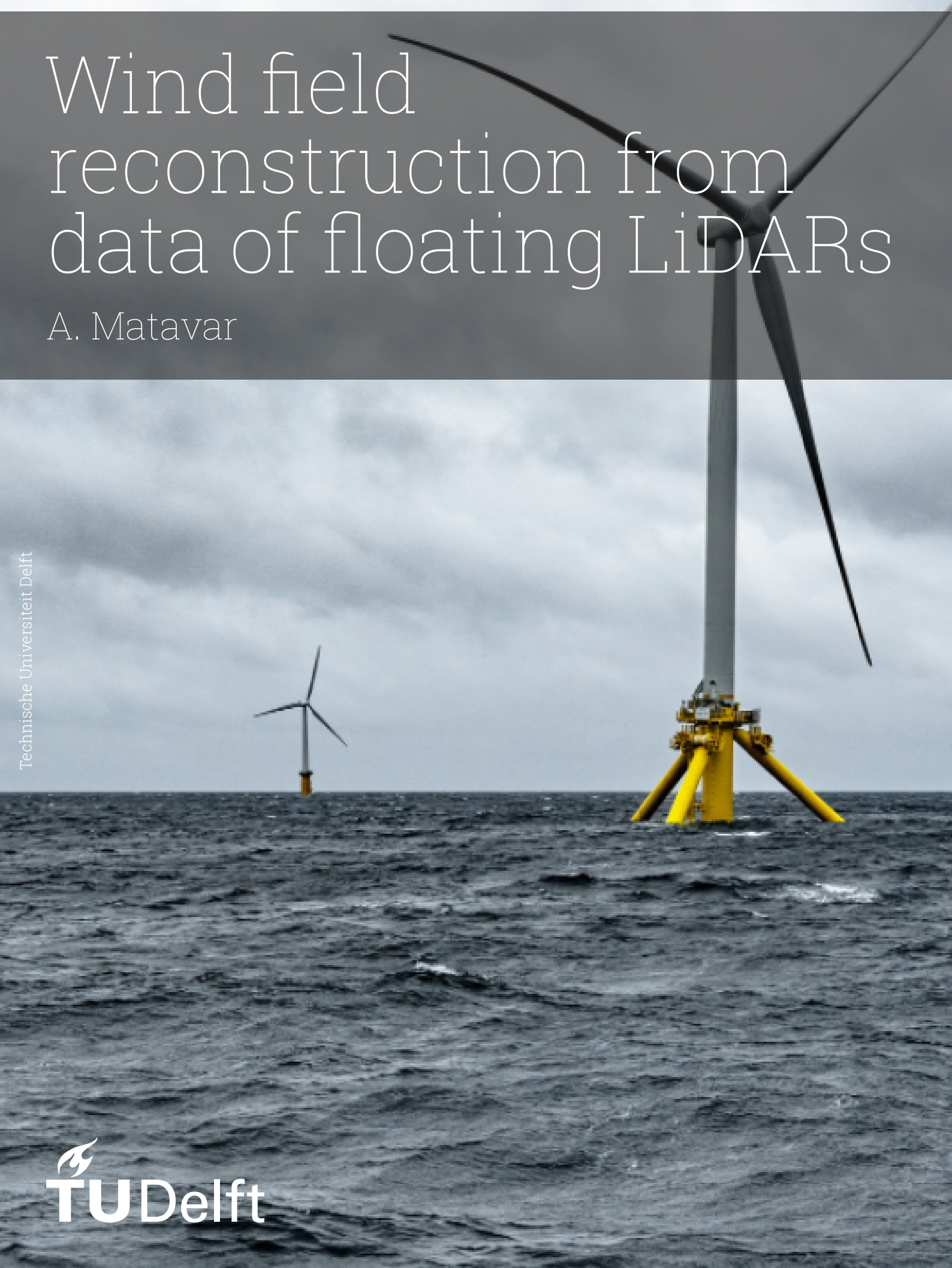
Wind field reconstruction from data of floating LiDARs

A. Matavar



Technische Universiteit Delft

TU Delft

Wind field reconstruction from data of floating LiDARs

by

A. Matavar

Final version - 20/08/2024

Student number:	5856809	
Project duration:	November 2023 – August 2024	
Thesis committee:	Dr. ir. W. Bierbooms,	TU Delft, supervisor
	Prof. dr. ir. A. Viré.,	TU Delft
	Dr. R. Dwight,	TU Delft
	Dr. N. Warncke,	Siemens Gamesa Renewable Energy

TU Delft

Abstract

Floating offshore wind turbines are growing to be significant parts of the renewable energy system that will help meet the energy demand of a growing population with limited resources. Such floating structures offer the opportunity of harnessing wind power at deeper sea than fixed-bottom counterparts. However, they remain a novel concept and, as all emerging technologies do, they still require considerable progress before their learning curve enables complete operational maturity of the concept. An important part of prototype certification and development of wind turbine projects is the validation of loads and power curve.

For this, an accurate representation of the wind field parameters is needed at the site of the turbine, which requires a sensing device. Due to the complexity of installing met masts at deep seas, nacelle-mounted lidars appear as a promising option. They suffer however from the motion of the wind turbine in such environments, altering the velocities measured by the lidar, but also their measurement heights. The scope of this report was to study how to correct for this motion, account for it in the wind field reconstruction and measure its impact.

The method developed is novel in the sense that it tracks in real time the location of the lidar, its orientation, velocity and angular velocity. The impact of the lidar motion was first assessed through simulations using an OpenFAST floater model. The results show that existing methods that do not account precisely for the lidar's motion perform reasonably well in the simulated case, and that in particular the lidar's velocity interference has a negligible impact on the reconstructed mean quantities. However, the developed method has a significant advantage, being that it does not seem to suffer from the lidar's motion and that it allows for uncertainty quantification of the reconstructed profile. What was seen however is that for turbulence reconstruction, the lidar's velocity interference still has significant impact on the spectrum, and thus will impact the quality of the reconstructed wind velocity fluctuations.

Regarding real applications, nacelle-mounted lidars are being tested in floating prototypes, paired with Inertial Navigation System devices. The output of these devices are generally offset in time with the outputs of the lidar, which creates a need for interpolation of the motion data. This was also one of the main scopes of this project. It was shown that interpolating position, velocity and acceleration using Quintic Hermite basis functions is a robust option that can be used with noisy signals. Next, for the interpolation of orientation and angular velocity, this was, as expected, a much more complex problem. Some novel quaternion interpolation methods were developed and presented here. These methods are promising but they need further study on the optimal positioning of the quaternionic nodes, involving least-squares fitting as it was done in the real number case for position, velocity and acceleration. Challenges remain in this regard, especially requiring uncertainty quantification of the reconstruction when using imperfect motion data from an inertial device.

Preface

This Master's thesis carried out with Siemens Gamesa represents the culmination of my degree at TU Delft. This project has been marked by numerous challenges, characteristic of a first experience in thesis work. It has been an immensely enriching period in many aspects.

I had the chance to work with passionate people, eager to share their knowledge. These ten months dedicated to this thesis are summarized in this report, which serves as a showcase for the hard work done to achieve my goals. This journey would not have been possible without the help, advice, support, and exchanges with all the people I met during this project.

I would like to extend my special thanks to my supervisor, M. Norbert Warncke. He generously shared his experience and professional expertise, while always providing precise advice. Norbert consistently gave me his time when I needed it, offered guidance, and showed great pedagogy in teaching me more rigorous working methods. I am deeply grateful for his warm welcome to the team.

I would also like to express my gratitude to Wim Bierbooms for his continuous support throughout this project, as well as for his advice and feedback, which allowed me to gain perspective and improve week by week. Thank you for the time spent and the thoughtful guidance.

A big thank you also to the entire SGRE team, both in France and the Netherlands. Thank you to all the students and mentors with whom I had the chance to collaborate, whether closely or from afar, for sharing our projects and the fascinating discussions we had. It was an enriching experience to work with such dedicated individuals, driven by a desire to share.

I would also like to express my special thanks to my fellow students and the SGRE team at the The Hague office for welcoming me and for creating such a pleasant atmosphere. I am confident that the project will be in good hands with Matti continuing the work on this topic.

I would also like to thank Paul Deglaire for the trust he placed in me throughout this project, for involving me in the life of the company, and for his confidence in my ability to carry this work forward.

Finally, I wish to express my deep gratitude to my friends, my parents, and my partner for their unwavering support and constant interest in my project and its progress.

A. Matavar
Delft, August 2024

Contents

Nomenclature

Abbreviations

ACF	Auto-Correlation Function
CW	Continuous-Wave
DFT	Discrete Fourier Transform
DOF	Degree of freedom
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
FOWT	Floating Offshore Wind Turbine
FWHM	Full Width at Half Maximum
GHG	Greenhouse gas
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LCOE	Levelized Cost Of Energy
LO	Local Oscillator
LOS	Line of sight
MRU	Motion Reference Unit
MSE	Mean Square Error
MSL	Mean Sea Level
NED	North-East-Down
PDF	Probability Density Function
PSD	Power Spectral Density
RMSE	Root Mean Square Error
SDG	Sustainable Development Goal
Slerp	Spherical linear interpolation
SNR	Signal-to-Noise Ratio
TEM	Transverse Electromagnetic Mode

Greek Symbols

β	Backscatterer coefficient ($\mathrm{m}^{-1}\,\mathrm{sr}^{-1}$)
η	Efficiency (-)

λ	Wavelength (m)
ϕ	Wave or signal phase (rad)

Roman Symbols

c	Speed of light $(\mathrm{m\,s^{-1}})$
E	Electric field $(\mathrm{V\,m^{-1}})$
F	Focus range (m)
f	Frequency (Hz)
H	Set of quaternions (-)
H_1	Set of unit quaternions (-)
I	Intensity $(\mathrm{W\,m^{-2}})$
k	Wave number $(\mathrm{rad\,m^{-1}})$
P	Power (W)
q	Unit rotation quaternion (-)
R	Lidar sensing range (m)
ΔR	Range resolution (m)
$\boldsymbol{R}$	Rotation matrix (-)
t_r	Round-trip time of flight (s)
T_{ext}	Atmospheric extinction (-)
V_r	Radial velocity $(\mathrm{m\,s^{-1}})$
w	Beam radius (m)
w_0	Beam waist (m)
z_R	Rayleigh length (m)

1

Introduction

Among the 17 Sustainable Development Goals (SDG) set by the United Nations in 2015, one can find ”Affordable and clean energy” [80]. First, the development of affordable energy goes hand in hand with an increasing access to energy for the worldwide population. Importantly, more access to energy, and in particular to electricity has been generally linked to economic development [24]. Thus, it can be argued that participating in the improvement of energy technologies concerns many more that only one SDG. Especially, this link between electricity use and development tends to be more prevalent for developing countries. In fact, the UN estimates that at the current pace, about 660 million people will still lack access to electricity in 2030, mainly in developing countries [80]. The need for energy affordability becomes here more apparent. Since international public financing for clean energy in developing countries continues to decline, one way of accelerating the electricity access rate in these countries is to make the installation and the maintenance of energy production facilities cheaper. In some technologies already, such as wind energy conversion systems, emerging countries play a significant role and are expected to make a rapid and global transition [41]. This is the case for example for Philippines and Vietnam, that are among the leading manufacturing countries of wind turbine. For developed countries, what is at stake is to make the already widespread energy cheaper and more sustainable, through electrification and the expansion of renewable energy. In the European Union, this matter is tackled by the *European Green Deal*. While the objective is to reduce the greenhouse gas (GHG) emissions by at least 55% by 2030, the EU needs to continue to ensure accessible electricity for households, especially in times where the electricity prices recently reached an all-time high [75].

Secondly, now that the importance of affordable energy was addressed, the need for so-called clean energy will be approached. An energy source is considered clean if its impact on the environment is relatively low, so it is highly related to sustainability. If the measure of affordability is straight forward through the price indicator, a quantitative measurement for cleanliness of energy is harder to define. For this, an indicator that can be used is the life-cycle GHG emission intensity, that is for a certain technology the quantity of GHG emitted per unit of energy produced, on average during its lifetime. The unit is generally in tonne CO_2e/GWh. The emission of GHG into the atmosphere is critical for the global warming of the planet around the world, which in turn has tremendous effects on lands and population. According to the International Panel on Climate Change, limiting the global warming to 1.5°C after 2030 -or if not, 2°C- would limit the global impact. Currently, an abrupt acceleration of efforts is needed to achieve that goal [56]. In 2022 in the EU, 39% of electricity was generated by renewable sources and the ambition is to reach 45% by 2030 [75]. Given their advantages, solar and wind energy are in the spotlight. A third objective for energy production can finally be defined, as it was done by the European Commission: security of energy. This means having diverse and resilient energy systems, that are able to respond to extreme events, whether they are natural, geopolitical or technological.

In all these regards, and reaching the object of this project, wind energy is a promising source. Regardless of the metal dependency problem, wind turbines have the potential to provide secure energy for their lifetime of around 20 years. Secondly, one can question the sustainability of wind power, in comparison to other energy conversion systems. According to a report from the World Nuclear Association [13], wind energy ranks as the lowest emitting source with hydropower, at 26 tonnes CO_2e/GWh.

To put this number into perspective, coal power plants typically operate emitting more than 800 tonnes CO_2e/GWh. Consequently, wind energy seems extremely pertinent when focusing on the sustainability criterion.

Finally, considering the Levelized Cost Of Energy (lifetime costs of a system divided by its energy production), onshore wind energy was in 2018 and in the EU the energy source with the lowest average price. Indeed, onshore wind energy price was around 60 €/MWh [25], followed by offshore wind energy with 85 €/MWh. Regarding other common energy sources, the same report commissioned by the European Directorate-General for Energy noted a price of 87 €/MWh for utility-scale solar PV, 90 €/MWh for coal-fired power plants and 95 €/MWh Combined Cycle Gas Turbines plants (before the recent gas crisis). A final relevant comparison is that with nuclear power, for which the lowest LCOE of the G20 countries was attained in China, with 67 €/MWh. Thus, wind energy naturally seems to be an excellent source of energy.

Leading to the actual content of this project, where do floating lidars stand in the development of wind energy? A LiDAR (Light Detection And Ranging) is a remote sensing technology that can measure the wind speed at a given range, often several hundreds of meters away. To be more specific, in the context of this report, floating lidars should be seen as lidars mounted onto the nacelle of a Floating Offshore Wind Turbine (FOWT), and not on buoys for example. The objective of measuring the wind with nacelle-mounted lidars in this case is twofold. The first is for FOWT certification, namely power curve and loads validation. Power curves have a key role for different applications in wind industry, such as turbine selection, capacity factor estimation, energy assessment and forecasting (Annual Energy Prediction) [5]. Having a lidar placed on the nacelle of a floating turbine can help determine and verify the real power curve: associating the measured wind speed at hub height and the actual power produced. Then for loads, the goal is similar, it is to associate the measured and reconstructed wind field (mean speed, turbulence, shear...) from lidar measurements to the real loads that the turbine undergoes. Nacelle-mounted lidars have proven to be efficient in this regard [20] in the general case, but no so much for floating platforms, which remains a novel technology. In conclusion, nacelle-mounted lidars are extremely useful for certification of FOWTs and validation of the current models for floating turbines. The second objective is to use lidars for turbine control, adapting in real time the behaviour of the wind turbine to the surrounding environmental conditions to optimize production and reduce loads. Again, lidars have proven to be a good candidate [72].

Finally, an interest of using lidars is to avoid using the traditional met mast technology for wind sensing. These towers are especially challenging to install offshore and their height is limited, whereas hub heights are getting larger and larger. Additionally, it was reported that the use of lidar over met masts can cause about 40% cost savings. Thus, not only do nacelle-mounted lidars are a powerful tool for the development of floating wind energy, they also represent a cheaper alternative to some industry standards. As it is often the case, emerging technologies face number of challenges, the main one here being that the motion at the nacelle of a FOWT has non-negligible impact on the lidar measurements. This leads to the research question, containing the essence of the project.

Research questions:

What instruments, data and models are needed to correct the influence of a floating lidar's motion so that the incoming wind field characteristics are accurately reconstructed?

This question can then be divided in different sub-questions, that outline the construction of the project:

- Does the lidar simulator constitute a good model of the measurements of a moving lidar in a turbulent atmosphere?
- Through simulations and prescribed wind fields, do the lidar measurements and the motion-correction algorithm enable to reconstruct the wind field? How do other methods perform?
- What criteria should be used to draw conclusions on the right method to use (performance, cost of having an inertial device in the nacelle, cost of storing the data, cost of having this algorithm in place...)?
- What is the performance of the discussed interpolation methods and motion-correction methods?

Report structure

The following chapters are aimed at answering these questions and at introducing all concepts necessary for the reader. Chapter 2 will introduce the necessary knowledge on pulsed lidar theory, motion representation, interpolation and wind field simulation. Then, Chapter 3 dives into the treatment of real data from the sensor available in the nacelle of a prototype floating wind turbine. It also shows how the velocity of the lidar can be corrected in the first point-wise approximation, using data from such an inertial device. This project has focused on obtaining simulation results, thus it needed a way to simulate pulsed lidar measurements. Building on top of an existing model, Chapter 4 shows the properties of this simulator and how some simple assumed motions affect the time series of the lidar measurements. Additionally, this chapter shows some first results on the uncertainties of the chosen model of the lidar's motion, which are obtained by fitting the measurements to the introduced interpolation method for position, velocity and acceleration.

Following some prescribed motion that uses simplifying assumptions, it is natural to move on to more realistic simulated turbine behaviour and wind field reconstruction examples. This will be the focus of Chapter 5. Reflecting on the performance of wind field reconstruction methods should come in pair with economical and practical matters, especially in terms of system integration. These two parts will also be addressed in the same chapter. Finally, the conclusions that can be drawn for these results as a way to answer the research questions will be presented in Chapter 6. Recommendations on future work and on how to treat the results of this project shall also be found in the same chapter.

