of Authors:** | Midas Gossye |
| | Sunyou Hwang |
| | Bart Remes, Ir. |
| **Order of Authors Secondary Information:** | |
| **Abstract:** | Applications of Unmanned Aerial Vehicles (UAV's) are often limited by flight endurance. To address the limitation of endurance, we propose a regenerative soaring method in this paper. The atmospheric energy from updrafts generated by obstacles such as hills and ships can be harvested by UAV's using a regenerative electric drivetrain. With fixed-wing aircraft, the vehicle can hover with specific wind conditions, and the battery can be recharged in the air while wind hovering. In order to research the feasibility of this regenerative soaring method, we present a model to estimate hovering locations and the amount of extractable power using the proposed method.<br><br>The resulting modular regeneration simulation tool can efficiently determine the possible hovering locations and provide an estimate of the power regeneration potential for each hovering location, given the UAV's aerodynamic characteristics and wind-field conditions. Furthermore, a working regenerative drivetrain test setup was constructed and characterised that showcased promising conversion efficiencies and can be incorporated into existing UAV's easily. |
| **Response to Reviewers:** | Copyright transfer form and source files are now included as requested |

Unmanned Systems, Vol. 0, No. 0 (2013) 1–12
© World Scientific Publishing Company

# Developing a Modular Tool to Simulate Regeneration Power Potential Using Orographic Wind-hovering UAV's

M. Gossye[a,*] S. Hwang[a], and B.D.W. Remes[a]

[a] *MAVLab, Delft University of Technology, Kluyverweg 1, 2629HS Delft, the Netherlands*
*E-mail: midasgossye@gmail.com*

Applications of Unmanned Aerial Vehicles (UAV's) are often limited by flight endurance. To address the limitation of endurance, we propose a regenerative soaring method in this paper. The atmospheric energy from updrafts generated by obstacles such as hills and ships can be harvested by UAV's using a regenerative electric drivetrain. With fixed-wing aircraft, the vehicle can hover with specific wind conditions, and the battery can be recharged in the air while wind hovering. In order to research the feasibility of this regenerative soaring method, we present a model to estimate hovering locations and the amount of extractable power using the proposed method. The resulting modular regeneration simulation tool can efficiently determine the possible hovering locations and provide an estimate of the power regeneration potential for each hovering location, given the UAV's aerodynamic characteristics and wind-field conditions. Furthermore, a working regenerative drivetrain test setup was constructed and characterised that showcased promising conversion efficiencies and can be incorporated into existing UAV's easily.

*Keywords*: Wind-hovering; Regenerative soaring; Regenerative drivetrain

## 1. Introduction

UAV's are performing more and more diverse missions every year, but are often limited by the maximum achievable endurance and/or range. Using the principle of orographic soaring to extend the range and endurance of UAV's has already been extensively researched, often based on techniques used by various bird species that have been observed.[1,2] However, conventional orographic soaring techniques do have some limitations that limit their usability in certain environments and conditions.

With traditional soaring methods, the only energy-storage mediums are the potential energy (altitude) and kinetic energy (airspeed) of the aircraft. The associated aircraft state variables, altitude & airspeed, are often desired to stay constant to be able to take advantage of the favourable conditions to gain energy from the atmosphere.[3] A great example of this is when one is, for instance, soaring upwind along a ridge trying to take advantage of the updrafts it generates. It is possible to store the gained energy in the form of altitude, but the higher the altitude, the weaker the updrafts are from the obstacle. At a certain altitude, the updrafts become so weak that the glider is barely able to maintain altitude without losing airspeed. Once this "ceiling altitude" is reached, it is not possible to store any more energy. It is possible to trade the potential energy for kinetic energy, and dive back down to the original altitude while gaining airspeed. The aircraft is now positioned once again in a region with stronger updrafts. However, due to the increased airspeed, the glider has a higher sink-speed which may render it unable to gain energy from the updrafts anymore.

Regenerative soaring introduces another energy storage medium to store harvested energy from the environment. The regenerative soaring method was first proposed by Paul MacCready already back in 1998 .[4] Instead of having to change the altitude and/or airspeed to be able to store energy, an on-board energy accumulator in the form of a rechargeable battery can harvest the energy through the use of a regenerative drivetrain. This means that the aircraft can stay positioned in the altitude region where the most favourable updrafts are present, and keep its optimum airspeed.

One problem with the suggested regenerative orographic soaring methods is that a long ridge or hill range is required to take advantage of this, such that the aircraft can fly straight along the ridge in the most favourable updraft regions for an extended period of time. It would be beneficial if small UAV's could also use the updrafts present around smaller, single objects such as a small hill, a building or a ship on the open sea. This could be achieved by altering the orographic regenerative soaring methods by

[*] Delft, the Netherlands

applying a technique called wind hovering or static hovering.

Achieving static hovering while using the orographic soaring method (called wind hovering) is a topic found in a minimal amount of research.

Fisher [5] introduced the concept of a "feasible soaring region", a spatial region inside a wind-field where wind hovering is possible for a given wind-speed. A point in the wind-field is deemed feasible for wind hovering if the local vertical wind component/updraft velocity is larger or equal than the minimum sink speed of the aircraft when flying at zero ground speed in the wind field. In their paper, the feasibility of having a fixed-wing UAV autonomously hover in the updraft region of a hill and a building was investigated. The paper concluded with the experimental results proving that a small UAV can indeed apply wind-hovering techniques to statically hover in the favourable updraft region. Our research tries to determine if this wind hovering UAV concept can be extended by adding a regenerative drivetrain, which would in theory combine both the advantages of wind hovering and regenerative soaring techniques. In order to research the feasibility and achievable power levels of this concept, a simplified model was created that can estimate the best case available regen power while hovering, given the aerodynamic parameters of the UAV and the flow conditions of the wind-field. Furthermore, a test-setup was devised to measure the real-life efficiency of a regenerative drivetrain with standard components such as a Brushless DC electric (BLDC) motor and Lithium Polymer (LiPo) battery typically found on small electric UAV's.

The remainder of this paper is structured as follows: Section 2 introduces the wind field estimation method, Section 3 describes how to calculate extractable power generated by the wind field, Section 4 presents how the feasible soaring locations and generated power at each location are determined. Section 5 presents some details on the regenerative drivetrain test setup and results. Finally, Section 6 gives a summary of the presented work and discusses future work that can expand this concept.

## 2. Wind-field estimation

To be able to determine the power available in the wind-field, it is first vital to have a good understanding of the wind-field. To achieve this, a wind-field estimation tool is required that can simulate the flow around various simple obstacles. The following subsections will describe what methods are available to achieve this and how the wind-field estimation program was implemented.

### 2.1. *Methods*

There exist numerous methods to estimate the behaviour of air around obstacles, greatly varying in complexity and required computational power. The most common choice lately has been to use a complex Computational Fluid Dynamics (CFD) simulation package like ANSYS fluent, openFOAM, etc. The CFD simulations performed with these packages require a large amount of computational power and are very complex to set-up. It was opted to first search for another method as a basis of the wind-field estimator. Langelaan used a simplified potential flow method to find the wind field upwind of an idealised circular shaped hill, as was presented in [6], to gain a better understanding of the general behaviour of the wind-field and to estimate the ideal location relative to the circular hill for ridge soaring. This methodology sparked the idea to use potential flow theory to estimate the flow field present upwind of the hill.

### 2.2. *Potential flow estimator*

The standard potential flow equations describing the idealised flow around circular and oval shaped obstructions were used as a basis.

The equations used to determine the flow-field are listed below, with $U_\infty$ being the free-stream velocity, $R$ the radius of the circular hill and $r$ the distance between the aircraft and the centre of the hill. $\theta$ represents the angle between the horizontal axes and the radial of the aircraft (see Figure 1):

$$u_r = \left[1 - \frac{R^2}{r^2}\right] U_\infty \cos\theta \tag{1}$$

$$u_\theta = -\left[1 + \frac{R^2}{r^2}\right] U_\infty \sin\theta \tag{2}$$

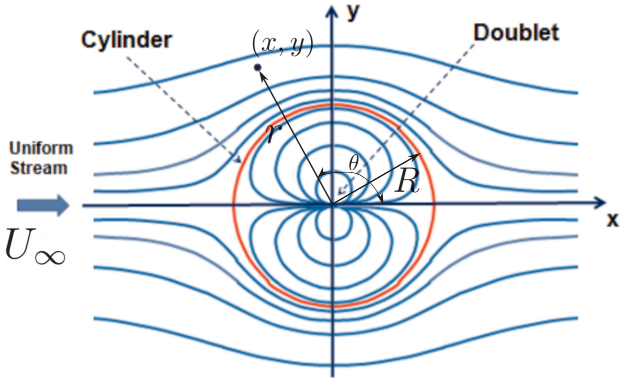


Fig. 1: Potential flow field around cylinder

Transforming the polar velocity components into cartesian velocity components results in the following velocity functions for the x and y components:

$$u_x = \cos\theta \cdot u_r - \sin\theta \cdot u_\theta \tag{3}$$

$$u_y = \sin\theta \cdot u_r + \cos\theta \cdot u_\theta \tag{4}$$

$$with \quad \theta = \arctan\frac{y}{x} \tag{5}$$

Another set of equations for oval shaped hills can also be selected, which correspond to the equations representing the flow-field over a rankine oval, as described in [7]:

$$x_{stag}^2 - a^2 - \frac{ma}{\pi U_\infty} = 0 \tag{6}$$

$$\Leftrightarrow m = \frac{\pi U_\infty}{a}(x_{stag}^2 - a^2) \tag{7}$$

$$u_x(x,y) = U_\infty + \frac{m}{2\pi}\left[\frac{x+a}{(x+a)^2+y^2} - \frac{x-a}{(x-a)^2+y^2}\right] \tag{8}$$

$$u_y(x,y) = \frac{my}{2\pi}\left[\frac{1}{(x+a)^2+y^2} - \frac{1}{(x-a)^2+y^2}\right] \tag{9}$$

Where the x-coordinate of the stagnation point $x_{stag}$ and the x-coordinate of the focal point $a$ determine the geometry of the oval shaped hill.

These equations were then altered with a simplified boundary/shear layer model equation to include an estimate of the Atmospheric Boundary Layer.

Equation 10 shows the used model that alters the vertical wind-speed distribution with a logarithmic function to try to estimate the Atmospheric Boundary Layer.

One problem arises by using this simple model to estimate the varying wind speeds in the boundary layer, the function is only able to estimate the boundary layer effects to the horizontal wind-speed over flat terrain. It has been proven though that the log wind-profile can produce accurate results even above non-flat terrain in certain circumstances at higher altitudes above the obstacle.[8] The log wall function can certainly be applied to the regions of the flow that are not greatly affected by the presence of the hill (mainly upwind of the hill-side). The proposed boundary layer model will however most likely not predict the boundary layer effects close to the hill surface. It was still opted to use this model for the entire hill region since the resulting flow patterns are more closely resembling real-life wind conditions where the flow velocity decreases close to the surface due to friction. If more accurate flow behaviour needs to be predicted close to the surface of the hill, a CFD simulation including models for laminar and turbulent boundary layer behaviour would be more applicable.

$$u(z_2) = u(z_1)\frac{\ln((z_2-d)/z_0)}{\ln((z_1-d)/z_0)} \tag{10}$$

## 3. Power contours

Now that a wind field estimate is available around differently sized obstacles, it is time to determine the feasible power that can be extracted at each point.

Before going into the details of the ability of the UAV to perform wind hovering at each location, it is helpful to first estimate the theoretical maximum power that can be extracted at each location assuming the UAV can maintain to hover at that location indefinitely. In this case, the energy harvesting UAV can essentially be modelled as a Horizontal Axis Wind Turbine (HAWT) where the upstream wind velocity is equal to the total wind velocity at the location of the UAV in the wind field. This is not totally accurate, since this assumes that the upstream wind velocity is constant along the axis of the propeller, but since the propeller dimensions of small UAV are at least an order of magnitude smaller than the obstacle dimensions it can be assumed that this will only have a very minor effect.

To be able to determine the theoretical maximum power that a HAWT can extract from the wind stream, it is evident to first have a closer look into the so called Betz law:

### 3.1. *Betz law*

One of the most famous theories concerning wind turbine theory is the Betz law (also called Betz condition or limit).

Simply put, it states that even an ideal wind turbine that contains no centre hub and has an infinite number of blades that cause no additional drag (e.g. skin friction drag) can only extract roughly 59 % of the power available in the wind stream.[9] To be able to achieve a continuous power extraction flow, it is evident that a continuous mass flow of air must pass through the propeller/turbine disc. For this to occur, both the incoming and outgoing flow must have a positive flow velocity. If, hypothetically, the turbine was able to extract all of the available energy from the incoming flow, the flow past the disc area should have a velocity of zero (otherwise there would still be unextracted energy present). Having a zero fluid flow velocity at the exit of the turbine, directly means that no mass flow can be present, so no power can be extracted at these conditions.

Using the continuity equation, Euler's theorem and kinetic energy equations the following ideal power limit following the Betz law can be derived [9]:

$$P_{ideal} = \frac{16}{27}\frac{1}{2}\rho S_{turb} V_{air}^3 \tag{11}$$

This first estimate for the maximum theoretical power can be used as a basis to generate the power contours for the wind field. The following assumptions have to be kept in mind though:

- The wind turbine is assumed to not have a hub, the entire disc area region only contains blades
- It has an infinite number of blades that cause no additional drag (e.g. skin friction drag, induced drag due to tip vortices)
- The incoming flow is assumed to be constant, laminar and axial to the wind turbine axis
- No swirl is generated, the outgoing flow is also flowing axial to the wind turbine axis
- The air is considered to be an incompressible fluid

### 3.2. *Using Betz law to generate potential power contours*

Figure 2 shows the ideal maximum power at every location in the wind field that could be extracted from a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a wind-turbine with a rotor disc area of 0.1 m$^2$. It basically represents the absolute ideal maximum power that a regenerating UAV could achieve at every point in the wind field if static hovering can be achieved at that point and if the turbine can operate at its maximum power operating point, which will obviously not be the case for the majority of the wind field.

It is logical that the highest ideal power estimates are located directly above the hill since this is where the wind speeds are the highest (for the idealised potential flow case). It can be seen however that close to the surface of the hill the power figure is lower since this region has a lower velocity due to the added boundary layer wall function.

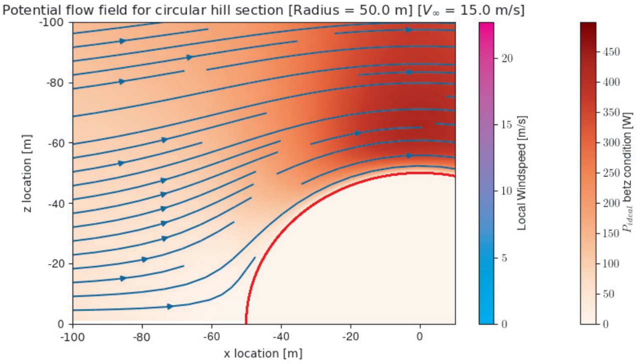


Fig. 2: Ideal power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.1 m$^2$

## 4. Determining hovering locations & power generation potential

Now that the absolute maximum theoretical power that can be extracted at each point in the wind field is known, the next step is to determine if the UAV can actually hover at that location, and if so, what power fraction should be extracted from the turbine to generate the required drag equalising the "thrust" generated by gravity, to enable the hovering to be stable?

To be able to answer this question, the equations governing the longitudinal flight dynamics of a hovering UAV needs to be studied.

### 4.1. *Longitudinal hovering flight dynamics*

The following equations (Equations 12, 13 and 14) express the system of differential equations for longitudinal flight dynamics (following from the Free-Body Diagram (FBD) given in Figure 3), including a non-zero wind, in the air-path reference system, as stated in the avian inspired energy harvesting paper [10]:

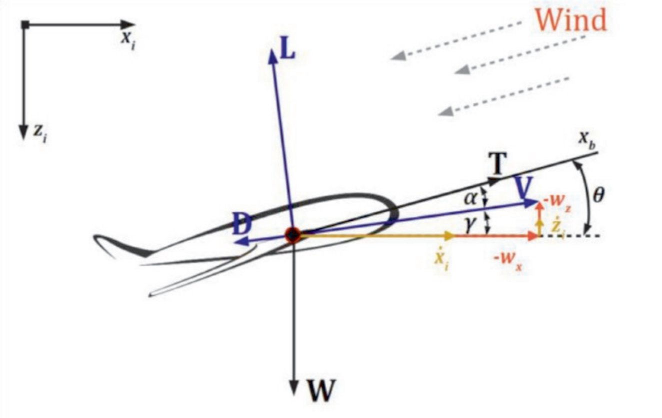


Fig. 3: FBD Air path reference system longitudinal flight dynamics (courtesy of [10])

$$T - D - W\sin\gamma = \frac{W}{g}\left(\dot{V}_{air} + \dot{u}_x\cos\gamma - \dot{u}_z\sin\gamma\right) \tag{12}$$

$$-L + W\cos\gamma = \frac{W}{g}\left(-\dot{V}_{air}\dot{\gamma} + \dot{u}_x\sin\gamma + \dot{u}_z\cos\gamma\right) \tag{13}$$

$$M = \ddot{\theta}I_{yy} \tag{14}$$

The equilibrium equations governing the balance of forces required for a UAV to hover in a steady state can be easily derived by setting the time derivative of the airspeed and both wind speed components (horizontal and vertical) to zero. The thrust force is also replaced with a (negative) turbine drag force which will represent the additional variable drag generated by propeller/motor drivetrain that can act as a turbine. To avoid possible confusions between the total drag force (encompassing both the aircraft and turbine drag forces) and the drag force purely generated due to the aerodynamic properties of the aircraft, the symbol $D$ which represented the latter was replaced by $D_{AC}$. Lastly, it is assumed that all of the forces acting on the aircraft are acting at the CG, meaning no moments are generated. This results in the following system of equations:

$$\begin{cases} -D_{turb} - D_{AC} - W\sin\gamma & = 0 \\ -L + W\cos\gamma & = 0 \end{cases} \tag{15}$$

An altered FBD which reflects the changes and simplifications made is shown in Figure 4.

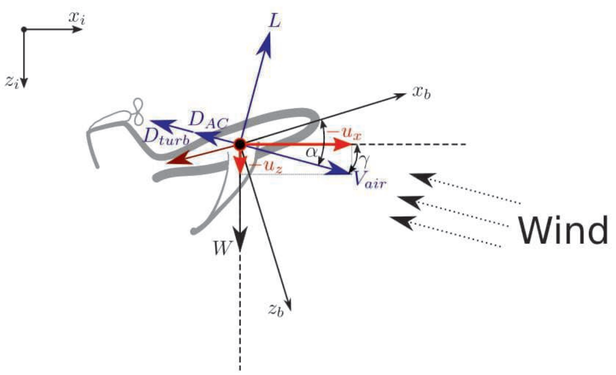


Fig. 4: FBD Air path reference system longitudinal hovering flight dynamics

### 4.2. *Estimating turbine drag*

Assuming that the turbine behaves as an ideal wind turbine as discussed in Subsection 3.1, it can be assumed that the wind only exerts a net axial force on the rotor. This means the useful power that the wind turbine extracts can be written as the product of this axial force ($D_{turb}$) and the air velocity at the rotor disc/turbine ($V_{turb}$): $P_{turb} = D_{turb} \cdot V_{turb}$. Furthermore, when the rotor is operating at the theoretical maximum efficiency conditions Betz proved that the air velocity at the rotor disc/turbine must be equal to two thirds of the incoming air velocity.[9] Using these equations and observations, it is possible to derive a simple expression for the estimated drag produced by an ideal turbine which is shown below (see Figure 5 for diagram with variables):

$$P_{ideal} = \frac{16}{27}\frac{1}{2}\rho S_{turb} V_{air}^3 \tag{16}$$

$$D_{turb} = \frac{P_{ideal}}{V_{turb}} \tag{17}$$

$$V_{turb} = \frac{2}{3} V_{air} \tag{18}$$

Substituting Equation 18 in Equation 17:

$$D_{turb} = \frac{P_{ideal}}{\frac{2}{3} V_{air}} \tag{19}$$

Finally. substituting Equation 19 in Equation 16 results in an equation expressing the estimated turbine drag ($D_{turb}$) in terms of incoming airspeed ($V_{air}$) and rotor disc area ($S_{turb}$):

$$\boxed{D_{turb} = \frac{1}{2}\frac{2}{9}\rho S_{turb} V_{air}^2} \tag{20}$$

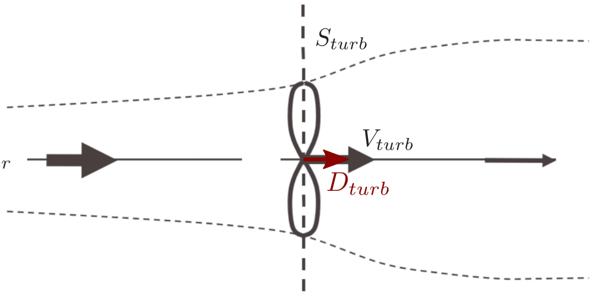


Fig. 5: Ideal wind turbine diagram

### 4.3. *Finding the required lift and drag coefficients for hovering*

Now that both the systems of equations describing the force equilibrium during hovering flight and an estimate for the turbine drag are found, it is possible to derive a set of equations that determine the required lift and drag coefficients to enable static hovering.

Following from the system of equations that describes the force equilibrium during hovering flight derived in Subsection 4.1 (Equation 15), the required lift and drag terms can be expressed as follows:

$$\begin{cases} L & = W\cos\gamma \\ D_{turb} + D_{AC} & = -W\sin\gamma \end{cases} \tag{21}$$

Rewriting this system of equations in terms of the lift and drag coefficients, substituting $D_{turb}$ with Equation 20 and dividing both sides by $\frac{1}{2}\rho V_{air}^2 S$ results in the following system of equations:

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\cos\gamma \\ \frac{2}{9}\frac{S_{turb}}{S} + C_{D,AC_{hover}} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\sin\gamma \end{cases} \tag{22}$$

The resulting non-dimensionalised contribution of the turbine to balance the horizontal force equilibrium (the bottom row of Equation 22), $\frac{2}{9}\frac{S_{turb}}{S}$, can be thought of being the maximum achievable drag coefficient of the turbine, since multiplying this figure by $\frac{1}{2}\rho V_{air}^2 S$ results in the ideal maximum drag caused by the turbine. Setting $C_{D_{turb}} = \frac{2}{9}\frac{S_{turb}}{S}$ results in the following system of equations:

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\cos\gamma \\ C_{D_{turb}} + C_{D,AC_{hover}} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\sin\gamma \end{cases} \tag{23}$$

Next, the sine and cosine of the flight path angle ($\gamma$) can be substituted with the fractions $\frac{u_z}{V_{air}}$ and $\frac{u_x}{V_{air}}$ respectively. This can be done because the velocity of the UAV with respect to the inertial reference frame is assumed to be zero during stable hovering. This means that the airspeed vectors magnitude and direction is purely determined by the local wind speed vectors (see Figure 4).

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\frac{u_x}{V_{air}} \\ C_{D_{turb}} + C_{D,AC_{hover}} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\frac{u_z}{V_{air}} \end{cases} \tag{24}$$

Finally, if the lift-drag polar can be estimated using the following standard equation relating the drag and lift coefficient to each other:

$$C_{D,AC} = C_{D_0} + \frac{C_L^2}{\pi Ae} \tag{25}$$

And substituting this equation in Equation 24:

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\frac{u_x}{V_{air}} \\ C_{D_{turb}} + C_{D_0} + \frac{C_{L_{hover}}^2}{\pi Ae} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\frac{u_z}{V_{air}} \end{cases} \tag{26}$$

This leaves a system of equations that can easily be solved for both the required lift coefficient ($C_{L_{hover}}$), and turbine drag coefficient $C_{D_{turb}}$ if the local air speed (which is equal to the wind speed magnitude during hovering), horizontal and vertical wind speed components are known.

Some important observations can be made from the final equations:

- The required lift coefficient, determined by the first part of Equation 26, should be less than the maximum lift coefficient of the aircraft. If this would not be the case, the aircraft would effectively stall when trying to achieve these conditions.
- The drag that the clean aircraft itself can provide is fixed by the required operating point on the lift-drag polar. If the required drag coefficient is lower than this value, the aircraft will not be able to achieve hovering equilibrium, even if the turbine is fully switched off or assumed to not be present;
- At specific wind speed and direction conditions, the clean aircraft will be able to provide just the right amount of drag at a certain required lift coefficient to satisfy both equilibrium equations, the turbine doesn't need to be switched on, and no power can be regenerated, since $C_{D_{turb}}$ will have to be equal to 0.
- At wind conditions where more drag is required than the clean aircraft itself can provide, the turbine needs to be switched on to close the "drag deficit" and equalise both terms of the second part of Equation 26. If the required extra drag from the turbine is less than its ideal maximum, the regen drivetrain should regulate the drawn power from the turbine in such a way that the drag provided by the turbine satisfies the equations.
- There exist another specific set of wind conditions where the required drag from the turbine to achieve hovering equilibrium will be equal to the maximum drag that the turbine ideally can provide. Note that although the maximum amount of power (imposed by the Betz limit) that can be drawn from the turbine in this scenario at the specific conditions, it is not necessarily the optimum resulting in the maximum amount of regeneration power, since the regeneration power also depends on the wind speed and other locations in the wind-field might exist where not all ideally available power can be extracted, but due to a higher wind speed the total regenerated power potential is still higher.

With the finalised equations for the turbine drag coefficient and above observations in mind, the calculation of the regen power contours can now be performed.

### 4.4. *Regen power contour calculation*

The finalised equations presented in the previous subsection enable one to determine if static hovering is achievable (given the local wind conditions at a certain point in the wind-field and aircraft parameters). If this is the case, the corresponding static hovering power regeneration potential can be calculated for that point.

The resulting equations can be used to determine both the required lift coefficient ($C_{L_{hover}}$), and combined drag coefficients (one being the turbine drag coefficient $C_{D_{turb}}$, the other being the drag coefficient of the aircraft $C_{D,AC_{hover}}$) to enable stable static hovering.

This function determines if the UAV is theoretically able to statically hover with zero ground speed at each point of the calculated wind field. At each potential hover location, the required additional drag and power needed from the turbine is calculated as well as the angle of attack.

First, the required lift coefficient to satisfy the hovering equilibrium equations is calculated:

$$C_{L_{hover}} = \frac{W}{0.5 \cdot \rho \cdot V_{air}^2 \cdot S} \cdot \frac{u_x}{V_{air}} \tag{27}$$

If the resulting lift coefficient is larger than the maximum achievable lift coefficient ($C_{L_{max}}$), the aircraft would stall if it tried to approach the conditions required for hovering and the corresponding point in the wind field will have a zero power regeneration potential using static hovering since hovering cannot be achieved.

Next, the total required drag coefficient to enable hovering ($C_{D_{turb}} + C_{D,AC_{hover}}$) is calculated:

$$C_{D_{turb}} + C_{D,AC_{hover}} = \frac{W}{0.5 \cdot \rho \cdot V_{air}^2 \cdot S} \cdot \frac{u_z}{V_{air}} \tag{28}$$

For the aircraft to be able to achieve static hovering, the combined required drag coefficient can not be smaller than the minimum achievable total drag coefficient. This minimum achievable total drag coefficient is equal to the clean aircraft's drag coefficient, since the least amount of drag will be generated when no additional turbine drag is generated (hence $C_{D_{min}} = C_{D,AC_{hover}} = C_{D_0} + \frac{C_{L_{hover}}^2}{\pi Ae}$).

The combined required drag coefficient can also not be larger than the maximum achievable drag coefficient, which is equal to the clean aircraft's drag coefficient plus the maximum achievable turbine drag coefficient. As stated in the previous subsection, the maximum achievable turbine drag coefficient can be estimated using the Betz limit and is equal to $C_{D_{turb,max}} = \frac{2}{9}\frac{S_{turb}}{S}$. Summarising, the acceptable combined required drag coefficient bounds to achieve static hovering leads to the following expression:

$$C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae} \leq C_{D_{turb}} + C_{D,AC_{hover}} \leq C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae} + \frac{2}{9}\frac{S_{turb}}{S} \tag{29}$$

If the total required drag coefficient falls within these bounds and the required lift coefficient is not larger than the maximum lift coefficient (as stated earlier), it can be assumed that the aircraft can achieve static hovering, and a valid power regeneration potential can be calculated.

The resulting required turbine drag coefficient to achieve stable static hovering can be calculated as follows:

$$C_{D_{turb,hov}} = C_{D_{turb}} + C_{D,AC_{hover}} - \left(C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae}\right) \tag{30}$$

The corresponding turbine drag generated during hovering can easily be found by multiplying the turbine drag coefficient with $0.5 \cdot \rho \cdot V^2_{air} \cdot S$:

$$D_{turb,hov} = 0.5 \cdot \rho \cdot V^2_{air} \cdot S \cdot C_{D_{turb,hov}} \tag{31}$$

Finally, by rearranging Equation 19 the estimated turbine power can be found:

$$P_{turb.hov} = \frac{2}{3} \cdot V_{air} \cdot D_{turb,hov} \tag{32}$$

### 4.5. *Results*

By incorporating the finalised turbine drag and power equations and only populating the values for locations where hovering is deemed feasible by satisfying the maximum lift coefficient constraint and conditions set in Equation 29, power contour plots can be generated for any given wind-field. This results in figures like the one shown below (Figure 6):

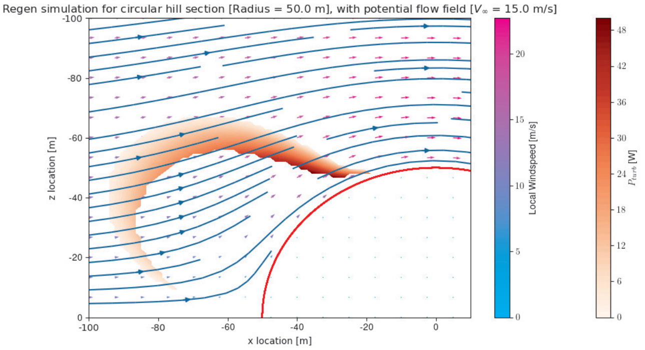


Fig. 6: Regen power contour plot for a $15\,\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of $50\,\mathrm{m}$ for a rotor disc area of $0.1\,\mathrm{m}^2$ using aerodynamic parameters in Appendix A

It can immediately be seen that the estimated maximum amount of power that can be regenerated using the turbine while hovering is roughly 1 order of magnitude lower than the ideal Betz limit power contour graph of the entire wind-field (see Figure 2). The primary reason for this is that the UAV is unable to statically hover with these conditions at the point in the wind-field that has the maximum potential power, which is the point with the highest wind velocity.