2

Theory

In this chapter, the theoretical background for this project will be presented. The first section introduces the lidar technology and details how lidars can measure accurately the wind field characteristics using moving particles in the atmosphere. Then, the second section dives deeper into the wave characteristics of the light emitted and backscattered. In particular, it will explain the concept of a Gaussian beam and how lidars perfom spatial averaging to retrieve the wind velocity. Next, the third section treats the use of quaternions to represent the motion of the lidar with 3 degrees of freedom in rotation. The other three translational degrees of freedom are rather simple to represent and will be shortly discussed in the same section. Since the data is only available in time series, the interpolation of the motion channels between time steps will be covered in the fourth section. Finally, the last part will characterize the properties of a wind field in order to simulate it. Using this knowledge, it will then be possible to move on to introducing the problem and to simulations.

2.1. Doppler Wind Lidar

The LIght Detection And Ranging (lidar) operating principle and first measuring attempts trace back to the 1930s, when laser did not exist yet. It is the invention of the laser in 1960 that permitted the rapid development of modern lidar technology [84] and today, virtually all lidars use a laser as the light source [54]. While automotive lidars are still the dominant application of the technology, this section will only focus on wind lidars, based on the Doppler effect.

The first lidar laser-based optical systems aimed at remote wind-sensing were demonstrated in the second half of the 1960s [31]. Historically, the first applications included for example detection of clear air turbulence ahead of an airplane and detection of wind hazards around an airport [54]. Since then, lidars have progressively become more prominent as a wind measurement technique. Especially, in the wind energy field, they have demonstrated an accuracy similar to calibrated cup anemometers to measure wind speed [73]. Thus, lidars are a promising alternative to traditional expensive met masts. In the next subsections, the principle of operation of a lidar and the different wind lidar technologies will be detailed.

2.1.1. Principle

Doppler wind lidars transmit laser light into the atmosphere and receive a portion of that light back, when it is backscattered by moving particles (aerosols and molecules) in the air. Since these particles are moving and because of the Doppler effect, the backscattered light has a frequency shift compared to the incident light. By processing the signal received by the lidar, this Doppler shift can be estimated and it provides the wind velocity along the line of sight.

Lidars are made of three main components: the laser source, the receiver, and the optical systems for pointing the lidar (transmitter). The transmitter and receiver are installed in a telescope. An illustration of these parts can be found in Figure 2.1. There, the configuration is called *bistatic* because the transmit/receive optics are separated. When the transmitter and receiver are collocated, the configuration is *monostatic*. Let us now derive how the Doppler shift relates to the wind velocity.

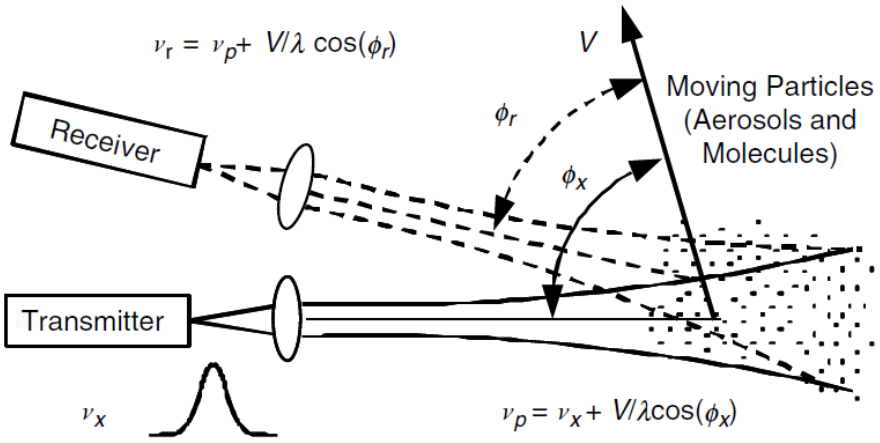


Figure 2.1: Bistatic wind lidar concept and simplified setup [31]

As can be seen in Figure 2.1, the transmitted laser beam with frequency $f_x = \frac{c}{\lambda}$, with c the speed of light and λ the optical wavelength of the laser beam, is scattered by a volume of particles. The mean velocity of these particles V is equal to the mean wind speed [31]. Taking into account the motion of these particles relative to the fixed transmitter and receiver, the Doppler shift at the receiver end can be expressed as $f_{Doppler} = \frac{V(cos(\phi_x)+cos(\phi_r))}{\lambda}$. In this expression, ϕ_x and ϕ_r are respectively the angles between the mean wind speed vector and the line of sight of the transmitter and that of the receiver. For a monostatic configuration, ϕ_x=ϕ_r=ϕ, so the Doppler shift simplifies to Equation 2.1, where $V_r = Vcos(\phi)$ is the radial velocity (also called line-of-sight velocity), or the projection of the velocity onto the transmitter/receiver line of sight.

$$f_{Doppler} = \frac{2V_r}{\lambda} \tag{2.1}$$

2.1.2. Lidar equation

This first lidar equation allows to express the collected optical power in the receiver in terms of system parameters and atmospheric and particle parameters, with simplifications in terms of the light behavior. It is useful to have an equation for the detected signal power to describe how individual lidar techniques retrieve the frequency shift from the backscattered signal and how performant these measurements are [31, 84]. The following derivation is highly inspired by the *Wind Lidar* section of the book edited by *Fujii, T. and Fukushi, T.* [31]. It is applied to a lidar using a pulsed laser.

To simplify, a monostatic lidar will be considered, with a range resolution ΔR, that can be expressed as a function of the laser pulse length ΔT and the speed of light c (see Figure 2.2 and Equation 2.2). The target echoing the signal is at a range R, expressed in Equation 2.3 in terms of t_r (round-trip pulse time-of-flight) and c. The lidar equation predicts the amount of optical signal power, P_s, collected by the receiver aperture from a target echo at range R, through a cascade of multiplicative operators on the laser pulse energy, $E_x = P_x \Delta T$, with P_x the incident pulse power.

First, the received power comes as the product of the backscattered radiant intensity of the target $\psi(W/sr)$ and the perception solid angle of the lidar, that is $\frac{A_r}{R^2}(sr)$, where A_r is the receiver aperture area. The unit sr is for steradian, it is an unit of solid angle, so the dimension of ψ is a power per unit of solid angle. In other words, the lidar collects the light that is backscattered by the target and that is visible to the receiver. Along the receive path (from the target to the receiver), the backscattered signal is attenuated by atmospheric extinction T_{ext}. Consequently, a first expression of P_s is Equation 2.4.

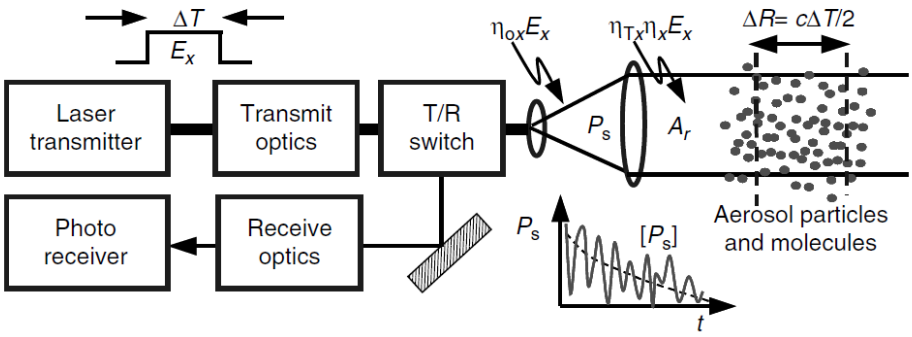


Figure 2.2: Simplified diagram of a monostatic pulsed lidar [31]

$$\Delta R = \frac{c\Delta T}{2} \tag{2.2}$$

$$R = \frac{ct_r}{2} \tag{2.3}$$

$$P_s = T_{ext}\,\psi\frac{A_r}{R^2} \tag{2.4}$$

$$\psi = \eta T_{ext}\,E_x\frac{\beta c}{2} \tag{2.5}$$

Then, since ψ is the backscattered intensity of the target, it is proportional to the incident pulse power reaching the target. To account for the most important physical phenomena, this incident power P_x has to be multiplied by the overall system efficiency η, atmospheric extinction T_{ext} along the transmit path (from the transmitter to the target) and a term representing the portion of the signal reaching the target that is backscattered. This term is defined as the product of a coefficient, the volume backscatterer coefficient $\beta(m^{-1}sr^{-1})$, and the target volume, which is here the resolution ΔR. Given the simple relation between P_x and E_x, and Equation 2.2, a final form for ψ is found in Equation 2.5 . A simplified illustration of the laser beam reaching the targeted particles is Figure 2.2. There, the system losses are separated in two, transmit optical losses η_{ox} and transmit truncation loss η_{Tx}. Altogether, Equation 2.6 is the final lidar equation used for the power collected by the receiver aperture within a given range gate. There is actually another factor that depends on the pulse shape function, and which explains the time-dependency of P_s seen in Figure 2.2. This term is highest at the time when the pulse illuminates the most the particles contained in the considered range-gate. The further the pulse travels from the range gate, the less particles contained in it are illuminated and thus the less power will be collected corresponding to that range gate. Equation 2.6 can be considered at the time of maximal range-gate illumination.

$$P_s = \eta E_x T_{ext}^2\,\frac{c\beta}{2}\,\frac{A_r}{R^2} \tag{2.6}$$

To represent how low the backscattered power usually is, it is useful to take an example for atmospheric and lidar parameters found in literature. Lidar parameters can be taken for a $2\,\mu\mathrm{m}$ pulsed Doppler lidar by CTI Technologies [84]. It uses a pulse energy of $2\,\mathrm{mJ}$ (with a pulse length of around $400\,\mathrm{ns}$, that is an incident pulse peak power of around $5\,\mathrm{kW}$), a receiver aperture of $90\,\mathrm{cm}^2$ and a lower limit of sensing range of $500\,\mathrm{m}$. A monostatic system efficiency of 50% can then be taken [31]. Now for atmospheric parameters, in clear boundary-layer air, it can be assumed that $\beta = 10^{-8}m^{-1}sr^{-1}$ [64]. Finally, the calculation of the atmospheric extinction is more challenging. The formula for T_{ext}^2 at a sensing range R and wavelength λ is given in Equation 2.7, where α is the atmospheric extinction *coefficient* in km^{-1} [84]:

$$T_{ext}^2 = \exp\Big(-2\int_0^R \alpha(\lambda, z)dz\Big) \tag{2.7}$$

It was found that using an extensive molecular and aerosol database in a rural area under $1\,\mathrm{km}$ altitude [31], the atmospheric extinction coefficient is around $1\,\mathrm{km}^{-1}$ for a lidar wavelength of $2\,\mu\mathrm{m}$. Thus, it comes that $T_{ext}^2 \approx 0.37$ for a sensing range of $500\,\mathrm{m}$. It should be considered that this value is to be approached with caution because the measurement conditions for the database are certainly different than that of the lidar environment (rural area in the summer with $23\,\mathrm{km}$ visibility). Inputing all of these parameters into Equation 2.6, the backscattered power is estimated to be $\boldsymbol{P_s \approx 2.0 * 10^{-11} W}$. This highlights the need for a high sensitivity in the detector. Note that in literature, a value of $5 * 10^{-14} W$ was reported with an incident power of $1\,\mathrm{W}$ [64], so the calculation is correct in terms of order or magnitude. The actual technique used by lidars to detect that backscattered intensity and retrieve the Doppler frequency is not a priority for this project, but can be found in Appendix A.1. What is more important however is to detail the laser optics, found in the next part.

2.2. Laser optics and scanning patterns

This section will focus on the wave characteristics of the light beam that is emitted from the lidar, to understand how it interacts with particles in the atmosphere. In particular, Subsection 2.1.2 omitted how the incident light focuses at a particular range by avoiding interaction with other particles as much as is feasible. This section's aim is to explain how the focus is realized with laser properties and the two different types of coherent lidars. They are classified in two categories according to the emitted laser output: *Continuous-Wave* (CW) lidars and *pulsed* lidars. Both have different advantages that Subsections 2.2.2 and 2.2.3 will attempt to explain. The project having focused on pulsed lidars, much more attention is paid to the technology.

2.2.1. Lasers and Gaussian beams

Lidars use laser light because of its special properties that allow accurate sensing. In fact, they showcase high degree of spatial and temporal coherence [62]. High spatial coherence allows for high directionality and radiation focused on small spots while high temporal coherence means a narrow optical spectrum with extremely well-defined frequency (and wavelength). Most lasers, including in lidars, emit quasi-monochromatic infrared light. Finally, an other advantage of lasers used in pulsed lidars is that they can emit light in the form of short pulses, so even a low pulse energy can lead to a high pulse power.

A laser consists of an optical amplifier placed within an optical resonator, visualized in Figure 2.3. The amplifier is a pumped active medium, so-called gain medium [68]. This medium, when it is "pumped", i.e. supplied with energy, increases the amplitude of the light field while maintining its phase at each round trip of the light inside of the resonator. This gain in amplitude is performed using the principle of *stimulated emission*, hence the name Light Amplification by the Stimulated Emission of Radiation. The underlying concept is that a photon in a given mode can induce an atom in an upper energy level to undergo a transition to a lower energy level and, in the process, to emit a clone photon into the same mode as the initial photon (same frequency, direction, and polarization) [68]. For extracting a laser beam as a useful output of the device, the right mirror, for example (see Figure 2.3), acts as an output coupler, transmitting some percentage (say 10%) of the intracavity power [62].

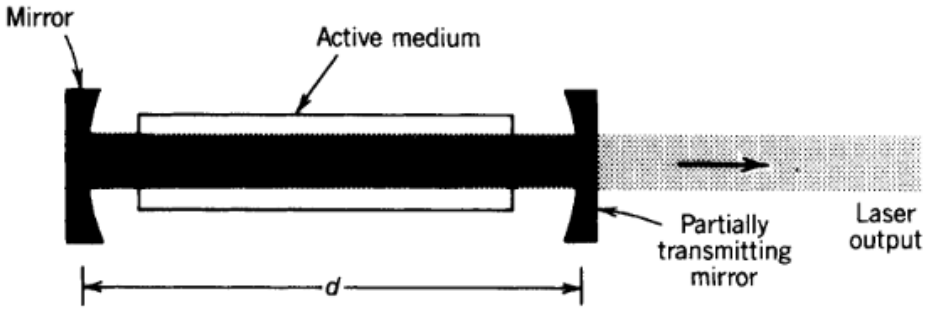


Figure 2.3: Formation of the laser beam in the resonator. The output is extracted through the partially transmitting mirror [68].

The light contained inside of a stable resonator is present in resonator modes, or Transverse Elec-

tromagnetic Modes (TEM), and so is the light escaping through the transmitting mirror (then emitted through the telescope of the lidar). These modes are denoted $TEM_{l,m}$ where each pair of indices (l,m) defines a tranverse mode with an associated spatial distribution [68]. The lowest order mode $TEM_{0,0}$ is the Gaussian beam, presented below and sketched in Figure 2.4. Lasers are often designed to operate on a single transverse mode; for lidars this is the (0,0) Gaussian mode because it has the smallest beam diameter and can be focused to the smallest spot size. It is also interesting to note that the Gaussian beam is a solution of the wave equation in the paraxial approximation, meaning that the beam has a low divergence, that is the wavefront normals are close to parallel to the axis of propagation [68]. This can be visualized in Figure 2.4.

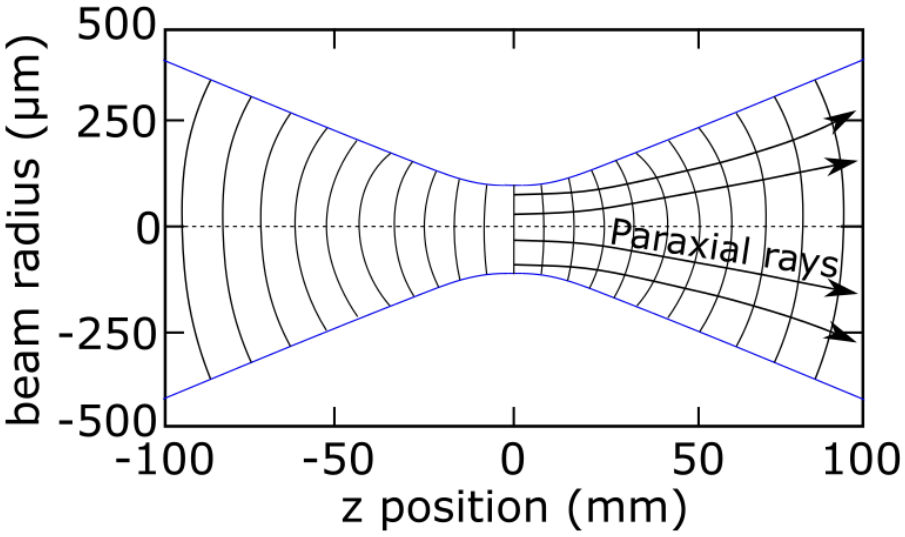


Figure 2.4: Propagation of a Gaussian beam and evolution of the beam radius. The so-called beam waist is situated at $z = 0$. Axis units are extracted from [62]

The paraxial approximation for Gaussian beams is a small-angle approximation. It has been shown [1] that Gaussian beams are valid in the paraxial approximation when the wavelength is smaller than the beam waist multiplied by a factor 0.2π. This is largely the case for commercial lidars. For example, the ZephIR lidar in its range of operation uses a beam waist smaller than $10\,\mathrm{mm}$ and a wavelength of $1.55\,\mu\mathrm{m}$ [17].

The transverse profile of the Gaussian beam's electric field distribution can be described by Equation 2.8 [37]:

$$E(x,y,s) = \frac{E_0}{w(s)} \exp\left(\frac{-(x^2+y^2)}{w(s)^2}\right) \exp\left(\mathrm{i}k\frac{x^2+y^2}{2R(s)^2}\right) \tag{2.8}$$

where $k = \frac{2\pi}{\lambda}$ is the wave number , s is the axial distance from focus F (and $z = F - s$ with z the distance from the telescope aperture), $w(s)$ is the "beam $1/e^2$ intensity radius" and $R(s)$ is the curvature radius of the wavefronts. These parameters can be further expressed as Equations 2.9 and 2.10 :

$$w(s) = w_0\sqrt{1 + (\frac{s}{z_R})^2} \tag{2.9}$$

$$R(s) = s + \frac{z_R^2}{s} \tag{2.10}$$

where w_0 is the beam waist and z_R is the Rayleigh length. It is the distance at which the beam radius has reached $\sqrt{2}w_0$, or where the cross-sectional area of the beam has doubled. The expression for z_R as a function of the wavelength λ and w_0 is given in Equation 2.11. All these beam parameters are represented in Figure 2.5 . Furthermore, the divergence angle $\theta = \frac{\lambda}{\pi w_0}$ characterises the beam geometry where $s >> z_R$ (far-field region) and where the beam radius evolves nearly linearly. One can

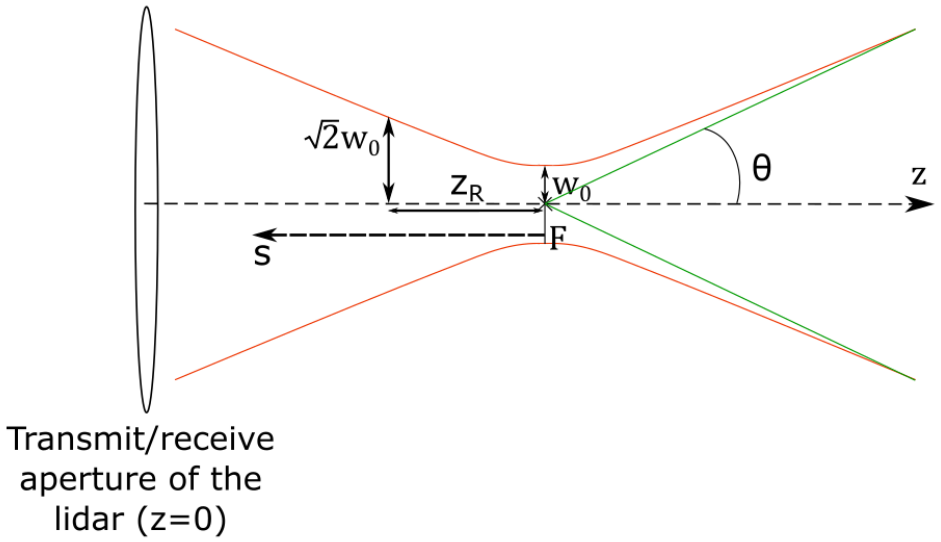


Figure 2.5: Beam geometry when leaving the telescope aperture of the lidar. The beam is focused around F, where one can see the beam waist w_0.

also see that in the far-field region, the wavefront resembles that of a spherical wave while in the near-field (close to $s = 0$), it is similar to that of a plane wave (see Figure 2.4). Another important property is that the curvature radius $R(s)$ is minimum at $s = z_R$, so the wavefront curvature is the greatest at the Rayleigh length.

$$z_R = \frac{\pi w_0^2}{\lambda} \tag{2.11}$$

Then, the variation of the beam intensity in cartesian coordinates when it propagates along the z-axis can be expressed as Equation 2.12 [62]. It is a function of the beam power $P_0 = I_0 \frac{\pi w_0^2}{2}$, with I_0 the on-axis intensity at the beam waist. In Equation 2.12, it becomes clear why $w(s)$ is called the "beam $1/e^2$ intensity radius". In fact, when the radius in the transverse plane of the beam is equal to $w(s)$, that is $x^2 + y^2 = w(s)^2$, the last term becomes equal to $\exp(-2) = 1/e^2$.

$$I(x, y, s) = |E(x, y, s)|^2 = \frac{P_0}{\pi w(s)^2/2} \exp\left(-2\frac{x^2 + y^2}{w(s)^2}\right) \tag{2.12}$$

Since Equation 2.12 represents the spatial propagation of the beam, it can be used to describe where the backscattered power comes from in space. To be more clear, one can observe that the maximum on-axis beam intensity is obtained at the beam waist ($s = 0$). From this point, the radial intensity decreases exponentially. In fact, when the radius of the beam is equal to $1.5\, w_0$ (or $x^2 + y^2 = (1.5\, w_0)^2$), the intensity is around 1% of its value at the center. The further the particles are from the beam waist, the more radially spread out the beam radius is and the lower the intensity of the incident light is. From this comes the fact that the particles at the beam waist (i.e. at the focus point) are the most irradiated and are the most likely to scatter light back to the receiver. This seems intuitive since one wants the particles where the beam is focused to contribute the most to the signal, because they are the ones containing the information about the wind velocity at the desired range. In fact, it would be ideal if the particles in the focus point plane would be the only ones contributing to the received signal, because this would lead to an almost singular frequency shift measurement, and thus an exact wind velocity measurement. In this report, this optimal case will be referred to as the point-wise approximation.

However, due to the actual shape of the beam, it is in practice impossible to get the backscattered radiation only from the focus point and radiation from other regions of space around the line of sight also contribute to the received signal with different intensities [69]. Considering this, to model how a lidar computes the line-of-sight velocity from the backscattered signal, each point in space can be given an appropriate weight that is maximum at the focus point and reduces with the intensity of the beam

reducing. Mathematically, that is the the 3D convolution integral of the wind velocity field (projected onto the line-of-sight) with the intensity profile of the beam $I(x, y, s)$. Generally in lidar theory, only the dimension s is considered, since the radial spread is so low. Then, the weighted average radial velocity can be written as an integral along the line of sight of the product of a weighting function W and the projected velocity field $\boldsymbol{v}$ evaluated at each point of the line of sight, see Equation 2.13.

$$\overline{v_r} = \int_{-\infty}^{+\infty} W(s)\, \boldsymbol{n} \,.\, \boldsymbol{v}(s) ds \tag{2.13}$$

where $\boldsymbol{n}$ is the line-of-sight unit directional vector. The function W reaches its maximum at $s = 0$ or $z = F$ with F the focus distance. Here, one can only see the spatial averaging effect of the beam in the longitudinal dimension but the weighting function can also incorporate the transversal spread of the beam, hence the term ”sensing volume” encapsulating the 3D nature of the problem. For the point-wise approximation this function is indeed a Dirac delta function with a peak at the focus point. Then in this much simplified case, Equation 2.13 becomes the basic Equation 2.14.

$$\overline{v_r} = \boldsymbol{n} \,.\, \boldsymbol{v}(z = F) \tag{2.14}$$

It is important to note that in practice, the calculation of $\overline{v_r}$ is done in the frequency domain by performing Fourier Transforms on the detector voltage output signal, and then squaring them to obtain power spectra. When averaged for a given measurement time, these so-called ”Doppler spectra” show a peak at a frequency that is made to correspond to a radial velocity through Equation 2.1. On a modelling level however, convolution formulas for the measurement of $\overline{v_r}$ are widely used.

Afterwards, across this framework, it is fundamental to delve into what happens when the lidar itself is moving with a certain speed compared to an Earth-fixed referential. If at the time where the beam arrives back at the aperture of the lidar, it has a velocity $\boldsymbol{v_{lidar}}$, then it only measures the projection onto the line of sight of the wind velocity minus the velocity of the lidar, which is the projected wind velocity with respect to the lidar. In other words the lidar measures, at its position, what is ”seen” by the moving turbine. For wind field reconstruction, because one needs the wind speed with respect to an Earth-fixed referential, and not with respect to the moving turbine, this then becomes a problem. This effect is depicted in Figure 2.6. The case 1) is for a still lidar while the case 2) is for a moving lidar. The figure is for a lidar pointing upwards but same principle applies for a lidar pointing horizontally upwind.

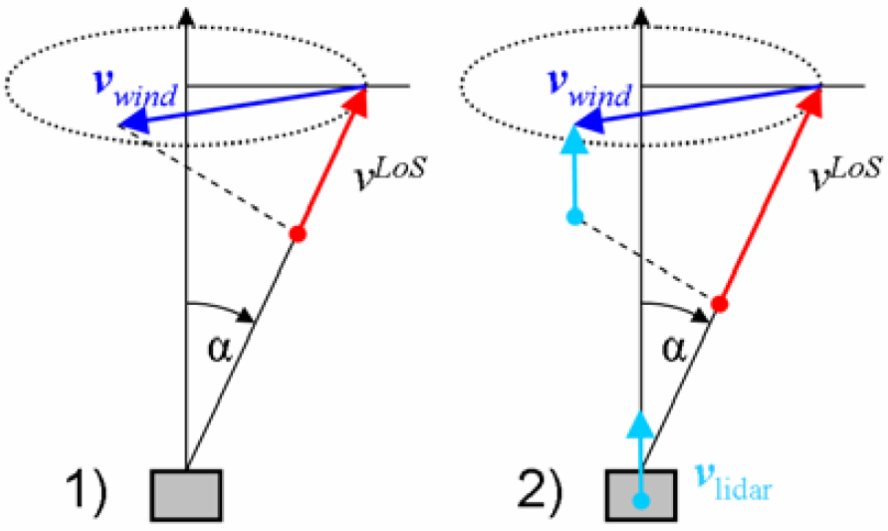


Figure 2.6: Error on the wind velocity measurement caused by the interference of the system velocity [87]

For a moving lidar, Equation 2.13 becomes Equation 2.15. And in the point-wise approximation, the weighted measured velocity is simply $\overline{v_{r,motion}} = \boldsymbol{n} \,.\, (\boldsymbol{v}(z = F) - \boldsymbol{v_{lidar}})$.

$$\overline{v_r}_{,motion} = \int_{-\infty}^{+\infty} W(s)\, \boldsymbol{n}\, .\, (\boldsymbol{v}(s) - \boldsymbol{v_{lidar}})\, ds \tag{2.15}$$

Note that it does not really matter if $\boldsymbol{v_{lidar}}$ is considered when the beam is released or when the backscattered signal comes back. Indeed, considering that the lidar beam is sent into the atmosphere at $t = 0$ when the lidar has a velocity $\boldsymbol{v_{lidar}}(t = 0) = \boldsymbol{v^0_{lidar}}$, the backscattered signal from a distance R comes back when the lidar has a velocity $\boldsymbol{v_{lidar}}(t = \frac{2R}{c}) = \boldsymbol{v^1_{lidar}}$. For the difference of taking into account $\boldsymbol{v^0_{lidar}}$ or $\boldsymbol{v^1_{lidar}}$ to be significant for the measurement, the magnitude of their difference would need to be for example in the order of magnitude of $0.01\,\mathrm{ms}^{-1}$ or greater. This means that, with $\Delta t = \frac{2R}{c}$:

$$\frac{\|\boldsymbol{v^1_{lidar}} - \boldsymbol{v^0_{lidar}}\|}{\Delta t} \geq c \times \frac{0.01}{2R}$$

Given that the left-hand side of the inequality is approximately the acceleration magnitude of the lidar, and taking $R = 700\,\mathrm{m}$ (maximum range of the WindCube pulsed lidar [86]), this would mean that the acceleration magnitude of the lidar is greater than $2000\,\mathrm{ms}^{-2}$. The reality for a FOWT would never even be close to this since for example at the nacelle, the limit state criterion for acceleration can be $0.2\,g$ or $1.962\,\mathrm{ms}^{-2}$ [46], so a thousand times smaller than what would be required for the error between considering $\boldsymbol{v^0_{lidar}}$ or $\boldsymbol{v^1_{lidar}}$ to be significant.

In the general case, W is a function that depends on the Gaussian beam intensity found in Equation 2.12 and that integrates to one. A further development of W can be made depending on the type of lidar. This will be explained in the following parts on CW and pulsed lidars. It will also be detailed how the wind field is reconstructed in these two different technologies that use different ”scanning patterns”. Finally, it should be noted that Equation 2.13 is derived from the first moment of the spectral density of the echo signal [2]. A more complete approach would then be to build the radial velocity from the Doppler spectrum of the received signal but this is not supposed to be in the scope of this report.

2.2.2. Continuous-Wave (CW) lidars

The simplest form of lidar uses a CW laser, it emits light continuously into the atmosphere, which is the typical operational mode of a laser. The main advantage is to combine simplicity with high sensitivity at relevant ranges for wind energy [64], making CW lidars robust and at relatively low cost. An example of commercial CW wind lidar is the former ZephIR, now ZX TM. It can operate with a good performance at short ranges, up to a few hundred meters away from the device [64]. Especially, the ZephIR lidar showed significantly better sensing accuracy than a pulsed lidar under $80\,\mathrm{m}$ [15]. Another advantage of CW lidars is a typically higher frequency of measurements, with for example up to 50 measurements per second. Pulsed lidars on the other hand will provide generally more in the order of magnitude of 1 measurement per second, but at several ranges at a time. To measure at a different range, CW lidars need to be refocused. Due to the application of the project to a pulsed lidar, little is needed on the theory of CW lidars, and the focus will be made on the pulsed lidar technology.

2.2.3. Pulsed lidars

Pulsed lidars emit regularly spaced emissions of highly collimated light energy (as opposed to focused light in the case of CW lidars) [64]. For a single pulse, since light travels at a known speed c, precision timing circuits in a pulsed lidar are able to isolate the backscattered signals to a period of time that corresponds to a specific segment of radial distance along the beam, called the *range gate*. In practice, consider that the pulse is sent at a time t_1 and the backscattered signal returns at a time t_2. This means that the range gate (corresponding part of the illuminated space that backscatters the emitted light) is centered around a distance $R = \frac{(t_2 - t_1)c}{2}$ from the aperture. Here, division by a factor 2 comes from the fact that the light travels from the lidar to the target, and back to the lidar. Note that the word ”centered” is used, because outside of the point-wise approximation, the beam reaches a volume of particles in space, and not a point. For a pulsed lidar, this can be seen as a cylinder, the length of which is determined by the pulse length. The lidar then has to capture this illuminated volume by dividing the

return signals into bins that have an appropriate size, corresponding to different parts of the line of sight. The sizing of these bins correspond to a certain measurement time, and is in the same order of magnitude as the measurement volumes. These aspects of the signal processing are depicted in Figure 2.7 and will be discussed further. The heterodyne detection addressed previously is reflected by the use of two photodetectors.

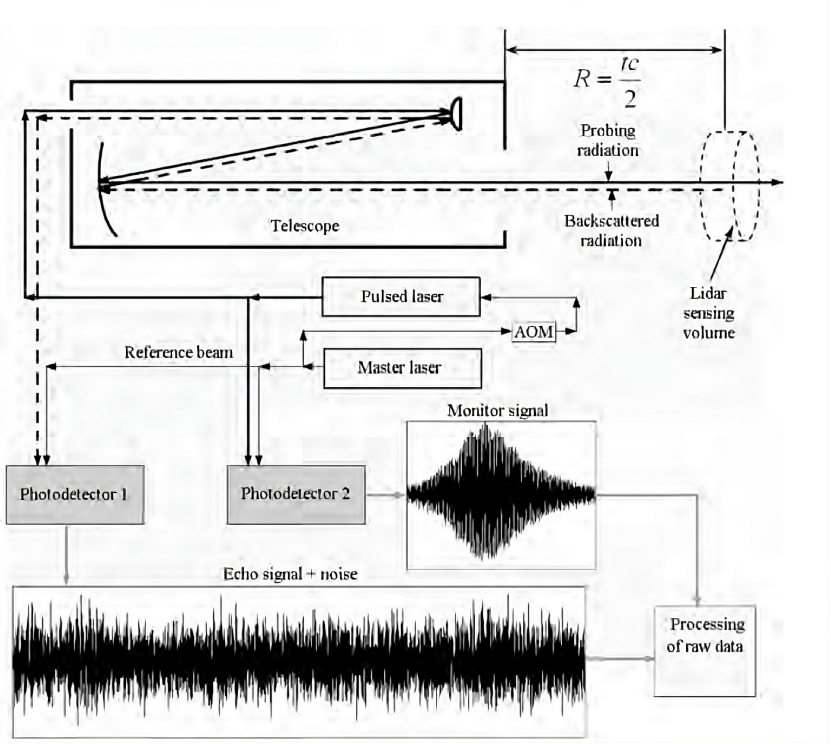


Figure 2.7: Principle of operation of a pulsed lidar and propagation of the sensing volume [2]

It follows that an advantage of the pulsed lidar is that from one single emission of light, the device can reconstruct the wind velocity at several ranges, whereas for a CW lidar, a change of focus would be needed. However, this also results in the fact that the sampling rate for a pulsed lidar is inherently lower than for a CW lidar. In fact, the pulse emissions should be spread out over time in order not to mix the sensing volumes associated with two distinct pulses. To give some concrete examples, the $1.54\,\mu\mathrm{m}$ nacelle-mounted pulsed lidar WindCube developed by Vaisala has a sampling rate of $1\,\mathrm{Hz}$ and can measure wind for 20 different ranges per pulse emission [86]. This means that per second, the user can get the wind velocity at 20 different distances away from the turbine. In comparison, the ZX TM nacelle-mounted CW lidar can give 50 measurements per second, at one specific range [89]. It can give measurements at 17 horizontal ranges in total but that requires a change of focus.

Now, to understand the optics of a pulsed lidar and how the beam propagates and interacts with particles, it is useful to determine its power output. Notably, the emitted pulses have a temporal nature (usually a pulse duration of less than $1\,\mu\mathrm{s}$) and the lidar discriminates the position of the backscattering particles with respect to time-of-flight. Thus, it is relevant to first express the power of the beam as a function of time rather than in space coordinates such as in Equation 2.12. In the case of a pulsed Doppler wind lidar, the power of the probing beam as a function of time is well described by equation 2.16. Note that this is also a Gaussian function.