Power contour plots were calculated for a range of conditions, such as different wind-speeds, hill-sizes, rotor disc areas, UAV masses, etc. The resulting plots showed the expected behaviour for the change in conditions.

## 5. Regenerative drivetrain

As was briefly touched upon in the previous section, a special kind of electric drivetrain is required which enables the power flow to be reversed in certain conditions. In normal conditions the battery provides the power to the motor to propel the vehicle, but in the regen mode this power flow is reversed and the motor now acting as a generator can be used as a wind-turbine to supply power back to the battery.

### 5.1. *Shortcomings of existing setups from prior research*

Most of the previous research into this area focused on determining if this method was suitable to use in light electric road vehicles (e.g. e-scooters) to increase the total driving range of a typical ride that includes sporadic braking. Apart from relative increases in driving ranges, no actual performance data on this setup could be found. It was also identified that in most experiments lead-acid batteries were used which are unsuitable for UAV due to their low energy mass density of around 35-40 Wh/kg (compared to 150-180 Wh/kg for LiPo batteries as described in [11]).

In order to be able to evaluate the performance of using this kind of combined regen controller for the proposed use case in a UAV with LiPo batteries, additional research and tests will be required. Instead of trying to model and develop a prototype regen controller with the required functionality from scratch however, it was decided to first perform a comprehensive search if a possible suitable existing Electronic Speed Controller (ESC) could be found that can be easily reprogrammed and altered to achieve the desired functionality. Once a suitable candidate is found, it can be used to perform tests to try to estimate the efficiency of the regen drivetrain.

### 5.2. *VESC ESC as combined ESC/regen controller*

After a comprehensive search a possible suitable existing ESC controller was found that even has built-in functionality to be used as a regen controller: the VESC.

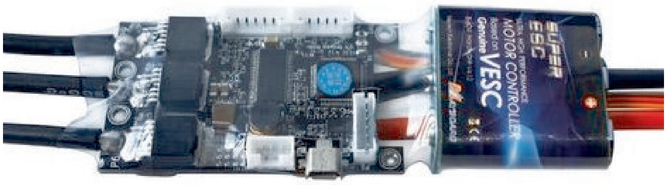


Fig. 7: VESC ESC (courtesy of *electricboardsolutions.com* )

The VESC (depicted in Figure 7) is a fully open-source ESC originally designed to be used in electric skateboards. Over time, it has been further developed to be used in a variety of applications and includes additional features such as datalogging, added control interfaces (such as CAN bus & UART), and (most importantly) regen capability.

The VESC was found to be ideally suited as an all-in-one combined regen controller and ESC for UAV's, due to it's compatibility with recharging LiPo batteries in the regen mode. Because of the open-source nature of the design it is possible to easily alter the firmware and do hardware modifications.

In order to test the performance and power efficiency of the regenerative drivetrain, a test-setup with the VESC will be used that enables the input and output power to be measured while performing the regen function.

### 5.3. *Test setup description*

The general test-setup diagram is shown in Figure 8. On the left side, a brushless motor is connected to an ESC that drives this motor. The power input of the ESC is connected to a lab bench power supply such that the input voltage can be varied and the output current monitored. The throttle/power setting of the ESC can be adjusted by moving the sticks on a transmitter connected to the receiver (RX) that sends the ESC the required motor throttle signal. This setup basically emulates a propeller providing mechanical power to the shaft of the mechanically coupled brushless motor on the right.

The two brushless motors are connected together with a mechanical shaft coupler. The second brushless motor that is being driven by the first one is connected to the VESC that can be reprogrammed with a custom firmware that employs the regenerative control strategy. This second motor is mounted on a motor benchmarking device called RC benchmark that measures the torque on the motor shaft. The VESC's output is connected to a rechargeable LiPo battery through an ammeter such that the recharging current can be monitored. The VESC has built-in current, voltage and RPM sensors to monitor the brushless motor RPM and braking current, and battery voltage. All of these parameters are logged on a pc by using an interfacing adapter based on an STM32 microcontroller. The desired braking current can be adjusted using a rotary potentiometer on the transmitter that is mapped to a channel and also sent to the RX. The STM32 does the necessary conversion of the PPM signal to the serial commands required by the VESC to set a custom braking current for the regen functionality. Finally, a digital oscilloscope is also connected to the two gate drives of the switching MOSFETs of one of the motor phases to allow for an analysis to be carried out of the switching behaviour of the VESC when operating in the regen mode. The built-up test setup in the lab is shown below in Figure 9.

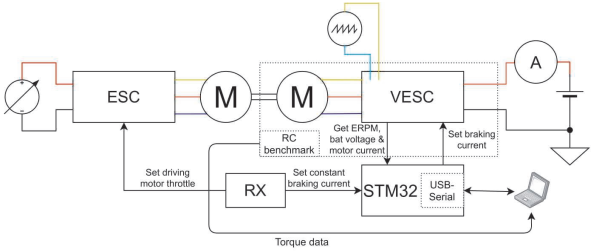

A
ESC
M
M
VESC
RC benchmark
Get ERPM, bat voltage & motor current
Set braking current
Set driving motor throttle
RX
Set constant braking current
STM32
USB-Serial
Torque data

Fig. 8: Regenerative drivetrain test-setup diagram

The specifications of the used equipment can be found in Table 1.

### 5.4. *Expected results and outcome*

Various tests will be conducted where the regenerating brushless motor will be subjected to a range of RPMs, and requested regen current & power.

The following parameters will be logged:

- Braking current ($I_{brake}$)
- Motor RPM (RPM)
- Battery regen current ($I_{bat}$)
- Battery voltage ($V_{bat}$)
- Motor torque ($T$)
- Driving motor/ESC voltage ($V_{supply}$)
- Driving motor/ESC current ($I_{supply}$)

With all of these parameters logged, it is possible to

| | |
|---|---|
| Lab bench power supply | AFX-9660SB 0-30V 0-3A |
| Driving motor | Turnigy Aerodrive D2826/10 1400 KV |
| Driving ESC | YEP 40 A |
| Driven motor | EMAX GT2215/10 1100 KV |
| regen ESC | Maytech MTVESC50A VESC compatible |
| Torque measuring device | RC Benchmark Series 1580 |
| LiPo battery | 3D Robotics 3s 9C 5000 mAh |

Table 1: Used equipment specifications

determine the approximate total drivetrain efficiency from the measurements to have a better estimate available of the practically achievable regen drivetrain efficiency, encompassing both the BLDC motor acting as a generator and the regen ESC. This can be done by comparing the mechanical power present at the coupled motor shaft to the electrical charging power of the battery. The mechanical and electrical power figures can easily be calculated from the measured parameters using the following formulae:

$$P_{mech} = T \cdot 2\pi \frac{RPM}{60} \tag{33}$$

$$P_{bat} = V_{bat} \cdot I_{bat} \tag{34}$$

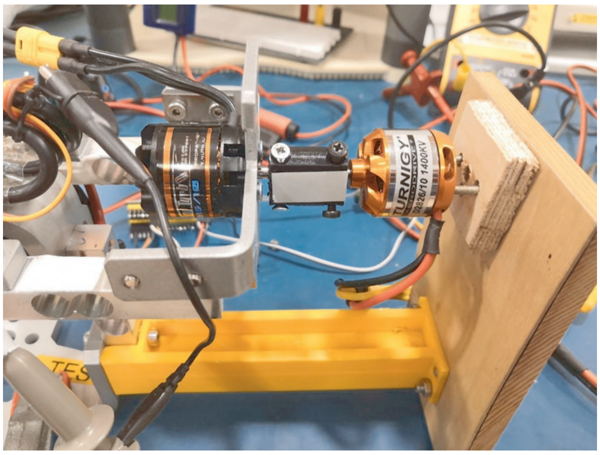


Fig. 9: regen drivetrain test-setup, with driving motor on the right and driven motor on the left, mounted on the RC Benchmark torque measuring device

### 5.5. *Observations during the test*

The main tests that were conducted involved determining the efficiency of the regen drivetrain for a range of RPMs and mechanical power. For each run, the driving motor ESC was set to keep the BLDC running at a constant RPM using the driving ESC's governer mode. For each constant RPM the requested braking current was gently increased, which lead to both the battery recharging power and the mechanical power on the motor shaft to increase. It was unfortunately not possible to choose constant measuring points for all RPMs with the same braking current or recharging power due to induced instabilities in the system probably caused by a conflict of the braking current control-loop of the VESC and the RPM governor of the driving ESC at certain operating conditions. Instead, it was opted to try to approach similar measuring points that resulted in stable readings to hopefully increase the accuracy of the measured data.

Another anomaly that was first observed was that the battery discharge current reported by the VESC didn't correspond with the external ammeter. After performing some additional measurements and checks it was found that the VESC had a near constant current reading error of around 0.25 A - 0.3 A when the brake current is set high enough to actually start the regen mode, which can be clearly seen in the graphed out measurements shown in Figure 10. The rest of the reported parameters logged by the VESC (motor RPM, battery voltage and motor current) were also able to be checked with a multimeter and did show good correspondence to the measured data.

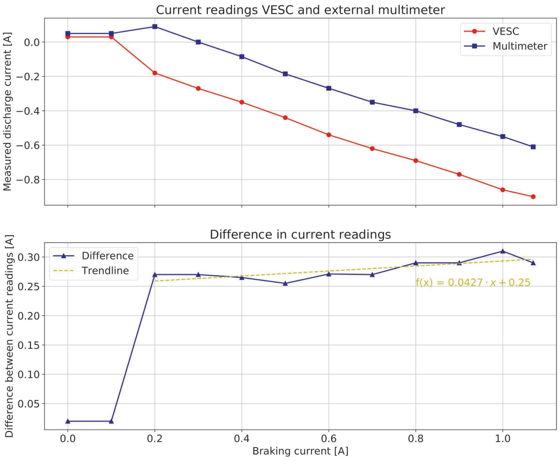
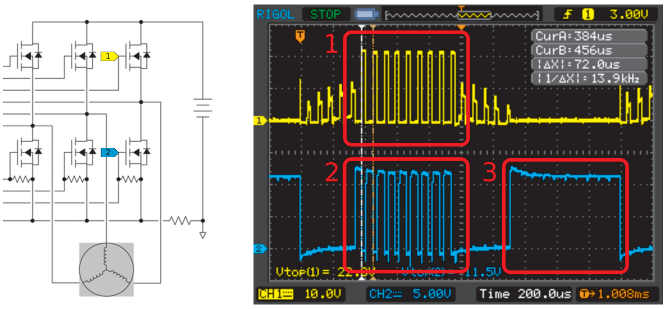


Fig. 10: VESC current measurement discrepancy

During the test the gate drive signals of the VESC of both the high- and low-side switching MOSFETs of one of the motor phases were probed to investigate the exact switching behaviour of the regen mode. Instead of using (body-)diodes to perform the diode functions, the VESC uses active rectification. This means it uses the MOSFETs as semi-ideal diodes. This complicates the switching pattern and control logic, but can increase the overall efficiency by virtually eliminating the voltage drop due to the low on-resistance characteristics of power MOSFETs reducing the power dissipation significantly when compared to a regular diode.[12] An additional drawback of the active rectification approach is that a small amount of power is required every time the switching element has to change state, which at high frequencies and certain conditions may offset the power dissipation savings. The observed switching behaviour, together with the probing locations, is shown in Figure 11. Three distinct switching regimes can be observed in Figure 11b:

(1) Active rectification, allowing current flow to the battery when higher voltage is being generated by the inductors
(2) Switching element of boost converter
(3) Connecting phase to ground, making sure there is a complete current path



(a) Probing locations channel 1 & 2

(b) Oscilloscope screenshot of MOSFET gate switching pattern

Fig. 11: Switching behaviour of high- and low-level MOSFETs of motor phase during regen operation of VESC

### 5.6. *Results*

With all of the necessary data gathered and processed, the measurements at the different RPM ranges were grouped together and it's efficiency calculated using the formulae described in the test setup expected results and outcomes subsection (Subsection 5.4). The result of this for the first measurement series performed with a constant RPM of around 5000 is shown in Table 2.

The efficiency results of the drivetrain for the various RPM runs are plotted in Figure 12.

As was expected, the efficiency of the regen drivetrain varies considerably for different power levels & RPMs. This can be explained by the fact that the total efficiency figure is a combination of the efficiency of the BLDC motor to convert the mechanical shaft power to electrical power, and the electrical efficiency of the regen controller to convert the complex 3-phase lower AC-voltage to a higher DC-voltage suitable to safely charge the LiPo battery.

It can be clearly seen that the efficiency is very low at lower power levels. At very low power levels all of the regenerated power is used to supply the VESC itself, leaving little or no additional power for the battery to be recharged. This explains the zero efficiency number at low power values, since all of the recuperated energy is used to drive the regen controller itself, which in some cases is not even sufficient requiring still some power from the battery to drive the VESC.

The performance of the regen drivetrain generally increases with increasing mechanical input power levels. The very rapid increase in performance at the beginning of the curves can be partly explained by the phenomenon that was just described, but there are probably also other factors at play. At very low generator currents and power levels (BLDC) motors have a very low efficiency, which rapidly increases when the generator current only rises slightly.[13] The efficiency gains for higher input powers do taper off at a certain point. The optimum efficiency point will depend on both the characteristics of the BLDC motor being used as the generator, as well as the performance characteristics of the VESC switching hardware.

The final observation that can be made is that overall higher RPMs do seem to result in slightly higher efficiency figures. This can partly be explained by the fact that higher RPMs result in a higher input voltage to the VESC regen controller, which in turn has to boost the input voltage by a lower fraction to still achieve a high enough output voltage to charge the batteries. This means that for it to achieve the same charging power as with a lower RPM, less input current is required which leads to less heat losses due to resistance present in the windings, switching elements, etc.

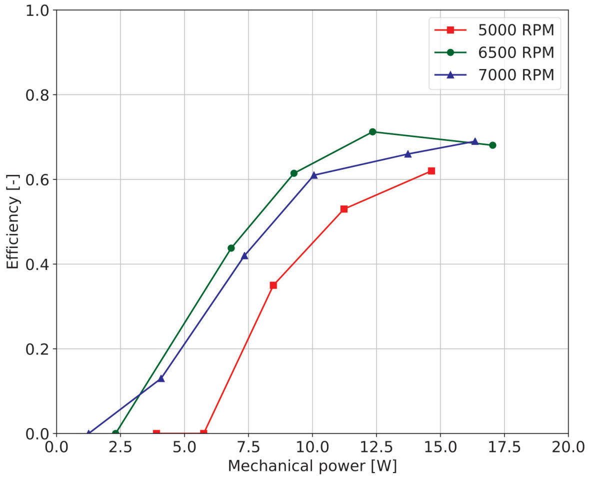


Fig. 12: Efficiency figures calculated from VESC test runs for different RPMs

## 6. Conclusion

A simplified wind-field model around obstacles such as circular and oval shaped hills was constructed based on potential flow theory. A model was developed to determine the

| # | Time stamp | $I_{supply}$[A] | $V_{supply}$ [V] | $P_{supply}$ [W] | $I_{bat}$ VESC [A] | $-I_{bat}$ (multi) [A] | Torque [N*m] |
|---|---|---|---|---|---|---|---|
| 1 | 13/11/2020 17:23 | 0.72 | 8.9 | 5.7 | 0.01 | $-0.049$ | -0.0075 |
| 2 | 13/11/2020 17:24 | 0.99 | 9.0 | 8.9 | $-0.23$ | $-0.007$ | -0.011 |
| 3 | 13/11/2020 17:25 | 1.38 | 8.9 | 13.6 | $-0.47$ | 0.248 | -0.0162 |
| 4 | 13/11/2020 17:26 | 1.82 | 8.9 | 18.4 | $-0.73$ | 0.491 | -0.0215 |
| 5 | 13/11/2020 17:27 | 2.30 | 8.9 | 24.5 | $-0.98$ | 0.750 | -0.028 |

| # | ERPM | $V_{bat}$ [V] | $I_{brake}$ [A] | RPM | $P_{mech}$ [W] | $P_{bat}$ [W] | $\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | -34805 | 12.0 | 0.00 | 4972 | 3.905 | $-0.588$ | 0% |
| 2 | -34935 | 12.0 | 0.60 | 4991 | 5.749 | $-0.084$ | 0% |
| 3 | -34957 | 12.1 | 1.25 | 4994 | 8.472 | 3.001 | 35% |
| 4 | -34918 | 12.2 | 1.95 | 4988 | 11.23 | 5.990 | 53% |
| 5 | -34976 | 12.2 | 2.69 | 4997 | 14.65 | 9.150 | 62% |

Table 2: Measured & calculated data for 5000 RPM run

maximum theoretical regeneration power if wind-hovering orographic soaring techniques are applied, given a certain wind-field and aerodynamic characteristics of the UAV. The resulting modular simulator program is able to determine the hovering locations and gives an estimate of the maximum achievable regeneration power. For the source wind-field, either the simplified potential-flow based model can be used, which needs very little computational power making it suitable to be even run on on-board processors of UAV's, or a wind-field generated by other more advanced software or even from a measurement field. The tool should allow anyone to easily get an estimate of the feasibility of the regenerative hovering soaring method in their particular application.

Furthermore, a working regenerative drivetrain test-setup that could be incorporated in a small electric UAV was tested and proved that with commercial source-able components a regenerative electric drivetrain can be constructed that has a mechanical input power to recharging power efficiency of up to 70 % at it's optimum operating conditions.

For future work, additional simulations could be carried out by simulating real-life conditions. The model can then be validated by performing a flight-test in these conditions with the devised regenerative drivetrain setup.

## Acknowledgments

We would like to thank our colleagues from the Never-Landing Drone working group for their continued support, suggestions, ideas and feedback.

### Appendix A UAV aerodynamic parameters used for simulation

| | | |
|---|---|---|
| $S$ | 1 | $\mathrm{m}^2$ |
| $C_{L_\alpha}$ | 5.7 | $\mathrm{rad}^{-1}$ |
| $\alpha_{0L}$ | -4 | ° |
| $A$ | 6 | - |
| $e$ | 0.8 | - |
| $C_{D_0}$ | 0.05 | - |

Table A.1: UAV aerodynamic parameters for simulation

### References

[1] N.R.J Lawrance and J.J. Acevedo and J.J Chung and J.L. Nguyen and D. Wilson and S. Sukkarieh, Long Endurance Autonomous Flight for Unmanned Aerial Vehicles, *AerospaceLab Journal.* **8** (2014).

[2] N.R.J Lawrance and S. Sukkarieh, Long Wind energy based path planning for a small gliding unmanned aerial vehicle, in *AIAA Guidance, Navigation, and Control Conference and Exhibit,* (Illinois, Chicago, 2009).

[3] R. Carvalho, Development of the regenerative soarer: Theoretical and Practical aspects, in *Congress of the International Council of the Aeronautical Sciences (ICAS),* (Minas Gerais, Belo Horizonte, 2018).

[4] P. MacCready, Regenerative battery-augmented soaring in *Self-Launching Sailplane Symposium,* (New-York, Elmira, 1998).

[5] A. Fisher and M. Marino and R. Clothier and S. Watkins and L. Peters and J.L. Palmer, Emulating avian orographic soaring with a small autonomous glider, *Bioinspiration and Biomimetics.* **11**(1) (2015).

[6] J.W. Langelaan, Long distance/duration trajectory optimization for small UAV's, in *AIAA Guidance, Navigation, and Control Conference,* (South Carolina, Hitlon Head, 2007).

[7] P.K. Kundu and I.M. Cohen, Chapter 6.4 in *Fluid Mechanics* (Academic Press, San Diego, California, 2002)

[8] S. Besio and A. Mazzino and C.F. Ratto, Long Endurance Autonomous Flight for Unmanned Aerial Vehicles, *Boundary-Layer Meteorology.* **107**(1) (2003).

[9] M. Ragheb and M. Adam, Wind Turbines Theory - The Betz Equation and Optimal Rotor Tip Speed Ratio in *Fundamental and Advanced Topics in Wind Power* (Intech, 2011), pp. 21.

[10] N. Gavrilovic and A. Mohamed and M. Marino and S. Watkins and J.M. Moschetta and E. Benard, Avian-inspired energy-harvesting from atmospheric phenom-

ena for small UAV's *Bioinspiration and Biomimetics.* **14**(1) (2019).

[11] G.J. May and A. Davidson and B. Monahov, Lead batteries for utility energy storage: A review, *Journal of Energy Storage* **15** (2018) pp. 145–157

[12] Texas Instruments, *Basics of Ideal Diodes* (TI, 2019), pp.1–22

[13] U. Kafader, *Maxon Motors as Generators* (Maxon group, 2020)

# 3 Research Questions, Objectives and Sub-goals

The following two sections will cover the research questions and the research objective respectively.

## 3.1. Research Question(s)

The main research question for the thesis project can be stated as follows:

> "Can a UAV harvest power from the environment by employing regenerative static hovering soaring techniques around obstacles such as hills and ships?"

In order to find an answer to this question the project can be split-up into two major parts: one focusing on the amount of atmospheric power/energy available in the wind-field around an obstacle, and the other focusing on how much of the atmospheric power available from the wind-field around the obstacle can feasibly be extracted using a regenerative drivetrain. Hence the main research question can be split-up into the following 2 sub-questions:

1. What is the relation between the wind conditions and obstacle geometry to the amount of power available for energy harvesting in the wind-field around that obstacle?
2. How much power can feasibly be extracted from the wind field around an obstacle with a regenerative drivetrain using static hovering soaring techniques?

## 3.2. Research Objective

The main research objective of this thesis is to:

> “Achieve unlimited endurance for a UAV by using a combination of static orographic soaring with a regenerative drivetrain”

The main objective can be subdivided into the following sub goals:

- "Gain a good understanding and develop a simple model of the behaviour of a wind field around obstacles that are prime candidates for static soaring such as hills, dunes and ships."

  This objective is vital to be able to determine the energy harvesting potential of the wind-field. A good knowledge about the behaviour is necessary to be able to draw conclusions about how much power generating potential there is in the wind-field and to find the location of the optimum power points.

  The next objective follows naturally from the previous one, and is stated as follows:

- "Develop a model that predicts the amount of atmospheric power that can be harvested by a UAV using static wind hovering for a given flow field around an object."

In order to be able to answer the main research question if it is possible to achieve unlimited endurance by harvesting energy from the wind field, it is vital to know how much power is available from the wind field. It is this power balance between harvested available power and required power to fly that will determine the final outcome.

However, since some of the power will be lost during the necessary conversion of voltages that takes place in the regenerative drivetrain controller, the efficiency of the drivetrain should be determined before a final conclusion can be drawn if the final power balance checks out and unlimited endurance can be achieved. Hence the next objective is to:

- "Determine the feasible efficiency of the regenerative drivetrain for various power levels and operating conditions, to determine the feasible amount of power that can be used to recharge to the on-board battery."

  The ultimate proof that the suggested regenerative static soaring method is actually able to extend a UAVs endurance to possibly infinity is of course to build a proof-of-concept prototype. This (ambitious) optional final objective can be stated as follows:

- "Create a proof-of-concept regenerative static soaring UAV that is able to show the scientific community that regenerative static wind hovering is a feasible technique to extend the endurance of UAVs."

# 4

# Soaring & regenerative flying techniques

Soaring can be best described as applying various flight techniques to harvest energy from the atmosphere to stay in the air as long as possible. It allows birds and glider aircraft to cover large distances without the need for a propulsive power source, such as having to flap their wings or have an on-board engine with a propeller. Even when adding regenerative drivetrains, rechargeable batteries and other technologies into the mix the same core principles that make it possible for vultures to soar for hours without flapping their wings once will apply to the proposed novel regenerative UAV designs. For this reason it is of vital importance to first have a firm grasp on the basic soaring techniques and principles, which can then later be extended to include hovering and regenerative techniques.

## 4.1. Soaring techniques overview

All types of soaring can be divided into two main categories: static & dynamic soaring. The definitions of and differences between these types of soaring techniques are explained below.

### 4.1.1. Static soaring

With static soaring an aircraft can gain energy by making use of updrafts (currents of rising air resulting in a positive vertical wind component) that can be created by natural or artificial phenomena. Static soaring techniques depend on vertical movements of the atmosphere, which can be generated by thermal updrafts (caused by differences in temperature) or wind that gets deflected upwards due to natural or artificial obstructions [5]. If the vertical displacement velocity of the air is larger than the minimum sink speed of the aircraft, it becomes possible to use static soaring to gain altitude without losing airspeed (and vice versa). Static soaring can be further subdivided into thermal soaring and orographic soaring.

With **thermal soaring** energy gains are made solely by using the thermal updrafts, which are caused by the fact that air with a higher temperature has a lower mass density, which causes volumes of warmer air to rise. The temperature differences required to generate pockets of warmer air are caused by the uneven heating of various land surfaces [9].

**Orographic soaring** is the general term used for soaring where the upward deflection of the wind-stream is caused by natural or man-made objects. Examples include the updrafts produced on the windward side of hills and ridges (often called ridge or slope lift).

### 4.1.2. Dynamic soaring

The second general category of soaring techniques, **dynamic soaring**, uses varying wind gradients to extract energy from the wind-field. These varying wind gradients can be caused by different features and phenomena, such as the shear layer present on the leeward side of hills, the surface boundary layer caused by friction, or time-varying winds such as gusts [4]. The term "dynamic" is very applicable here, since in most of these wind-fields no net energy gains can be generated without performing some sort of cyclic manoeuvre that requires the UAV to constantly change the heading, airspeed and/or altitude. Since the main objective is to hover statically above an obstacle to extract energy the dynamic soaring method will not be discussed further in this thesis. However, this method might be an interesting starting point for further research to increase the endurance along the mission route in certain circumstances.

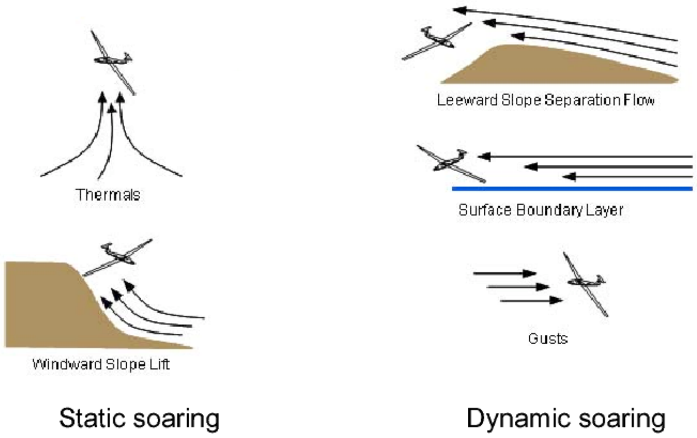


Figure 4.1: Overview of types of soaring (courtesy of [4])

## 4.2. Orographic soaring & wind hovering techniques

Using the principle of orographic soaring to extend the range and endurance of UAVs has already been extensively researched, often based on techniques used by various bird species that have been observed [13, 19, 20]. Achieving static hovering while using the orographic soaring method (called wind hovering) is a topic that was found to only be covered by a very small amount of research.

The following sections will focus on the discussion of the process and results presented in papers connected to orographic soaring and wind hovering, that were deemed useful for creating a better understanding on how regenerative wind hovering can be achieved and modelled.

### 4.2.1. Orographic soaring model for flying upwind off (idealised) hills

Langelaan described in his Trajectory Optimization for Small UAVs paper [18] how by using ridge/orographic soaring, which employs the large area of updrafts which are present upwind of a hill/mountain range, the range can be extended considerably. A model was created that is able to estimate the total energy gained, given the UAV's flight characteristics and current conditions. From this it is possible to determine the optimal flight path to maximise the energy gained and to estimate the remaining range of the UAV once it exits the area with updrafts. Using a simplified potential flow model to calculate the wind-field around a circular shaped hill, the local vertical updraft velocity around the hill can be estimated by Equation 4.1 [18]. It was found that the optimum soaring region, where the highest updraft velocities are present, is concentrated around the $\eta = 45$ degree radial from the hill on the upwind side (see Figure 4.2).

$$w_z = -2w_\infty \frac{R^2}{r^2} \cos\eta sin n\eta \tag{4.1}$$

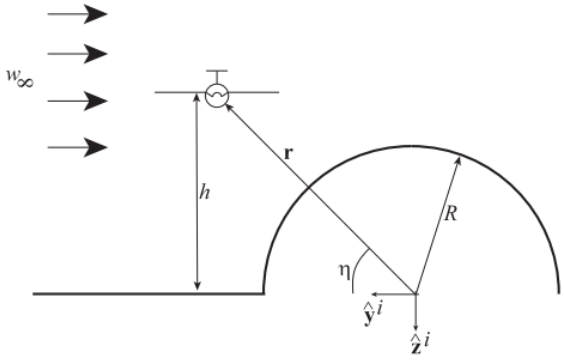


Figure 4.2: Plane soaring along circular shaped hill (courtesy of [18])

If the UAV is soaring along upwind of a ridge with favourable wind conditions that generates a large enough local vertical updraft velocity ($w_z$), that is higher than the minimum sink-speed of the aircraft ($\dot{h}_{min}$), the UAV will be able to maintain altitude and airspeed without the help of an engine. There are some possible caveats though, since the minimum sink-speed of an aircraft is not a constant, it varies with airspeed, aircraft mass and wing surface area (often combined in the form of wing loading). Practically, this means that the UAV might have to slow down or speed up first to be able to gain useful energy afterwards [12]. The sink-speed airspeed relationship is often called the glider speed polar. Figure 4.3 shows an example of this relationship for a UAV and a high-performance sailplane with varying wing loadings.

Langelaan deduced a relatively simple equation that can be used to estimate the minimum sink-speed (given the UAV's aerodynamic characteristics and airspeed) of the UAV ($\dot{h}_{min}$) [18]. When combined with the standard simplified drag polar model (see Equation 4.2), this results in a fairly simple relation shown in Equation 4.3.

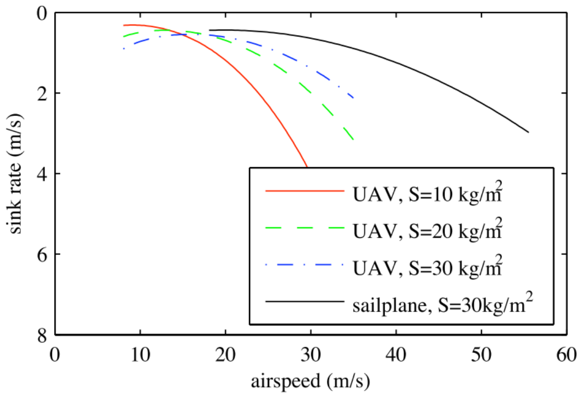


Figure 4.3: Example of glider speed polar (sink-rate vs airspeed) for small UAVs with varying wing loadings and a high-performance sailplane (courtesy of [18])

$$C_D = C_{D_0} + \frac{{C_L}^2}{\pi Ae} \tag{4.2}$$

$$\dot{h}_{min} = \frac{\rho S C_{D_0}}{2mg} \cdot {V_{air}}^3 + \frac{2mg}{\rho S V_{air} \pi Ae} \tag{4.3}$$

### 4.2.2. Wind hovering models and techniques

As stated earlier, far less research is available on the topic of statically soaring/hovering using updrafts, often called wind hovering. With wind hovering, a constant position is held relative to the ground. The position can be held indefinitely as long as a favourable updraft region is present in that location to keep the aircraft from losing energy. Fisher [13] introduced the concept of a "feasible soaring region", a spatial region inside a wind-field where wind hovering is possible for a given windspeed (see Figure 4.4). A point in the wind-field is deemed feasible for wind hovering if the local vertical wind component/updraft velocity is larger or equal than the minimum sink speed of the aircraft when flying at zero ground speed in the wind field.

The Fisher [13] paper concluded with the experimental results, describing that a UAV managed to hover autonomously in the updrafts of a hill for an extended period of time. The second experiment where hovering was attempted in front of a man-made obstacle (a building in this case), the flow was found to be too turbulent to enable stable hovering. Nevertheless, this research shows that wind hovering with a UAV is possible in practice.

The works described in this section can be used as a basis for finding the optimum hovering location for UAvs with regenerative drivetrains, which up until now has not been covered yet in literature.

## 4.3. Regenerative soaring

The concept of incorporating regenerative drivetrains (which will be discussed in more detail in Chapter 5) in soaring strategies to potentially increase the range of (glider) aircraft was first proposed by Paul MacCready in 1998 [22]. He introduced the basic idea of using the energy gained from flying in thermals and regions with

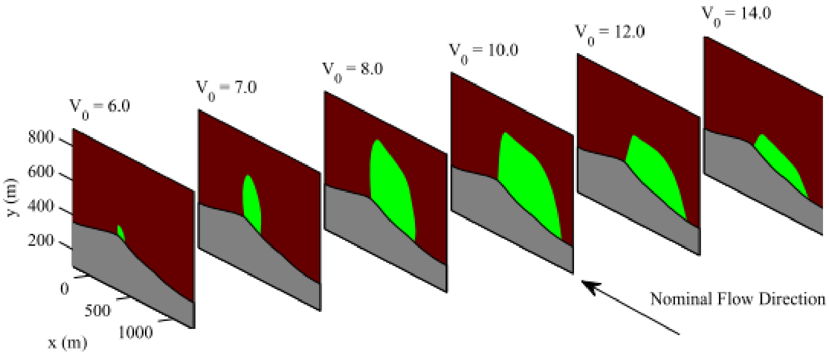


Figure 4.4: Illustration of feasible hovering region of a hill for different wind speeds (courtesy of [13])

updrafts to charge an on-board battery in combination with a regenerative electric drivetrain. He called this new soaring strategy "regenerative soaring".

The principle behind regenerative soaring is quite simple. With traditional soaring methods the only energy-storage mediums are the potential energy (altitude) and kinetic energy (airspeed) of the aircraft. The associated aircraft state variables, altitude & airspeed, are often desired to stay constant to be able to take advantage of the favourable conditions to gain energy from the atmosphere [7]. A great example of this is when one is for instance soaring upwind along a ridge to try to take advantage of the updrafts it generates. It is possible to store the gained energy in the form of altitude, but the higher the altitude the weaker the updrafts are from the obstacle. At a certain altitude the updrafts become so weak that the glider is barely able to maintain altitude without losing airspeed. Once this "ceiling altitude" is reached, it is not possible to store any more energy. It is possible to trade the potential energy for kinetic energy, and dive back down to the original altitude while gaining airspeed. The aircraft is now positioned once again in a region with stronger updrafts, but due to the increased airspeed the glider has a higher sink-speed (see Figure 4.3) which may render it unable to still gain energy from the updrafts.

With regenerative soaring, an additional energy storage medium is added in the form of a rechargeable battery. The rechargeable battery can be charged in-flight by using a propeller and electric motor as a wind-turbine. Energy can be stored without having to alter the airspeed and/or altitude. This enables an aircraft to keep flying in the more optimal regions while harvesting energy. This added benefit is highlighted in Figure 4.5, where the updrafts are not strong enough at the altitude where the glider enters the area with favourable soaring conditions. With a conventional soaring strategy it is not possible to gain any additional energy because the glider is already above the altitude with large enough updrafts. With the regenerative soaring technique in mind however, the glider can descend and circle along the ridge to first fully charge the on-board battery, after which it can use the region with updrafts to climb again to the ceiling altitude. With the battery of the regenerative system now fully charged it can be used as an on-demand power source, by using the electrical motor and the propeller to provide thrust. This enables a glider to still continue to fly even when outside of thermals or updrafts at any altitude. There are still limits of course, once the battery is fully charged no extra energy can be stored by using this method. Secondly, there is also a limit on the maximum amount of power that can be extracted because of the limitations of the regenerative drive system like for instance the maximum battery charging current. Lastly, the conversions from atmospheric energy (updrafts), mechanical energy (torque and RPM on motor shaft), electrical energy and chemical energy (battery) wastes more energy than if one were to directly convert it to gain altitude or airspeed.

MacCready looked at all of the system components required for enabling regenerative soaring and concluded that with modern technologies it is feasible to design a well performing glider that can use this system to achieve longer range and endurance [22].

Philip Barnes of the Pelican Aero Group used the concepts highlighted in MacCready's publications and designed a concept glider (dubbed the regenosoar) capable of regenerative aerodynamic braking [2]. His research provided some highly useful insights on for example the basic design of the electric drivetrain, which will be discussed later in Chapter 5, and a description of the design challenges encountered when trying to optimise a drivetrain for both regular and and regen flight performance. Unfortunately, no experiments were conducted to prove the feasibility of the proposed engineering solutions.

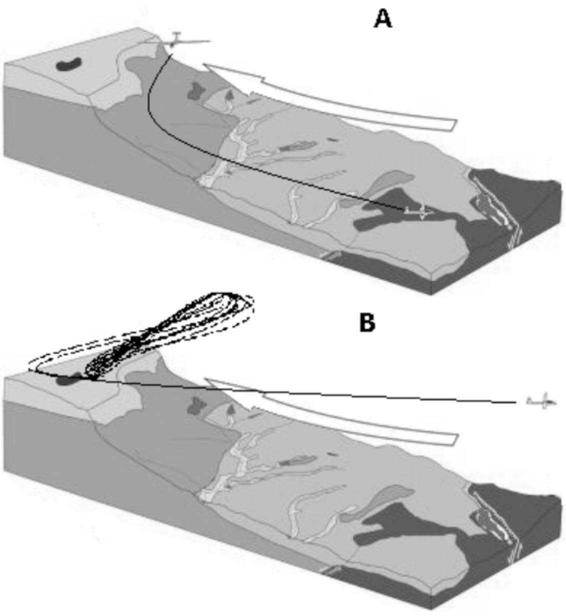


Figure 4.5: Example of conventional orographic soaring (A) vs regenerative orographic soaring (B) (courtesy of Carvalho [7])

There exists already one commercial application of the aerodynamic regen soaring/braking concept: the Pipistrel Alpha Electro. The regen drive system on the alpha electro is designed to be used as an aerodynamic brake during descents, where part of the dissipated energy can be used to recharge the batteries [11]. By using this concept, more practice patterns can be flown at an airfield with a single battery charge because of the recuperated energy from the steeper descents made possible by the regen drivetrain. The plane uses a custom designed "windprop", which is a specially designed propeller that is efficient both in the regular flight /propeller region and in the regen/wind turbine region [11]. A Power Control Unit regulates the electric braking torque applied to the propeller to maximize the regen power. The power electronics and drivetrain are tightly coupled to the main flight computer and the cockpit instruments via power and communication busses that enable precise control & management of the regen system (as can be seen in Figure 4.7).

Figure 4.6: Pipistrel alpha electro (Image courtesy of Pipistrel California)

The regenerative drivetrain itself will be discussed in more detail in Chapter 5.

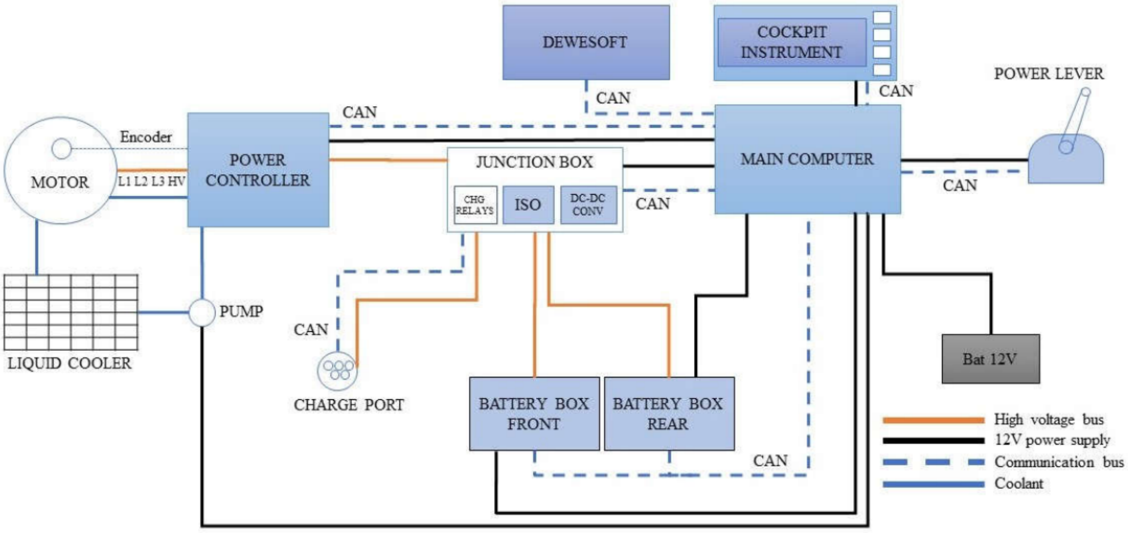


Figure 4.7: Pipistrel Alpha Electro drivetrain functional diagram (courtesy of [11])

# 5 Regenerative drivetrain

In order for the harvested atmospheric energy to be converted into electrical energy that can be stored in an accumulator (such as a rechargeable battery), an appropriate drivetrain needs to be chosen/designed. A regenerative drivetrain enables the energy flow to be reversed if necessary, by allowing the electric motor to act as a generator to recharge the accumulator. Fortunately, there has been a lot of research into this type of drivetrains, especially in the area of electric cars. The regenerative drivetrain for UAVs can be split-up into 3 major components: propeller/wind turbine (often called windprop), motor/generator, power controller/electronics and energy accumulator/battery. The following sections will discuss each of these components in detail.

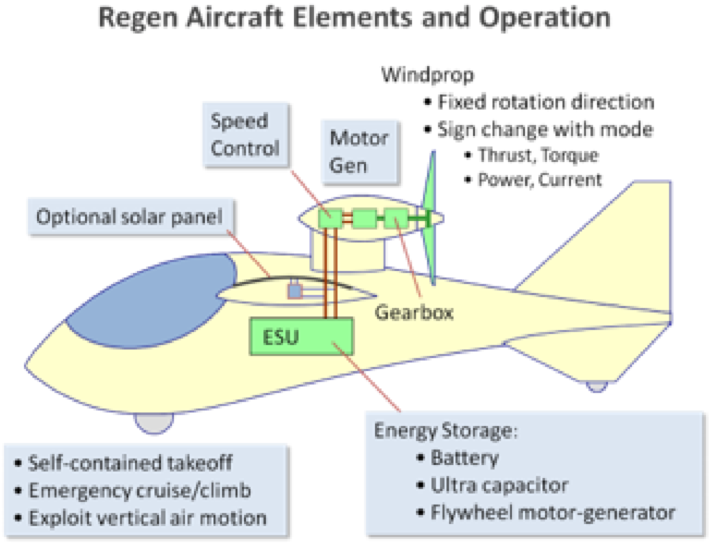


Figure 5.1: Regen drivetrain overview (courtesy of Barnes [2])

## 5.1. Windprop

Philip Barnes' paper [2] explored the idea of using the same propeller, normally used to propel the aircraft during regular flight, as a windturbine while operating in the regen mode. A common name for these combined propellers/wind turbine rotors is windprop. At first thought, one might be inclined to assume that the rotational direction of a propeller needs to be reversed in order for it to act as a windturbine. This proves to be a false assumption however. The operating mode is purely determined by the angle of attack of the relative wind with respect to the blade sections:

- Propeller: When the relative wind (W) has a positive Angle of Attack. Higher RPMs cause the angle of attack to increase (for a constant forward flight speed), providing more thrust force (as long as the propeller is not stalled). A propelling torque has to be supplied to the propeller axis.

- Pinweel: When the relative wind (W) is in line with the chord-line and the Angle of Attack is zero. No thrust or additional (pressure) drag force is generated, although the skin friction drag is still present of course. No braking torque is applied and no propelling torque is supplied to the propeller axis.
- Turbine: When the relative wind (W) has a negative Angle of Attack. Lower RPMs cause the absolute angle of attack to increase (for a constant forward flight speed), providing a higher drag force. A braking torque has to be applied to the propeller axis.

Figure 5.2 gives a graphical overview of the discussed operating modes.

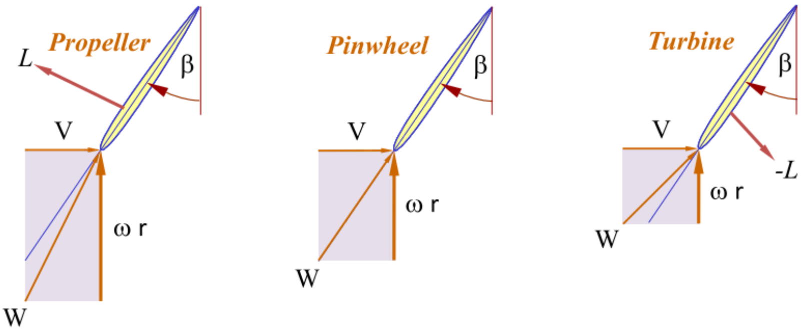


Figure 5.2: Different modes of operation of windprop (courtesy of Barnes [2])

Every propeller can act as a wind-turbine by purely exerting a braking torque to the shaft that can be used to generate electrical power by a generator. The main drawback however is that it is very challenging to design a propeller/wind-turbine that is both efficient in the propelling region, as in the turbine region. Methods have already been researched on how a windprop can be best designed with both propelling and regenerating efficiency in mind. An excellent example is the windprop designed for the Pipistrel Alpha Electro, where an iterative procedure was used to find the optimal airfoil shapes and chord/twist blade distribution [11].

## 5.2. Motor/generator

Most UAVs with an electric drivetrain use BLDC motors. These types of motors are essentially low-voltage synchronous 3-phase AC motors. The stator of the motor consists of wound coils over a ferromagnetic core making up the motor coils which are connected in a 3-phase star connection. The rotor features permanent magnets and an output shaft. By commutating the motor coils in the correct sequence (for which a specialised ESC (Electronic Speed Controller) is used), it will result in a rotation of the rotor. The frequency of the commututation of the coils will directly determine, together with the amount of turns and the way the motor coils are wound, the rotational frequency of the output shaft. Figure 5.3 shows the typical construction of a BLDC motor used in small UAVs.

BLDC motors are ideally suited to be used in a range of electric vehicles due to their high efficiency, high power densities (both in terms of mass and volume), and long lifespan (due to the lack of carbon brushes that can wear out) [8].

Last but not least these types of motors can be used in regenerative drivetrains as they can be act as a generator as well, although a rectifier is required to convert the 3-phase AC-voltage into DC [17].

## 5.3. Regen controller

The converted electrical energy needs to be conditioned and regulated before it can be used to recharge a battery such as a LiPo (Lithium-ion Polymer battery). The need to have a device that conditions and regulates the generated electrical energy becomes apparent. This controller device will be called the regenerative controller. In the following subsections, the operation and working principles of the regenerative controller will be described.

### 5.3.1. Function requirements

The conditioning/regulation controller will first have to convert the generated 3-phase AC voltage from the BLDC to DC. The resulting DC voltage after rectification will, in most cases, be lower than the battery voltage.

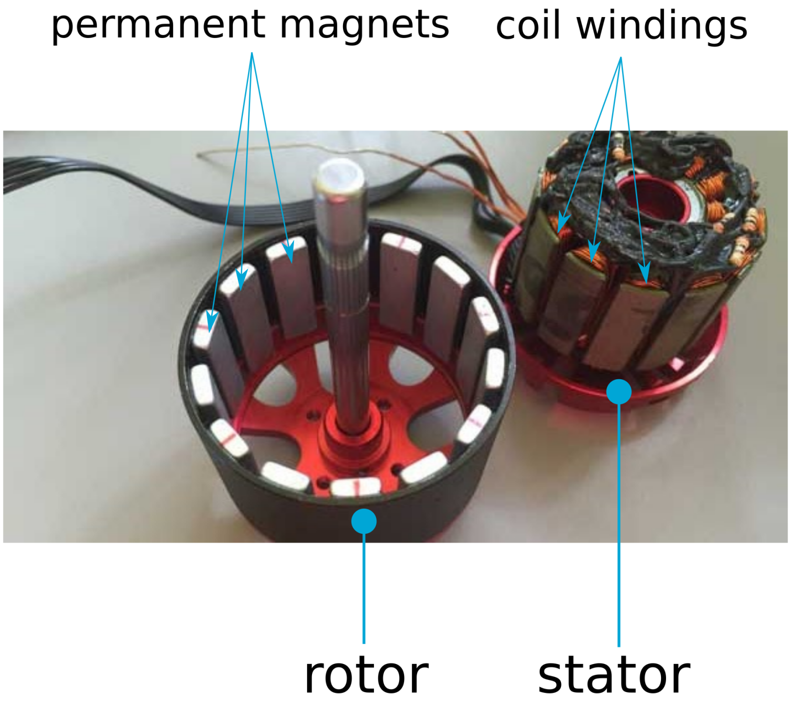


Figure 5.3: BLDC motor construction

This can be explained by the fact that a BLDC motor generated voltage (caused by the back-emf ($\mathscr{E}$) of the windings) is linearly related to it's RPM with the so called $K_v$ constant:

$$RPM = K_v \cdot \mathscr{E} \tag{5.1}$$

$$\mathscr{E} = \frac{RPM}{K_v} \tag{5.2}$$

This means that, unless switchable winding elements or other devices are introduced that can dynamically change the $K_v$ ratio of a BLDC, a higher RPM will always result in a higher generated back-emf voltage on the motor/generator windings.

Combined with the fact that fixed pitch propeller/turbine blades have to spin at a lower rotational velocity in the turbine regime than in the propelling regime (previously discussed from Barnes' paper [2]), the regenerated voltage from the turbine will be lower than the battery voltage. Thus the regen controller needs to include a boost-converter to be able to recharge the LiPo.

Lastly, the regen controller needs to include a LiPo charge controller that can monitor the battery state and control the battery charging current. Both the DC boost converter and the LiPo charge controller should be able to interface with the flight control system so the amount of regen power requested can be changed (such that the drag the turbine generates can be controlled to enable hovering) and, the battery charge state can be communicated to the flight system. An overview of the interconnections of the discussed converter/controller blocks of the regen controller, together with the ESC that is already present to drive the BLDC for providing thrust, is shown in Figure 5.5.

### 5.3.2. Combining regen controller functionality with the ESC

Xu [29] proposed a simple and lightweight regenerative braking drivetrain power controller design, where the same switching components used to drive the connected brushless electric motor can also be used to convert the generated electric power (when the brushless motor is acting as a generator) to charge the connected battery. The biggest issue that has to be overcome with these drivetrains is that the generated voltage by the brushless motor is lower than the battery voltage. This requires the regenerative drivetrain to incorporate some form of boost converter to boost the generated voltage to the battery voltage required to be able

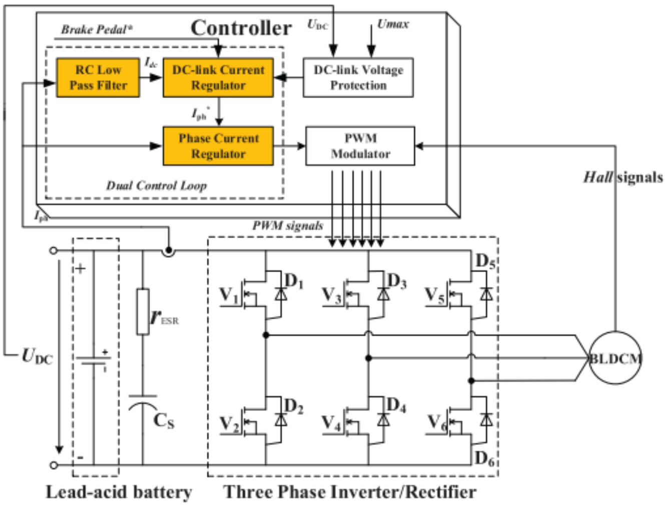


Figure 5.4: Regenerative drivetrain architecture presented in Xu [29] paper

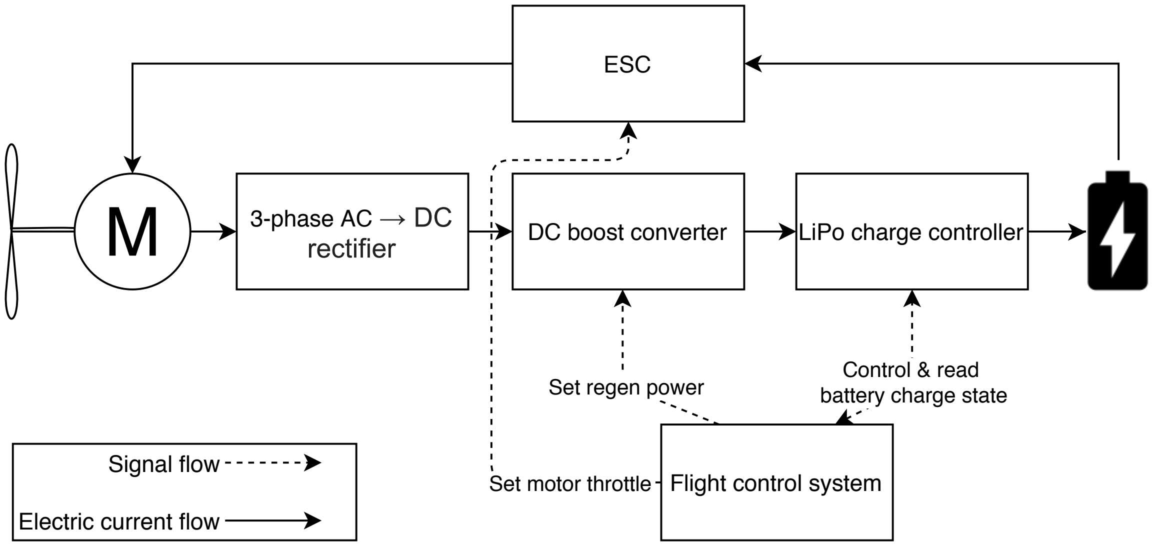


Figure 5.5: Block diagram proposed regen drivetrain

to charge the battery. The Xu [29] paper solved this problem in a very elegant way by using the inductive properties of the motor windings and the switching transistors necessary to drive the engine when not in regenerative mode, to form the required boost converter. This minimises the component count and weight of the system.

In order to take a closer look at how the switching mechanism works, it is first helpful to model each motor winding as an ideal inductor, resistor, and voltage source (which represents the generated back-emf). The simplified connection diagram detailing how the MOSFET switching elements are connected to the idealised motor winding components is shown in Figure 5.6a. The figure also shows the two different current paths during the boost switching phase using two of the three motor windings as the inductor. Essentially, the switching MOSFETs and (internal) body diodes switch and conduct in the same way as a regular boost converter circuit. Comparing the regenerative controller circuit from Figure 5.6a, to a conventional boost converter circuit (see Figure 5.6b):

- the input voltage source of the converter are the two active back-emf idealised voltage sources providing power, $e_a$ and $e_b$;
- the inductor is the series inductance of motor windings a and b, represented by $L_a$ and $L_b$;

- the switching element is MOSFET $V_2$;
- the diode that allows the higher voltage generated by the inductance of the coil to flow to the battery is $D_1$, which can be the internal body diode of MOSFET $V_1$;
- finally, diode $D_4$, which can be the internal body diode of MOSFET $V_4$, makes sure there is always a complete current path to the voltage source.

By altering the duty-cycle of the switching element, which in this case is MOSFET $V_2$, the output voltage can be regulated just as with a normal boost converter circuit. The main additional complication is that the due to the constantly changing signs and magnitudes of the generated back-emfs of the motor windings, depending on the current position of the rotor of the BLDC motor, the switching pattern has to cycle through the 3 different sets of MOSFETs to ensure that there is always a positive current flow to the battery.

The paper concluded with the experimental results proving the feasibility of this type of regenerative drivetrain. This means that by only adding some additional current and voltage sensors, together with custom firmware that enables the regen switching pattern of the switching elements of the ESC, the ESC can also act as the regen controller.

### 5.3.3. Shortcomings of existing setups from prior research

Most of the previous research into this area focused on determining if this method was suitable to use in light electric road vehicles (e.g. escooters) to increase the total driving range of a typical ride that includes sporadic braking. Apart from relative increases in driving ranges, no actual performance data on this setup could be found. It was also identified that in most experiments lead-acid batteries were used which are unsuitable for UAVs due to their low energy mass density of around 35-40 Wh/kg (compared to 150-180 Wh/kg for LiPo batteries [23]).

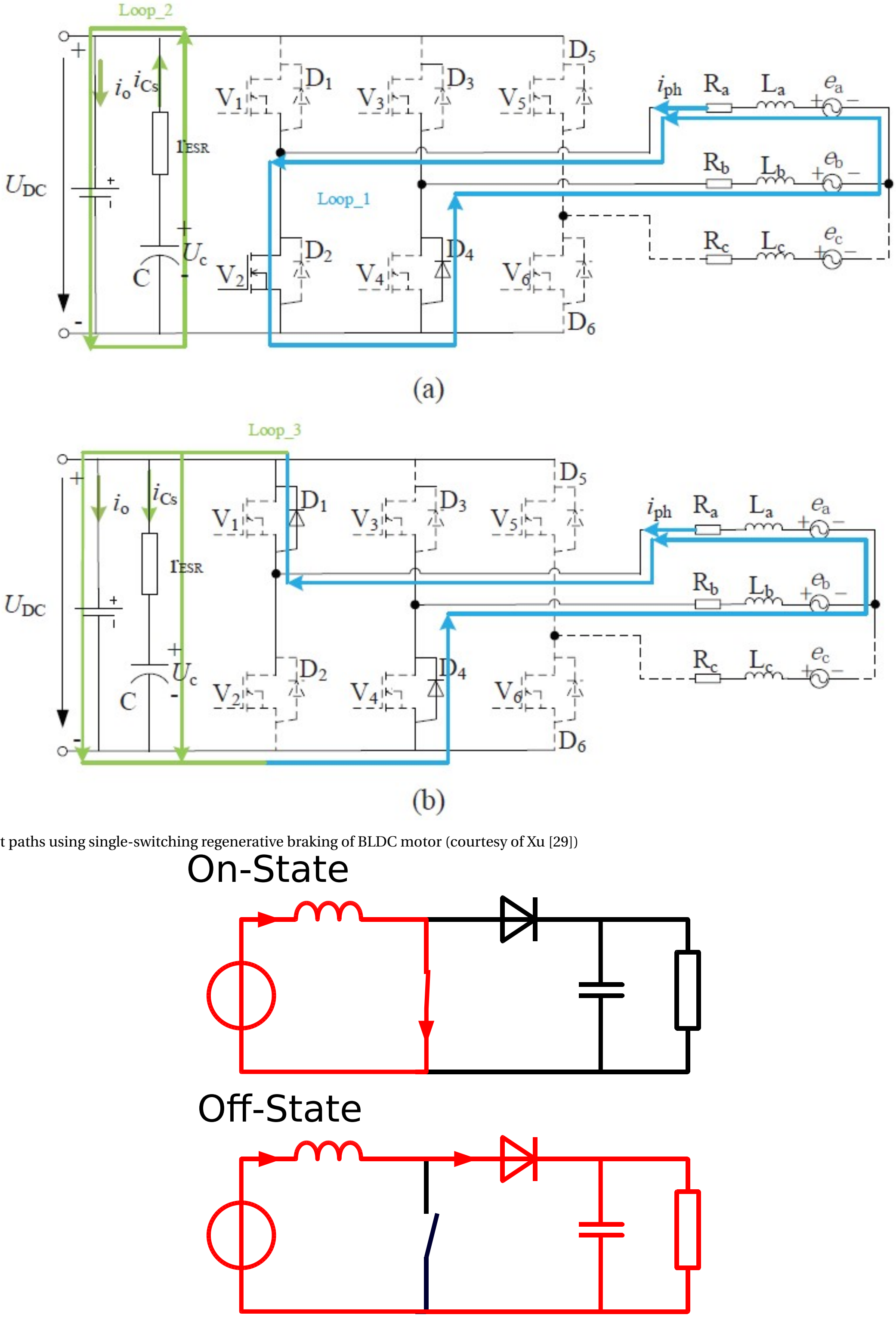
In order to be able to evaluate the performance of using this kind of combined regen controller for the proposed usecase in a UAV with LiPo batteries, additional research and tests will be required. Instead of trying to model and develop a prototype regen controller with the required functionality from scratch however, it was decided to first perform a comprehensive search if a possible suitable existing ESC controller could be found that can be easily reprogrammed and altered to achieve the desired functionality. Once a suitable candidate is found, it can be used to perform tests to try to estimate the efficiency of the regen drivetrain.

### 5.3.4. VESC ESC as combined ESC/regen controller

After a comprehensive search a possible suitable existing ESC controller was found that even has built-in functionality to be used as a regen controller: the VESC.

The VESC is a fully open-source ESC originally designed to be used in electric skateboards. Over time, it has been further developed to be used in a variety of applications and includes additional features such as datalogging, added control interfaces (such as CAN bus & UART), and (most importantly) regen capability.

The VESC was found to be ideally suited as an all-in-one combined regen controller and ESC for UAVs, due to it's compatibility with recharging LiPo batteries in the regen mode. Because of the open-source nature of the design it is possible to easily alter the firmware and do hardware modifications.



(a) Current paths using single-switching regenerative braking of BLDC motor (courtesy of Xu [29])

(b) Standard boost converter circuit diagram (source `https://commons.wikimedia.org/wiki/File:Boost_operating.svg`)

Figure 5.6: Comparison between regen boost switching and standard boost converter switching

## 5.4. VESC test setup

In order to test the performance and power efficiency of the regenerative drivetrain, a test-setup will be used that enables the input and output power to be measured while performing the regen function.