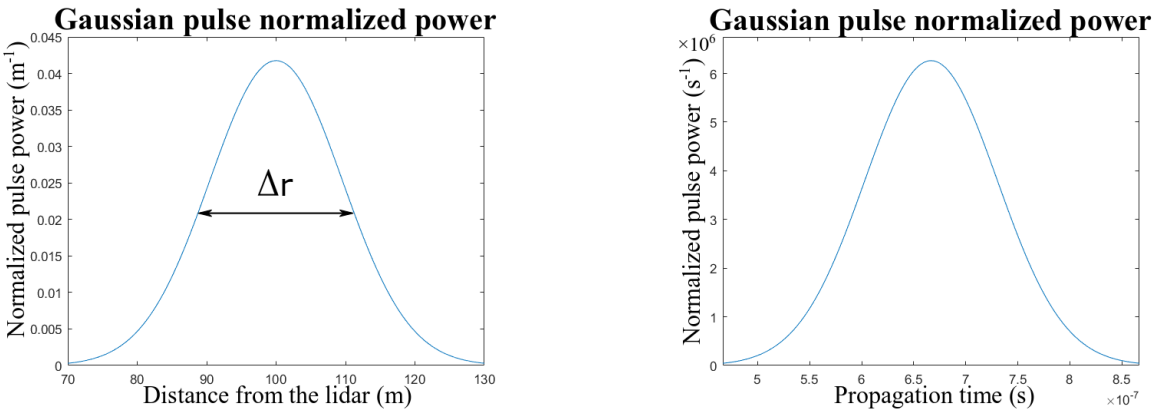
$$P_p(t) = \frac{E_p}{\sqrt{\pi}\sigma_p}\exp(-\frac{t^2}{\sigma_p^2}) \qquad \sigma_p = \frac{\tau_p}{2\sqrt{\ln(2)}} \tag{2.16}$$

In Equation 2.16, E_p is the pulse energy and τ_p is the pulse duration. For example, in the case of the WindCube lidar, the pulse energy is of $100\,\mu\mathrm{J}$ and the pulse duration is $150\,\mathrm{ns}$ [44]. For the rest of this section, it is also useful to define the normalized laser pulse power $P_{p,n}(s^{-1})$, see equation 2.17.

$$P_{p,n}(t) = \frac{1}{\sqrt{\pi}\sigma_p} \exp(-\frac{t^2}{\sigma_p^2}) \tag{2.17}$$

Following this, to study the specificity of pulsed lidars, one can look into its characteristic range-weighting function W, as described in Equation 2.13. To express W for pulsed lidars in the same form as Equation 2.13, it is possible to do a transformation from time to space coordinates. Indeed, considering a pulse released at $t = 0$, the instantaneous radial wind velocity detected at a distance R from where the pulse is emitted can be either expressed as a function of time or spatial coordinates. For a range R, $v_r(z = R) = \boldsymbol{n} \,.\, \boldsymbol{v}(z = R) = v_r(t = \frac{2R}{c})$. Thus, by assimilating the time t to the space position $z = \frac{ct}{2}$ and $r_p = \frac{c\sigma_p}{2}$, one can write Equation 2.18 [30] for a beam propagating in the z-direction. In this case, the radial spread of the beam is ignored, as it is an assumption generally made. This implies that anywhere the term ”sensing volume” or anything related is used, in practice it will actually be a ”sensing line”.

$$P_{p,n}(z) = \frac{1}{\sqrt{\pi}r_p} \exp(-\frac{z^2}{r_p^2}) \tag{2.18}$$



(a) Normalized power as a function of distance

(b) Normalized power as a function of time

Figure 2.8: Normalized Gaussian pulse power for a range-gate centered at $R = 100m$ and $\tau_p = 150\,\mathrm{ns}$

For a specific case and for a range-gate centered at a distance $R = 100m$ from the lidar, it is possible to represent the corresponding pulse’s spatial and time extent. The round-trip time-of-flight for light to reach that point is $\frac{2R}{c} \approx 667ns$. Figure 2.8 depicts this for the pulse duration of the WindCube previously stated ($150\,\mathrm{ns}$). As expected, the peak power of the pulse is at the center of the range-gate in Figure 2.8a, which corresponds to $t = \frac{2R}{c}$ in Figure 2.8b. In the WindCube user manual [86], it is specified that the functional probed length is $30\,\mathrm{m}$. This length is also usually referred to as the ”range-gate length”, it is the distance that the laser-illuminated aerosol volume moves during the observation time [28]. The relationship between the range-gate length Δp and the observation time τ is simply given by Equation 2.19. Here, $\tau = 200ns$. In Figure 2.8a, the x-axis delimitation was chosen such that the extreme left side corresponds to $R - \Delta p$ (here, $70\,\mathrm{m}$) and the extreme right side to $R + \Delta p$ (here, $130\,\mathrm{m}$). The measurement time is the difference in round-trip time-of-flight between light that reaches one extreme side of the range-gate and light that reaches the other. Because of the round-trip effect, the light that reaches the furthest part of the range-gate has travelled twice the range-gate length (hence the factor 2 in Equation 2.19). So, similarly, in Figure 2.8b the corresponding extremities are $\frac{2R}{c} - \tau$ and $\frac{2R}{c} + \tau$. This shows that the measurement time is chosen to roughly fit the illuminated volume.

$$\Delta p = \frac{\tau c}{2} \tag{2.19}$$

Now, it is natural to question how to quantify the laser-illuminated aerosol volume. By nature, this depends on the spatial extent of the beam. Indeed, as can be seen in Figure 2.8 , the particles situated at the center of the range-gate are the most illuminated at that time whereas the ones situated at a distance Δp from the center are barely illuminated. A common measure for this effect is the ”Full Width

at Half Maximum" (or FWHM), written Δr in what follows. As the name suggests, it is the width of the curve where the pulse power on both symmetric sides is half of the value at the peak. Mathematically, it can be expressed as Equation 2.20. It is also shown in Figure 2.8a.

$$\Delta r = \sqrt{\ln(2)}\, c\, \sigma_p \tag{2.20}$$

In the case previously mentioned, $\Delta r = 22.5m$, which verifies to be in the same order of magnitude as the range-gate length. In reality, there are many particles inside of the sensing volumes that backscatter light at different times as the pulse travels the range-gate. This is depicted in Figure 2.9.

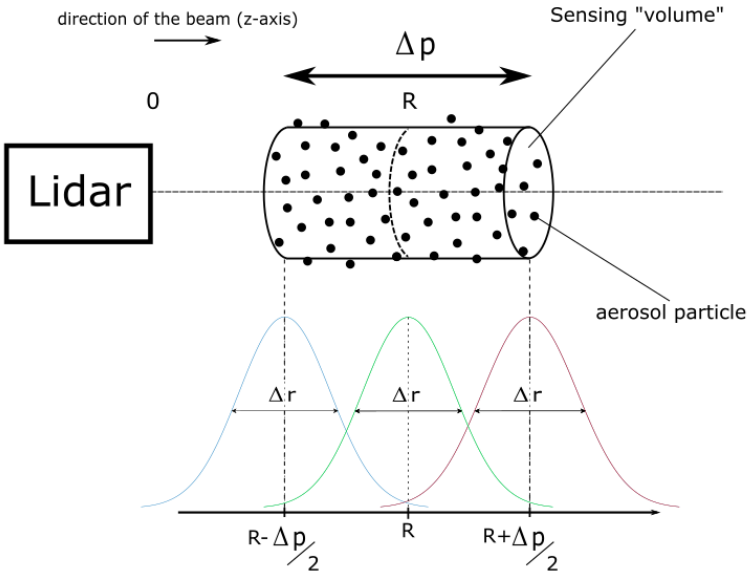


Figure 2.9: Normalized pulse profile of the light when it reaches 3 parts of the range-gate(the two extremities and the center)

The backscattered blue signal will be the first to reach the detector back while the red will be the last. The figure is not scaled for better visualizing purposes, the actual overlapping of the pulse profiles is shown in Figure 2.10. Finally, when considering the range-gate, an important parameter is the range resolution ΔR. It refers to the ability of the lidar to to distinguish between two range-gates. Consider now Figure 2.10. At the time when the peak of the pulse power arrives at $R - \frac{\Delta p}{2}$ (represented by the blue curve), it strongly illuminates the particles contained within the space defined by its FWHM (blue double-end arrow), so until $R - \frac{\Delta p}{2} - \frac{\Delta r}{2}$. The same effect happens on the right side. Thus, the illumination of the pulse in a particular range gate "leaks" on the extremities with a length scale of $\frac{\Delta r}{2} + \frac{\Delta r}{2} = \Delta r$. This means that the actual volume illuminated in the measurement time is in reality not Δp but much closer to $\Delta p + \Delta r$. For example, the particles situated at a distance $R - \frac{\Delta p}{2} + \frac{\Delta r}{2}$ from the lidar are as likely to backscatter the light from the pulse when it matches the blue pulse in Figure 2.10 as those situated at $R - \frac{\Delta p}{2} - \frac{\Delta r}{2}$. This is why the range resolution is $\Delta R = \Delta p + \Delta r$ [29].

Example of a moving pulsed lidar

Shortly aside of the theory for the pulsed lidar optics, it will be introduced what happens to the position of the beam when the lidar is moving. For example in the case of the Windcube pulsed lidar (placed on the nacelle of a wind turbine), there are 4 beams: 2 upper beams and two lower beams. In the lidar frame, the beams have an horizontal angle of $15°$ and an elevation angle of $5°$ [86]. That can be visualized in Figure 2.11. Consider a 2-dimensional case with one simple motion, where the turbine only moves by rotating around the Y-axis, so in the plane defined by Figure 2.11b : this motion is called pitching motion tracked by a so-called *pitch angle*, not to mistake for the pitch angle of the turbine blades.

What is the order of magnitude of that pitch angle for FOWTs? In literature, it is found that pitch motion can be as a first approximation modeled by an harmonic oscillator [34]. One of the cases

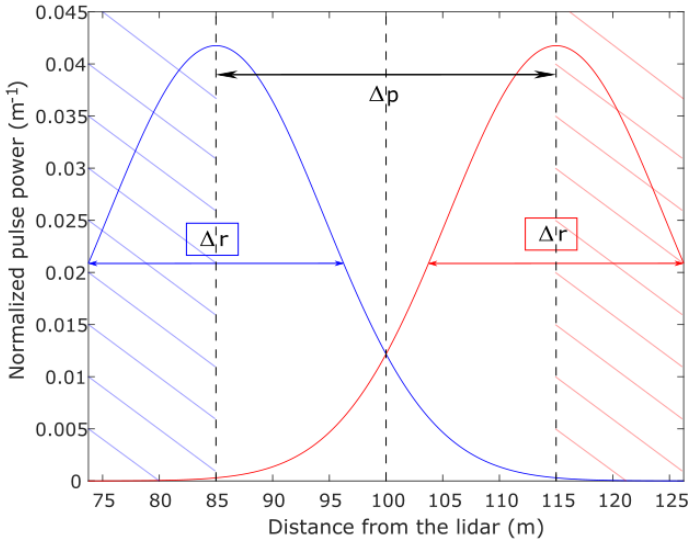
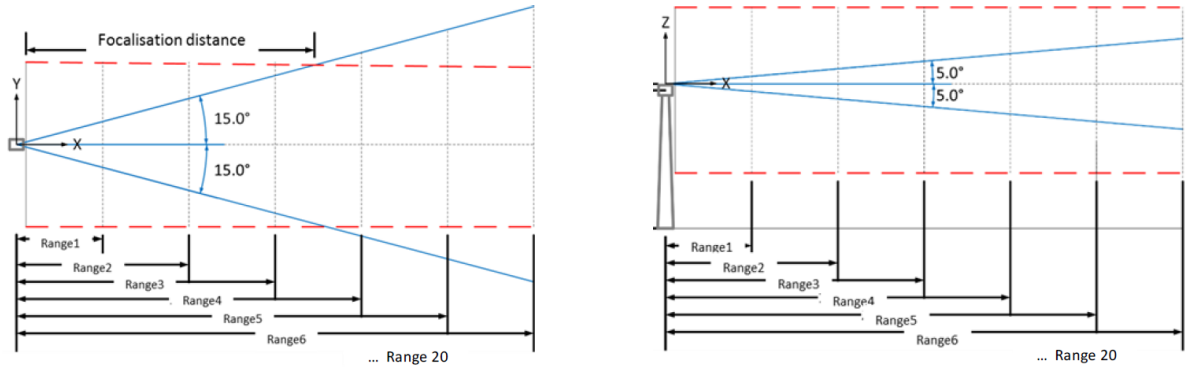


Figure 2.10: Actual overlapping of the normalized pulse profiles and sensing resolution



(a) Top view

(b) Side view

Figure 2.11: Top and side views of the Windcube beam configuration with range-gates [86]

comprises a pitch amplitude of $5°$. This means that if one of the upper beams initially has an elevation angle of $5°$, by this additional pitch angle, at the maximum of the amplitude, its elevation angle becomes $10°$. It will now come clear why it is important to track the position of the lidar in time. One can evaluate this impact on one of the upper beams for a particular range gate, for example at $300\,\mathrm{m}$, if the lidar is placed at $80\,\mathrm{m}$. The initial altitude to which the beam points is equal to $H_1 = 80 + 300\ \sin(5°) \approx 106\,\mathrm{m}$ and the altitude after being pitched due to the motion of the turbine is $H_2 = 80 + 300\ \sin(10°) \approx 132\,\mathrm{m}$. If the pitching motion of the turbine is not taken into account, significant errors on the altitude of the measurements will be made! What does that mean in terms of wind speed?

One can consider a case of simple wind field where the wind speed at a height z is given by the power law defined in Equation 2.21, with U_H the wind speed at the height H and α the vertical wind shear exponent.

$$U(z) = U_H \left(\frac{z}{H}\right)^{\alpha} \tag{2.21}$$

A quite extreme but possible case for offshore wind can be $U_H = 10\,\mathrm{ms}^{-1}$ at $H = 60\,\mathrm{m}$ and $\alpha = 0.3$ [8]. In the two previous cases, without pitching motion, the wind speed measured would correspond (before projection onto the line of sight) to an altitude where the theoretical wind speed is $U_1 \approx 11.9\,\mathrm{ms}^{-1}$ and after pitching that would be $U_2 \approx 12.7\,\mathrm{ms}^{-1}$. Thus, that makes a difference of around $0.8\,\mathrm{ms}^{-1}$. This

is the order of magnitude of the error one should expect if the changing position and orientation of the lidar is not accounted for, which is not acceptable. On top of this positioning error, there is the error of the lidar velocity on the VLOS measurements. The next section will address these kinematic aspects.

Finally, one aspect that differentiates a CW lidar and a pulsed lidar is the expression of the weighting function that models the spatial averaging of the measurement. In the case of a pulsed lidar, the line-of-sight velocity measured can also be modelled by the convolution at the range R of the radial velocity with the weighting function. But in this case, the range-weighting function is itself a convolution of the normalized pulse profile $P_{p,n}$ and a windowing function. The bounds of this windowing function match that of the considered range gate, and this convolution models the range-gating of the lidar's detector output. This is not done when modelling CW lidars because one beam only has one range, which corresponds to the focus distance. An example of windowing function is a normalized rectangle of width equal to the range-gate dimensions Δp [28]. The estimated expression of the radial velocity measured by the lidar that follows is found in Equation 2.22 and that of the weighting function in Equation 2.23 [30].

$$\overline{v_r}(R) = \int_{-\infty}^{+\infty} W(R-s)\, \boldsymbol{n} \,.\, \boldsymbol{v}(s) ds \tag{2.22}$$

$$\begin{aligned} W(r) &= \frac{1}{\Delta p} \int_{-\Delta p/2}^{\Delta p/2} P_{p,n}(r+s) ds \\ &= \frac{1}{2\Delta p} \left(Erf\Big(\frac{r+\Delta p/2}{\sqrt{2} r_p}\Big) - Erf\Big(\frac{r-\Delta p/2}{\sqrt{2} r_p}\Big) \right) \end{aligned} \tag{2.23}$$

The function Erf is the standard error function defined as follows.

$$Erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2) dt$$

2.3. Kinematics and representation of the rotational attitude of an object with quaternions

When a lidar is mounted on the nacelle of an offshore wind turbine or when it is placed on a buoy, it is subject to considerable motion compared to when it is placed still on the ground. This section is dedicated to the representation of this motion in space and its impact on the wind profile measurements. The motion of the lidar in space is considered in the wind turbine system, that moves relatively to the inertial earth frame. It is characterized by a position vector, a velocity vector and an acceleration vector. These last two vectors are usually decomposed into a translational and a rotational component. To represent rotation mathematically, rotation matrices and Euler angles are traditionally used. In this project, quaternions are used instead given their advantages compared to Euler angles, that will be addressed in the first subsection. An introduction to the basic properties of quaternion can be found in Appendix A.2, and a way of visualizing them in Appendix A.4 . Finally, the second subsection will define more clearly how the motion of the lidar and its six degrees of freedom are modelled mathematically, to be used as inputs for the numerical algorithm.

2.3.1. Quaternions and rotation

It can be understood that 4 dimensions are needed to represent a rotation in space. Indeed, to know how a 3-D vector is rotated from its initial coordinates to its new coordinates, one only needs to know the angle of the rotation (in $\mathbb{R}$ so 1 dimension) and the axis of the rotation (in $\mathbb{R}^3$ so 3 dimensions), for a total of 4 dimensions [16]. This is where quaternions become extremely handy.

Let $q \in H_1$, $q = [\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})\, \boldsymbol{n}]$ and $\boldsymbol{r}$ a vector of $\mathbb{R}^3$. In "quaternion form", $r = [0, \boldsymbol{r}]$. Then, the new vector $R\boldsymbol{r}$ from Equation 2.24 is the vector $\boldsymbol{r}$ rotated by the angle θ about the axis $\boldsymbol{n}$.

$$R\boldsymbol{r} = q\, r\, q^{-1} \tag{2.24}$$

This is shown in Figure 2.12. Note that applying the same formula but with q^{-1} on the left and $(q^{-1})^{-1} = q$ on the right yields the opposite rotation about the same axis (see Figure 2.13) while applying $-q$ yields the exact same rotation. Indeed, $-q\, r\, (-q)^{-1} = -q\, r[-(q^{-1})] = q\, r\, q^{-1}$. Another key property of quaternions is that given another unit quaternion p, replacing q by pq in Equation 2.24 represents a rotation by q followed by a rotation by p. Such a direct mapping is not possible with Euler angles.

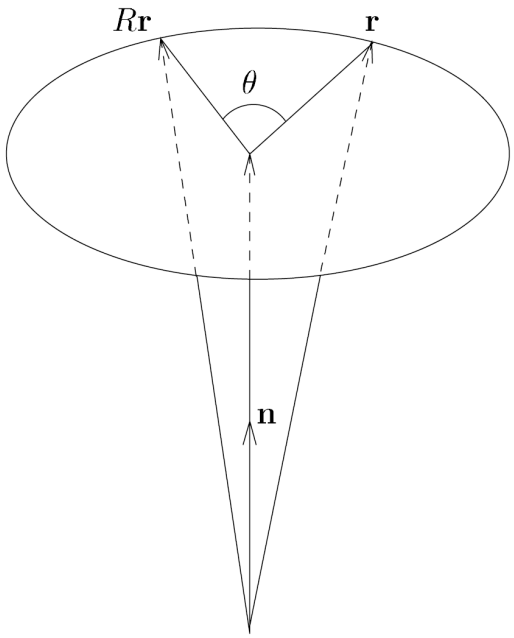


Figure 2.12: Rotation of $\boldsymbol{r}$ by θ about $\boldsymbol{n}$ [16]. The resulting vector $R\boldsymbol{r}$ is given by Equation 2.24. The norm of $\boldsymbol{r}$ is of course conserved.

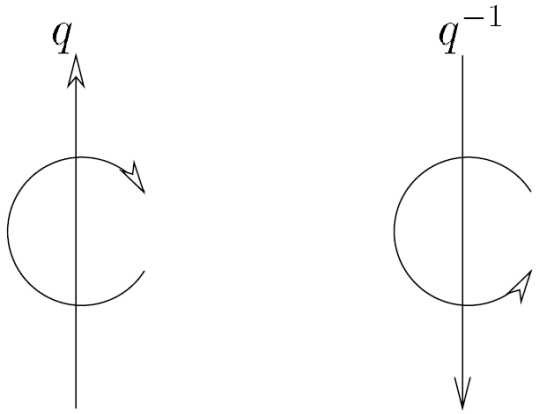


Figure 2.13: Opposite rotation obtained by changing the order of the multiplication in Equation 2.24 [16].

Since this is a crucial part of the work, it is valuable to show an example and how quaternions compare to more traditional Euler angles and rotation matrices. For example, how to represent the rotation of the vector $[0, 0, 1]$ along a circle defined around the vector $[1, 1, 1]$? In this case, the normalized axis of rotation is $[\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$. For an angle θ, the quaternion corresponding to the rotation is simply:

$$q(\theta) = \Big[\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})\, [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]\Big]$$

And the rotated vector is obtained by applying formula 2.24 to the vector $[0, 0, 1]$. Note that it is straightforward. For a range of angles, this rotation is shown in Figure 2.14, where the black vector is the axis of rotation.

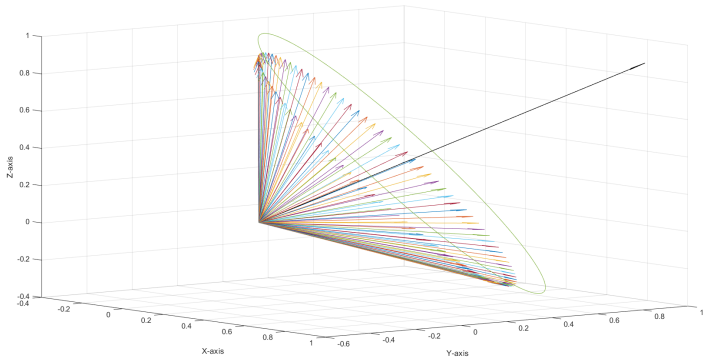


Figure 2.14: Example of a rotation around the axis $[1, 1, 1]$

How would one describe that rotation using traditional Euler angles and rotation matrices? First, as a reminder Euler angles describe a rotation by three successive rotations around 3 successive axes, for example starting with typical cartesian X-Y-Z axes. The problem is that it is not as straightforward as quaternions. The advantage however is that they are easier to understand and represent. Also, they depend on the convention for the order of rotations, since each rotation modifies the initial vectors.
For example, if the convention is *ZYX*, then the Euler angle rotation {γ, β, α} is this chain of rotation: first a rotation of γ around the Z-axis, then of β around the new Y-axis and finally of α around the newer X-axis. If the set is for example {$\frac{\pi}{2}$,$\frac{\pi}{4}$,0} and the initial vector is $[1, 1, 0]$ normalized (see Figure 2.15 for the initial vector and axes), the rotation performed is first a rotation of $\frac{\pi}{2}$ about Z (see Figure 2.16a), with which the X and Y axes also rotate. The new axes are in dotted green lines. The actual initial Cartesian basis is still in blue. The vector of rotation that was in the X direction is now in the -Y direction while the one that was in the Y direction is now in the X direction. And then, the second rotation is about the new Y-axis (which has become the initial X-axis) by $\frac{\pi}{4}$. The result is again in red and the rotated Euler axes are in dotted green. There is no third rotation since that last angle was set to 0. Now let us consider a YZX convention with the set {$\frac{\pi}{4}$,$\frac{\pi}{2}$,0}. So again, one has a rotation of $\frac{\pi}{4}$ around the Y-axis and of $\frac{\pi}{2}$ around the Z-axis. But the axes change in the course of the rotation sequence, so the result is different! This is shown in Figure 2.17. With the first convention the coordinates of the final vector are $[\frac{1}{\sqrt{2}}, 0.5, 0.5]$ and with the second $[0.5, -\frac{1}{\sqrt{2}}, 0.5]$.

Mathematically, to calculate V' the rotation of the vector V by the set of Euler angles γ, β, α in ZYX convention, one has to apply $V' = R\,V$ with R the rotation matrix defined by Equation 2.25 [21]. Note how the 9 components of the matrix are needed whereas only 4 components are needed when describing rotation with a quaternion. Here, only the rotation of vectors are mentioned and not of an object. The difference is that a vector has only a direction, no ”orientation” whereas an object does. That orientation can however be defined by the direction of three vectors that define the frame of the body, for example the green dotted vectors in the above figures. And the principle of applying the rotation is exactly the same as applying a rotation to a single vector, but to all 3 vectors of the body frame separately.

$$R = \begin{bmatrix} \cos(\beta)\cos(\gamma) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma) - \cos(\beta)\sin(\gamma) & \sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\beta)\cos(\gamma) & \sin(\alpha)\cos(\beta) \\ \cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\beta)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma) - \sin(\beta)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix} \quad (2.25)$$

That matrix would change given that the convention changes. Thus, Euler angles suffer from ambiguity due to different conventions. But if the convention is known and fixed, that problem should be solved. However, there still are disadvantages of using Euler angles.

Consider the previous case of rotations displayed in Figure 2.14 and remember how ”easy” it was to represent with quaternions. Now one can try to formulate the same rotation with Euler angles/rotation

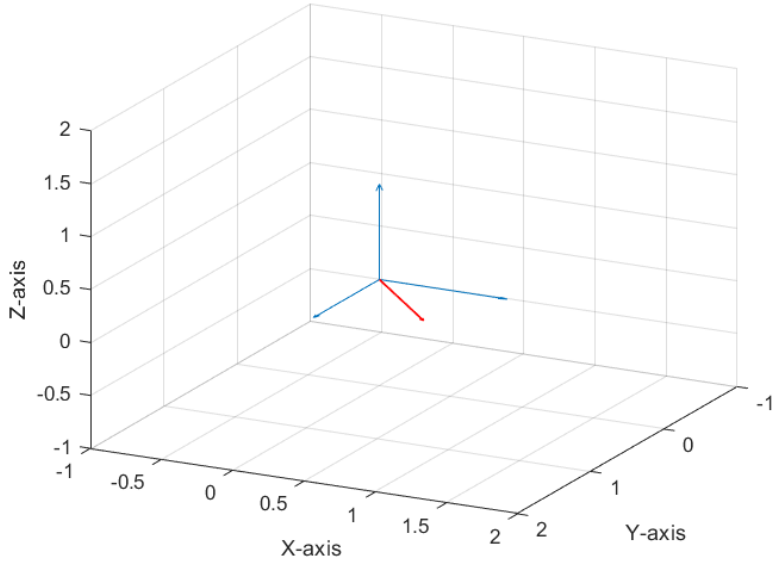
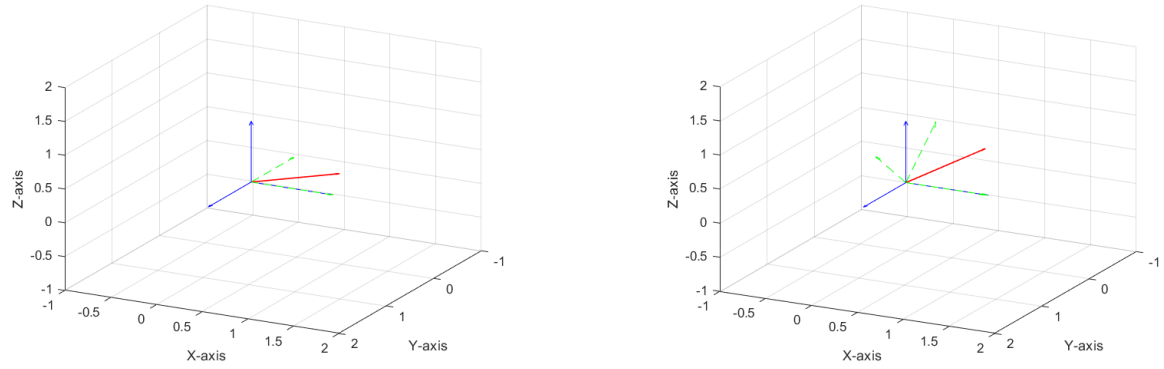


Figure 2.15: Initial position of the vector in red and of the basis in blue (matching the Cartesian basis)



(a) First rotation of $\frac{\pi}{2}$ about the Z-axis

(b) Second rotation of $\frac{\pi}{4}$ about the new Y-axis

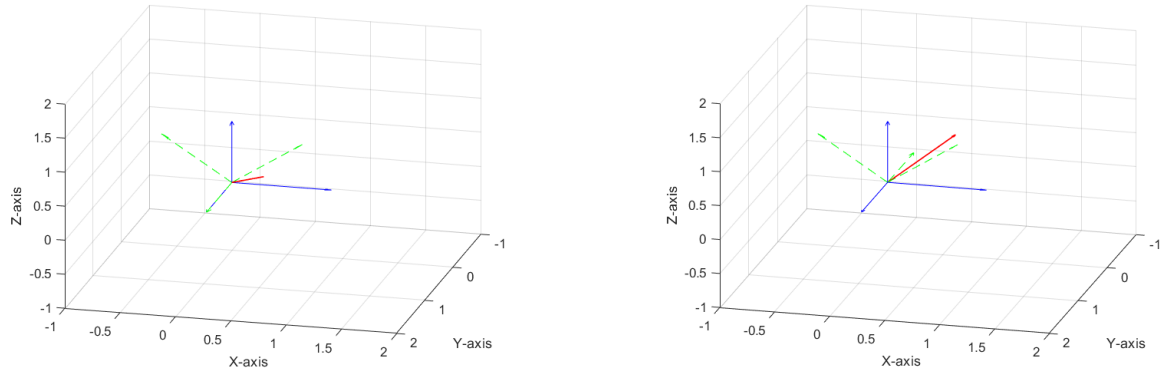
Figure 2.16: Rotation with Euler angles and ZYX convention

matrices. There exists a correspondence between Euler angles and quaternions that is given in Appendix A.3. The initial vector $[0, 0, 1]$ rotated by $\frac{\pi}{4}$ around $[1, 1, 1]$ corresponds to the rotation performed by the unit quaternion $\left[\cos(\frac{\pi}{8}), \sin(\frac{\pi}{8})\ [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]\right]$. According to Equation A.7, the corresponding Euler angles in ZYX convention are:

$$\gamma \approx 0.1786\pi$$
$$\beta \approx 0.1005\pi$$
$$\alpha \approx 0.1786\pi$$

So, even for such a simple rotation, there is no simple form for this set of Euler angles. Finally, one last argument to make a case for the use quaternions (or rather, against the use of Euler angles) is the problem of *Gimbal lock*.

It is challenging to explain Gimbal lock on a plane paper, but one can simply understand that Gimbal lock happens on measurement unit systems (such as gyroscopes) that use so-called gimbals to perform rotation with Euler angles. In certain configurations, these gimbals align and the system loses one degree of freedom : two of the three gimbals are ”locked” together. It is a well-know issue in the



(a) First rotation of $\frac{\pi}{4}$ about the Y-axis

(b) Second rotation of $\frac{\pi}{2}$ about the new Z-axis

Figure 2.17: Rotation with Euler angles and YZX convention

aerospace industry [16]. Quaternions avoid that problem since they only use one rotation instead of three consecutive ones. In this subsection, quaternions were represented by the effect they have on 3D vectors, so by representing the initial vector and then the vector rotated by the quaternion. This is one way of visualizing quaternions. In the next subsection, another way of visualizing them as objects themselves, instead of the operation they perform on other objects (namely 3D vectors), will be detailed. This is useful for visualizing interpolation for example. However, visualizing quaternions is not an easy task because they are 4-dimensional. And even if unit quaternions can be ”reduced” to 3 dimensions, they will still be visualized on this 2-dimensional support.

2.3.2. Defining kinematics of the problem

The aim of this whole section is to be able to represent efficiently the orientation and rotational motion of the lidar. In this subsection, it will be described how to do so with quaternions and at the same time, the linear part of the motion will be addressed : position and linear velocity/acceleration. The goal is to first know with as much precision as possible the position/orientation of the lidar in time and space so the position it is pointing to can be determined. Then, one also wants to determine the velocity of the lidar that needs to be reconstructed because the lidar only measures the wind velocity with respect to the lidar. So, the knowledge of three physical parameters is crucial:

- $\boldsymbol{p}(t)$: position of the lidar in the global frame
- $q(t)$: orientation of the lidar in quaternion form such that if one lidar beam has a direction determined by the unit vector $\boldsymbol{b}$ in the lidar frame (for pulsed lidars this is not time-dependent), then in the global frame, its direction is $q(t)\ \boldsymbol{b}\ q^{-1}(t)$ [1]. Since the lidar is rigidly fixed to the turbine, its orientation in time compared to that at $t = 0$ in the global frame is the same as any other component rigidly fixed to the turbine, in particular the Inertial Navigation System (INS).
- $\frac{d}{dt}\left(\boldsymbol{p}(t)\right)$: velocity of the lidar in the global frame

The position/orientation is represented at a time t is shown in Figure 2.18 with $\boldsymbol{p}(t) = [x(t), y(t), z(t)]$. It is fundamental to understand the correspondence between the body frame of the lidar and that of the INS and also with the global frame. A INS is typically comprised of an Inertial Measurement Unit (IMU) that can give linear acceleration, velocity, as well as angular acceleration, velocity and orientation. The INS has additional sensors that can also provide real-time position from GNSS (Global Navigation Satellite System) data. The lidar and the INS frames undergo the same rotations (under the rigid-body hypothesis). Given the initial position offset between the lidar and the INS is known, the position of the lidar is simply the position of the INS with a correction for that initial offset rotated. So the position of the INS, its orientation (which are directly output from the device) and the initial offset between the

[1] Remember that to simplify, a quaternion [0,$\boldsymbol{v}$] associated to any vector of $\mathbb{R}^3$ $\boldsymbol{v}$ will simply be written as $\boldsymbol{v}$. So, if quaternions are involved, any vector should be understood as its quaternion form with a null scalar part.

lidar and the INS are enough to get the lidar position. Regarding velocity, the lidar and INS have different velocities/accelerations but they have the same angular velocity/acceleration. Thus again, the INS outputs and some additional information on the initial position offset with the lidar is enough to determine the kinematic parameters of the lidar in time (in the rigid-body assumption between INS and lidar!). Now how does the INS frame relate to the global earth frame?

The quaternion $q(t)$ represents the orientation of the rotating body frame (INS for example) with respect to an inertial observer at the time t. That could be represented by the position and orientation of the initial INS frame. In this frame, the beams have a constant position. See Figure 2.18 for an example of the rotated frame $R(t)$. If one writes $\boldsymbol{e_1}^0, \boldsymbol{e_2}^0, \boldsymbol{e_3}^0$ the vectors of the orthonormal earth inertial frame R_0, then the 3 vectors of the rotated frame in the coordinates of the fixed frame at the time t can be expressed as:

$$\begin{cases} \boldsymbol{e_1}(t) &= q(t)\, \boldsymbol{e_1}^0\, q^{-1}(t) \\ \boldsymbol{e_2}(t) &= q(t)\, \boldsymbol{e_2}^0\, q^{-1}(t) \\ \boldsymbol{e_3}(t) &= q(t)\, \boldsymbol{e_3}^0\, q^{-1}(t) \end{cases}$$

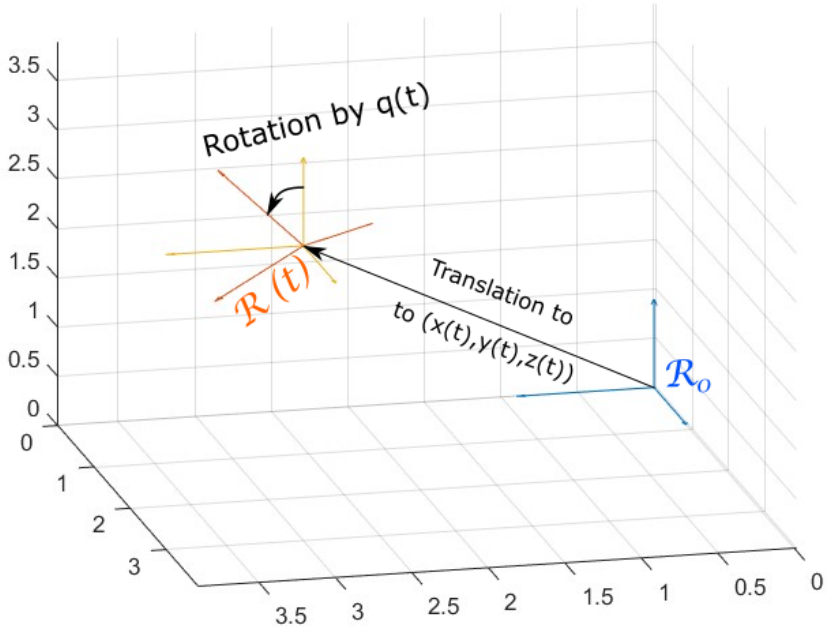


Figure 2.18: $R(t)$ is the frame R_0 after a translation and a rotation by $q(t)$

In the fixed global frame, one can express the quaternion first derivative $\dot{q}(t)$ as a function of the quaternion $q(t)$ and the angular velocity of the body to earth $\boldsymbol{\omega_e}(t)$, see Equation 2.26 [4]. With respect to the body angular velocity in the rotating frame $\boldsymbol{\omega_b}(t)$, one can also write Equation 2.27 [88] and the relation between the two is found in Equation 2.28. Note that this symbolizes a rotation opposite to that represented by $q(t)$. From now on, the body angular velocity $\boldsymbol{\omega_b}(t)$ will be used.

$$\dot{q}(t) = \frac{1}{2}\, \boldsymbol{\omega_e}(t)\, q(t) \tag{2.26}$$

$$\dot{q}(t) = \frac{1}{2}\, q(t)\, \boldsymbol{\omega_b}(t) \tag{2.27}$$

$$\boldsymbol{\omega_b}(t) = q^{-1}(t)\, \boldsymbol{\omega_e}(t)\, q(t) \tag{2.28}$$

Then, one can also express the quaternion second derivative which comes from $\ddot{q}(t) = \frac{1}{2}\frac{\partial}{\partial t}\left(q(t)\boldsymbol{\omega_b}(t)\right)$. From the product rule and the use of Equation 2.26, Equation 2.29 comes. The quaternion form of the angular velocity and acceleration should be considered in the equation, with a null scalar part.

$$\ddot{q}(t) = \frac{1}{4}\left(q(t)\,\boldsymbol{\omega_b}(t)\,\boldsymbol{\omega_b}(t) + 2\,q(t)\,\dot{\boldsymbol{\omega}}_{\boldsymbol{b}}(t)\right) \tag{2.29}$$

Now that all quantities are defined, the need for interpolation can be explained.