Figure 5.7: VESC ESC (courtesy of `electricboardsolutions.com`)

### 5.4.1. Test setup description

The general test-setup diagram is shown in Figure 5.8. On the left side, a brushless motor is connected to an ESC (Electronic Speed Controller) that drives this motor. The power input of the ESC is connected to a lab bench power supply such that the input voltage can be varied and the output current monitored. The throttle/power setting of the ESC can be adjusted by moving the sticks on a transmitter connected to the RX (receiver) that sends the ESC the required motor throttle signal. This setup basically emulates a propeller providing mechanical power to the shaft of the mechanically coupled brushless motor on the right.

The two brushless motors are connected together with a mechanical shaft coupler. The second brushless motor that is being driven by the first one is connected to the VESC that can be reprogrammed with a custom firmware that employs the regenerative control strategy. This second motor is mounted on a motor benchmarking device called RC benchmark that measures the torque on the motor shaft. The VESC's output is connected to a rechargeable LiPo battery through an ammeter such that the recharging current can be monitored. The VESC has built-in current, voltage and RPM sensors to monitor the brushless motor RPM and braking current, and battery voltage. All of these parameters are logged on a pc by using an interfacing adapter based on an STM32 microcontroller. The desired braking current can be adjusted using a rotary potentiometer on the transmitter that is mapped to a channel and also sent to the RX. The STM32 does the necessary conversion of the PPM signal to the serial commands required by the VESC to set a custom braking current for the regen functionality. Finally, a digital oscilloscope is also connected to the two gate drives of the switching MOSFETs of one of the motor phases to allow for an analysis to be carried out of the switching behaviour of the VESC when operating in the regen mode.

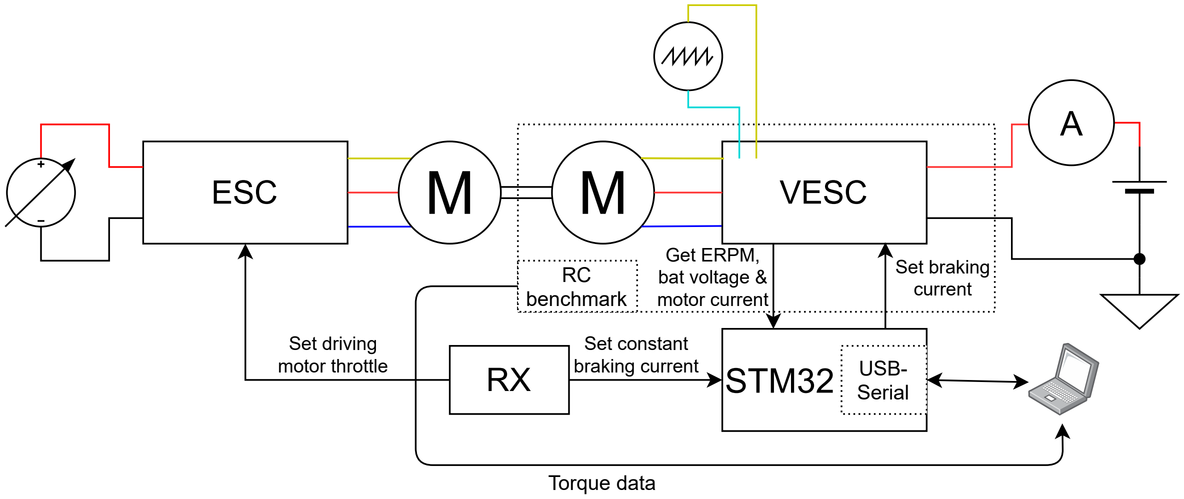


Figure 5.8: Regenerative drivetrain test-setup diagram

The specifications of the used equipment can be found in Table 5.1.

| | |
|---|---|
| Lab bench power supply | AFX-9660SB 0-30V 0-3A |
| Driving motor | Turnigy Aerodrive D2826/10 1400 KV |
| Driving ESC | YEP 40 A |
| Driven motor | EMAX GT2215/10 1100 KV |
| Regen ESC | Maytech MTVESC50A VESC compatible |
| Torque measuring device | RC Benchmark Series 1580 |
| LiPo battery | 3D Robotics 3s 9C 5000 mAh |

Table 5.1: Used equipment specifications

### 5.4.2. Expected results and outcome

Various tests will be conducted where the regenerating brushless motor will be subjected to a range of RPMs, and requested regen current & power.

The following parameters will be logged:

- Braking current ($I_{brake}$)
- Motor RPM (RPM)
- Battery regen current ($I_{bat}$)
- Battery voltage ($V_{bat}$)
- Motor torque ($T$)
- Driving motor/ESC voltage ($V_{supply}$)
- Driving motor/ESC current ($I_{supply}$)

With all of these parameters logged, it is possible to determine the approximate total drivetrain efficiency from the measurements to have a better estimate available of the practically achievable regen drivetrain efficiency, encompassing both the BLDC motor acting as a generator and the regen ESC. This can be done by comparing the mechanical power present at the coupled motor shaft to the electrical charging power of the battery. The mechanical and electrical power figures can easily be calculated from the measured parameters using the following formulae:

$$P_{mech} = T \cdot 2\pi \frac{RPM}{60} \tag{5.3}$$

$$P_{bat} = V_{bat} \cdot I_{bat} \tag{5.4}$$

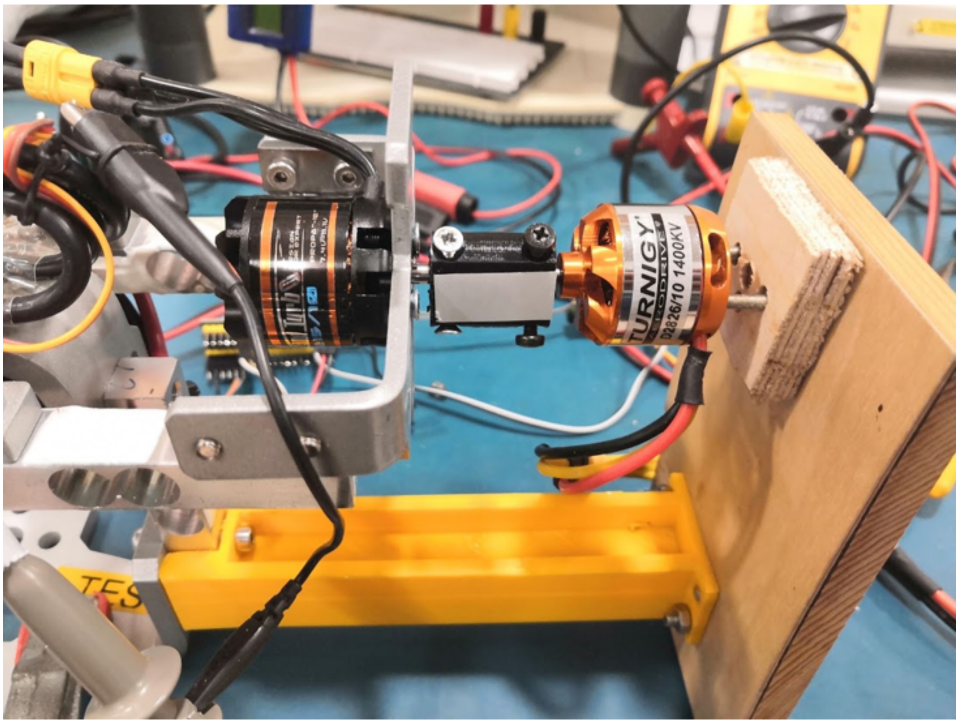


Figure 5.9: Regen drivetrain test-setup, with driving motor on the right and driven motor on the left, mounted on the RC Benchmark torque measuring device

### 5.4.3. Observations during the test

After long and thorough preparations the final performance tests were able to be conducted in the MAVlab. The main tests that were conducted involved determining the efficiency of the regen drivetrain for a range of RPMs and mechanical power. For each run, the driving motor ESC was set to keep the BLDC running at a constant RPM using the driving ESC's governer mode. For each constant RPM the requested braking current was gently increased, which lead to both the battery recharging power and the mechanical power on the motor shaft to increase. It was unfortunately not possible to choose constant measuring points for all RPMs with the same braking current or recharging power due to induced instabilities in the system probably caused by a conflict of the braking current control-loop of the VESC and the RPM governor of the driving ESC at certain operating conditions. Instead, it was opted to try to approach similar measuring points that resulted in stable readings to hopefully increase the accuracy of the measured data.

Another anomaly that was first observed was that the battery discharge current reported by the VESC didn't correspond with the external ammeter. After performing some additional measurements and checks it was found that the VESC had a near constant current reading error of around 0.25 A - 0.3 A when the brake current is set high enough to actually start the regen mode, which can be clearly seen in the graphed out measurements shown in Figure 5.10. The rest of the reported parameters logged by the VESC (motor RPM, battery voltage and motor current) were also able to be checked with a multimeter and did show good correspondence to the measured data.

During the test the gate drive signals of the VESC of both the high- and low-side switching MOSFETs of one of the motor phases were probed to investigate the exact switching behaviour of the regen mode. The switching pattern was found to be roughly equivalent to the one described in subsection 5.3.2, with one major alteration. Instead of using (body-)diodes to perform the diode functions, the VESC uses active rectification. This means it uses the MOSFETs as semi-ideal diodes. This complicates the switching pattern and control logic, but can increase the overall efficiency by virtually eliminating the voltage drop due to the low on-resistance characteristics of power MOSFETs reducing the power dissipation significantly when compared to a regular diode [28]. An additional drawback of the active rectification approach is that a small amount of power is required every time the switching element has to change state, which at high frequencies and certain conditions may offset the power dissipation savings. The observed switching behaviour, together with the probing locations, is shown in Figure 5.11. Three distinct switching regimes can be observed in Figure 5.11b:

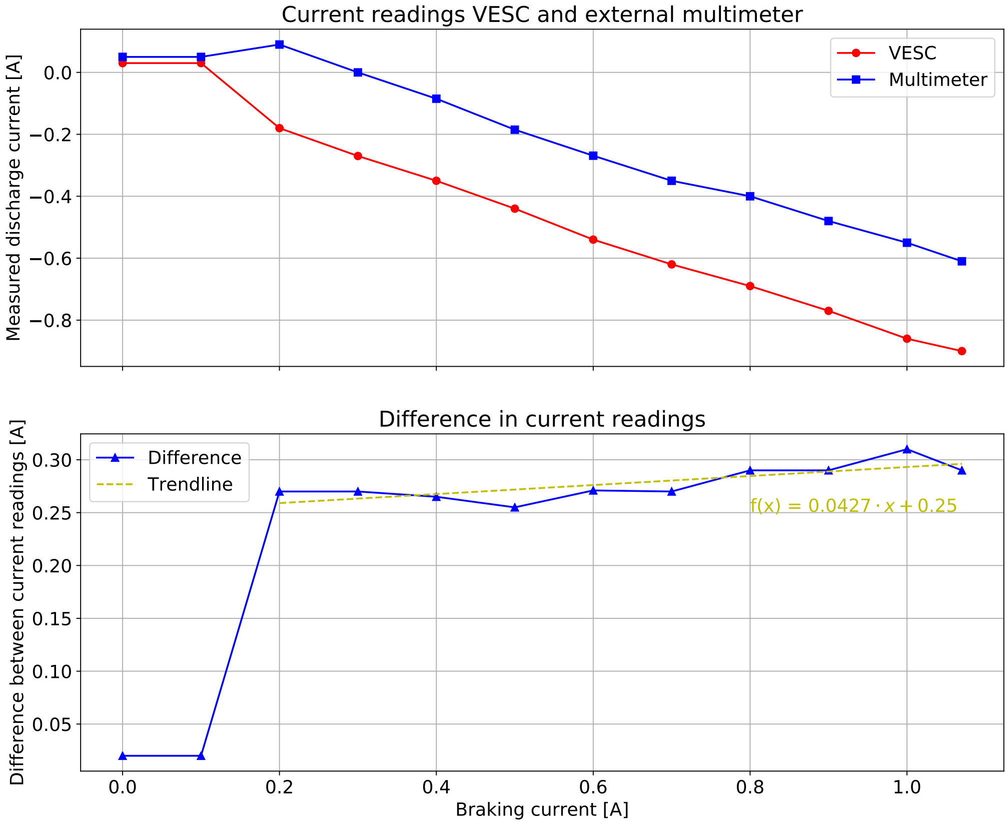


Figure 5.10: VESC current measurement discrepancy

1. Active rectification, allowing current flow to the battery when higher voltage is being generated by the inductors

2. Switching element of boost converter

3. Connecting phase to ground, making sure there is a complete current path

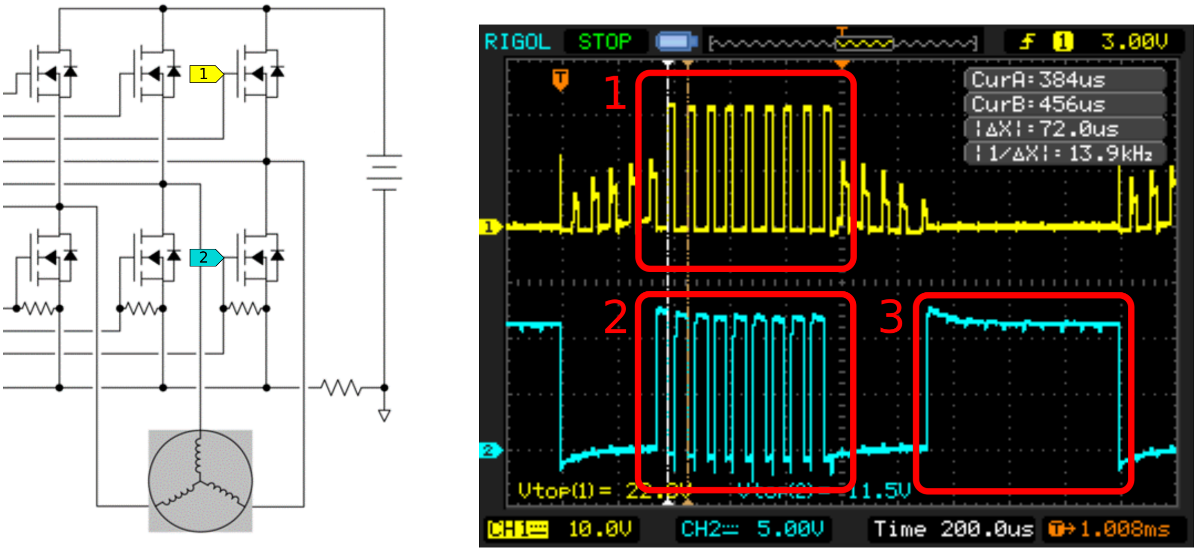
### 5.4.4. Results

With all of the necessary data gathered and processed, the measurements at the different RPM ranges were grouped together and it's efficiency calculated using the formulae described in the test setup expected results and outcomes subsection (subsection 5.4.2). The result of this for the first measurement series performed with a constant RPM of around 5000 is shown in Table 5.2. All of the other data that was logged is available in Appendix A.

The efficiency results of the drivetrain for the various RPM runs are plotted in Figure 5.12.

As was expected, the efficiency of the regen drivetrain varies considerably for different power levels & RPMs. This can be explained by the fact that the total efficiency figure is a combination of the efficiency of the BLDC motor to convert the mechanical shaft power to electrical power, and the electrical efficiency of the regen controller to convert the complex 3-phase lower AC-voltage to a higher DC-voltage suitable to safely charge the LiPo battery.

It can be clearly seen that the efficiency is very low at lower power levels. At very low power levels all of the regenerated power is used to supply the VESC itself, leaving little or no additional power for the battery to be recharged. This explains the zero efficiency number at low power values, since all of the recuperated energy is



(a) Probing locations channel 1 & 2

(b) Oscilloscope screenshot of MOSFET gate switching pattern

Figure 5.11: Switching behaviour of high- and low-level MOSFETs of motor phase during regen operation of VESC

used to drive the regen controller itself, which in some cases is not even sufficient requiring still some power from the battery to drive the VESC.

The performance of the regen drivetrain generally increases with increasing mechanical input power levels. The very rapid increase in performance at the beginning of the curves can be partly explained by the phenomenon that was just described, but there are probably also other factors at play. At very low generator currents and power levels (BLDC) motors have a very low efficiency, which rapidly increases when the generator current only rises slightly [17]. The efficiency gains for higher input powers do taper off at a certain point. The optimum efficiency point will depend on both the characteristics of the BLDC motor being used as the generator, as well as the performance characteristics of the VESC switching hardware.

The final observation that can be made is that overall higher RPMs do seem to result in slightly higher efficiency figures. This can partly be explained by the fact that higher RPMs result in a higher input voltage to the VESC regen controller, which in turn has to boost the input voltage by a lower fraction to still achieve a high enough output voltage to charge the batteries. This means that for it to achieve the same charging power as with a lower RPM, less input current is required which leads to less heat losses due to resistance present in the windings, switching elements, etc.

| # | Time stamp | $I_{supply}$[A] | $V_{supply}$ [V] | $P_{supply}$ [W] | $I_{bat}$ VESC [A] | $-I_{bat}$ (multi) [A] | Torque [N*m] |
|---|---|---|---|---|---|---|---|
| 1 | 13/11/2020 17:23 | 0.72 | 8.9 | 5.7 | 0.01 | $-0.049$ | -0.0075 |
| 2 | 13/11/2020 17:24 | 0.99 | 9.0 | 8.9 | $-0.23$ | $-0.007$ | -0.011 |
| 3 | 13/11/2020 17:25 | 1.38 | 8.9 | 13.6 | $-0.47$ | 0.248 | -0.0162 |
| 4 | 13/11/2020 17:26 | 1.82 | 8.9 | 18.4 | $-0.73$ | 0.491 | -0.0215 |
| 5 | 13/11/2020 17:27 | 2.30 | 8.9 | 24.5 | $-0.98$ | 0.750 | -0.028 |

| # | ERPM | $V_{bat}$ [V] | $I_{brake}$ [A] | RPM | $P_{mech}$ [W] | $P_{bat}$ [W] | $\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | -34805 | 12.0 | 0.00 | 4972 | 3.905 | $-0.588$ | 0% |
| 2 | -34935 | 12.0 | 0.60 | 4991 | 5.749 | $-0.084$ | 0% |
| 3 | -34957 | 12.1 | 1.25 | 4994 | 8.472 | 3.001 | 35% |
| 4 | -34918 | 12.2 | 1.95 | 4988 | 11.23 | 5.990 | 53% |
| 5 | -34976 | 12.2 | 2.69 | 4997 | 14.65 | 9.150 | 62% |

Table 5.2: Measured & calculated data for 5000 RPM run

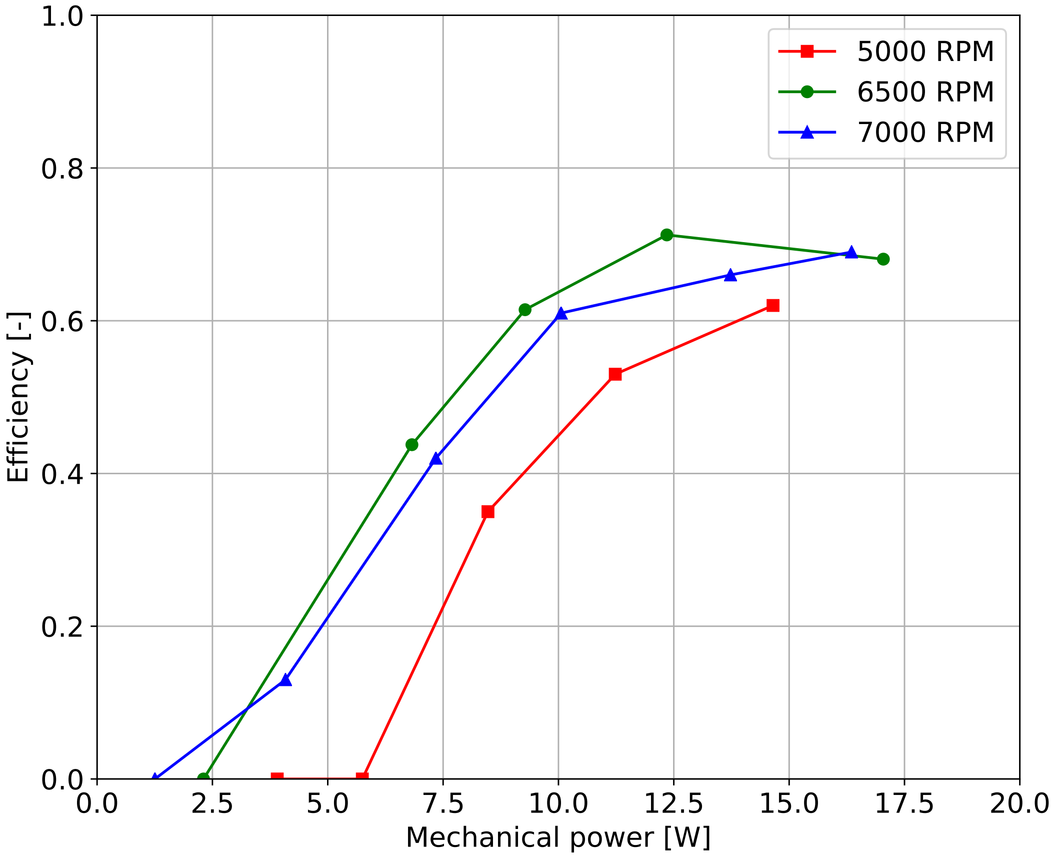


Figure 5.12: Efficiency figures calculated from VESC test runs for different RPMs

# 6 Wind-field power regen simulator

In order to help answer the main research question posed in Section 3.1, it would be helpful to have a simulated environment that models the wind-field around obstacles. The tool should be able to give an accurate estimation of the air-flow (both direction and magnitude) at the upwind side, and enable the user to vary the conditions such as upstream wind-speed, obstacle shape & size, surface roughness, etc.

Once the wind-field parameters are known for the given conditions, it would be beneficial to also get an estimate of the power that could be extracted using static wind hovering at each location in the wind-field. The tool could generate a power-contour map based on the UAV's performance characteristics to help get insights on the optimal location to hover in order to maximise the regen power.

The following sections will cover the process on how the models incorporated into the tool were derived, how the tool functions, and a detailed discussion of the different results generated by the tool for different input conditions.

## 6.1. Wind-field estimation

To be able to determine the power available in the wind-field, it is first vital to have a good understanding of the wind-field. To achieve this, a wind-field estimation tool is required that can simulate the flow around various simple obstacles. The following subsections will describe what methods are available to achieve this and how the wind-field estimation program was implemented.

### 6.1.1. Methods

There exist numerous methods to estimate the behaviour of air around obstacles, greatly varying in complexity and required computational power. A modern and accurate approach that has been gaining popularity for a while is to use a CFD analysis. CFD is a very broad term, it is a technique that uses numerical computations of equations that describe the flow of fluids [16]. In most cases, a CFD analysis uses the Navier-Stokes equations with a varying number of simplifications and assumptions that depend on the conditions and required accuracy as a basis. These simplified partial differential equations are then solved numerically for a finite number of predetermined grid points. The whole process of having to slice-up the geometry of the obstacles into a finite number of grid points (called meshing), selecting the appropriate simplifications and numerical solvers to achieve a reasonably accurate result with a limited processing time, and set-up the correct initial conditions is quite tedious. Some of the aforementioned steps also have to be repeated each time the conditions (such as free-stream wind speed) and obstacle geometry change. Gui

Because of the aforementioned time-consuming steps required to set-up the CFD simulation for different conditions and hill sizes, it was opted to search for another method that allows these variables to be easily changed. As was covered in the literature review chapter, Langelaan used a simplified potential flow method to find the wind field upwind of an idealised circular shaped hill [18] (see Subsection 4.2.1). This methodology sparked the idea to use potential flow theory to estimate the flow field present upwind of the hill.

### 6.1.2. Potential flow

Potential flow theory assumes that the fluid is incompressible irrotational, and inviscid [1]. An overview of the consequences and validity of these assumptions to the proposed wind-field estimation case are summarised

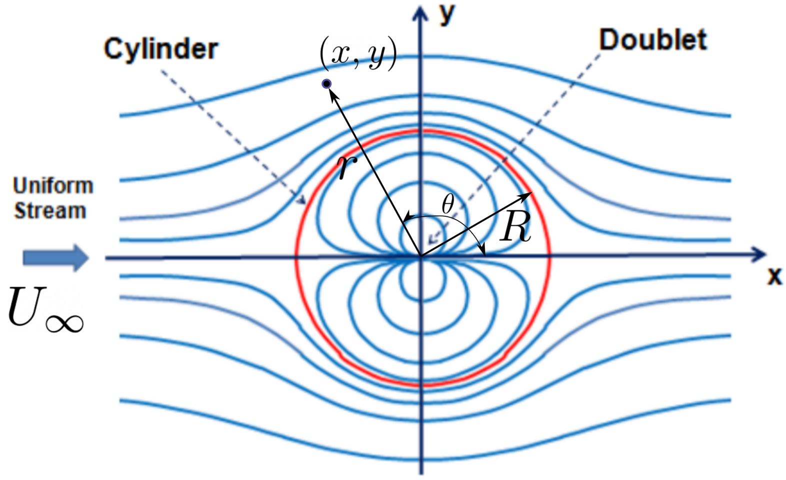


Figure 6.1: Potential flow field around cylinder

below:

- **Incompressible:** valid assumption, highest flow velocity is well below Mach 0.3
- **Irrotational:** Assumes vorticity of the flow is zero. Valid assumption for regions of the flow that don't experience turbulence.
- **Inviscid:** Assumes that the fluid does not have any viscosity. In practice this means that no flow separation occurs, and no shear forces are present between layers of air causing the skin friction drag to be zero. This assumption is valid in the layers of the flow where the flow is attached to the object. For a smooth surfaced obstacle, it can be assumed that the flow will stay attached up until the point where the height decreases, causing flow separation due to an adverse pressure gradient. This means this assumption is valid for a uniform wind-field upstream, combined with a smooth surfaced obstacle, only for the upwind side and with some distance from the surface to be outside of the boundary layer.

Circular hill
The potential flow field around a cylinder is defined by superimposing a uniform stream of velocity ($U_\infty$) and a doublet at the center of the cylinder with he correct magnitude such that the stagnation points of the resulting flow are present at the boundary of the cylinder [6]. The derived algebraic equations for the resulting velocity components (radial and cartesian) are shown below:

$$u_r = \left[1 - \frac{R^2}{r^2}\right] U_\infty \cos\theta \tag{6.1}$$

$$u_\theta = -\left[1 + \frac{R^2}{r^2}\right] U_\infty \sin\theta \tag{6.2}$$

Transforming the polar velocity components into cartesian velocity components results in the following velocity functions for the x and y components:

$$u_x = \cos\theta \cdot u_r - \sin\theta \cdot u_\theta \tag{6.3}$$

$$u_y = \sin\theta \cdot u_r + \cos\theta \cdot u_\theta \tag{6.4}$$

$$with \;\; \theta = \arctan\frac{y}{x} \tag{6.5}$$

Oval hill
The potential flow field for an oval shaped obstacle is called a rankine oval. The velocity components in x and y direction for each point in the field can be algebraically determined using the following equations [27]:

$$x_{stag}{}^2 - a^2 - \frac{ma}{\pi U_\infty} = 0 \tag{6.6}$$

$$\Leftrightarrow m = \frac{\pi U_\infty}{a}(x_{stag}{}^2 - a^2) \tag{6.7}$$

$$u_x(x,y) = U_\infty + \frac{m}{2\pi}\left[\frac{x+a}{(x+a)^2+y^2} - \frac{x-a}{(x-a)^2+y^2}\right] \tag{6.8}$$

$$u_y(x,y) = \frac{my}{2\pi}\left[\frac{1}{(x+a)^2+y^2} - \frac{1}{(x-a)^2+y^2}\right] \tag{6.9}$$

The x-coordinate of the stagnation point $x_{stag}$ and the x-coordinate of the focal point $a$ determine the geometry of the oval (see Figure 6.2).

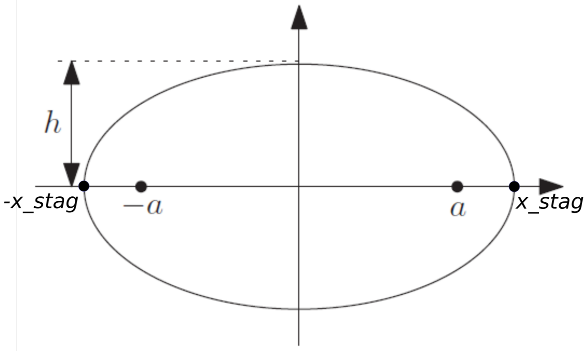


Figure 6.2: Oval geometry parameters

### 6.1.3. Boundary/shear layer

The final assumption that was discussed at the beginning of the previous subsection, assuming inviscid flow, is most likely to cause the majority of the inaccuracies. Real-life wind flows over terrain always feature a so called boundary layer (sometimes called shear layer), a region of the flow where viscosity effects play a significant role in determining the behaviour of the flow. The atmospheric boundary layer present above flat terrain most often features a horizontal wind-gradient with a zero wind-speed at the surface and converging to the free-stream velocity at higher altitudes (see Figure 6.3). Additionally, because of the surface imperfections due to grass, trees, etc on hills a plethora of hard-to predict turbulent flows will occur in regions of the boundary layer.

The undisturbed laminar atmospheric boundary layer can however be estimated using so called wall functions. A commonly used model to predict the wind gradient of atmospheric winds over oceans and undisturbed terrain is the logarithmic wind profile [10]. The simplified logarithmic wind profile equation is shown below in Equation 6.10. By altering the wind field model by including the wind profile also adds an additional modelling variable, the surface roughness ($z_0$). It will be beneficial to study the effects of varying surface roughnesses to the optimum power location.

$$u(z_2) = u(z_1)\frac{\ln((z_2-d)/z_0)}{\ln((z_1-d)/z_0)} \tag{6.10}$$

One problem arises by using this simple model to estimate the varying wind speeds in the boundary layer, the function is only able to estimate the boundary layer effects to the horizontal wind-speed over flat terrain. It has been proven though that the log wind-profile can produce accurate results even above non-flat terrain [3] in certain circumstances at higher altitudes above the obstacle. The log wall function can certainly be

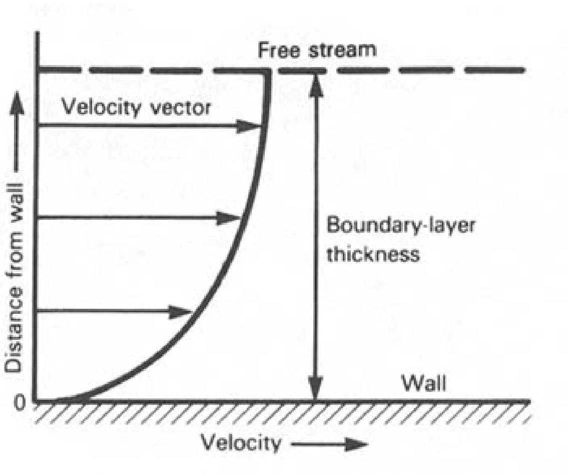


Figure 6.3: Boundary layer (courtesy of NASA)

applied to the regions of the flow that are not greatly affected by the presence of the hill (mainly upwind of the hill-side). The proposed boundary layer model will however most likely not predict the boundary layer effects close to the hill surface. It was still opted to use this model for the entire hill region since the resulting flow patterns are more closely resembling real-life wind conditions where the flow velocity decreases close to the surface due to friction. If more accurate flow behaviour needs to be predicted close to the surface of the hill, a CFD simulation including models for laminar and turbulent boundary layer behaviour would be more applicable.

### 6.1.4. Output of wind field generating program

All of the methods and formulas presented above were used to program a potential flow field generator for both circular and oval shaped hills. The parameters that can be adjusted include the free-stream wind velocity ($U_\infty$), hill radius, and surface roughness.

The following figure shows an example output of the program, namely, the vector field and streamlines for a $15\,\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m.

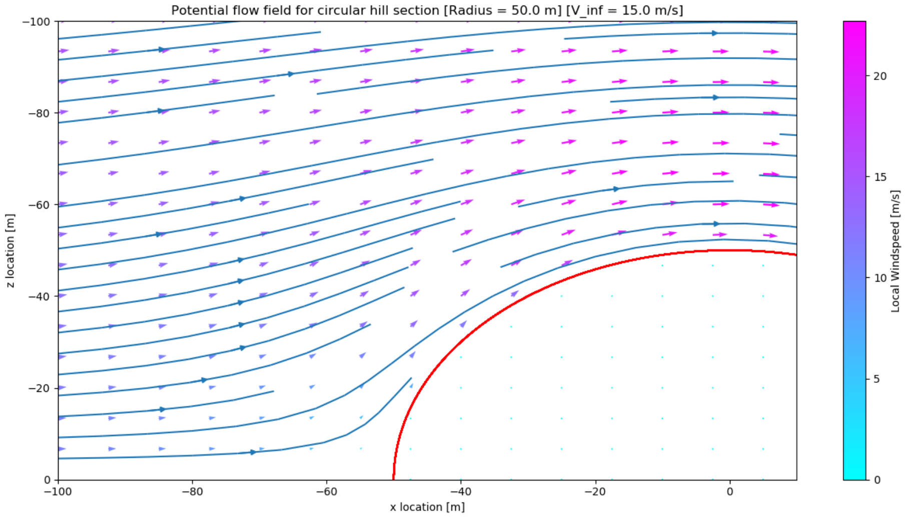


Figure 6.4: Output plot of wind field program for a $15\,\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m

## 6.2. CFD verification

To be able to assess if the proposed potential flow based methods can reliably estimate flows around idealised hills, it was decided to make a CFD analysis as well for a few select cases. The potential flow and CFD results can then be compared to each other and serve as a verification check for the model. It also verifies the compatibility of the power contour estimation tool with possible future simulations made in CFD packages.

### 6.2.1. CFD package selection

The openFOAM CFD package was selected to perform the CFD verification. It is a fully open-source set of CFD software tools available to everyone without a license. This will enable anyone to replicate the cases and alter the used initial conditions, solvers, geometry, etc.

### 6.2.2. CFD case setup

For the base CFD verification cases, it was opted to search for a similar case in the vast library of available examples for the openFOAM CFD package. A useful case which was included in one of the example files was the "Wind around buildings" case, a study of the resulting urban wind-flow when a uniform undisturbed flow-field has to pass through various urban obstacles. The main solver method used for this case was simpleFOAM, a steady-state solver for incompressible, turbulent flow based on the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm [24]. The solver methods for this case were studied and found to be used in various other related cases such as a tutorial on how to use openFOAM to study urban wind flow, with an example on how the Atmospheric Boundary Layer can also be incorporated into the simulation [26]. These two cases were used as a basis to construct the solver dictionaries to perform the verification CFD base case.

Besides the solver method, the geometry & meshing algorithm also needs to be determined. For the base-case, a cylindrical hill with a radius of 50 m was also used. A rectangular volume of 300x200x100 m was created, and a 50 m cylinder was removed with a distance of 100 m from one of the edges. Using this base geometry, a basic blockmesh (using the `BlockMesh` function) with some additional refinement at the edges of the cylinder was created using the `SnappyHexMesh` function, resulting in the final mesh for the simulation that is shown in Figure 6.5.

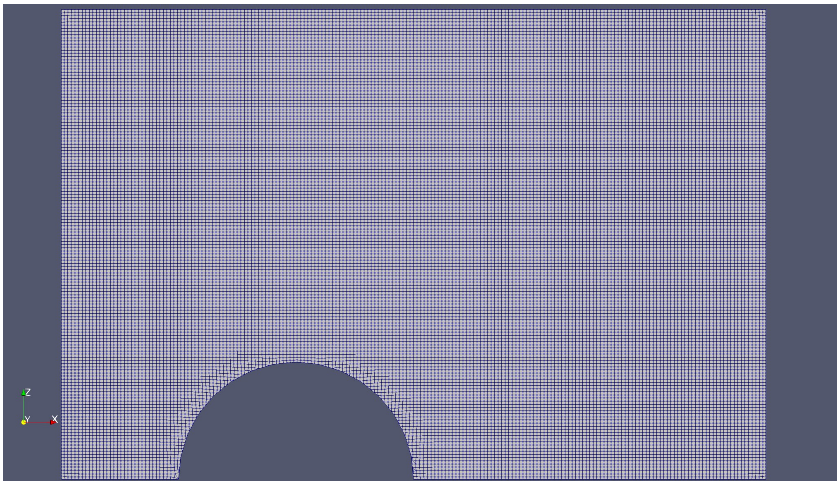


Figure 6.5: Resulting mesh to be used for wind-field model verification testing

Next, the initial conditions were determined. The setup of the velocity, pressure, epsilon & k-field initial conditions were based off of the urban wind flow openFOAM paper, with the omission of the source field for the tree canopy. They are tabulated below in Table 6.1.

The wind-velocity can be set by altering the parameters connected to the `atmBoundaryLayerInletVelocity` function. This entails the $U_{ref}$ (reference velocity), $Z_{ref}$ (the height above ground where the desired reference velocity should be present), and $z_0$ (surface roughness parameter).

An initial run using the `simpleFOAM` solver with the setup as described resulted in the following wind velocity magnitude and stream-lines plot shown in Figure 6.6

| | |
|---|---|
| Velocity field ICs | |
| Inlet | atmBoundaryLayerInletVelocity |
| Outlet | inletOutlet |
| Hill & floor | uniformFixedValue |
| Top | slip |
| Pressure field ICs | |
| Inlet | zeroGradient |
| Outlet | uniformFixedValue |
| Hill & floor | zeroGradient |
| Top | slip |
| Epsilon field ICs | |
| Inlet | atmBoundaryLayerInletEpsilon |
| Outlet | inletOutlet |
| Hill & floor | epsilonWallFunction |
| Top | slip |
| k field ICs | |
| Inlet | atmBoundaryLayerInletK |
| Outlet | inletOutlet |
| Hill & floor | kqRWallFunction |
| Top | slip |

Table 6.1: Boundary functions used for initial conditions CFD analysis

### 6.2.3. Comparison of CFD and Potential flow based model

Now the CFD simulation is running with the proper boundary conditions, a comparison can be made between the simulated flow field generated by the altered potential-flow model, and the openFOAM simpleFOAM algorithm based CFD anaylsis.

The same conditions listed below were simulated using both methods. In order to determine the differences of the vector fields generated by both methods, it is useful to perform a vector subtraction. The resulting difference vector field will ideally result in a point field, indicating a perfect match. If a difference exists for a certain grid point, the difference vector at that location will show both the difference in magnitude, and direction.

It can be observed from Figure 6.7 that the vector field approaches a point field quite well in the region in front of the hill, this is to be expected since this region should still contain mostly laminar flow where a potential flow model is still able to predict realistic actual flow conditions. Even closer to the hill itself on the upwind-side the altered potential flow model shows to correspond quite well.

The largest deviations between the two models are present on top and behind the hill. This can be explained by the fact that the actual flow starts to detach from the surface and transitions to a turbulent regime, which the potential flow model cannot take into account. Most importantly, the two models show a good correspondence on the upwind side of the hill both closer to the surface of the hill, and further away. This is also the region where the feasible regen region is expected to be.

## 6.3. Power contours

Before going into the details of the ability of the UAV to perform wind hovering at each location, it is helpful to first estimate the theoretical maximum power that can be extracted at each location assuming the UAV can maintain to hover at that location indefinitely. In this case, the energy harvesting UAV can essentially be modelled as a HAWT where the upstream wind velocity is equal to the total wind velocity at the location of the UAV in the wind field. This is not totally accurate, since this assumes that the upstream wind velocity is constant along the axis of the propeller, but since the propeller dimensions of small UAVs are at least an order of magnitude smaller than the obstacle dimensions it can be assumed that this will only have a very minor effect.

To be able to determine the theoretical maximum power that a HAWT can extract from the wind stream, it is evident to first have a closer look into the so called Betz law:

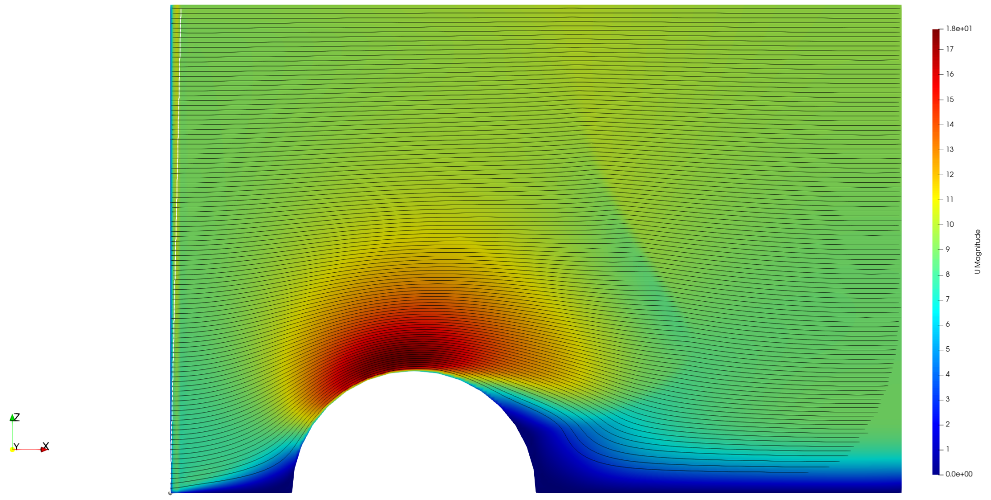


Figure 6.6: Result of CFD simulation for 50 m hill case with 15 m s$^{-1}$ $U_{ref}$ wind-speed, surface roughness parameter ($z_0$) of 0.6 m

### 6.3.1. Betz law

One of the most famous theories concerning wind turbine theory is the Betz law (also called Betz condition or limit).

Simply put, it states that even an ideal wind turbine that contains no centre hub and has an infinite number of blades that cause no additional drag (e.g. skin friction drag) can only extract roughly 59 % of the power available in the wind stream [25]. For power to continuously be able to be extracted, it is evident that a continuous mass flow of air must pass through the propeller/turbine disc. For this to occur, both the incoming and outgoing flow must have a positive flow velocity. If, hypothetically, the turbine was able to extract all of the available energy from the incoming flow, the flow past the disc area should have a velocity of zero (otherwise there would still be unextracted energy present). Having a zero fluid flow velocity at the exit of the turbine, directly means that no mass flow can be present, so no power can be extracted at these conditions.

Using the continuity equation, Euler's theorem and kinetic energy equations the following ideal power limit following the Betz law can be derived [25]:

$$P_{ideal} = \frac{16}{27}\frac{1}{2}\rho S_{turb} {V_{air}}^3 \tag{6.11}$$

This first estimate for the maximum theoretical power can be used as a basis to generate the power contours for the wind field. The following assumptions have to be kept in mind though:

- The wind turbine is assumed to not have a hub, the entire disc area region only contains blades
- It has an infinite number of blades that cause no additional drag (e.g. skin friction drag, induced drag due to tip vortices)
- The incoming flow is assumed to be constant, laminar and axial to the wind turbine axis
- No swirl is generated, the outgoing flow is also flowing axial to the wind turbine axis
- The air is considered to be an incompressible fluid

### 6.3.2. Using Betz law to generate potential power contours

The equation presented in the previous subsection was calculated for every point in the wind field from Figure 6.4 to generate a potential power contour.

Figure 6.8 shows the ideal maximum power at every location in the wind field that could be extracted from a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a windturbine with a rotor disc area of 0.1 m$^2$. It basically represents the absolute ideal maximum power that a regenerating UAV could achieve at every point in the wind field if static hovering can be achieved at that point and if the turbine can operate at its maximum power operating point, which will obviously not be the case for the majority of the wind field.

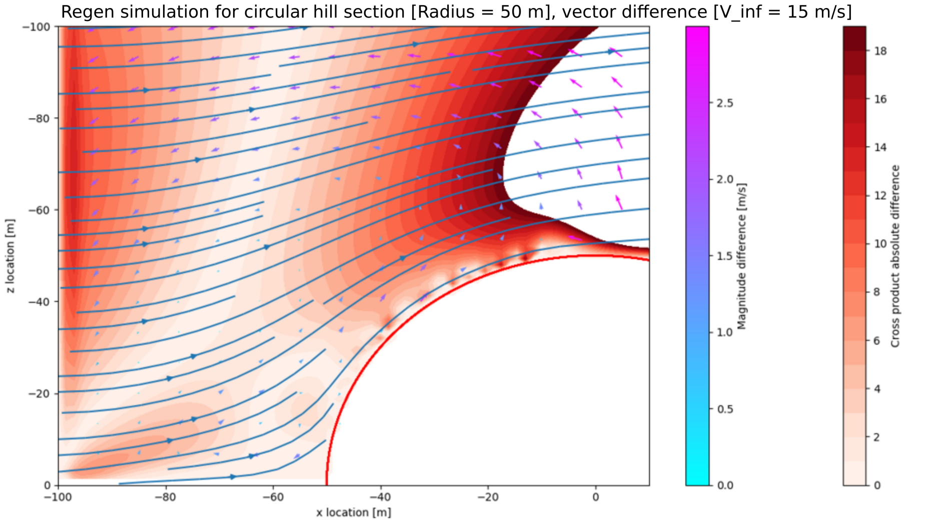


Figure 6.7: Vector difference plot between CFD calculations and altered potential flow mode

It is logical that the highest ideal power estimates are located directly above the hill since this where the wind speeds are the highest (for the idealized potential flow case). It can be seen however that close to the surface of the hill the power figure is lower since this region has a lower velocity due to the added boundary layer wall function.

## 6.4. Hovering locations

Now that the absolute maximum theoretical power that can be extracted at each point in the wind field is known, the next step is to determine if the UAV can actually hover at that location, and if so, what power fraction should be extracted from the turbine to generate the required drag to enable the hovering to be stable?

To be able to answer this question, the equations governing the longitudinal flight dynamics of a hovering UAV needs to be studied.

### 6.4.1. Longitudinal hovering flight dynamics

The following equations (Equations 6.12, 6.13 and 6.14) express the system of differential equations for longitudinal flight dynamics (following from the FBD given in Figure 6.9), including a non-zero wind, in the air-path reference system [14]:

$$T - D - W\sin\gamma = \frac{W}{g}\left(\dot{V}_{air} + \dot{u}_x\cos\gamma - \dot{u}_z\sin\gamma\right) \tag{6.12}$$

$$-L + W\cos\gamma = \frac{W}{g}\left(-\dot{V}_{air}\dot{\gamma} + \dot{u}_x\sin\gamma + \dot{u}_z\cos\gamma\right) \tag{6.13}$$

$$M = \ddot{\theta} I_{yy} \tag{6.14}$$

The equilibrium equations governing the balance of forces required for a UAV to hover in a steady state can be easily derived by setting the time derivative of the airspeed and both wind speed components (horizontal and vertical) to zero. The thrust force is also replaced with a (negative) turbine drag force which will represent the additional variable drag generated by propeller/motor drivetrain that can act as a turbine. To avoid possible confusions between the total drag force (encompassing both the aircraft and turbine drag forces) and the drag force purely generated due to the aerodynamic properties of the aircraft, the symbol $D$ which represented the latter was replaced by $D_{AC}$. Lastly, it is assumed that all of the forces acting on the aircraft are acting at the CG, meaning no moments are generated. The simplifications and alterations are shown

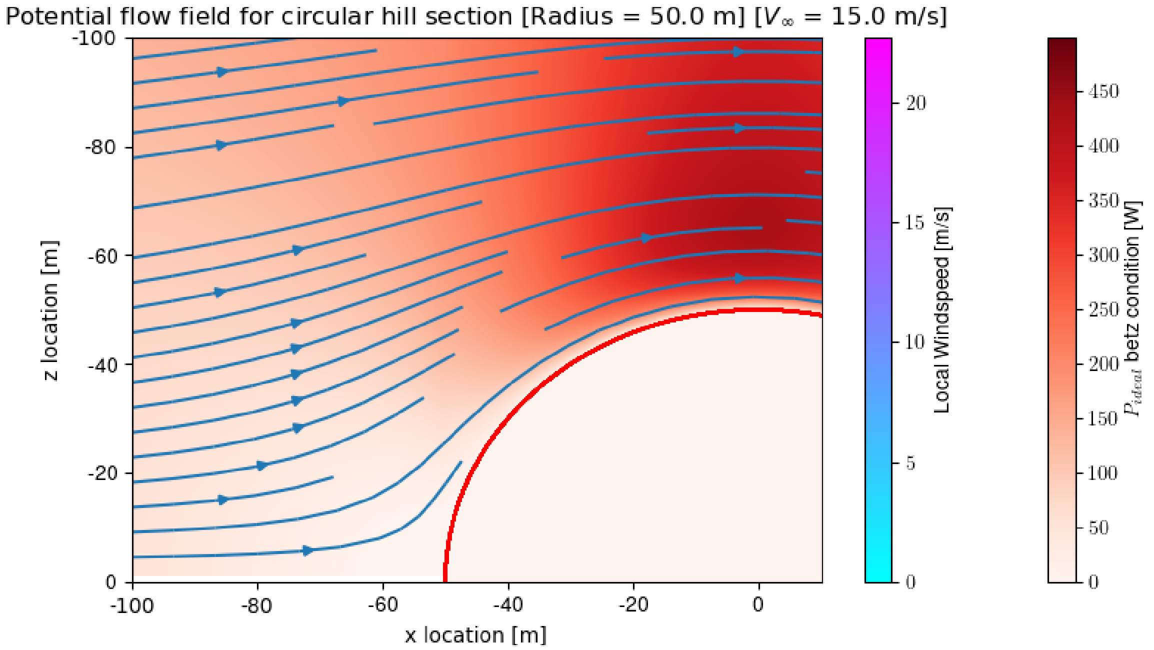


Figure 6.8: Ideal power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.1 m$^2$

below in Equations 6.15 and 6.16. An altered FBD which reflects the changes and simplifications made is shown in Figure 6.10.

$$\cancelto{-D_{turb}}{T} - D_{AC} - W\sin\gamma = \frac{W}{g}\left(\cancelto{0}{\dot{V}_{air}} + \cancelto{0}{\dot{u}_x}\cos\gamma - \cancelto{0}{\dot{u}_z}\sin\gamma\right) \tag{6.15}$$

$$-L + W\cos\gamma = \frac{W}{g}\left(-\cancelto{0}{V_{air}\dot{\gamma}} + \cancelto{0}{\dot{u}_x}\sin\gamma + \cancelto{0}{\dot{u}_z}\cos\gamma\right) \tag{6.16}$$

This results in the following system of equations:

$$\begin{cases} -D_{turb} - D_{AC} - W\sin\gamma & = 0 \\ -L + W\cos\gamma & = 0 \end{cases} \tag{6.17}$$

### 6.4.2. Estimating turbine drag

Assuming that the turbine behaves as an ideal wind turbine as discussed in Subsection 6.3.1, it can be assumed that the wind only exerts a net axial force on the rotor. This means the useful power that the wind turbine extracts can be written as the product of this axial force ($D_{turb}$) and the air velocity at the rotor disc/turbine ($V_{turb}$): $P_{turb} = D_{turb} \cdot V_{turb}$. Furthermore, when the rotor is operating at the theoretical maximum efficiency conditions Betz proved that the air velocity at the rotor disc/turbine must be equal to two thirds of the incoming air velocity [25]. Using these equations and observations, it is possible to derive a simple expression for the estimated drag produced by an ideal turbine which is shown below:

$$P_{ideal} = \frac{16}{27}\frac{1}{2}\rho S_{turb} {V_{air}}^3 \tag{6.18}$$

$$D_{turb} = \frac{P_{ideal}}{V_{turb}} \tag{6.19}$$

$$V_{turb} = \frac{2}{3}V_{air} \tag{6.20}$$

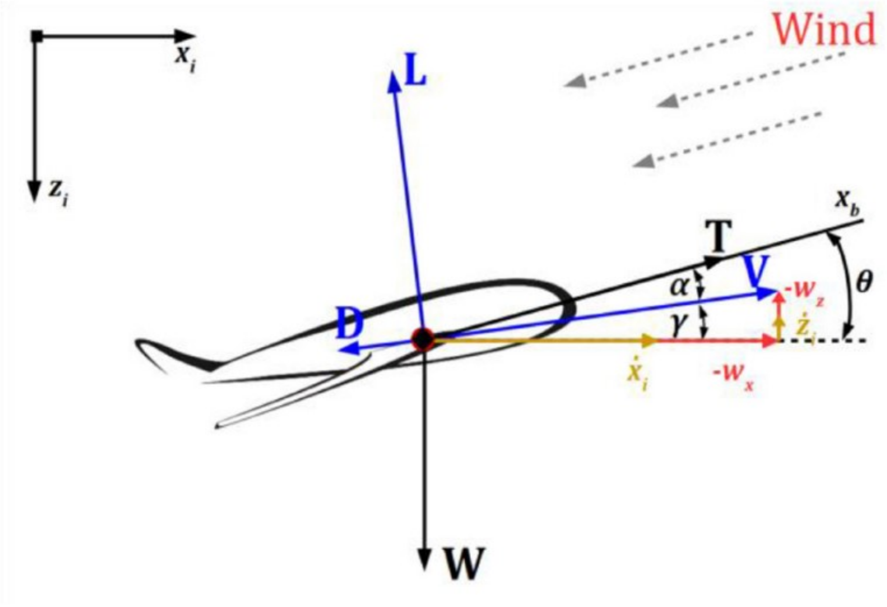


Figure 6.9: FBD Air path reference system longitudinal flight dynamics (courtesy of [14])

Substituting Equation 6.20 in Equation 6.19:

$$D_{turb} = \frac{P_{ideal}}{\frac{2}{3}V_{air}} \tag{6.21}$$

Finally. substituting Equation 6.21 in Equation 6.18 results in an equation expressing the estimated turbine drag ($D_{turb}$) in terms of incoming airspeed ($V_{air}$) and rotor disc area ($S_{turb}$):

$$\boxed{D_{turb} = \frac{1}{2}\frac{2}{9}\rho S_{turb} {V_{air}}^2} \tag{6.22}$$

### 6.4.3. Finding the required lift and drag coefficients for hovering

Now that both the systems of equations describing the force equilibrium during hovering flight and an estimate for the turbine drag are found, it is possible to derive a set of equations that determine the required lift and drag coefficients to enable static hovering.

Following from the system of equations that describes the force equilibrium during hovering flight derived in Subsection 6.4.1 (Equation 6.17), the required lift and drag terms can be expressed as follows:

$$\begin{cases} L & = W\cos\gamma \\ D_{turb} + D_{AC} & = -W\sin\gamma \end{cases} \tag{6.23}$$

Rewriting this system of equations in terms of the lift and drag coefficients results in the following system:

$$\begin{cases} \frac{1}{2}\rho V_{air}^2 S C_{L_{hover}} & = W\cos\gamma \\ \frac{1}{2}\frac{2}{9}\rho S_{turb} {V_{air}}^2 + \frac{1}{2}\rho V_{air}^2 S C_{D,AC_{hover}} & = -W\sin\gamma \end{cases} \tag{6.24}$$

Dividing both sides by $\frac{1}{2}\rho V_{air}^2 S$:

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\cos\gamma \\ \frac{2}{9}\frac{S_{turb}}{S} + C_{D,AC_{hover}} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\sin\gamma \end{cases} \tag{6.25}$$

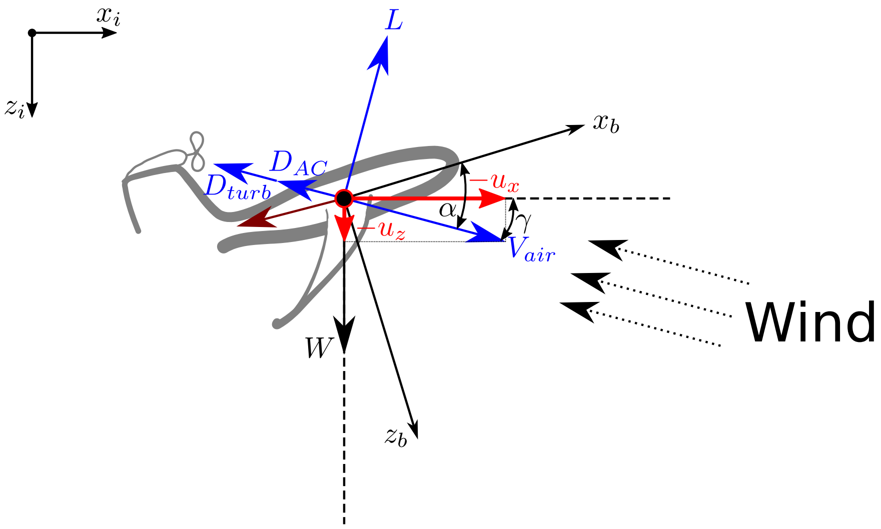


Figure 6.10: FBD Air path reference system longitudinal hovering flight dynamics

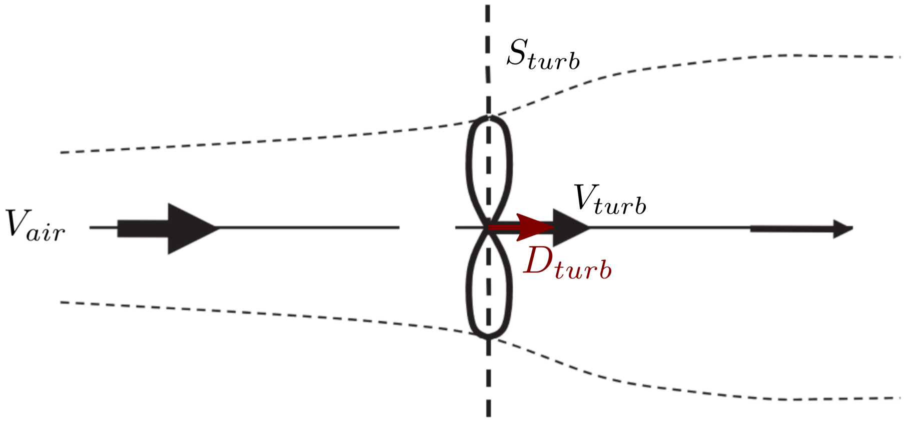


Figure 6.11: Ideal wind turbine diagram

The resulting non-dimensionalised contribution of the turbine to balance the horizontal force equilibrium (the bottom row of Equation 6.25), $\frac{2}{9}\frac{S_{turb}}{S}$, can be thought of being the maximum achievable drag coefficient of the turbine, since multiplying this figure by $\frac{1}{2}\rho V_{air}^2 S$ results in the ideal maximum drag caused by the turbine. Setting $C_{D_{turb}} = \frac{2}{9}\frac{S_{turb}}{S}$ results in the following system of equations:

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\cos\gamma \\ C_{D_{turb}} + C_{D,AC_{hover}} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\sin\gamma \end{cases} \tag{6.26}$$

Next, the sine and cosine of the flight path angle ($\gamma$) can be substituted with the fractions $\frac{u_z}{V_{air}}$ and $\frac{u_x}{V_{air}}$ respectively. This can be done because the velocity of the UAV with respect to the inertial reference frame is assumed to be zero during stable hovering. This means that the airspeed vectors magnitude and direction is purely determined by the local wind speed vectors (see Figure 6.10).

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S}\frac{u_x}{V_{air}} \\ C_{D_{turb}} + C_{D,AC_{hover}} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S}\frac{u_z}{V_{air}} \end{cases} \tag{6.27}$$

Finally, if the lift-drag polar can be estimated using the following standard equation relating the drag and lift coefficient to each other:

$$C_{D,AC} = C_{D_0} + \frac{C_L^2}{\pi A e} \tag{6.28}$$

And substituting this equation in Equation 6.27:

$$\begin{cases} C_{L_{hover}} & = \frac{W}{\frac{1}{2}\rho V_{air}^2 S} \frac{u_x}{V_{air}} \\ C_{D_{turb}} + C_{D_0} + \frac{C_{L_{hover}}^2}{\pi A e} & = -\frac{W}{\frac{1}{2}\rho V_{air}^2 S} \frac{u_z}{V_{air}} \end{cases} \tag{6.29}$$

This leaves a system of equations that can easily be solved for both the required lift coefficient ($C_{L_{hover}}$), and turbine drag coefficient $C_{D_{turb}}$ if the local air speed (which is equal to the wind speed magnitude during hovering), horizontal and vertical wind speed components are known.

Some important observations can be made from the final equations:

- The required lift coefficient, determined by the first part of Equation 6.29, should be less than the maximum lift coefficient of the aircraft. If this would not be the case, the aircraft would effectively stall when trying to achieve these conditions.
- The drag that the clean aircraft itself can provide is fixed by the required operating point on the lift-drag polar. If the required drag coefficient is lower than this value, the aircraft will not be able to achieve hovering equilibrium, even if the turbine is fully switched off or assumed to not be present;
- At specific wind speed and direction conditions, the clean aircraft will be able to provide just the right amount of drag at a certain required lift coefficient to satisfy both equilibrium equations, the turbine doesn't need to be switched on, and no power can be regenerated, since $C_{D_{turb}}$ will have to be equal to 0.
- At wind conditions where more drag is required than the clean aircraft itself can provide, the turbine needs to be switched on to close the "drag deficit" and equalise both terms of the second part of Equation 6.29. If the required extra drag from the turbine is less than its ideal maximum, the regen drivetrain should regulate the drawn power from the turbine in such a way that the drag provided by the turbine satisfies the equations.
- There exist another specific set of wind conditions where the required drag from the turbine to achieve hovering equilibrium will be equal to the maximum drag that the turbine ideally can provide. Note that although the maximum amount of power (imposed by the Betz limit) that can be drawn from the turbine in this scenario at the specific conditions, it is not necessarily the optimum resulting in the maximum amount of regeneration power, since the regeneration power also depends on the wind speed and other locations in the wind-field might exist where not all ideally available power can be extracted, but due to a higher wind speed the total regenerated power potential is still higher.