2.4. Interpolating translation and orientation

Fundamentally, the lidar gives as an output a time series of $V_r(t)$ at n different time steps $t_0, t_1, ..., t_n$. For the pulsed lidar of this project, the sampling rate is of $1\,\mathrm{Hz}$ [86], so the spacing between the time steps is of $1\,\mathrm{s}$. Note that these outputs alternate between each line-of-sight and there are 4 lines-of-sight in this case (4 different beam apertures) so for one specific line-of-sight, the spacing is actually of $4\,\mathrm{s}$ (or $0.25\,\mathrm{Hz}$). And, all the INS outputs are given with a different sampling rate and the time steps *do not correspond* with that of the lidar. Typically here, the output rate of the INS is $25\,\mathrm{Hz}$ (so one output every $0.04\,\mathrm{s}$) and the position/orientation data is noisy and suffers from discontinuity because of the filters used (more on this in the next chapter). Thus, interpolation of the motion data is needed to reconstruct position/velocity at any time step t_i of the VLOS measurements.

Now there are some physical constraints on the interpolated quantities, in particular for position and orientation. Let the position of the object (for example INS or lidar) be represented in time by $\boldsymbol{p}(t)$ and its orientation by $q(t)$. The quaternion $q(t)$ relates to the body angular velocity $\boldsymbol{\omega_b}(t)$ through Equation 2.27 and to the angular acceleration through Equation 2.29.

And $\boldsymbol{p}(t)$ relates to the body velocity through $\boldsymbol{V_b}(t) = q^{-1}(t)\boldsymbol{V_e}(t)q(t)$ where $\boldsymbol{V_e}(t) = \frac{d\boldsymbol{p}(t)}{dt}$ is the velocity of the object in a global inertial frame. Then, $\dot{\boldsymbol{V}}_{\boldsymbol{b}}(t)$ is the linear acceleration in the body (non-inertial) frame. Dynamically, these quantities are linked to the moments and forces applied to the object through the classical equations of motion 2.30 [21]. The first line is the linear equation while the second line is the angular equation. I_n is the inertia tensor ($\mathrm{kg\,m^2}$) and m is the mass of the body.

$$\begin{aligned} m\,\dot{\boldsymbol{V}}_{\boldsymbol{b}}(t) + \boldsymbol{\omega_b}(t) \times (m\,\boldsymbol{V_b}(t)) &= \boldsymbol{F_{ext}}(t) \\ I_n(t)\dot{\boldsymbol{\omega}}_{\boldsymbol{b}}(t) + \boldsymbol{\omega_b}(t) \times (I_n(t)\boldsymbol{\omega_b}(t)) &= \boldsymbol{M_{ext}}(t) \end{aligned} \tag{2.30}$$

In the scope of the project, the dynamics of the system are not studied but at least these two equations should be verified. Since the external forces and moments applied are continuous it comes that all quantities on the left hand sides of Equations 2.30 should be continuous. That implies in particular the continuity of the partial time derivative of position $\ddot{\boldsymbol{p}}$ (linear acceleration) and of $\ddot{q}$. In mathematical terms, the functions of time $\boldsymbol{p}$ and q need to be C^2 - continuous, i.e. twice differentiable with continuous second derivatives. Thus, any function used for interpolating $\boldsymbol{p}$ and q that is not C^2 - continuous would be physically incorrect. For piece-wise defined functions, this creates a problem particularly at the interpolation nodes. Note that for example, using linear interpolation for position would be fundamentally wrong, because that would imply constant velocity and null acceleration. For simplicity or because high order methods have their own limitations, a lower order method that does not respect the dynamic continuity prerequisite is also an option. In the next two subsections, the interpolation of translation and of orientation will be treated separately, especially because interpolating orientation is significantly more challenging, because it is a 4-dimensional problem.

2.4.1. Quintic Hermite interpolation of translation

This section will be focused on the linear part of Equations 2.30, so the 3 translational degrees of freedom. The angular part will be treated in the next, so the 3 rotational degrees of freedom. Position can be interpreted as a parametric function of time with 3 components $\boldsymbol{p}(t) = [x(t), y(t), z(t)] \in \mathbb{R}^3$. The motion outputs come from the INS which are then used to determine the attitude of the lidar but that needs further transformations. At first, $\boldsymbol{p}$ should be understood as $\boldsymbol{p_{INS}}$.

The outputs of the INS are $\boldsymbol{p}(t_i)$, $\dot{\boldsymbol{p}}(t_i)$ and $\ddot{\boldsymbol{p}}(t_i)$ where i ranges from 0 to n, with $n+1$ the number of time stamps and $t_0 < t_1 < ... < t_i < ... < t_n$. In parallel, the lidar measures the line-of-sight velocity at a time t that falls in between two time steps of the INS data : there is t_i such that $t \in [t_i; t_{i+1}]$. One method of interpolation that allows for C^2 - continuity of the interpolated quantity and an error with the actual quantity (that itself needs to be C^2) in the order of magnitude of $(t_{i+1} - t_i)^6$ is Quintic-Hermite interpolation. This method uses 6 basis functions, referred to as *Quintic-Hermite basis functions* [82],

that will be noted ξ_0, ξ_1, ξ_2, ξ_3, ξ_4 and ξ_5. Their expressions are given in Equation 2.32 [82]. For a time $t \in [t_i; t_{i+1}]$, the interpolated position $\hat{\boldsymbol{p}}(t)$ with Quintic-Hermite interpolation is given by Equation 2.31 [82] with $\Delta t = t_{i+1} - t_i$.

$$\begin{aligned}\hat{\boldsymbol{p}}(t) = \boldsymbol{p}(t_i)\,\xi_0\left(\frac{t-t_i}{t_{i+1}-t_i}\right) + \dot{\boldsymbol{p}}(t_i)\,\Delta t\,\xi_1\left(\frac{t-t_i}{t_{i+1}-t_i}\right) + \ddot{\boldsymbol{p}}(t_i)\,\frac{\Delta t^2}{2}\,\xi_2\left(\frac{t-t_i}{t_{i+1}-t_i}\right) \\ + \boldsymbol{p}(t_{i+1})\,\xi_3\left(\frac{t-t_i}{t_{i+1}-t_i}\right) + \dot{\boldsymbol{p}}(t_{i+1})\,\Delta t\,\xi_4\left(\frac{t-t_i}{t_{i+1}-t_i}\right) + \ddot{\boldsymbol{p}}(t_{i+1})\,\frac{\Delta t^2}{2}\,\xi_5\left(\frac{t-t_i}{t_{i+1}-t_i}\right)\end{aligned} \tag{2.31}$$

$$\forall\, s \in [0;1], \begin{cases} \xi_0(s) = (1-s)^3\,(6s^2+3s+1) \\ \xi_1(s) = (1-s)^3\,s\,(3s+1) \\ \xi_2(s) = (1-s)^3\,s^2 \\ \xi_3(s) = s^3\,(6s^2-15s+10) \\ \xi_4(s) = s^3\,(1-s)\,(3s-4) \\ \xi_5(s) = s^3\,(1-s)^2 \end{cases} \tag{2.32}$$

One can then define the interpolated position using piece-wise defined functions. Let $t \in [t_0; t_n]$ and $i \in [\![0; n-1]\!]$. Then, h_0^i, h_1^i, h_2^i, h_3^i, h_4^i, h_5^i are new basis functions defined such that (recall that for each i, $\Delta t_i = t_{i+1} - t_i$):

- $h_0^i(t) = \begin{cases} \xi_0(\frac{t-t_i}{t_{i+1}-t_i}) & \text{if } t \in [t_i; t_{i+1}] \\ 0 & \text{else.} \end{cases}$
- $h_1^i(t) = \begin{cases} \Delta t_i\,\xi_1(\frac{t-t_i}{t_{i+1}-t_i}) & \text{if } t \in [t_i; t_{i+1}] \\ 0 & \text{else.} \end{cases}$
- $h_2^i(t) = \begin{cases} \frac{\Delta t_i^2}{2}\,\xi_2(\frac{t-t_i}{t_{i+1}-t_i}) & \text{if } t \in [t_i; t_{i+1}] \\ 0 & \text{else.} \end{cases}$
- $h_3^i(t) = \begin{cases} \xi_3(\frac{t-t_i}{t_{i+1}-t_i}) & \text{if } t \in [t_i; t_{i+1}] \\ 0 & \text{else.} \end{cases}$
- $h_4^i(t) = \begin{cases} \Delta t_i\,\xi_4(\frac{t-t_i}{t_{i+1}-t_i}) & \text{if } t \in [t_i; t_{i+1}] \\ 0 & \text{else.} \end{cases}$
- $h_5^i(t) = \begin{cases} \frac{\Delta t_i^2}{2}\,\xi_5(\frac{t-t_i}{t_{i+1}-t_i}) & \text{if } t \in [t_i; t_{i+1}] \\ 0 & \text{else.} \end{cases}$

To be able to rewrite Equation 2.31 in the desired form, so-called *control points* will be used. Let $t \in [t_i; t_{i+1}]$ and i the corresponding index. The control points $\boldsymbol{p_{i,k}}$ with $k \in [\![0; 5]\!]$ are defined as follows (there are a total of $6\,n$ control points):

$$\begin{cases} \boldsymbol{p_{i,0}} = \boldsymbol{p}(t_i) \\ \boldsymbol{p_{i,1}} = \dot{\boldsymbol{p}}(t_i) \\ \boldsymbol{p_{i,2}} = \ddot{\boldsymbol{p}}(t_i) \\ \boldsymbol{p_{i,3}} = \boldsymbol{p}(t_{i+1}) \\ \boldsymbol{p_{i,4}} = \dot{\boldsymbol{p}}(t_{i+1}) \\ \boldsymbol{p_{i,5}} = \ddot{\boldsymbol{p}}(t_{i+1}) \end{cases}$$

Now, the form of Equation 2.31 can be changed into the much simpler Equation 2.33

$$\hat{\boldsymbol{p}}(t) = \sum_{k=0}^{5} \boldsymbol{p_{i,k}}\,h_k^i(t) \tag{2.33}$$

Then, the expression for $\hat{\boldsymbol{p}}$ for any $t \in [t_0; t_n]$ is found in Equation 2.34.

$$\hat{\boldsymbol{p}}(t) = \sum_{i=0}^{n-1}\sum_{j=0}^{2}\left(\frac{\partial^j \boldsymbol{p}}{\partial t^j}(t_i)\,h_j^i(t) + \frac{\partial^j \boldsymbol{p}}{\partial t^j}(t_{i+1})\,h_{j+3}^i(t)\right) \tag{2.34}$$

It appears easier to write this Equation as a matrix product. In this case, only a 1-by-$6\,n$ matrix for the values of the basis functions and a $6\,n$-by-1 matrix for the control points are needed. The matrix form of Equation 2.34 is Equation 2.35.

$$\hat{\boldsymbol{p}}(t)=\begin{bmatrix} h_0^0(t) & h_1^0(t) & h_2^0(t) & h_3^0(t) & h_4^0(t) & h_5^0(t) & \dots & h_0^{n-1}(t) & h_1^{n-1}(t) & h_2^{n-1}(t) & h_3^{n-1}(t) & h_4^{n-1}(t) & h_5^{n-1}(t) \end{bmatrix} \begin{bmatrix} \boldsymbol{p_{0,0}} \\ \boldsymbol{p_{0,1}} \\ \boldsymbol{p_{0,2}} \\ \boldsymbol{p_{0,3}} \\ \boldsymbol{p_{0,4}} \\ \boldsymbol{p_{0,5}} \\ \vdots \\ \boldsymbol{p_{n-1,0}} \\ \boldsymbol{p_{n-1,1}} \\ \boldsymbol{p_{n-1,2}} \\ \boldsymbol{p_{n-1,3}} \\ \boldsymbol{p_{n-1,4}} \\ \boldsymbol{p_{n-1,5}} \end{bmatrix} \tag{2.35}$$

In practice, the position of the lidar needs to be interpolated at different times, which here will be the time stamps corresponding the lidar VLOS measurements, that will be designated as T_l with $l \in [\![0; m]\!]$ and each $T_l \in [t_0; t_n]$. All these interpolated quantities can be represented as a column vector. That can again be written in a matrix form, see Equation 2.36. The 1-by-$6\,n$ left matrix has become a $m+1$-by-$6\,n$ matrix while the right matrix for the control points remains the same.
Note that the matrix of the basis function values is almost filled with zeros, since per row, only the 6 basis functions that are defined on the corresponding interval can have non-zero values. For the derivatives of the interpolated function with respect to time, the Equation looks almost exactly the same, the difference being that the basis functions derivatives with respect to time are now evaluated. Control points stay the same.

$$\begin{bmatrix} \hat{\boldsymbol{p}}(T_0) \\ \hat{\boldsymbol{p}}(T_1) \\ \dots \\ \hat{\boldsymbol{p}}(T_{m-1}) \\ \hat{\boldsymbol{p}}(T_m) \end{bmatrix} = \begin{bmatrix} h_0^0(T_0) & h_1^0(T_0) & h_2^0(T_0) & \dots & h_3^{n-1}(T_0) & h_4^{n-1}(T_0) & h_5^{n-1}(T_0) \\ h_0^0(T_1) & h_1^0(T_1) & h_2^0(T_1) & \dots & h_3^{n-1}(T_1) & h_4^{n-1}(T_1) & h_5^{n-1}(T_1) \\ & & & \ddots & & & \\ h_0^0(T_{m-1}) & h_1^0(T_{m-1}) & h_2^0(T_{m-1}) & \dots & h_3^{n-1}(T_{m-1}) & h_4^{n-1}(T_{m-1}) & h_5^{n-1}(T_{m-1}) \\ h_0^0(T_m) & h_1^0(T_m) & h_2^0(T_m) & \dots & h_3^{n-1}(T_m) & h_4^{n-1}(T_m) & h_5^{n-1}(T_m) \end{bmatrix} \begin{bmatrix} \boldsymbol{p_{0,0}} \\ \boldsymbol{p_{0,1}} \\ \boldsymbol{p_{0,2}} \\ \vdots \\ \boldsymbol{p_{n-1,3}} \\ \boldsymbol{p_{n-1,4}} \\ \boldsymbol{p_{n-1,5}} \end{bmatrix} \tag{2.36}$$

One last valuable aspect to note is that if the values of the position and/or its first and/or its second derivative are not known at the nodes t_i, some control points can still be found by knowing the the measurements on the left hand side and fitting the control points to the values of the basis functions for an arbitrary set of fitting time nodes. This is basically the inverse problem.
For example, take the cubic function $f : t \longmapsto t^3$ and define two nodes at $t = 1$ and $t = 3$ (so $\Delta t_0 = 2$). For this example, one knows the value of f from 1.2 to 2.8 with a step of 0.2, but not of any derivative. According to Equation 2.36, there simply is:

$$\begin{bmatrix} f(1.2) \\ f(1.4) \\ \dots \\ f(2.6) \\ f(2.8) \end{bmatrix} = \begin{bmatrix} h_0^0(1.2) & h_1^0(1.2) & h_2^0(1.2) & h_3^0(1.2) & h_4^0(1.2) & h_5^0(1.2) \\ h_0^0(1.4) & h_1^0(1.4) & h_2^0(1.4) & h_3^0(1.4) & h_4^0(1.4) & h_5^0(1.4) \\ & & & \ddots & & \\ h_0^0(2.6) & h_1^0(2.6) & h_2^0(2.6) & h_3^0(2.6) & h_4^0(2.6) & h_5^0(2.6) \\ h_0^0(2.8) & h_1^0(2.8) & h_2^0(2.8) & h_3^0(2.8) & h_4^0(2.8) & h_5^0(2.8) \end{bmatrix} \begin{bmatrix} \boldsymbol{p_{0,0}} \\ \boldsymbol{p_{0,1}} \\ \boldsymbol{p_{0,2}} \\ \boldsymbol{p_{0,3}} \\ \boldsymbol{p_{0,4}} \\ \boldsymbol{p_{0,5}} \end{bmatrix}$$

And $\boldsymbol{p_{0,0}}, \boldsymbol{p_{0,1}}, \boldsymbol{p_{0,2}}, \boldsymbol{p_{0,3}}, \boldsymbol{p_{0,4}}, \boldsymbol{p_{0,5}}$ are the control points corresponding to these two nodes. According to the theory, they are estimates of respectively $f(1), f'(1), f''(1), f(3), f'(3), f''(3)$. Now one can calculate the values of the basis functions.

This is a problem of solving $A\,X = B$, where X is the column vector of the control points. Matlab can easily solve this and one gets $X^T = [1, 3, 6, 27, 27, 18]$. And these are exactly the values of f, f', f'' at the nodes 1 and 3. This can be generalized to any number of nodes.