With the finalised equations for the turbine drag coefficient and above observations in mind, the calculation of the regen power contours can now be performed.

### 6.4.4. Regen power contour calculation

The finalised equations presented in the previous subsection enable one to determine if static hovering is achievable (given the local wind conditions at a certain point in the wind-field and aircraft parameters). If this is the case, the corresponding static hovering power regeneration potential can be calculated for that point.

The resulting equations can be used to determine both the required lift coefficient ($C_{L_{hover}}$), and combined drag coefficients (one being the turbine drag coefficient $C_{D_{turb}}$, the other being the drag coefficient of the aircraft $C_{D,AC_{hover}}$) to enable stable static hovering.

This function determines if the UAV is theoretically able to statically hover with zero ground speed at each point of the calculated wind field. At each potential hover location, the required additional drag and power needed from the turbine is calculated as well as the angle of attack.

First, the required lift coefficient to satisfy the hovering equilibrium equations is calculated:

$$C_{L_{hover}} = \frac{W}{0.5 \cdot \rho \cdot V_{air}{}^2 \cdot S} \cdot \frac{u_x}{V_{air}} \tag{6.30}$$

If the resulting lift coefficient is larger than the maximum achievable lift coefficient ($C_{L_{max}}$), the aircraft would stall if it tried to approach the conditions required for hovering and the corresponding point in the wind field will have a zero power regeneration potential using static hovering since hovering cannot be achieved.

Next, the total required drag coefficient to enable hovering ($C_{D_{turb}} + C_{D,AC_{hover}}$) is calculated:

$$C_{D_{turb}} + C_{D,AC_{hover}} = \frac{W}{0.5 \cdot \rho \cdot V_{air}{}^2 \cdot S} \cdot \frac{u_z}{V_{air}} \tag{6.31}$$

For the aircraft to be able to achieve static hovering, the combined required drag coefficient can not be smaller than the minimum achievable total drag coefficient. This minimum achievable total drag coefficient is equal to the clean aircraft's drag coefficient, since the least amount of drag will be generated when no additional turbine drag is generated (hence $C_{D_{min}} = C_{D,AC_{hover}} = C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae}$).

The combined required drag coefficient can also not be larger than the maximum achievable drag coefficient, which is equal to the clean aircraft's drag coefficient plus the maximum achievable turbine drag coefficient. As stated in the previous subsection, the maximum achievable turbine drag coefficient can be estimated using the Betz limit and is equal to $C_{D_{turb,max}} = \frac{2}{9}\frac{S_{turb}}{S}$. Summarising the acceptable combined required drag coefficient bounds to achieve static hovering leads to the following expression:

$$C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae} \leq C_{D_{turb}} + C_{D,AC_{hover}} \leq C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae} + \frac{2}{9}\frac{S_{turb}}{S} \tag{6.32}$$

If the total required drag coefficient falls within these bounds and the required lift coefficient is not larger than the maximum lift coefficient (as stated earlier), it can be assumed that the aircraft can achieve static hovering, and a valid power regeneration potential can be calculated.

The resulting required turbine drag coefficient to achieve stable static hovering can be calculated as follows:

$$C_{D_{turb,hov}} = C_{D_{turb}} + C_{D,AC_{hover}} - \left( C_{D_0} + \frac{C^2_{L_{hover}}}{\pi Ae} \right) \tag{6.33}$$

The corresponding turbine drag generated during hovering can easily be found by multiplying the turbine drag coefficient with $0.5 \cdot \rho \cdot V_{air}{}^2 \cdot S$:

$$D_{turb,hov} = 0.5 \cdot \rho \cdot V_{air}{}^2 \cdot S \cdot C_{D_{turb,hov}} \tag{6.34}$$

Finally, by rearranging Equation 6.21 the estimated turbine power can be found:

$$P_{turb.hov} = \frac{2}{3} \cdot V_{air} \cdot D_{turb,hov} \tag{6.35}$$

## 6.5. Results

The equations described in the previous subsection were implemented into the power contour wind field program which results in a power contour plot that is only shown in locations where the UAV is able to statically hover.

Figure 6.12 shows the improved power contour plot with these additional calculations for the same situation as presented previously, 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.1 m$^2$ for approximated parrot DISCO UAV aircraft parameters:

In the following subsections, the input parameters to the model will be changed and the resulting power contour plot outputs will be compared to the expected behaviour based on the model equations. This also serves as a simple verification check of the regen model behaviour.

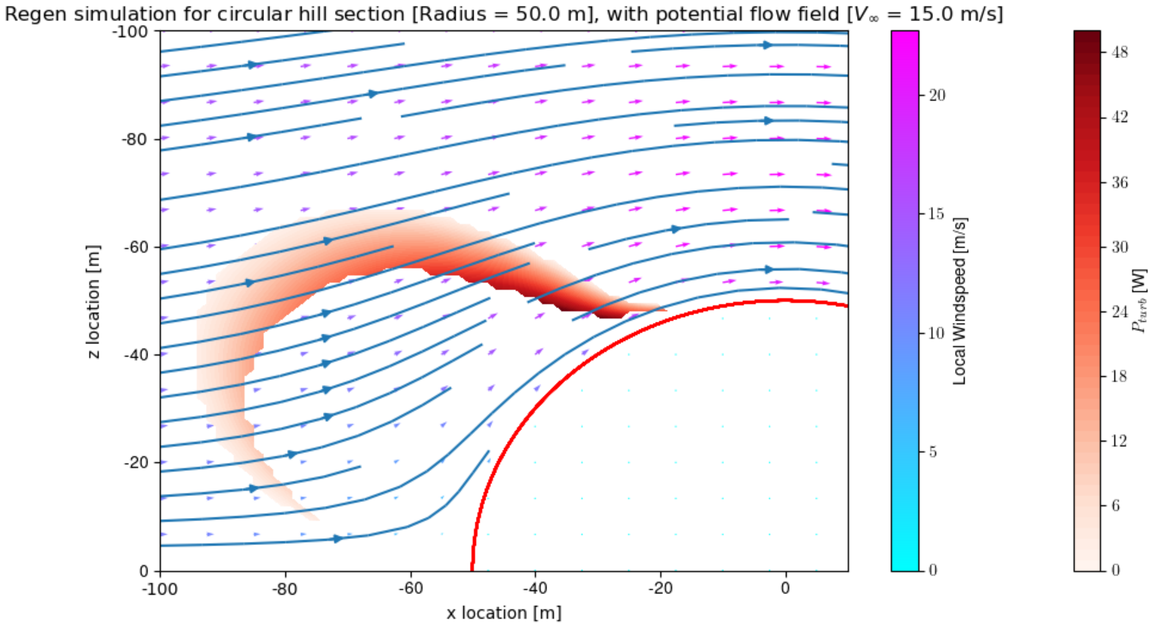


Figure 6.12: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.1 m$^2$ for approximated parrot DISCO UAV aircraft parameters

### 6.5.1. Comparing regen power estimate to Betz limit power available in wind-field

It can immediately be seen that the estimated maximum amount of power that can be regenerated using the turbine while hovering is roughly 1 order of magnitude lower than then the ideal Betz limit power contour graph of the entire windfield (Figure 6.8). The primary reason for this is that the UAV is unable to statically hover with these conditions at the point in the wind-field that has the maximum potential power, which is he point with the highest wind velocity.

### 6.5.2. Utilisation of turbine in hovering region

Another observation that can be made from this plot is that even within the feasible hovering region, higher wind speeds don't necessarily correspond to a higher regen power output.

This observation can probably be explained by the fact that the turbine's power generating potential can not be fully utilised at every point in the feasible hovering region because the equilibrium equations which govern static hovering would otherwise not be satisfied.

One way to check this theory is by looking at the calculated required turbine drag coefficient ($C_{D_{turb}}$) for each point in the feasible hovering region. The following Figure shows a $C_{D_{turb}}$ contour plot for the same conditions as Figure 6.12:

As can be clearly seen from the $C_{D_{turb}}$ contour plot, the required turbine drag coefficient to enable static hovering varies considerably depending on the location in the wind-field within the feasible hovering region. The outer edge of the feasible hovering region mostly contains points where the required turbine drag coefficient is close to zero, meaning that the generated drag by the aircraft itself is (almost) enough to balance the equilibrium hovering equations. The inner edge of the hovering region contour plot suggests that at these locations the aircraft needs all of the additional drag the turbine is able to generate to be able to satisfy hovering equilibrium. Trying to achieve hovering outside of the outer edge of the calculated hovering region will not be possible because the aircraft itself already produces too much drag to be able to satisfy the hovering equilibrium conditions, which will result in the aircraft drifting to the right side, back towards the hill. Trying to hover just outside of the inner edge of the feasible hovering region will also not be possible, but this time due to a lack of drag. Trying to hover in these locations will cause the aircraft to drift to the left side, away from the hill.

Taking a closer look at the corresponding equilibrium equation (Equation 6.26) reveals that the total required drag coefficient to enable hovering is related to the inverse of the total airspeed square, and to the sine

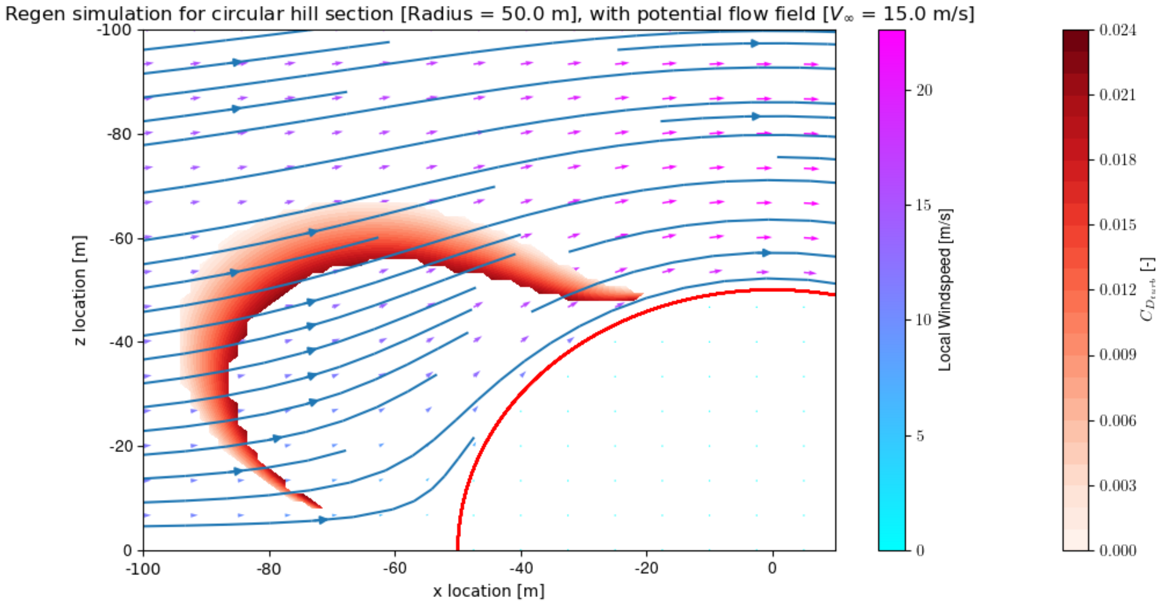


Figure 6.13: Required turbine drag coefficient ($C_{D_{turb}}$) contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.1 m$^2$ for approximated parrot DISCO UAV aircraft parameters

of the flight path angle, which during hovering is equivalent to angle between the horizontal and the wind vector (see Figure 6.10)

This indeed corresponds to the observations mentioned above and the result in the diagram, where in general the points along the outer edge with a near-zero required $C_{D_{turb}}$ to achieve static hovering have a higher local wind-speed than the closest point of the inner edge (where the required drag coefficient is higher).

### 6.5.3. Varying the rotor disc area

By varying the rotor disc area, the maximum achievable turbine drag coefficient ($C_{D_{turb}}$) changes since $C_{D_{turb}} = \frac{2}{9}\frac{S_{turb}}{S}$ (see Subsection 6.4.3). Since the maximum achievable total drag coefficient will affect the range of conditions for which hovering equilibrium can be achieved, it is expected that the shape and size of the feasible hovering region will change. Below, the power contour plots for the same conditions as used in the previous subsection are shown with varying rotor disc areas ranging from 0.05 m$^2$ to 0.2 m$^2$:

It can be seen from the various figures that the feasible hovering region grows significantly with increasing rotor disc areas. This is to be expected since a higher achievable drag coefficient (for the same lift coefficient) enables the UAV to hover in regions where it previously couldn't generate enough drag to satisfy the hovering equilibrium equations. The maximum estimated regen power is also increased significantly, which is expected since a higher turbine drag coefficient results in more power being able to be extracted from the turbine.

Taking a closer look at the difference between the different hovering regions for different rotor disc areas reveals that the location of the upper boundary of the curve stays constant, while the lower boundary of the hovering region is displaced such that the hovering region area increases. This is very logical since the location of the upper boundary is related to the minimum achievable drag coefficient (see Subsection 6.5.2). Since the minimum drag coefficient is assumed to not be affected by an increase in the rotor disc area, the location of the upper boundary of the region stays constant,

An interesting fact is that the plots seem to indicate that there is no single disadvantage in terms of flight performance for regen hovering flight by having a larger turbine disc area/diameter, since the maximum hovering region area increases as does the maximum regen power. It should be noted that the program assumes that the turbine produces zero drag when no regen power is drawn and that the propeller does not have a hub. In practice every turbine will have a non-zero drag even when pin-wheeling, and this drag is likely to increase for larger turbine diameters. Another factor that is not taken into account is that larger propellers also have an increased mass (assuming propellers are compared with a similar material and construction technique). The effect of a larger mass of the UAV to the hovering region and regen power estimations will be discussed in the next subsection.

### 6.5.4. Varying aircraft mass

Next a closer look will be taken at the consequences of altering the aircraft's mass to the hovering region and power output figures.

Looking at the equation to calculate the required lift coefficient to satisfy the hovering equilibrium (Equation 6.30) reveals that a higher aircraft mass/weight will require a higher lift coefficient to be able to hover at the same conditions. It should be noted that if the required lift coefficient increases to a higher value than the maximum lift coefficient, the aircraft can not hover at these conditions anymore since the UAV would stall if if tried to approach the required flight conditions for hovering. In this case the location associated with these conditions will not be a valid hovering point anymore in the wind-field. If the coefficient of lift is achievable, the corresponding required total drag coefficient to satisfy the hovering conditions can be calculated using Equation 6.31. This equation reveals that the total required drag coefficient will also increase with a higher aircraft weight. However, this does not necessarily translate into a higher required turbine drag coefficient since the clean aircraft drag coefficient is proportional to the required lift coefficient squared (which as discussed before increases with higher aircraft weight). Essentially, part of the increased drag coefficient margin to achieve static hovering due to a higher weight will always be used to counter the increased aircraft drag coefficient resulting from having to fly at a higher lift coefficient. Whether the remaining drag coefficient "budget" that can be used to perform regen using the turbine increases or decreases will depend on the specific conditions and aerodynamic characteristics of the airframe. This means that given a feasible hovering location at a nominal aircraft weight, if the mass increases the hovering point can become infeasible due to the required lift coefficient being higher than the maximum lift coefficient, or the point remains a feasible hovering location that will require a higher total drag coefficient which depending on the precise conditions and lift-drag polar of the airframe can result in higher or lower turbine drag coefficients which correspond to higher and lower regen power outputs respectively.

Another way to satisfy the hovering equilibrium equations with a higher weight is to fly at locations with a higher total wind-speed, which allows the lift and drag coefficients to stay in the same range. This means that points in the wind-field where the required lift and drag coefficient were too low to enable stable hovering become feasible. Since higher wind-speeds where hovering can be achieved correspond to higher regen power potentials, the maximum regen power that can be achieved in the wind-field will also likely increase. If of course no higher wind-speeds are present in the wind-field than where the UAV is able to hover with its current weight, increasing the mass will decrease the size of the hovering region until it eventually becomes impossible to hover at any point in the wind-field.

These effects can clearly be seen in the figures below, where the power contour plots are drawn for the approximated DISCO UAV aircraft parameters with a mass of 2, 3 & 10 kg in Figures 6.17, 6.18, 6.18 & 6.20 respectively. It can be seen that for increasing masses the maximum regen power increases slightly and that the feasible hovering region moves to regions with slightly higher wind-speeds. When the mass is drastically increased to e.g. 10 kg, the feasible hovering region drastically reduces in size due to the very limiting area where the wind-speeds are high enough for the aircraft not to stall due to its heavy mass. If the mass would be increased further the feasible hovering region would eventually reduce to a a very small area before it disappears entirely, because the aircraft is now too heavy to hover in a steady-state anywhere in the wind-field.

### 6.5.5. Varying the wind-speed

It is expected that for lower free-stream windspeeds, the hovering region will move towards regions where the magnitude of the the local wind-speed vs the free stream wind-speed is larger, to try to achieve the same airspeeds.

From Equation 6.30 governing the required lift coefficient to achieve hovering, it can be deduced that for the same location where the fraction of $\frac{u_x}{V_{air}}$ stays constant (since this is purely determined by the angle of the local wind-flow), a higher hovering lift coefficient is required for decreased wind-speeds. This also holds true for the total required drag coefficient.

A higher required lift coefficient means that the clean aircraft drag will be increased, provided that the required lift coefficient is even achievable before stalling. This can work in favour of achieving the equilibrium however, since a higher total drag coefficient is also required.

Figure 6.21 shows a regen region power plot for a $10\,\mathrm{m\,s^{-1}}$ wind flow with the standard hill-size of 50 m. When compared to the plot below it with the same conditions, except for a higher wind-speed of $15\,\mathrm{m\,s^{-1}}$, it can be observed that the hovering region moves further away from the hill. This corresponds to regions where the angle of the wind-field is more horizontal, since the aircraft would likely move forward into the wind if it didn't have a large enough head-wind component.

The maximum achievable regen power level is also lower, which is to be expected since the achievable power figure of the regen wind turbine are directly proportional to the local airspeed cubed.

## 6.6. Conclusion

In this chapter, the process on how a wind-field power regen simulator was created was discussed. A wind-field estimator model based on potential-flow methods was derived which is able to produce vector fields of circular and oval shaped hills with very little computational power required. This model was verified with a CFD simulation which is able to model the flow more accurately, but at a significant cost in terms of complexity to set-up the simulation and processing time. The CFD results showed a good correspondence with the potential-flow field model on the upwind-side of the hill, which is the area of interest for the hovering region.

A longitudal hovering dynamics flight model was derived based on wind-turbine theory and soaring dynamics. This model can predict if an aircraft with a set of basic aerodynamic parameters can achieve regen hovering given the local wind conditions, it also provides an estimate of the turbine drag and power figures.

The wind-field estimator model and longitudinal hovering dynamics flight model were then combined into a unified simulation tool which allows the user to generate wind-field diagrams with an overlayed regen power potential contour plots. These diagrams provide an easy overview where in the wind-field a UAV is able to hover, and what the expected best-case regen power levels are.

This tool was then extensively tested by generating plots for a variety of conditions, with varying aircraft parameters, and environment conditions. The effect of these changes to the power plots were analysed and discussed in detail, and the behaviour of the tool was found to be consistent with the theory and expected behaviour of an aircraft in these varying conditions.

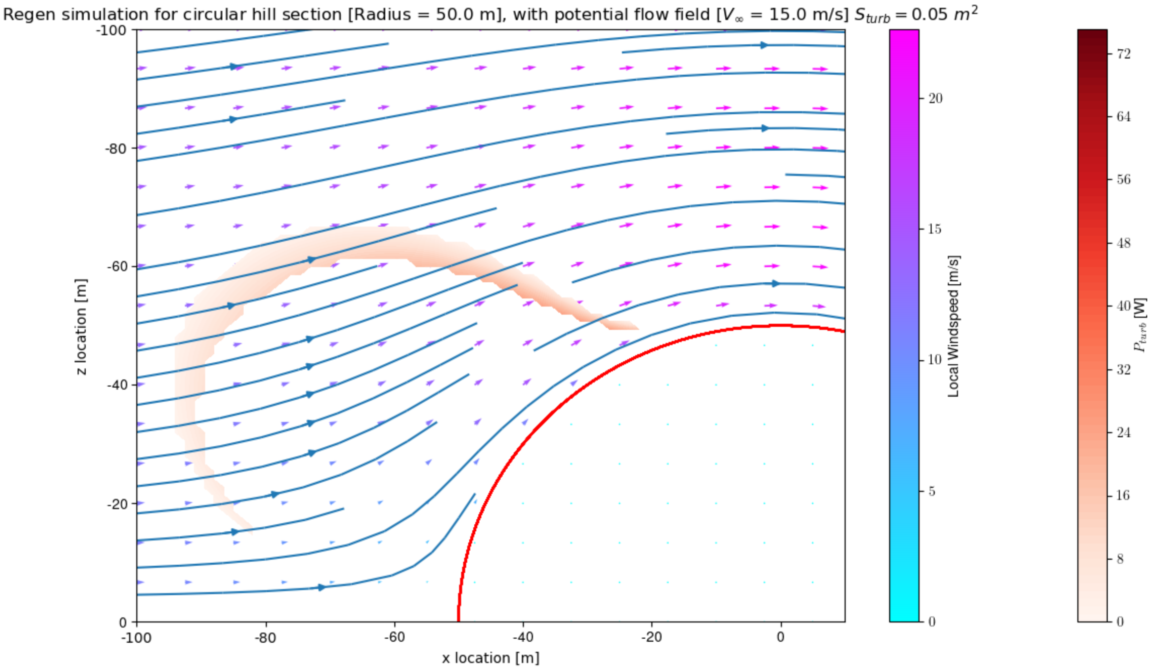


Figure 6.14: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.05 m$^2$ for approximated parrot DISCO UAV aircraft parameters

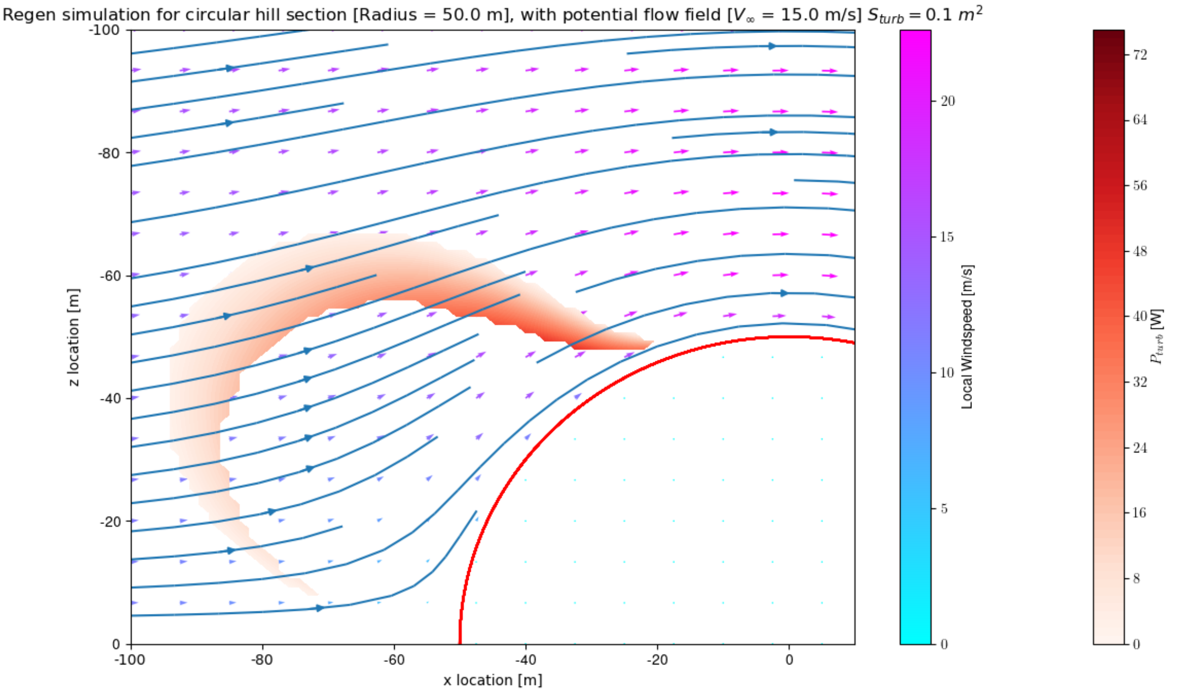


Figure 6.15: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.1 m$^2$ for approximated parrot DISCO UAV aircraft parameters

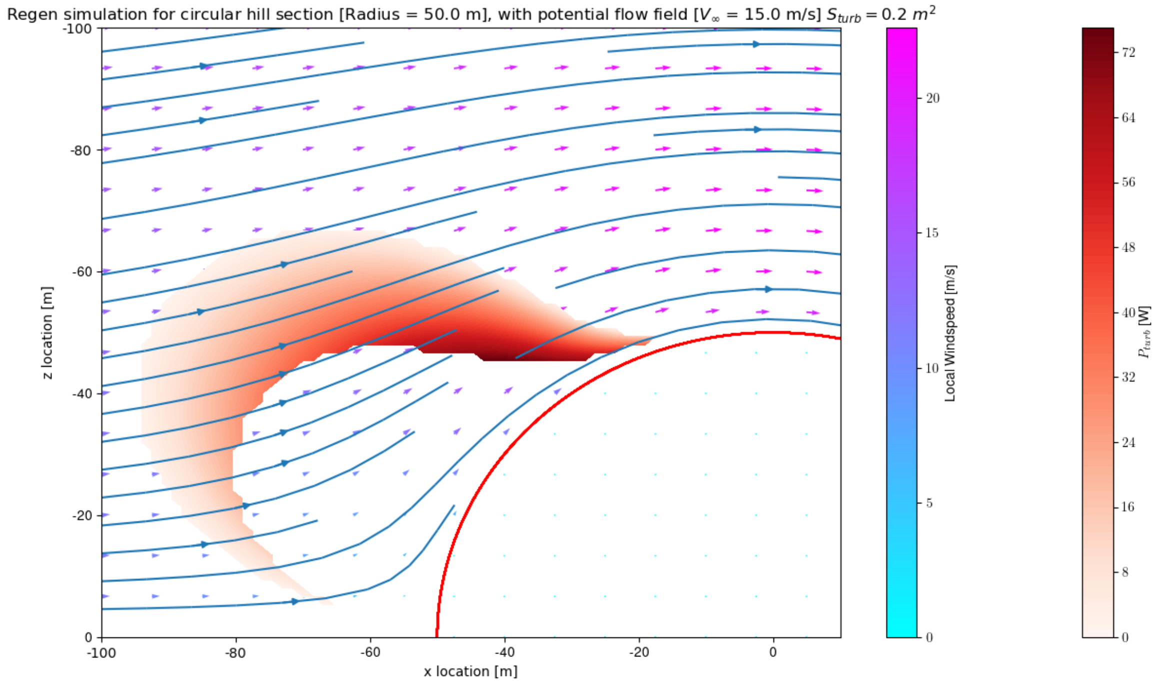


Figure 6.16: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for a rotor disc area of 0.2 m$^2$ for approximated parrot DISCO UAV aircraft parameters

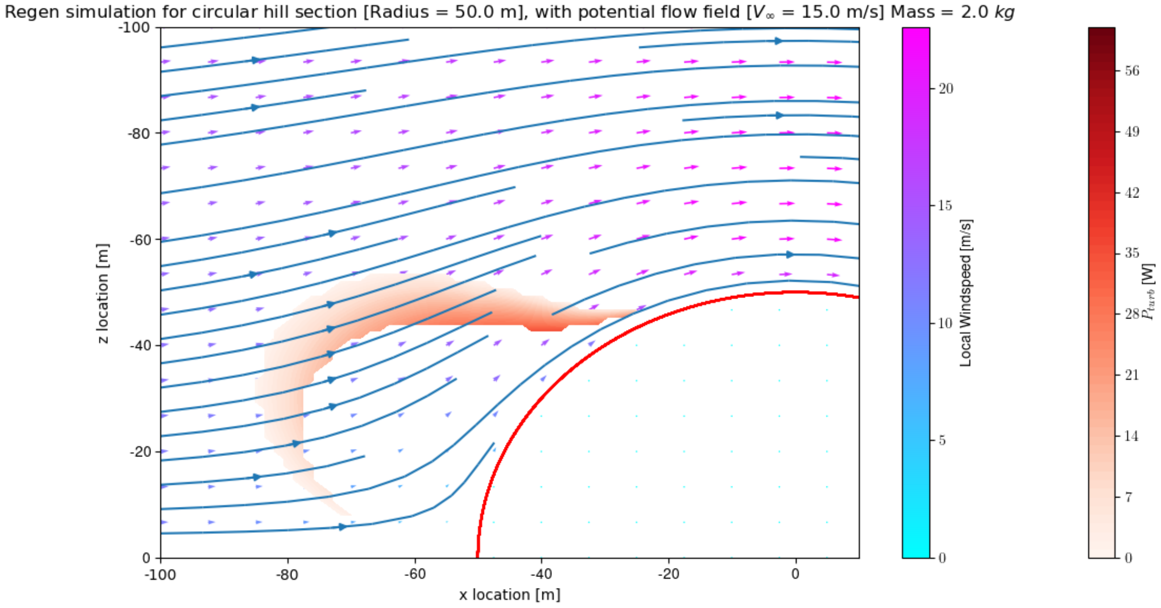


Figure 6.17: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for an aircraft mass of 2 kg for approximated parrot DISCO UAV aircraft parameters

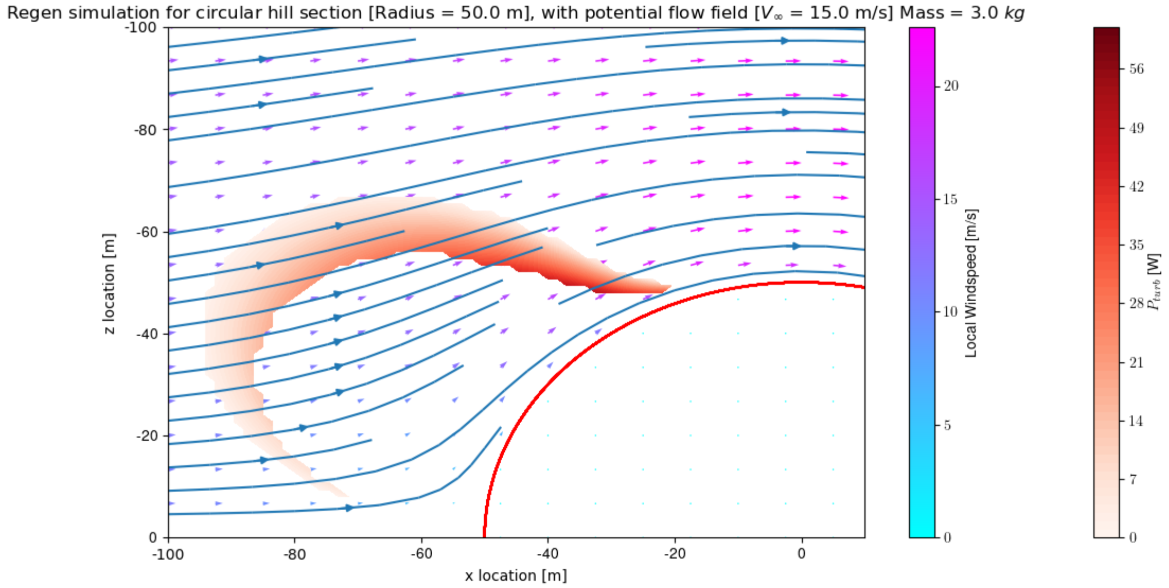


Figure 6.18: Regen power contour plot for a $15\,\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m for an aircraft mass of 3 kg for approximated parrot DISCO UAV aircraft parameters

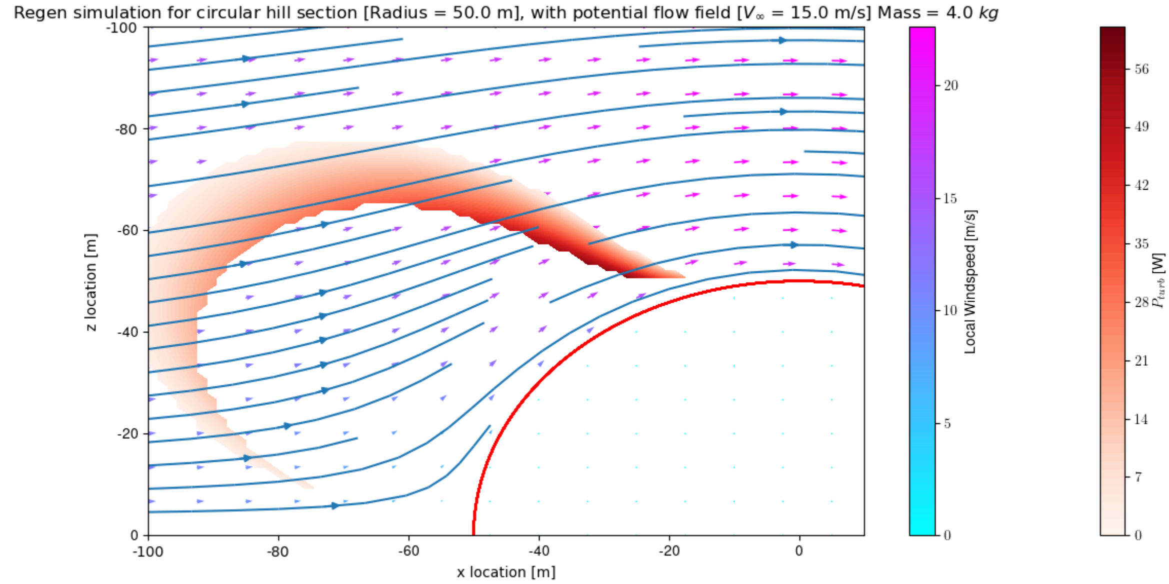


Figure 6.19: Regen power contour plot for a $15\,\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m for an aircraft mass of 4 kg for approximated parrot DISCO UAV aircraft parameters

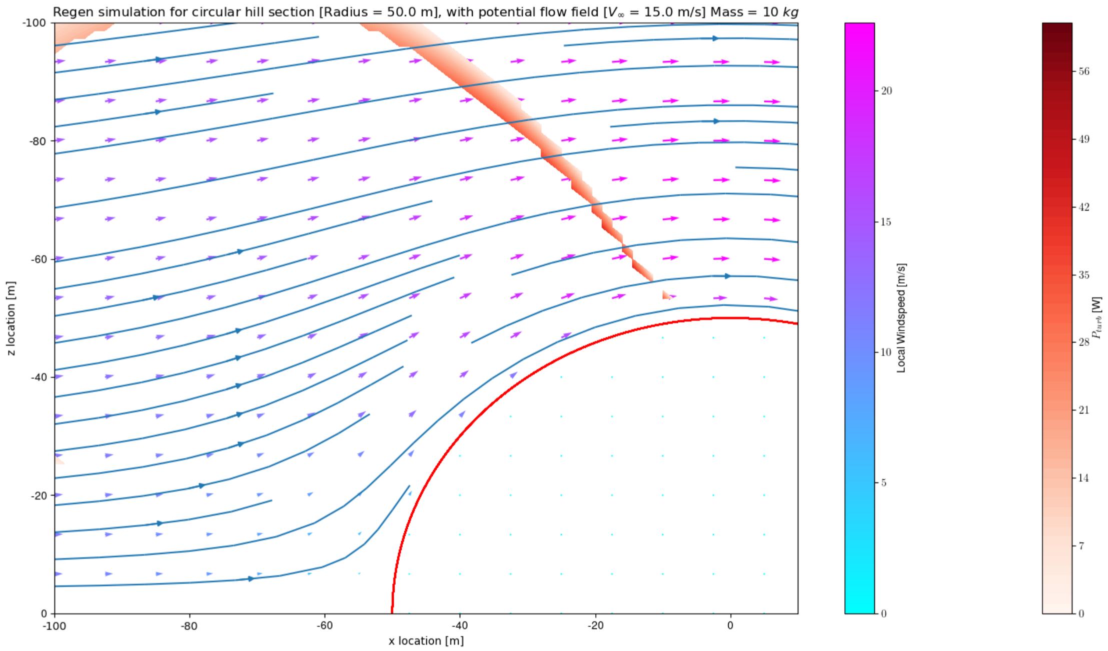


Figure 6.20: Regen power contour plot for a 15 $\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m for an aircraft mass of 10 kg for approximated parrot DISCO UAV aircraft parameters

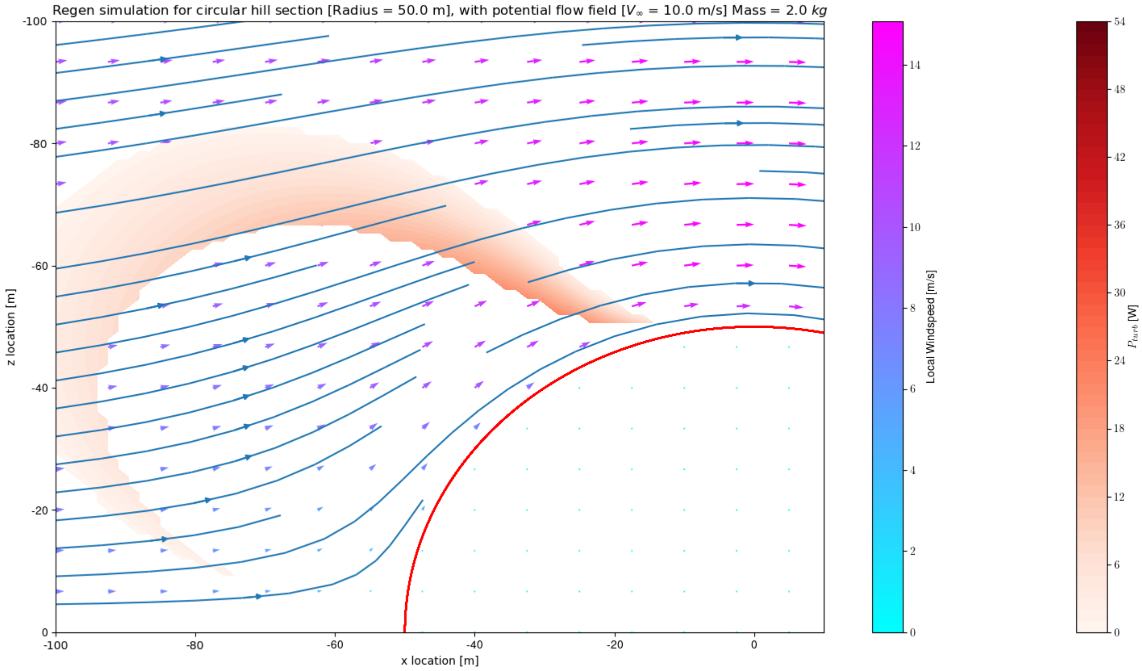


Figure 6.21: Regen power contour plot for a 10 $\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m for approximated parrot DISCO UAV aircraft parameters

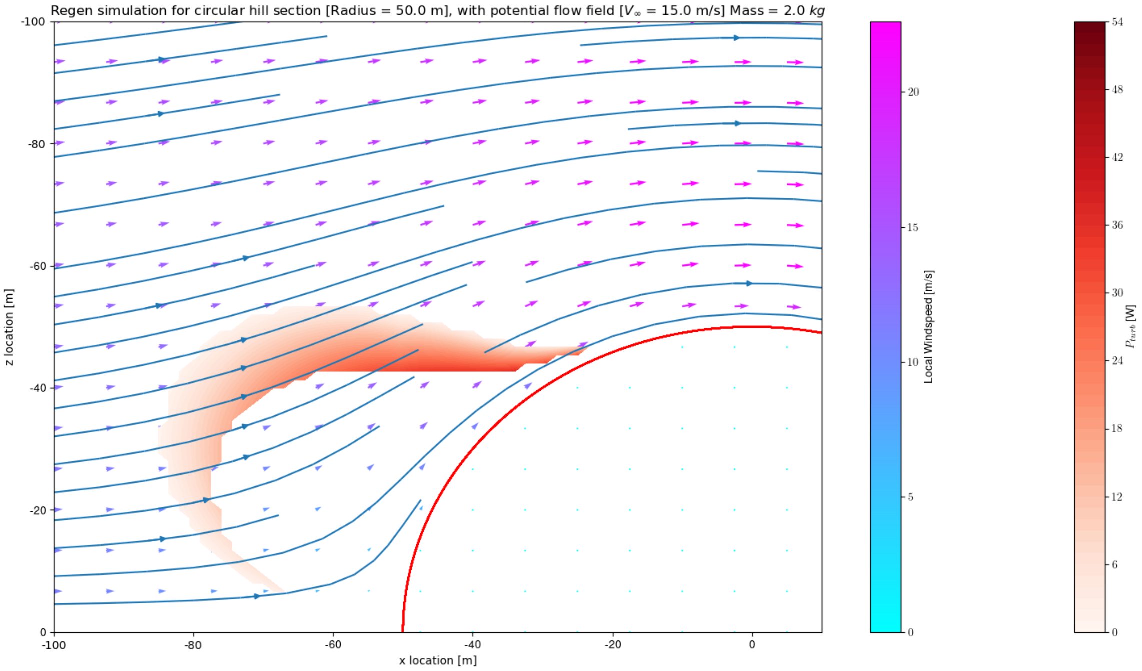


Figure 6.22: Regen power contour plot for a 15 $\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m for approximated parrot DISCO UAV aircraft parameters

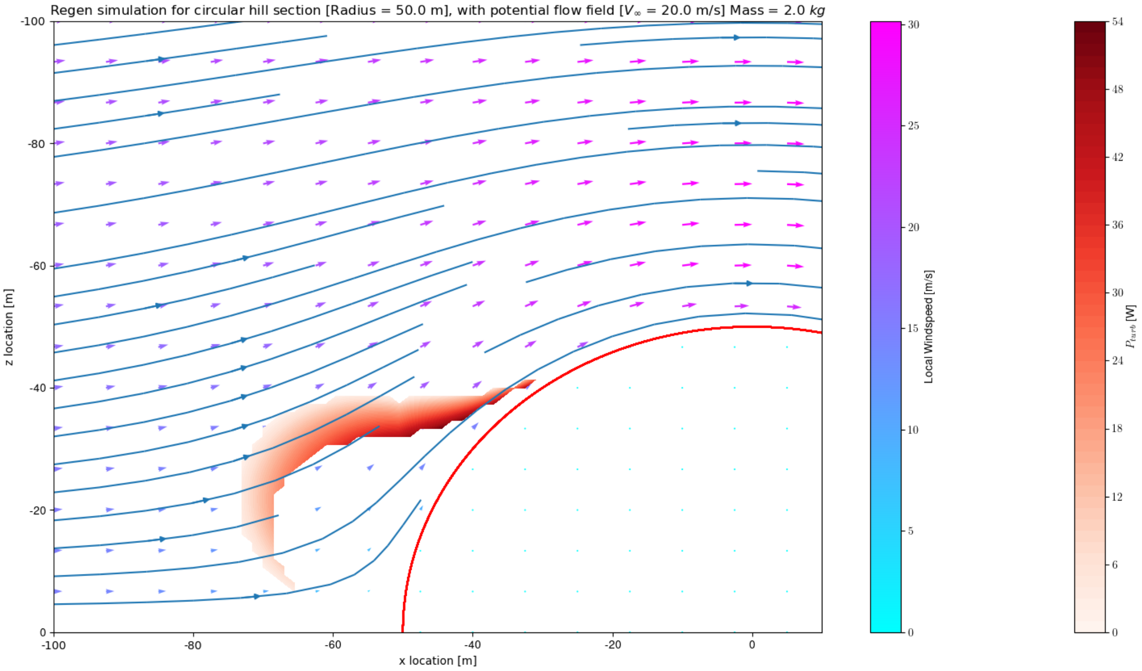


Figure 6.23: Regen power contour plot for a 20 $\mathrm{m\,s^{-1}}$ free-stream velocity over a circular hill section with a radius of 50 m for approximated parrot DISCO UAV aircraft parameters

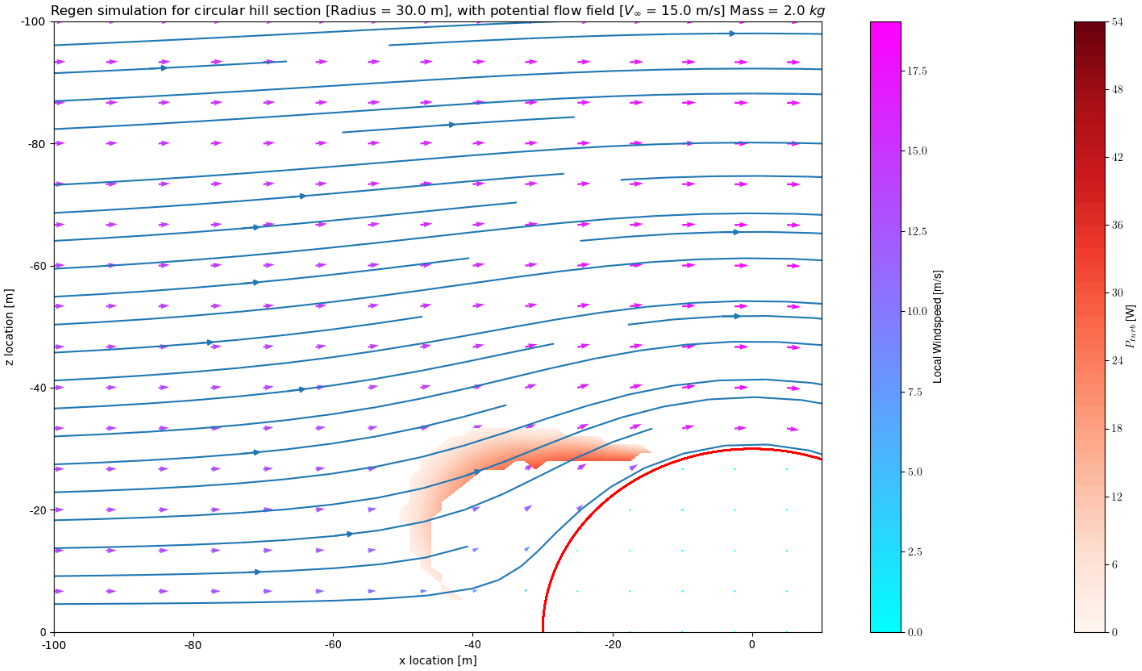


Figure 6.24: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 30 m for approximated parrot DISCO UAV aircraft parameters

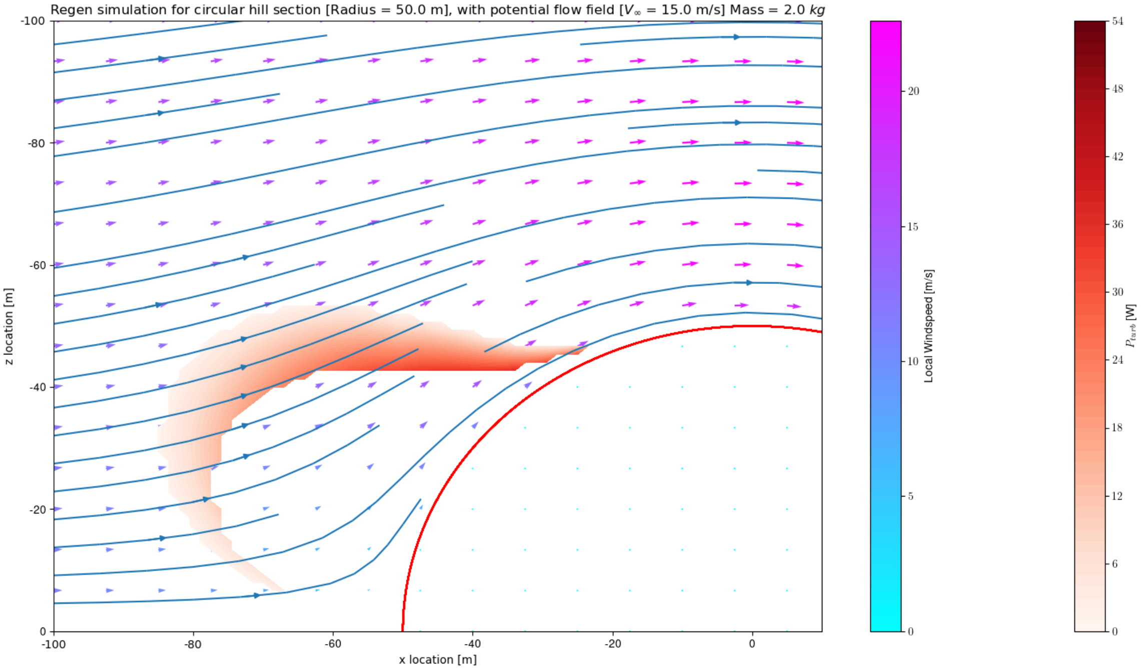


Figure 6.25: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 50 m for approximated parrot DISCO UAV aircraft parameters

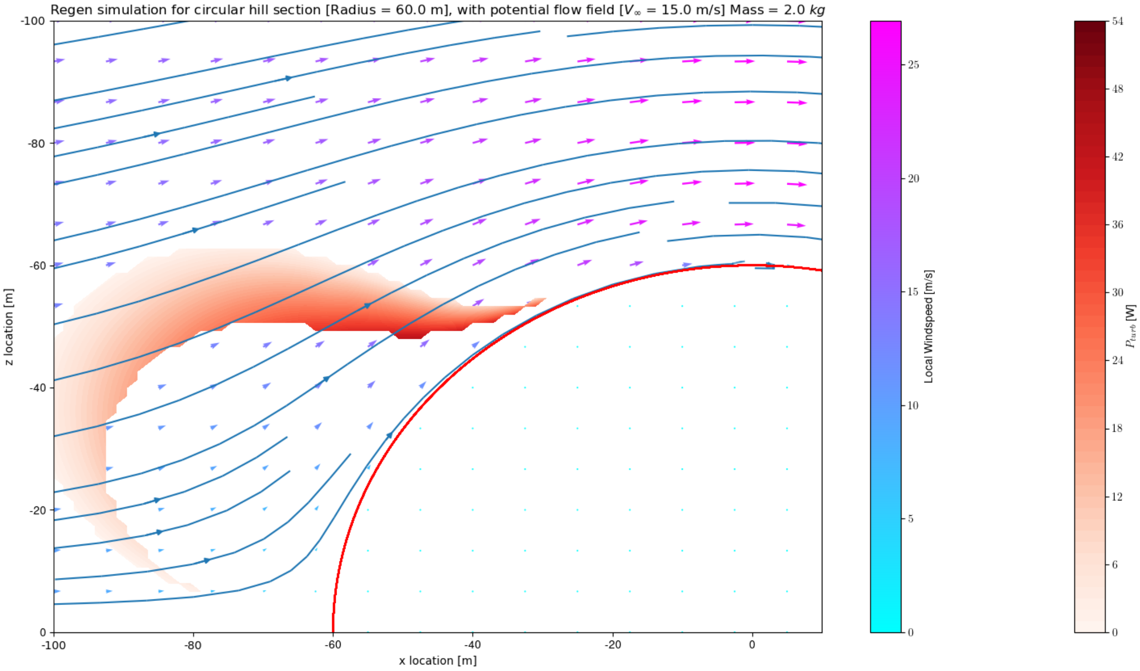


Figure 6.26: Regen power contour plot for a 15 m s$^{-1}$ free-stream velocity over a circular hill section with a radius of 60 m for approximated parrot DISCO UAV aircraft parameters

# 7
# Conclusion

In this report, the main research question, "Can a UAV harvest power from the environment by employing regenerative static hovering soaring techniques around obstacles such as hills and ships?", has been identified in Chapter 3.

In the following chapter, a comprehensive review has been made about the different types of soaring techniques and the state-of-the art research in terms of regen soaring. This proved that although regen soaring has even been deployed commercially in the form of the pipistrel electro, and utilising the updrafts around obstacles to achieve static hovering with UAVs was proven by researchers at the Melbourne institute of Technology [13], the combined concept has not been fully explored yet in literature.

From the previous research, the main components that make up the regenerative drivetrain were identified and its functionality described in Chapter 5. Furthermore, a test-setup for a proof-of-concept regen drivetrain was designed. This test-setup enabled the determination of the real-life efficiency of a regen drivetrain for UAVs. The experiments were conducted using a commercial of-the-shelf ESC, with additional supporting hardware that could change the regen braking characteristics on-the-fly, mounted on a test-bench which enabled the parameters such as torque and RPM to be determined. From this experiment it was found that the achievable efficiency for this kind of drivetrain can be up to 70%, but this figure is highly variable with regards to the input RPM of the BLDC motor, and the requested power setting. This test-setup proved that incorporating a regen drivetrain based on an ESC used in small electric RC aircraft is definitely possible.

Finally, in Chapter 6 the process on how a regen power potential simulator was created is described, comprising the wind-field estimator based on potential-flow theory to estimate the wind-flows around simple geometrical obstacles such as circular and oval hills, and the regen power estimator itself that can calculate the feasible hovering region and generate regen power contour plots to give an accurate representation of the optimum regen hovering locations.

A wind-field model was devised based on potential-flow theory and altered with a model that estimates the atmospheric boundary layer. This resulted in a model that is computationally light-weight and enables one to easily change input parameters and rerun the simulation for a variety of cases in a short time. This model was verified using a CFD simulation in the openFOAM CFD package. This also proved that the tool can be used with a wind-field from other sources than the built-in wind-field generator, like results from 3rd party CFD packages.

A regen hovering flight dynamics model was created by combining ideal wind-turbine theory with standard gliding flight dynamics equations, resulting in a set of equilibrium equations that describe the required conditions for achieving hovering. Additional equations were also derived which allows the regen power to be estimated based on the equilibrium conditions.

By combining the wind-field estimation and hovering flight dynamics models into a unified simulation package, a tool was created which can construct the regen power contour plots.

This tool was then used to run through a variety of different input conditions and scenarios, and all of the results were discussed in detail. The results conformed to the expected behaviour of the shift in location and power level shifts of the regen power region.

The resulting tool paves the way for future work, where additional simulations can be carried out to determine if trying to incorporate the suggested regen drivetrain system would result in satisfactory power levels which enables the range to be extended. With a UAV test-bench aircraft with a sufficient amount of instrumentation the model could be validated and further refined, and a control algorithm devised to achieve the regen static hovering in practice.

A technical research paper summarising the work presented in this thesis was accepted for publication in the journal of unmanned systems [15] and is included in Chapter 2.

# A

# Measurements VESC efficiency

| # | Time stamp | $I_{supply}$[A] | $V_{supply}$ [V] | $P_{supply}$ [W] | $I_{bat}$ VESC [A] | $-I_{bat}$ (multi) [A] | Torque [N*m] |
|---|---|---|---|---|---|---|---|
| 1 | 13/11/2020 17:23 | 0.72 | 8.9 | 5.7 | 0.01 | −0.049 | -0.0075 |
| 2 | 13/11/2020 17:24 | 0.99 | 9.0 | 8.9 | −0.23 | −0.007 | -0.011 |
| 3 | 13/11/2020 17:25 | 1.38 | 8.9 | 13.6 | −0.47 | 0.248 | -0.0162 |
| 4 | 13/11/2020 17:26 | 1.82 | 8.9 | 18.4 | −0.73 | 0.491 | -0.0215 |
| 5 | 13/11/2020 17:27 | 2.30 | 8.9 | 24.5 | −0.98 | 0.750 | -0.028 |

| # | ERPM | $V_{bat}$ [V] | $I_{brake}$ [A] | RPM | $P_{mech}$ [W] | $P_{bat}$ [W] | $\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | -34805 | 12.0 | 0.00 | 4972 | 3.905 | −0.588 | 0% |
| 2 | -34935 | 12.0 | 0.60 | 4991 | 5.749 | −0.084 | 0% |
| 3 | -34957 | 12.1 | 1.25 | 4994 | 8.472 | 3.001 | 35% |
| 4 | -34918 | 12.2 | 1.95 | 4988 | 11.23 | 5.990 | 53% |
| 5 | -34976 | 12.2 | 2.69 | 4997 | 14.65 | 9.150 | 62% |

Table A.1: Measured & calculated data for 5000 RPM run

| # | Time stamp | $I_{supply}$[A] | $V_{supply}$ [V] | $P_{supply}$ [W] | $I_{bat}$ VESC [A] | $-I_{bat}$ (multi) [A] | Torque [N*m] |
|---|---|---|---|---|---|---|---|
| 1 | 13/11/2020 18:41 | 0.67 | 11.98 | 8.0 | 0.02 | −0.052 | -0.0033 |
| 2 | 13/11/2020 18:42 | 1.28 | 11.94 | 15.1 | −0.51 | 0.262 | -0.0098 |
| 3 | 13/11/2020 18:43 | 1.61 | 11.91 | 19.1 | −0.74 | 0.500 | -0.0133 |
| 4 | 13/11/2020 18:42 | 1.98 | 11.89 | 23.5 | −0.97 | 0.765 | -0.0178 |
| 5 | 13/11/2020 18:47 | 2.37 | 11.86 | 27.8 | −1.19 | 1.000 | -0.0243 |

| # | ERPM | $V_{bat}$ [V] | $I_{brake}$ [A] | RPM | $P_{mech}$ [W] | $P_{bat}$ [W] | $\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | -46328 | 11.3 | 0.00 | 6618 | 2.287 | −0.5876 | 0% |
| 2 | -46474 | 11.4 | 0.94 | 6639 | 6.813 | 2.9868 | 44% |
| 3 | -46616 | 11.4 | 1.36 | 6659 | 9.275 | 5.700 | 61% |
| 4 | -46293 | 11.5 | 1.82 | 6613 | 12.327 | 8.7975 | 71% |
| 5 | -46933 | 11.6 | 2.26 | 6704 | 17.061 | 11.600 | 68% |

Table A.2: Measured & calculated data for 6500 RPM run

| # | Time stamp | $I_{supply}$[A] | $V_{supply}$ [V] | $P_{supply}$ [W] | $I_{bat}$ VESC [A] | $-I_{bat}$ (multi) [A] | Torque [N*m] |
|---|---|---|---|---|---|---|---|
| 1 | 13/11/2020 18:09 | 1.05 | 8.70 | 8.8 | 0.02 | −0.05 | -0.0017 |
| 2 | 13/11/2020 18:10 | 1.20 | 8.70 | 12.8 | −0.32 | 0.045 | -0.0055 |
| 3 | 13/11/2020 18:11 | 1.45 | 8.80 | 15.9 | −0.54 | 0.257 | -0.00988 |
| 4 | 13/11/2020 18:12 | 1.75 | 8.90 | 20.0 | −0.76 | 0.500 | -0.0134 |
| 5 | 13/11/2020 18:12 | 2.20 | 8.70 | 24.5 | −0.98 | 0.744 | -0.0185 |
| 6 | 13/11/2020 18:13 | 2.50 | 8.80 | 27.6 | −1.15 | 0.914 | -0.0220 |

| # | ERPM | $V_{bat}$ [V] | $I_{brake}$ [A] | RPM | $P_{mech}$ [W] | $P_{bat}$ [W] | $\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | -49313 | 12.0 | 0.00 | 7045 | 1.254 13 | −0.600 | 0% |
| 2 | -49631 | 12.0 | 0.56 | 7090 | 4.083 63 | 0.5400 | 13% |
| 3 | -49682 | 12.1 | 0.96 | 7097 | 7.343 22 | 3.1097 | 42% |
| 4 | -50168 | 12.2 | 1.37 | 7167 | 10.0569 | 6.1000 | 61% |
| 5 | -49608 | 12.2 | 1.85 | 7087 | 13.7295 | 9.0768 | 66% |
| 6 | -49687 | 12.3 | 2.17 | 7098 | 16.3529 | 11.2422 | 69% |