Test case

To test the work did on Quintic Hermite interpolation, test on a synthetic translation motion was made. For this, as it was introduced previously, the motion in the X, Y and Z directions were considered to oscillate harmonically, as was suggested in literature [34]. This test was done on non-uniform nodes. Here, a case of pure translation is to be derived. Usually, the words *surge*, *sway* and *heave* are used to describe translation respectively in the X, Y and Z direction. The position vector $\boldsymbol{p}(t) = [x(t), y(t), z(t)]$ is here defined with each coordinate having different amplitude and period as Equation 2.37.

$$\begin{aligned} x(t) &= 5\sin(2\pi t) \\ y(t) &= \sin(\frac{2\pi}{10}t) \\ z(t) &= 2\sin(\frac{2\pi}{12}t) \end{aligned} \tag{2.37}$$

The position being sinusoidal, the derivatives come easily and are also sinusoidal. One can see the actual curves for the position, the velocity and acceleration in Appendix A.5 for a time going from 0 to $20\,\mathrm{s}$. Examples of the interpolation are also given for two numbers of nodes non-uniformly distributed between 0 and $20\,\mathrm{s}$. As expected, for a higher number of nodes, the red interpolated curve matches more closely the real blue one until the difference is not perceptible anymore. Given a number of nodes N, a rough approximation of the length of the time interval between two nodes is in this case $\Delta t \approx \frac{20}{N-1}$ which is in the order of magnitude of $\frac{1}{N}$ times a constant. Since the error is proportional to the length of the time interval between two nodes to the power of 6, it is then proportional to N^{-6}. In a logarithmic space, that corresponds to a -6 slope. For the first derivative, the order of the basis functions reduces by 1 so the slope becomes -5 and for the second derivative -4. This has been verified for this case and is shown in Figure 2.19. The respective slopes are also shown and slightly shifted for better visualization.

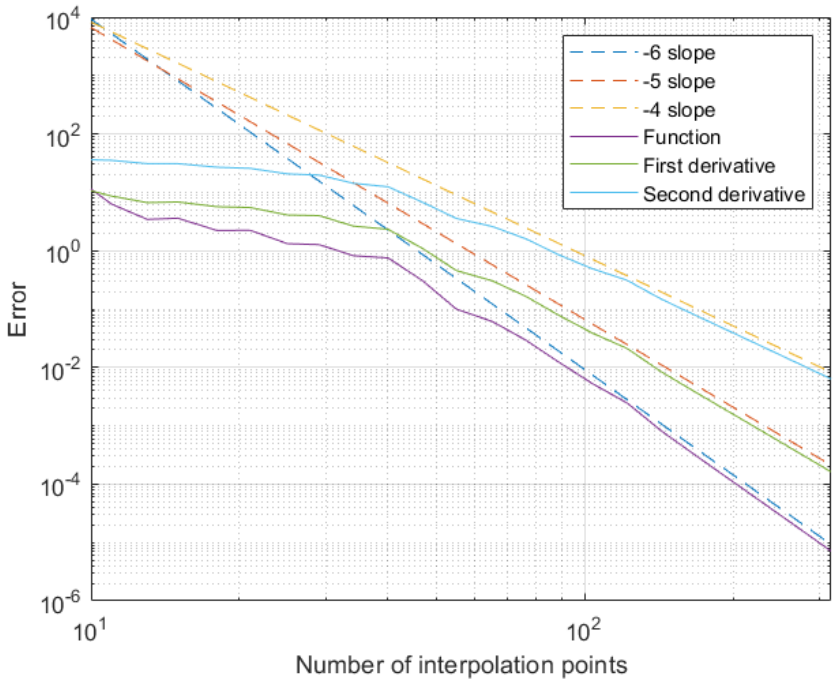


Figure 2.19: Verification of Quintic Hermite interpolation in this harmonic case

2.4.2. Interpolation of orientation

Following the introduction of quaternions in this report, they will be used to interpolate orientation. One other advantage they offer over interpolating with Euler angles is that only one interpolation is needed. On the other hand, Euler angle interpolation would require three separate interpolations, one for each of the three Euler angles. Moreover, separating these interpolations implies that all the Euler angles behave independently, which was shown not to be the case. Additionally, interpolating orientation matrices is too problematic [22]. Thus, interpolating orientation using quaternions appears to be the most viable option. Here, two methods will be derived and compared, each approach being analogous to the linear Hermite interpolation. The first one uses the Bézier curve construction scheme that uses a de Casteljau-like simple algorithm while the second one uses a blending of quaternion curves through product of exponentials and tool functions, called *cumulative basis functions*, that derive from Hermite basis polynomials. The two use similar control quaternions. While the Bézier curves offer simplicity of construction and implementation, the exponential blended curve offers simplicity of differentiation. Before anything, the simplest correct form of interpolation in quaternion space will be explained. It is called *Slerp*, for Spherical Linear Interpolation and is quite different from classic linear interpolation. Slerp interpolates smoothly along the shortest path between two quaternions [16] and its formula between two nodes unit quaternions q_0 and q_1 and for a given $h \in [0;1]$ is given by Equation 2.38. By rearranging Equation 2.38, one can also obtain a similar formula [16], which is given in Equation 2.39 and that will be used later. Note that the result is also a unit quaternion. This is the exponential form of Slerp. There also exists an equivalent formula that is built geometrically but it will have no use here.

$$Slerp(q_0, q_1, h) = q_0\,(q_0^*\,q_1)^h = q_0\,\exp(h\,\log(q_0^*\,q_1)) \tag{2.38}$$

$$Slerp(q_0, q_1, h) = (q_1\,q_0^*)^h\,q_0 = \exp(h\,\log(q_1\,q_0^*))\,q_0 \tag{2.39}$$

The path built by Slerp between two unit quaternion nodes on the projected unit sphere introduced before is shown in Figure 2.20 One encounters then the same problem as in $\mathbb{R}^3$. For a sequence of nodes, Slerp does not allow smooth interpolation and it is not continuously differentiable. This can be visualized in Figure 2.21 for 6 quaternion nodes. The way to correct this and have continuous differentiability is to use some control points. First, the Bézier quaternion curves will be introduced.

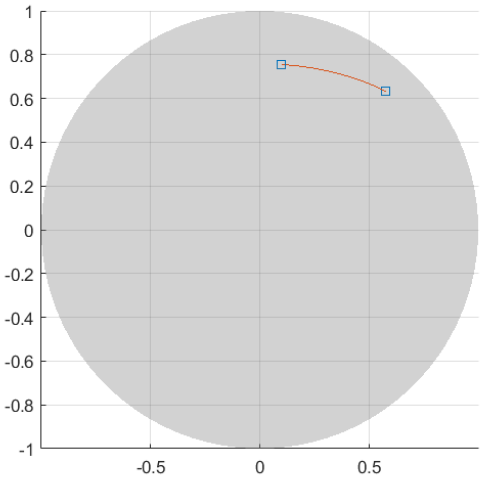


Figure 2.20: Slerp between two unit quaternions

Bézier quaternion curve

To start with, it is necessary to introduce the de Casteljau algorithm. It is used to build smooth curves, called Bézier curves. Typically, linear interpolation corresponds the Bézier curve of order 1, which uses 2 control points : the nodes. And a Bézier curve of order N uses N+1 control points, 2 that are the nodes and N-1 that are additional control points that will shape the curve. To understand how

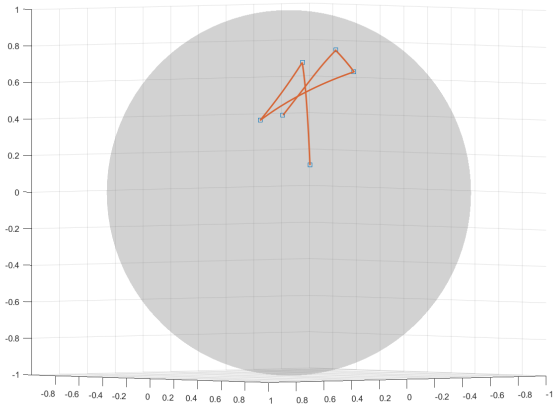


Figure 2.21: Slerp for a sequence of 6 unit quaternions

these control points help shape the curve, an example will be given for a Bézier curve of third order, drawn in Figure 2.22. The points at the nodes are $\boldsymbol{b_0}$ and $\boldsymbol{b_3}$. The two additional control points are $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$. In total, these 4 points create 3 segments : $[\boldsymbol{b_0}, \boldsymbol{b_1}]$, $[\boldsymbol{b_1}, \boldsymbol{b_2}]$ and $[\boldsymbol{b_2}, \boldsymbol{b_3}]$. The curve is built as such: (spherical) linear interpolation is done between $\boldsymbol{b_0}$ and $\boldsymbol{b_1}$, between $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ and between $\boldsymbol{b_2}$ and $\boldsymbol{b_3}$. That gives 3 intermediate points that make 2 segments. The previous step is repeated until there is only one segment, the linear interpolation of which gives the final point. This is done for each desired point in between $\boldsymbol{b_0}$ and $\boldsymbol{b_1}$, which gives that smooth shape of the Bézier curve. It is interesting to note that the previously defined Quintic Hermite interpolation can also be built using control points and the de Casteljau algorithm. It can also be used to build smooth quaternion curves.

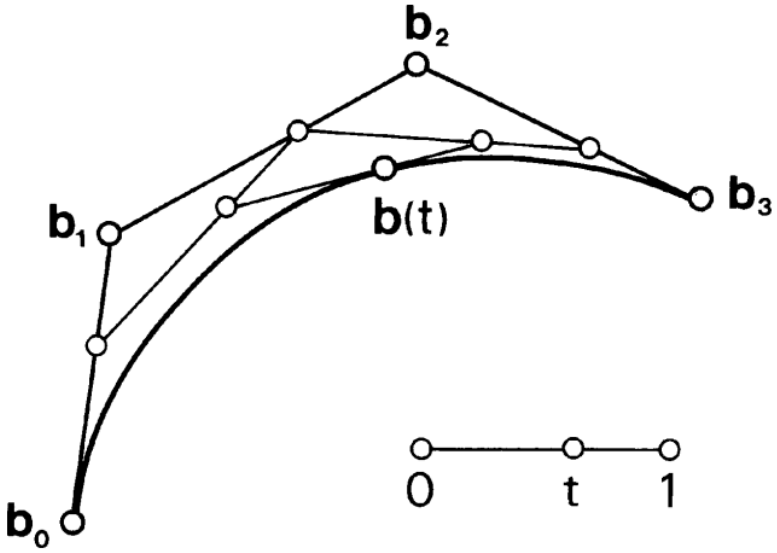


Figure 2.22: de Casteljau algorithm and Bézier curve for N=3 [6]

This algorithm was first proposed by K. Shoemake, prolific author in quaternion and animation [71]. It is similar to de Casteljau algorithm, where some control quaternions are chosen to smoothen the curve, the sub-segments of the curve defined by the control quaternions are iteratively interpolated with *Slerp*. An example is displayed in Figure 2.23, with the 4 control quaternions corresponding to the interpolation. The interpolated curve is not shown because it is too close to the real one.
The mathematical expression for the interpolated curve is a "cascaded" Slerp. Let $t \in [t_i; t_{i+1}]$, i the corresponding index and $h_i = \frac{t-t_i}{t_{i+1}-t_i}$. Then, the expression of the curve using 4 control points $q_{i,0}$,

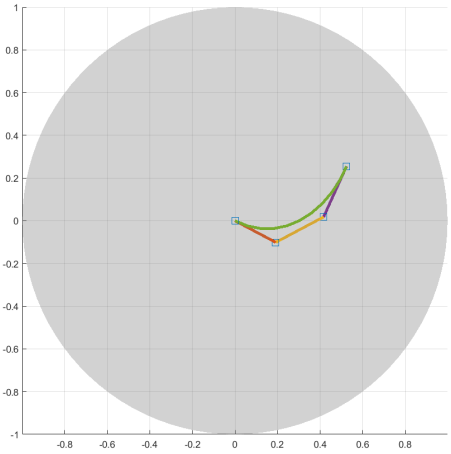


Figure 2.23: Control points for the quaternion curve between two nodes.

$q_{i,1}$, $q_{i,2}$, $q_{i,3}$ is given by Equation 2.40. Its decomposed and simpler to handle form is divided below.

$$\begin{aligned} \hat{q_4}(t) = Slerp(&Slerp(Slerp(q_{i,0}, q_{i,1},\ h_i), Slerp(q_{i,1}, q_{i,2},\ h_i),\ h_i),\\ &Slerp(Slerp(q_{i,1}, q_{i,2},\ h_i), Slerp(q_{i,2}, q_{i,3},\ h_i),\ h_i),\ h_i) \end{aligned} \tag{2.40}$$

$$\begin{aligned} \hat{q_4}(t) &= Slerp(p1(t), p2(t),\ h_i)\\ p1(t) = Slerp(r1(t), r2(t),\ h_i), \quad p2(t) &= Slerp(r2(t), r3(t),\ h_i)\\ r1(t) = Slerp(q_{i,0}, q_{i,1},\ h_i), \quad r2(t) &= Slerp(q_{i,1}, q_{i,2},\ h_i), \quad r3(t) = Slerp(q_{i,2}, q_{i,3},\ h_i) \end{aligned}$$

Now one has to find the expression of the control points as a function of the quaternion values at the nodes and its derivatives. First, it comes that $\hat{q_4}(t_i) = q(t_i)$. And by evaluating Equation 2.40 for $h_i = 0$, $\hat{q_4}(t_i) = q_{i,0}$. By doing the same operation at the second node for t_{i+1}, it follows that $q_{i,0} = q(t_i)$ and $q_{i,3} = q(t_{i+1})$. This can be seen in Figure 2.23, where the first and last control points are on the actual curve.

And for the control points in the middle, the derivative at the nodes should be used. First, by definition (see Equation 2.27), $\dot{q}(t_i) = \frac{1}{2}\, q(t_i)\, \boldsymbol{\omega_b}(t_i)$ and $\dot{q}(t_{i+1}) = \frac{1}{2}\, q(t_{i+1})\, \boldsymbol{\omega_b}(t_{i+1})$. The formula for the first derivative of that curve at the nodes was derived in [43] (here an alternative equivalent formula for Slerp is used):

$$\begin{aligned} \dot{\hat{q}}_4(t_i) &= \frac{\partial h_i}{\partial t}(t_i)\, 3\, q_{i,0}\ \log(q_{i,0}^{-1}\, q_{i,1}) = \frac{3}{t_{i+1} - t_i}\, q(t_i)\ \log(q^{-1}(t_i)\, q_{i,1})\\ \dot{\hat{q}}_4(t_{i+1}) &= \frac{\partial h_i}{\partial t}(t_{i+1})\, 3\, q_{i,3}\ \log(q_{i,2}^{-1}\, q_{i,3}) = \frac{3}{t_{i+1} - t_i}\, q(t_{i+1})\ \log(q_{i,2}^{-1}\, q(t_{i+1})) \end{aligned}$$

From the boundary condition for the first derivative at the nodes, the expression of the two middle control points comes as Equation 2.41.

$$\begin{aligned} q_{i,1} &= q(t_i)\ \exp(\frac{t_{i+1} - t_i}{6}\, \boldsymbol{\omega_b}(t_i))\\ q_{i,2} &= q(t_{i+1})\ \exp(-\frac{t_{i+1} - t_i}{6}\, \boldsymbol{\omega_b}(t_{i+1})) \end{aligned} \tag{2.41}$$

Even if that method gives good results, it has two problems. First, it is not C^2 - continuous [42]. Then, computing the derivative of that curve is far from straightforward because of its recursive Slerp form.

For the first problem, one could think of building a higher order Bézier quaternion that would be C^2 - continuous, typically using 6 control points instead of 4. However, for proving C^2 - continuity, an expression for the second derivative of the curve at the nodes should be used, which is much more complex than that of the first derivative [43]. To simplify the problem of the differentiation of the interpolating curve, a class of interpolating quaternion functions using product of exponentials was created [42] and will be the subject of the next section.