Table A.3: Measured & calculated data for 7000 RPM run

# B

# Regen field simulation code

```
"""
Regen power simulation in cylindrical/oval wind field


Author: Midas Gossye


"""

import numpy as np
from matplotlib import pyplot as plt
import csv
from matplotlib import ticker, cm
import os
from scipy import interpolate
# This function parses the turbine parameters from the MATLAB script
def get_turbine_params():

    if os.path.isfile('./D3_B2_P1.5_Airfoil4412_V1-40.csv'):
        pass
    else:
        os.system("turbinecalc.exe")
    raw_data = np.genfromtxt('D3_B2_P1.5_Airfoil4412_V1-40.csv', delimiter=",")

    #print(raw_data)

    split_idx = int((len(raw_data)) / 2) + 1

    tsrs = raw_data[0, :]
    winds = raw_data[1:split_idx, 0]
    #print(tsrs)
    #print(winds)
    powers = raw_data[1:split_idx, 1:]
    drags = raw_data[split_idx:, 1:]
    #print(powers)
    max_powers = np.amax(powers, 1)
    max_idcs = np.argmax(powers, 1)

    #print(max_idcs)

    drags = np.take(drags, max_idcs, axis=1)[:, 0]

    drag_interp = interpolate.interp1d(winds, drags, kind='quadratic',
     ↪  fill_value="extrapolate")

    return drag_interp
# Define general hill parameters

R_ridge = 50.0 # [m] Radius of the hill/ridge
U_inf = 15.0 # [m/s] Mean wind speed (negative to have flow flow right to left)
flow_type = 'circle' # Defines if hill is cylindrical or oval, options: 'oval' and
 ↪  'circle'

# Oval shaped hill parameters
a = 45 # m, defines loci x-position
x_stag = 67 # m, defines a-axis of standard oval
b = np.sqrt(x_stag**2 - a**2) # calculates b parameter of oval

surf_rough = 0.1 # m, surface roughness

oval_pointsx = np.array([])
oval_pointsz = np.array([])


def process_cfd():
    #test_arr = np.genfromtxt("SIMPLEFOAM_15ms_pot.csv", delimiter=",",
     ↪  skip_header=1)

    test_arr = np.genfromtxt("SIMPLEFOAM_15ms.csv", delimiter=",", skip_header=1)

    xs_csv = test_arr[:, 0]
    zs_csv = test_arr[:, 2]

    Winds_u_csv = test_arr[:, 3]
    Winds_v_csv = test_arr[:, 5]

    split_idx = np.where(test_arr[:, 1]==-50.7)
    #print("Split idx found in CSV:", split_idx)
    if split_idx[0][0]:
        split_idx = split_idx[0][0]
        print("Split idx found in CSV:", split_idx)
        xs_csv = xs_csv[:split_idx]
        zs_csv = zs_csv[:split_idx]
        Winds_u_csv = Winds_u_csv[:split_idx]
        Winds_v_csv = Winds_v_csv[:split_idx]



    x_step = 1.5
    z_step = 4.0 / 3.0

    # wind_u_mat = np.zeros((600, 150))
    # wind_v_mat = np.zeros((600, 150))

    wind_u_mat = np.empty((151, 201))
    wind_v_mat = np.empty((151, 201))
    wind_u_mat[:, :] = np.NaN
    wind_v_mat[:, :] = np.NaN

    max_x_idx = 0
    max_z_idx = 0

    for i in range(len(xs_csv)):
        x_idx = int(np.round((xs_csv[i] + 100.0) / x_step, 0))
        z_idx = int(np.round(zs_csv[i] / z_step, 0))
        print(x_idx, z_idx)
        wind_u_mat[z_idx, x_idx] = Winds_u_csv[i]
        wind_v_mat[z_idx, x_idx] = Winds_v_csv[i]
        max_x_idx = max(max_x_idx, x_idx)
        max_z_idx = max(max_z_idx, z_idx)

    xs_clean = np.arange(-100.0, 200 + x_step, x_step)
    zs_clean = np.arange(0, -201.33, -1.0 * z_step)

    xs_grid, zs_grid = np.meshgrid(xs_clean, zs_clean)

    #fig, ax = plt.subplots(1, 1)  # initialize pyplot figures (2 rows, 3 columns)
    #ax.invert_yaxis()  # invert y-axis on plot (represent z-axis)

    Total_wind_cfd = np.sqrt(np.power(wind_u_mat, 2) + np.power(wind_v_mat, 2))

    return xs_clean, zs_clean, xs_grid, zs_grid, wind_u_mat, wind_v_mat, Total_wind_cfd

def wind_comp(x_i, z_i, U):
    ## Wind field calculation function

    # Description: Given an x and z coordinate and the mean wind speed it calculates the wind field velocity components
    #               using potential flow theory and a logarithmic boundary layer

    if flow_type == 'oval':

        m = np.pi * U_inf / a * (x_stag ** 2 - a **2)
        v_x = U_inf + m / (2 * np.pi) * ((x_i + a) / ((x_i+0.001 + a) ** 2 + z_i ** 2) - (x_i+0.001 - a) / ((x_i+0.001 - a)**2 + z_i**2))
        v_z = -(m * z_i) / (2 * np.pi) * (1 / ((x_i+0.001 + a)**2 + z_i**2) - 1 / ((x_i+0.001 - a)**2 + z_i**2))

        ellipse_eq = (x_i ** 2) / (x_stag ** 2) + (z_i ** 2) / (b ** 2)
        if ellipse_eq < 1:
            return 0.0, 0.0

        if x_i < x_stag:

            z_ellipse = -b / x_stag * np.sqrt(x_stag ** 2 - x_i ** 2)

            mult_factor = (np.log(-(z_i - z_ellipse) / surf_rough)) / (np.log(-(-70 - z_ellipse) / surf_rough))
            if mult_factor < 0:
                print("z_ab =", -(z_i - z_ellipse), "mult =", mult_factor)
            if not (np.isnan(mult_factor)):
                v_x *= mult_factor
                v_z *= mult_factor

        else:
            if z_i < 0:
                z_ellipse = -0.01
                mult_factor = (np.log(-(z_i - z_ellipse) / surf_rough)) / (np.log(-(-70 - z_ellipse) / surf_rough))
                v_x *= mult_factor
                v_z *= mult_factor
            else:
                return 0.0, 0.0
        return v_x, v_z

    elif flow_type == 'circle':
        theta = np.arctan2(-z_i, x_i)
        r = np.sqrt(x_i**2 + z_i**2)

        global R_ridge
        u_r = (1-((R_ridge**2)/(r**2)))*U*np.cos(theta)
        u_th = -(1+((R_ridge**2)/(r**2)))*U*np.sin(theta)

        if R_ridge**2 - x_i**2 > 0:
            z_circle = -np.sqrt(R_ridge**2 - x_i**2)

        else:
            z_circle = -0.01
        mult_factor = (np.log(-(z_i - z_circle) / surf_rough)) / (np.log(-(-70 - z_circle) / surf_rough))
        u_x = np.cos(theta)*u_r - np.sin(theta)*u_th*mult_factor
        u_z = (np.sin(theta)*u_r + np.cos(theta)*u_th)*mult_factor


        if r<R_ridge:
            return 0.0,0.0

        return u_x, u_z

vwind_comp = np.vectorize(wind_comp) # Vectorizes the wind comp function so arrays can be parsed and returned by the new
                                     # vwind_comp function

def calc_jacobian(i, j, dx, dz, u_xs, u_zs):
    # calculates the jacobian of the wind field using the central difference formula for the
    # first spatial derivatives

    J = np.array([[(u_xs[j, i+1]-u_xs[j, i-1])/2*dx, (u_xs[j+1, i]-u_xs[j-1, i])/2*dz],
                  [(u_zs[j, i+1]-u_zs[j, i-1])/2*dx, (u_zs[j+1, i]-u_zs[j-1, i])/2*dz]])
    return J

def x_to_i(x_coor, max_i, x_coords):
    # converts x coordinate in meters to closest discrete i index of grid
    i = int(np.around(x_coor*(1/dx)) + x_coords[-1])
    if i>max_i:
        return max_i-2
    elif i < 0:
        return 0
    else:
        return i

def z_to_j(z_coor, max_j, z_coords):
    # converts z coordinate in meters to closest discrete j index of grid
    j = int(np.around(z_coor*(1/dz)) + z_coords[-1])
    if j>max_j:
        return max_j-2
    elif j < 0:
        return 0
    else:
        return j



x_vels = np.array([])
z_vels = np.array([])

## Specify 2D grid resolution [in meters]
dx = 0.5
dz = -(0.5)
## ======================================

## Create x and z coordinate array
x_coords = np.arange(-100, 200+dx, dx) # specifies range of x coords
z_coords = np.arange(0, -200+dz+0.1, dz)  # specifies range of z coords
## ======================================

xs, zs = np.meshgrid(x_coords, z_coords) # create meshgrid to construct wind velocity field

#Winds_u_PF, Winds_v_PF = vwind_comp(xs, zs, U_inf) # generates 2D matrices with horizontal and vertical wind velocity components
Winds_u, Winds_v = vwind_comp(xs, zs, U_inf) # generates 2D matrices with horizontal and vertical wind velocity components

#Total_wind = np.sqrt(np.power(Winds_u_PF, 2) + np.power(Winds_v_PF, 2)) # generates 2D matrix with total wind velocities
Total_wind = np.sqrt(np.power(Winds_u, 2) + np.power(Winds_v, 2)) # generates 2D matrix with total wind velocities

#x_coords, z_coords, xs, zs, Winds_u, Winds_v, Total_wind = process_cfd()

#Wind_u_diff = np.subtract(Winds_u, Winds_u_PF)
#Wind_v_diff = np.subtract(Winds_v, Winds_v_PF)

#magnitude_diff = np.sqrt(np.power(Wind_u_diff, 2) + np.power(Wind_v_diff, 2))

#cross_product_magnitudes = np.zeros(np.shape(Winds_u))

```

```python
#for i in range(np.shape(Winds_u)[0]):
#    for j in range(np.shape(Winds_u)[1]):
#        vect_1 = np.array([[Winds_u_PF[i, j],Winds_v_PF[i,j]]])
#        vect_2 = np.array([[Winds_u[i, j],Winds_v[i,j]]])
#        #print(vect_1)
#        #print(vect_1.shape)
#       # print(vect_2.shape)
#        cross_product_magnitudes[i, j] = np.abs(np.cross(vect_1, vect_2 ))

#print("coords CFD x", x_coords)
#print("coords CFD z", z_coords)

#print("Size u PF:", np.shape(Winds_u_PF))
#print("Size u CFD:", np.shape(Winds_u))

#print("Size u PF:", np.shape(Winds_v_PF))
#print("Size v CFD:", np.shape(Winds_v))

jacobians = np.zeros((len(x_coords), len(z_coords), 2,2)) # Initialize empty jacobian matrix

for i in range(1, x_coords.size-1):
    for j in range(1, z_coords.size-1):
        #print("i:", i)
        #print("j:", j)
        jacobians[i, j] = calc_jacobian(i, j, dx, dz, Winds_u, Winds_v)



if flow_type == 'circle':
    # generates cartesian coordinates where edge of hill is present (for circular shaped hill)
    thetas = np.arange(0, 180, 0.01)
    xs_hill = R_ridge*np.cos(thetas)
    zs_hill = -R_ridge*np.sin(thetas)

else:
    # generates cartesian coordinates where edge of hill is present (for oval shaped hill)
    ts = np.arange(0, 2 * np.pi, 0.01)
    u = np.tan(ts / 2)
    xs_hill = x_stag * (1 - u ** 2) / (u ** 2 + 1)
    zs_hill = -(2 * b * u) / (u ** 2 + 1)

print(Winds_u.shape)
print(Winds_v.shape)


## PLOTS FOR CHECKING DIFFERENCES BETWEEN WINDFIELDS ##

fig, ax = plt.subplots(1,1) # initialize pyplot figures (2 rows, 3 columns)
#ax[0][0].invert_yaxis() # invert y-axis on plot (represent z-axis)
ax.invert_yaxis()
skip=(slice(None,None,5),slice(None,None,5))

#magnitude_diff[magnitude_diff > 3] = np.nan
```

```
colormap_levels = np.arange(0, 10, 0.1)
#cp = ax.contourf(xs, zs, cross_product_magnitudes, colormap_levels, cmap='Reds')
#cbar = plt.colorbar(cp)
#ax[0][0].quiver(xs[skip], zs[skip], Winds_u[skip], Winds_v[skip], Total_wind[skip], cmap='Reds') # plots vector field
#qv1 = ax.quiver(xs[skip], zs[skip], Wind_u_diff[skip], Wind_v_diff[skip], magnitude_diff[skip],  cmap='cool', clim=(0.0, 10.0)) # plots vector field
qv1 = ax.quiver(xs[skip], zs[skip], Winds_u[skip], Winds_v[skip], Total_wind[skip], cmap='cool') # plots vector field
#cb1 = plt.colorbar(qv1)
#cb1.set_clim(0, 10.0)

#qv1 = ax.quiver(xs[skip], zs[skip], Winds_u[skip], Winds_v[skip], Total_wind[skip], cmap='cool') # plots vector field
#ax[0][0].streamplot(xs, zs, Winds_u, -Winds_v, density=1) # creates streamlines on plot
ax.streamplot(xs, zs, Winds_u, -Winds_v, density=1.5) # creates streamlines on plot
#ax[0][0].plot(xs_hill, zs_hill, '-r')
ax.plot(xs_hill, zs_hill, '-r')
#cbar.set_label("Cross product absolute difference")
## initial conditions for simulated aircraft in wind field
m = 2.0 # kg

S = 1.0 #0.2589 # m^2
rho = 1.225 # kg/m^3
g = 9.80665 # m/s^2
CL_alpha = 5.7 #4.8776 # 1/rad
alpha_0L = np.deg2rad(-4.0)#-1.07)
AR = 6#1.4224/0.3302
e = 0.8#0.9
CD_0 = 0.05#0.01631
W = m*g
V_a = 25 # initial airspeed for simulated aircraft m/s
gamma_a = -10*(np.pi/180) # deg/rad
dgamma_dt = 0 # rad/s
dt = 0.01 # s
t = 0 #s
x_i = 17
z_i = -5
D_prop = 100 # m
# prop diameter 1 m
Cdi_coef = 1/(np.pi*AR*e) # 1/pi*A*e

# P_maxs = (16/27)*(rho/2)* Total_wind**3 * (np.pi*D_prop**2)/4
#

S_turb_spec = 0.0314/0.2589 # 0.006 for smallest point at gridsize of 0.1


def calc_eq():
    C_L_req = W/(0.5*rho*np.power(Total_wind, 2)*S) * (np.abs(Winds_u)/Total_wind)
    C_D_req = W / (0.5 * rho * np.power(Total_wind, 2) * S) * (np.abs(Winds_v) / Total_wind)

    drag_turb_func = get_turbine_params()

    C_D_min_ach = CD_0+np.power(C_L_req, 2)*Cdi_coef
    C_D_max_ach = C_D_min_ach + 1*(2/9)*S_turb_spec#drag_turb_func(Total_wind)/(0.5* rho * np.power(Total_wind, 2) * S)#1*(2/9)*S_turb_spec

    C_D_turb = C_D_req - C_D_min_ach

    D_turb = 0.5*rho*np.power(Total_wind, 2)*S*C_D_turb

    alpha = C_L_req / CL_alpha + alpha_0L

    stall = (alpha > np.deg2rad(15)) & (alpha < np.deg2rad(-10))
    print("stall condition:", stall[stall == True])
    C_L_req[stall] = np.nan
    C_D_req[stall] = np.nan
    C_D_min_ach[stall] = np.nan
    C_D_max_ach[stall] = np.nan
    alpha[stall] = np.nan

    eq_points = (C_D_req > C_D_min_ach) & (C_D_req < C_D_max_ach)
    P_turb = D_turb*(2.0/3.0)*Total_wind
    P_turb[eq_points == False] = np.nan
    alpha[eq_points == False] = np.nan
    D_turb[eq_points == False] = np.nan
    C_D_turb[eq_points == False] = np.nan
    return P_turb, np.rad2deg(alpha), D_turb, C_D_turb

# vars to store
ts = np.array([])
V_as = np.array([])
x_is = np.array([])
z_is = np.array([])
W_xs_s = np.array([])
W_zs_s = np.array([])
Ps_turb = np.array([])
V_airs = np.array([])
#print(wind_comp(-11, -12, U_inf))

C_L_opt = np.sqrt(3*np.pi*AR*e*CD_0)
C_D_opt = CD_0 + C_L_opt**2/(np.pi*AR*e)

P_turbs, alphas_eq, D_turb, C_D_turb = calc_eq()


# while t < 1:
#
#     if np.isnan(x_i):
#         print("BIG ERROR")
#     i = x_to_i(x_i, len(x_coords)-1, x_coords)
#     j = z_to_j(z_i, len(z_coords)-1, z_coords)
#
#     U_x, U_z = wind_comp(x_i, z_i, U_inf)
#     V_air = np.sqrt(U_x**2 + U_z**2)
#     V_airs = np.append(V_airs, V_air)
#     U_z = -1*U_z
#     x_dot_i = V_a * np.cos(gamma_a) + U_x
#     z_dot_i = -V_a * np.sin(gamma_a) + U_z
#     W_xs_s = np.append(W_xs_s, U_x)
#     W_zs_s = np.append(W_zs_s, U_z)
#     #print(i, j)
#     gamma_test = np.arctan2(z_dot_i-U_z, x_dot_i -  U_x)
#     #print(gamma_test*180/np.pi)
#
#
#     M1 = np.array([np.sin(gamma_a), np.cos(gamma_a)])
#     M3 = np.array([x_dot_i, z_dot_i])
#     M3 = np.reshape(M3, (2,1))
#     temp_M = np.matmul(M1, jacobians[i,j])
#     #print("temp_M: ", temp_M)
#     #print("M3: ", M3)
#     MUL = np.matmul(temp_M, M3)
#
#     L_m = g*np.cos(gamma_a) + V_a*dgamma_dt - MUL
#
#     C_L = L_m*m / (0.5*rho*V_a**2*S)
#     C_D = CD_0 + (C_L**2)/(np.pi*AR*e)
#
#     Drag_m_turbine = (1/9*m) * rho * V_a**2 * np.pi * D_prop**2
#     P_turbine = Drag_m_turbine*m
#     Ps_turb = np.append(Ps_turb, P_turbine)
#
#
#     D_m = (1/(2*m)) * rho * V_a**2 * S * CD_0 + ((m*(L_m)**2)) / (0.5*rho*V_a**2 * S * np.pi * AR * e) + Drag_m_turbine
#     tan_gamma = (Drag_m_turbine + D_m)/(L_m)
#     gamma_goal = np.arctan2(Drag_m_turbine + D_m, L_m)
#
#     # if gamma_goal-gamma_test > 2*(np.pi/180):
#     #     dgamma_dt = -15*(np.pi/180)
#     # elif gamma_goal-gamma_test < -2*(np.pi/180):
#     #     dgamma_dt = 15 * (np.pi / 180)
#     # else:
#     #     dgamma_dt = 0
#
#     #print(gamma_goal*(180/np.pi))
#     N1 = np.array([np.cos(gamma_a), -np.sin(gamma_a)])
#     N3 = np.array([x_dot_i, z_dot_i])
#     N3 = np.reshape(N3, (2, 1))
#     temp_N = np.matmul(N1, jacobians[i,j])
#     #print("temp_N: ", temp_N)
#     #print("N3: ", N3)
#     NUL = np.matmul(temp_N, N3)
#     dVa_dt = -g*np.sin(gamma_a) - D_m - NUL
#
#     ts = np.append(ts, t)
#     V_as = np.append(V_as, V_a)
#     x_is = np.append(x_is, x_i)
#     z_is = np.append(z_is, z_i)
#
#     V_a += dVa_dt*dt
#     x_i += x_dot_i*dt
#     z_i += z_dot_i*dt
#     gamma_a += dgamma_dt*dt
```

```python
#     t += dt

def get_local_min_h_dot(V_loc):
    min_h_dot = (rho * S * CD_0) / (2 * m * g) * V_loc ** 3 + (2 * m * g * Cdi_coef) / (rho * S * V_loc)
    return min_h_dot

plt.rcParams.update({
    "text.usetex": True,
    "font.family": "sans-serif",
    "font.sans-serif": ["Helvetica"]})
#ax[0][0].plot(x_is, z_is)
ax.plot(x_is, z_is)
#min_h_dot = np.sqrt(W/(0.5*rho*S))*C_D_opt/(C_L_opt**(1.5))
#print(P_maxs)
P_turbs = np.ma.masked_where(Winds_v <= get_local_min_h_dot(Total_wind), P_turbs) # changed to P_maxs
P_turb_max = np.nanmax(P_turbs)

P_betz = (16/27) * 0.5 * rho * S_turb_spec*S * np.power(Total_wind, 3)
print("max P_turb:", P_turb_max)
#5 * round(P_turb_max/5)+0.1
D_turbs = np.ma.masked_where(Winds_v <= get_local_min_h_dot(Total_wind), D_turb)
print(P_turbs)
#colormap_levels = np.arange(0, 5 * round(P_turb_max/5)+0.1, 1.0)
colormap_levels = np.arange(0, 55, 1.0)
#colormap_levels2 = np.arange(0, 1 * round(np.nanmax(D_turbs)/1), 0.1)
#colormap_levels2 = np.arange(0, 0.025, 0.001)

#colormap_levels = 50
#cp = ax[0][0].contourf(xs, zs, P_turbs , colormap_levels,cmap='Reds')#locator=ticker.LogLocator(subs=0.5), cmap='Reds')
#cp = ax.contourf(xs, zs, P_turbs, colormap_levels, cmap='Reds')#locator=ticker.LogLocator(subs=0.5), cmap='Reds')
cp = ax.contourf(xs, zs, P_turbs, colormap_levels, cmap='Reds')
#cp = ax.contourf(xs, zs, C_D_turb, colormap_levels2, cmap='Reds')
#cp3 = ax.contourf(xs, zs, D_turbs, colormap_levels2, cmap='Reds')

xs_eq = np.ma.masked_where(np.isnan(P_turbs), xs)
zs_eq = np.ma.masked_where(np.isnan(P_turbs), zs)
#ax.plot(xs_eq, zs_eq, '*r')
cbar = plt.colorbar(cp)
#cbar3 = plt.colorbar(cp3)
cbar.set_label("$P_{turb}$ [W]")
#cbar.set_label("$C_{D_{turb}}$ [-]")
#cbar3.set_label("Max Turbine Drag [N]")

cbar2 = plt.colorbar(qv1)
cbar2.set_label("Local Windspeed [m/s]")

#ax[0][0].set_xlabel('x location [m]')
ax.set_xlabel('x location [m]')
#ax[0][0].set_ylabel('z location [m]')
ax.set_ylabel('z location [m]')
#str(np.around(S_turb_spec*100, 2)) + ' % of S'
```

```
#ax.set_title('Potential flow field power contours for circular hill section [Radius
 ↪  = {} m]'.format(R_ridge) + ' [$V_{\infty}$ = ' + str(U_inf) + ' m/s]')
ax.set_title('Regen simulation for circular hill section [Radius = {} m], with
 ↪  potential flow field'.format(R_ridge) + ' [$V_{\infty}$ = ' + str(U_inf) + ' m/s]
 ↪  ' + 'Mass = ' + str(m) + ' $kg$')
# ax[0][1].plot(x_is, V_as)
# ax[0][1].set_ylabel('Airspeed [m/s]')
# ax[1][0].plot(x_is, W_xs_s)
# ax[1][0].set_ylabel('Wx [m/s]')
# ax[1][0].set_xlabel('x location [m]')
# ax[1][1].plot(x_is, W_zs_s)
# ax[1][1].set_ylabel('Wz [m/s]')
# ax[1][1].set_xlabel('x location [m]')
# ax[0][2].plot(x_is, Ps_turb)
# ax[0][2].set_ylabel('P_turb [W]')
# ax[0][2].set_xlabel('x location [m]')
#ax.plot(oval_pointsx, oval_pointsz, '*b')
ax.set_xlim([-100, 10])
ax.set_ylim([0,-100])


plt.show()

np.savetxt("power_vals.csv", P_turbs, delimiter=",")
np.savetxt("total_winds.csv", Total_wind, delimiter=",")
np.savetxt("winds_u.csv", Winds_u, delimiter=",")
np.savetxt("winds_v.csv", Winds_v, delimiter=",")
```

# Bibliography

[1] John D. Anderson. *Fundamentals of aerodynamics*. McGraw-Hill, 5th edition, 2011. URL http://www.worldcat.org/isbn/9780073398105.

[2] John Philip Barnes. Regenerative electric flight: Synergy and integration of dual-role machines. *53rd AIAA Aerospace Sciences Meeting*, (January):1–15, 2015. doi: 10.2514/6.2015-1302.

[3] S. Besio, A. Mazzino, and C. F. Ratto. Local log-law-of-the-wall in neutrally-stratified boundary-layer flows. *Boundary-Layer Meteorology*, 107(1):115–142, 2003. ISSN 00068314. doi: 10.1023/A:1021599714936.

[4] Mark Boslough. Autonomous dynamic soaring. *IEEE Aerospace Conference Proceedings*, 2017. ISSN 1095323X. doi: 10.1109/AERO.2017.7943967.

[5] Jacques Boyer. The Soaring Flight of Birds Attained Mechanically. *Scientific American*, 99(4):63–63, 1908. ISSN 0036-8733. doi: 10.1038/scientificamerican07251908-63.

[6] Christopher Earls Brennen. Potential Flow around a Cylinder. *An Internet Book on Fluid Dynamics*, pages 2–4, 2004. URL http://brennen.caltech.edu/FLUIDBOOK/basicfluiddynamics/potentialflow/singularities/cylinder.pdf.

[7] Ramiro Carvalho. Development of the regenerative soarer: Theoretical and Practical aspects. In *31st Congress of the International Council of the Aeronautical Sciences*, number September, Belo Horizonte, 2018. URL https://www.researchgate.net/publication/328883394_DEVELOPMENT_OF_THE_REGENERATIVE_SOARER_THEORETICAL_AND_PRACTICAL_ASPECTS.

[8] Rahul Ganpati Chougale and C. R. Lakade. Regenerative braking system of electric vehicle driven by brushless DC motor using fuzzy logic. *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, 61(10):2167–2171, 2018. doi: 10.1109/ICPCSI.2017.8392101.

[9] Clarence D Cone. Thermal soaring of birds. *FRONTIERS OF ZEALOUS RESEARCH*, 50(1):180–209, 1962. URL http://www.jstor.com/stable/27838346.

[10] Stefan Emeis and Matthias Turk. Comparison of Logarithmic Wind Profiles and Power Law Wind Profiles and their Applicability for Offshore Wind Profiles. *Wind Energy*, pages 61–64, 2007. doi: 10.1007/978-3-540-33866-6{_}11.

[11] D Eržen, M Andrejašič, R Lapuh, J Tomažič, Č Gorup, and T Kosel. An Optimal Propeller Design for In-flight Power Recuperation on an Electric Aircraft. *Aviation Technology*, 2018. doi: 10.2514/6.2018-3206. URL http://arc.aiaa.org.

[12] Federal Aviation Administration. Glider Performance. In *FAA Glider handbook*, pages 5–16. 2013.

[13] Alex Fisher, Matthew Marino, Reece Clothier, Simon Watkins, Liam Peters, and Jennifer L. Palmer. Emulating avian orographic soaring with a small autonomous glider. *Bioinspiration and Biomimetics*, 11(1), 12 2015. ISSN 17483190. doi: 10.1088/1748-3190/11/1/016002.

[14] N. Gavrilovic, A. Mohamed, M. Marino, S. Watkins, J. M. Moschetta, and E. Benard. Avian-inspired energy-harvesting from atmospheric phenomena for small UAVs. *Bioinspiration and Biomimetics*, 14 (1), 2019. ISSN 17483190. doi: 10.1088/1748-3190/aaec61.

[15] M. Gossye, S. Hwang, and B. D. W. Remes. Developing a Modular Tool to Simulate Regeneration Power Potential Using Orographic Wind-Hovering UAV's. *Unmanned Systems*, 4 2022. ISSN 2301-3850. doi: 10.1142/S2301385022410047.

[16] Sreenivas Jayanti. *Computational Fluid Dynamics for Engineers and Scientists.* 2018. ISBN 9789402412178. doi: 10.1007/978-94-024-1217-8.

[17] Urs Kafader. maxon Motors as Generators – maxon Support, 2020. URL `https://support.maxongroup.com/hc/en-us/articles/360004496254-maxon-Motors-as-Generators`.

[18] Jack W. Langelaan. Long distance/duration trajectory optimization for small UAVs. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2007*, 4(August):3654–3667, 2007. doi: 10.2514/6.2007-6737.

[19] N R J Lawrance, J J Acevedo, J J Chung, J L Nguyen, D Wilson, and S Sukkarieh. Long Endurance Autonomous Flight for Unmanned Aerial Vehicles Aerial Robotics Long Endurance Autonomous Flight for Unmanned Aerial Vehicles. *AerospaceLab*, (8), 2014. doi: 10.12762/2014.AL08. URL `https://arc.aiaa.org/doi/pdf/10.2514/6.2009-6112`.

[20] Nicholas R.J. Lawrance and Salah Sukkarieh. Wind energy based path planning for a small gliding unmanned aerial vehicle. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2009. ISBN 9781563479786. doi: 10.2514/6.2009-6112. URL `https://arc.aiaa.org/doi/pdf/10.2514/6.2009-6112`.

[21] Michael J. Logan, Julio Chu, Mark A. Motter, Dennis L. Carter, Michael Ol, and Cale Zeune. Small UAV research and evolution in long endurance electric powered vehicles. *Collection of Technical Papers - 2007 AIAA InfoTech at Aerospace Conference*, 1(May):201–207, 2007. doi: 10.2514/6.2007-2730.

[22] Paul MacCready. Regenerative battery-augmented soaring. In *Self-Launching Sailplane Symposium*, 1998.

[23] Geoffrey J. May, Alistair Davidson, and Boris Monahov. Lead batteries for utility energy storage: A review. *Journal of Energy Storage*, 15:145–157, 2018. ISSN 2352152X. doi: 10.1016/j.est.2017.11.008. URL `http://dx.doi.org/10.1016/j.est.2017.11.008`.

[24] Openfoamwiki. SimpleFoam - OpenFOAMWiki. URL `https://openfoamwiki.net/index.php/SimpleFoam`.

[25] Magdi Ragheb and Adam M. Ragheb. Wind Turbines Theory - The Betz Equation and Optimal Rotor Tip Speed Ratio. In Rupp Carriveau, editor, *Fundamental and Advanced Topics in Wind Power*, page 21. Intech, 2011. ISBN 978- 953-307-508-2. doi: http://dx.doi.org/10.5772/57353. URL `https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics`.

[26] David Segersson. A tutorial to urban wind flow using OpenFOAM. *Proceedings of CFD with Open-Source Software*, pages 1–46, 2017. doi: http://dx.doi.org/10.17196/OS{_}CFD{\#}YEAR{_}2017. URL `http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2017/DavidSegersson/report_David_Segersson.pdf`.

[27] A A Sonin. 2 .25 Advanced Fluid Mechanics. (c):2–4, 2011.

[28] Texas Instruments. Basics of Ideal Diodes, 2019. URL `https://www.ti.com/lit/an/slvae57a/slvae57a.pdf?ts=1626199039333`.

[29] Jiaqun Xu and Haotian Cui. Regenerative brake of brushless DC motor for light electric vehicle. *2015 18th International Conference on Electrical Machines and Systems, ICEMS 2015*, (0):1423–1428, 2016. doi: 10.1109/ICEMS.2015.7385262.