Cumulative basis quaternion curve

This class of quaternion curves uses a product of exponentials with *tool functions*, that are a cumulative form of the *Bernstein* basis polynomials [42]. These Bernstein polynomials are linked with the Hermite basis polynomials introduced in the previous section. They are the tools for building the control points of the Hermite interpolation of any order. For example for the degree 5 corresponding to Quintic Hermite interpolation, there are 6 Bernstein basis polynomials that are functions of the Hermite ones introduced in Equation 2.32. In this part, only the degree 5 (so with 6 control quaternions) will be studied, because the degree 3 introduced in literature (4 control quaternions) [42] only allows for C^1 - continuity and not further. The expression of the 6 Bernstein basis polynomials for $i \in [\![0;5]\!]$ of degree 5 is found in Equation 2.42 [39].

$$\forall h \in [0;1], \beta_i(h) = \binom{5}{i}(1-h)^{5-i}\, h^i \tag{2.42}$$

As tool functions, this quaternion interpolating curve uses the cumulative form of this basis, which is simply a sum of these polynomials, as defined in Equation 2.43.

$$\forall h \in [0;1], \tilde{\beta}_i(h) = \sum_{j=i}^{5} \beta_j(h) \tag{2.43}$$

In particular, $\tilde{\beta}_0 = 1$ and $\tilde{\beta}_5 = \beta_5$. Now, it is possible to define the expression of the curve, with the control quaternions $q_0,\ q_1,\ q_2,\ q_3,\ q_4,\ q_5$ as Equation 2.44, with $t \in [t_i; t_{i+1}]$ and i the corresponding index : $h_i = \frac{t-t_i}{t_{i+1}-t_i}$.

$$\begin{aligned}\hat{q}(t) = q_0\ \exp(\log(q_0^{-1}\, q_1)\ \tilde{\beta}_1(h_i)))\ \exp(\log(q_1^{-1}\, q_2)\ \tilde{\beta}_2(h_i))\ \exp(\log(q_2^{-1}\, q_3)\ \tilde{\beta}_3(h_i))\\ \exp(\log(q_3^{-1}\, q_4)\ \tilde{\beta}_4(h_i))\ \exp(\log(q_4^{-1}\, q_5)\ \tilde{\beta}_5(h_i))\end{aligned} \tag{2.44}$$

It is also possible to write it with using $\Omega_k = \log(q_{k-1}^{-1}\, q_k)$ as Equation 2.45.

$$\hat{q}(t) = q_0 \prod_{k=1}^{5} \exp(\Omega_k\ \tilde{\beta}_k(h_i)) \tag{2.45}$$

Interestingly, for this class of curve of order 1 (2 control quaternions), the exact formula of Slerp is found. Now, the expression of the control quaternions needs to be found. For this, one can use the boundary conditions at the nodes. Let $t \in [t_i; t_{i+1}]$ and i the corresponding index, then:

$$\begin{cases}\hat{q}(t_i) = q(t_i)\\ \hat{q}(t_{i+1}) = q(t_{i+1})\\ \dfrac{\partial \hat{q}}{\partial t}(t_i) = \dot{q}(t_i)\\ \dfrac{\partial \hat{q}}{\partial t}(t_{i+1}) = \dot{q}(t_{i+1})\\ \dfrac{\partial^2 \hat{q}}{\partial t^2}(t_i) = \ddot{q}(t_i)\\ \dfrac{\partial^2 \hat{q}}{\partial t^2}(t_{i+1}) = \ddot{q}(t_{i+1})\end{cases}$$

Directly, it comes that $q_0 = q(t_i)$ and $q_5 = q(t_{i+1})$. Then, one needs the expression of the first and second derivatives of $\hat{q}$. For the first derivative, one needs to use that for example [42]:

$$\frac{\partial(\exp(\log(q_0^{-1}\, q_1)\ \tilde{\beta}_1(h_i)))}{\partial t} = \exp(\log(q_0^{-1}\, q_1)\ \tilde{\beta}_1(h_i))\ \log(q_0^{-1}\, q_1)\ \frac{\partial \tilde{\beta}_1(h_i)}{\partial t}$$

The expression for the first derivative of $\hat{q}$ is found in Equation A.10. The formula for the second derivative becomes even larger. Using the expression for $\hat{q}$, its derivatives, that of the cumulative basis functions and Equations 2.27, 2.29, the expression for the additional control points is found below in Equation 2.46. It is used that $(\exp(q))^{-1} = \exp(-q)$ because of the odd/even nature of the sinusoidal functions. An example of these control quaternions is shown in Figure A.13. The complete derivation is shown in Appendix A.6.2.

$$\begin{aligned} q_1 &= q(t_i)\ \exp(\frac{\Delta t_i\ \boldsymbol{\omega_b}(t_i)}{10}) \\ q_2 &= q_1\ \exp(\frac{\Delta t_i}{10}\ (\boldsymbol{\omega_b}(t_i) + \frac{\Delta t_i}{4}\ \dot{\boldsymbol{\omega}}_{\boldsymbol{b}}(t_i))) \\ q_4 &= q(t_{i+1})\ \exp(-\frac{\Delta t_i\ \boldsymbol{\omega_b}(t_{i+1})}{10}) \\ q_3 &= q_4\ \exp(-\frac{\Delta t_i}{10}\ (\boldsymbol{\omega_b}(t_{i+1}) - \frac{\Delta t_i}{4}\ \exp\Big(\frac{\Delta t_i\ \boldsymbol{\omega_b}(t_{i+1})}{10}\Big)\ \dot{\boldsymbol{\omega}}_{\boldsymbol{b}}(t_{i+1})\ \exp\Big(-\frac{\Delta t_i\ \boldsymbol{\omega_b}(t_{i+1})}{10}\Big))) \end{aligned} \tag{2.46}$$

Test case

Now it is possible to test these 2 classes of interpolating quaternion curves on a synthetic case. For this, a problem with only rotation was created, no translation. The rotation angle and the rotation axis are time-dependent and are sinusoidal functions. The rotation angle has an amplitude of $90°$ and a period of $16\,\mathrm{s}$ while the rotation axis has a period of $9\,\mathrm{s}$ (and an amplitude of 1 since it is a unit vector). The quaternion path followed by the corresponding sequence of quaternions is drawn for a time of $25\,\mathrm{s}$ in Figure 2.24. The corresponding sequence of rotations applied to the vector $[1, 1, 1]$ is shown in Appendix in Figure A.12. Interpolation of this curve was tested out with the two methods previously defined. The error plot for an increasing number of interpolation points is shown below in Figure 2.25. The error for the interpolation of the derivative with the quintic cumulative quaternion curve is also shown, not for the Bézier curve but this will be the topic of the next subchapter. According to the theory in $\mathbb{R}^3$, the error of third order interpolation should follow the -4 slope while for fifth order interpolation it should follow the -6 slope. Note that this appears to verify for the Bézier curve but not for the quintic cumulative interpolation curve! However, that cumulative basis form offers C^2 - continuity and gives reasonable error. As a comparison, Slerp is shown. Slerp being a first order interpolation method (using only quaternion curve values and not that of any derivative), it follows the expected -2 slope, the same as linear interpolation would in $\mathbb{R}^3$. Note that the de Casteljau algorithm also offers a better computational time.

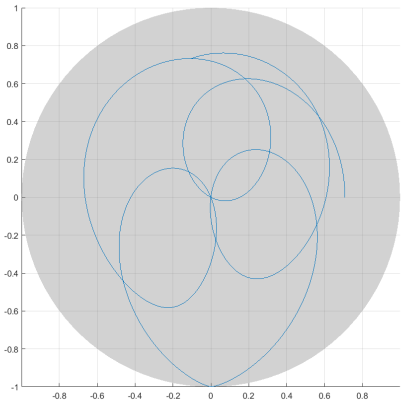


Figure 2.24: Quaternion path for this harmonic case and for $25\,\mathrm{s}$

Building the derivative of the cubic Bézier quaternion curve

One can see in Figure 2.25 that the cubic Bézier method for quaternion interpolation is more exact than the cumulative basis method. In particular, it follows the theoretical -4 slope close to perfection.

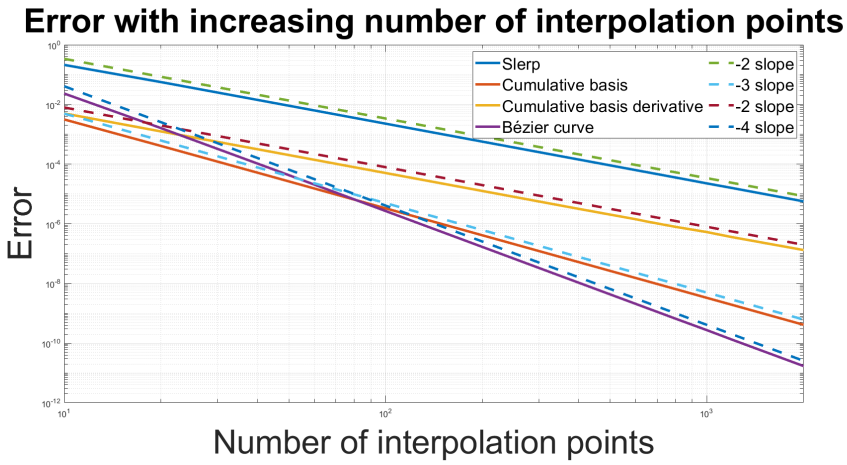


Figure 2.25: Verification of quaternion interpolation methods

Adding onto this the computational benefit of the cubic Bézier curve, it is natural to consider its use instead of the use of the cumulative basis curve. This would however mean allowing for discontinuity of the quaternion second derivative, i.e. of angular acceleration (see Equation 2.29). This contradicts the hypothesis made at the start of Section 2.4. Nevertheless, it is an acceptable compromise since only the first quaternion derivative, relating to angular velocity, is required to calculate the velocity of the lidar. Now, it remains to determine the derivative of the Bézier quaternion curve, which is possible but significantly more complex than that of the cumulative basis curve.

This derivation is made possible by the fact that any Bézier quaternion curve is in the end the ”blending” (defined below) of two unit quaternion two sub-curves. And for such a blending, a clear formula for the first derivative is defined [43]. In turn, these sub-curves are also a blending of two unit quaternion two sub-curves, etc. This is due to the fact a Bézier curve is built with nested Slerps. The blending of two unit quaternion curves (meaning that every quaternion belonging to this curve is a unit quaternion) is defined as Equation 2.47 [43], with $t \in [0;1]$. The function $f(t)$ is called a blending function and will take a trivial form in this applied case. Here, the unit quaternion curve q is the blending of the curves q_0 and q_1.

$$q(t) \ = \exp(f(t) \ \log(q_1(t)\, q_0^*(t)))\, q_0(t) \tag{2.47}$$

Provided that $f(t) \in [0;1]$, then this is exactly the expression of a Spherical Linear Interpolation between q_0 and q_1 by the factor $f(t)$ (see 2.39). This formula is developed in the case where the interpolation parameter is directly defined between 0 and 1. However in this case, the interpolation parameter is defined as a certain time t that is between two time nodes t_i and t_{i+1}. Hence, rather than defining $t \in [0;1]$, it will be defined as $t \in [t_i; t_{i+1}]$. Then, a blending function can be defined from this parameter as $f(t) = \frac{t-t_i}{t_{i+1}-t_i}$. This blending function is precisely what was defined previously as h_i.

Thus, using again the notations from 2.40 and the decomposition below it, one can conclude that the cubic Bézier curve $\hat{q_4}$ is simply the blending of the unit quaternion curves p_1 and p_2, with h_i serving as the blending function. Also, p_1 is the curve obtained by the blending of the two unit quaternion curves r_1 and r_2, and p_2 by that of r_2 and r_3. These three last curves are Slerps between two constant unit quaternions. Every sub-curve is a unit quaternion curve because they come from a Slerp. The blending function used is always the same. Now that the notion of blending quaternion curve was defined, what about its first derivative?

Given two unit quaternion curves q_0 and q_1, the first derivative of the blending curve q by the blending function f is given in Equation 2.48 [43], where $d\exp$ is a function mapping a 4-by-3 matrix to a $\mathbb{R}^3$ vector, defined in Appendix A. The formula for the derivative of the log of a unit quaternion curve is also given in Appendix A.

$$\begin{aligned}\dot{q}(t) = d\exp(f(t)\ \log(q_1(t)\,q_0^*(t)))\,.\,\Big(f'(t)\ \log(q_1(t)\,q_0^*(t)) \\ + f(t)\,\frac{d}{dt}(\log(q_1(t)\,q_0^*(t)))\Big)\,q_0(t) + \exp\Big(f(t)\ \log(q_1(t)\,q_0^*(t))\Big)\,\dot{q_0}(t)\end{aligned} \tag{2.48}$$

In this particular case, $f(t) = \frac{t-t_i}{t_{i+1}-t_i}$ and $f'(t) = \frac{1}{t_{i+1}-t_i}$. The last expression needed is that of the derivative of r_1, r_2 and r_3 and then, cascading, one can get to that of $\dot{\hat{q}_4}$.

Again, $r_1(t) = Slerp(q_{i,0}, q_{i,1},\ h_i) = Slerp(q_{i,0}, q_{i,1},\ \frac{t-t_i}{t_{i+1}-t_i})$. And it is known that, for $h \in [0;1]$ [16]:

$$\frac{d}{dh}\,Slerp(p,q,h) = Slerp(p,q,h)\ \log(p^*\,q)$$

Consequently, from the chain rule, Equation 2.49 comes for r_1 for example. Analogous formulas apply to r_2 and r_3.

$$\dot{r_1}(t) = \frac{d}{dt}\,Slerp(q_{i,0}, q_{i,1}, \frac{t-t_i}{t_{i+1}-t_i}) = Slerp(q_{i,0}, q_{i,1}, h_i)\,\frac{\log(q_{i,0}^*\,q_{i,1})}{t_{i+1}-t_i} \tag{2.49}$$

Although the analytical formula for $\dot{\hat{q}_4}$ remains out of reach, it can be obtained with the following algorithm steps at the time t:

- Compute the values of $r_1(t)$, $r_2(t)$, $r_3(t)$, $p_1(t)$ and $p_2(t)$ with the Equations developed in 2.40.
- Get the values of $\dot{r_1}(t)$, $\dot{r_2}(t)$ and $\dot{r_3}(t)$ by applying Equation 2.49 (or similar).
- Find $\dot{p_1}(t)$ by filling in the values of $r_1(t)$, $r_2(t)$, $\dot{r_1}(t)$ and $\dot{r_2}(t)$ in Equation 2.48. For $\dot{p_2}(t)$, the process is the same but with $r_2(t)$, $r_3(t)$, $\dot{r_2}(t)$ and $\dot{r_3}(t)$.
- Finally, $\dot{\hat{q}_4}(t)$ is obtained by the same process but with $p_1(t)$, $p_2(t)$, $\dot{p_1}(t)$ and $\dot{p_2}(t)$.

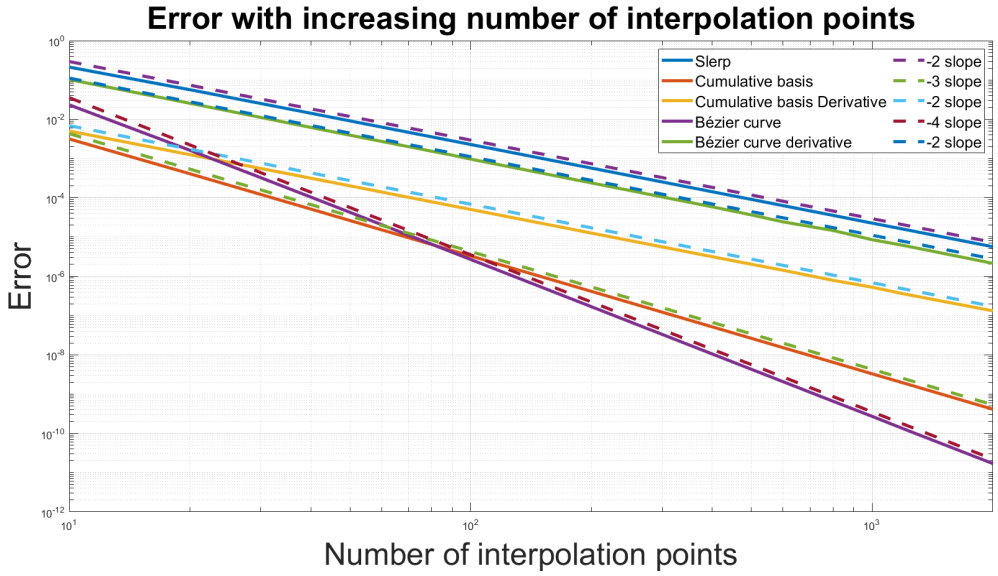


Figure 2.26: Verification of the error of the derivative of the Bézier curve

This algorithm has been tested to check its behavior in terms of error with respect to the number of interpolation points. The error plot was added to Figure 2.25. If the interpolation method was perfect, then as for in $\mathbb{R}^3$, the error between the real derivative of the curve and that of the interpolated derivative of the Cubic Bézier curve would follow a -3 slope. It is disappointing that this is not the case when looking at Figure 2.26. In fact, the slope of the error line of the Cubic Bézier derivative is of -2, the same as

Slerp, and the same as the derivative of the cumulative basis interpolation. The fact that this does not behave as expected could either be due to an error in the derivation or in the implementation.

One can notice that the error is always lower than the error of Slerp, multiplied by a certain factor < 1. This is reassuring because this means that using this method is still better than using the simplest low-order Slerp. However, it is always higher than the error of the cumulative basis derivative, multiplied by a factor > 1. Naturally, one can think of using the Bézier curve method for interpolation of the quaternions themselves, and using the cumulative basis derivative for interpolation of the quaternion derivatives. In the following, the Bézier curve will still be used because angular acceleration is not available as an output of the INS.

2.5. Wind Field simulation

2.5.1. Introduction and turbulence

The final goal of this project is to use the lidar motion-correction algorithm to be able to obtain wind velocity measurements as close to reality as possible, from which it is possible to extract wind field parameters and statistics. A wind field defines the 3D wind velocity vector $\boldsymbol{v}(\boldsymbol{x}, t)$ in a global Earth frame as a function of time and position. From only some measurements at certain points in time and space, reconstructing a wind field then allows to extrapolate the wind velocity at any desired time and position. The accuracy of the reconstructed wind field compared to reality will depend on hypotheses and modelling assumptions. They can come from mathematical equations governing fluid dynamics, and thus the wind flow, or that can be purely empirical. For example, typical characteristics of wind fields are that at higher heights, the horizontal wind speed is higher than at lower heights, due to less friction, and that at a solid boundary (at the ground), the wind velocity is 0 (no-slip condition). These two assumptions are far from sufficient to have a good modelling of the reality of the wind variation in time and space. There exists many ways to represent a wind field, with different levels of complexity. For some purposes such as energy production prediction, rather simple wind field models are enough. For example, one might only need to model the horizontal component of the wind velocity, or only its 10-minute averaged value. For other purposes such as load calculations, more complex wind fields are needed, where the lower-scale intricate fluctuations of the wind velocity are taken into account, that is a better modelling of *turbulence*. Turbulence in the context of fluid dynamics is not straightforward to define. It has for instance been defined as such [18]:

Incompressible hydrodynamic turbulence is a spatially complex distribution of vorticity which advects itself in a chaotic manner [...]. The vorticity field is random in both space and time, and exhibits a wide and continuous distribution of length and time scales.

Although this report does not aim to dive deep into fluid dynamics, it is useful to develop the previous definition, since without turbulence, it will be impossible to simulate and to reconstruct realistic wind fields. It seems first necessary to define what vorticity is. It is a quantity closely related to the wind velocity $\boldsymbol{v}$ by Equation 2.50.

$$\boldsymbol{\omega} = \nabla \times \boldsymbol{v} \tag{2.50}$$

where ∇ is the *nabla* operator. In Cartesian coordinates, Equation 2.50 can for instance be written

$$\boldsymbol{\omega} = \begin{bmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{bmatrix}$$

Turbulence is in fact a ”sea of eddies”, which are lumps of vorticity interacting with each other, and with a corresponding velocity. These eddies evolve in time and space (according the so-called *vorticity equation*), which makes the vorticity field (and also the velocity field) highly dependent on time and position. The term ”chaotic” is used instead of ”random” to highlight the fact that turbulent is extremely sensitive to initial conditions.

Finally, the meaning of "advects itself" is that, in a sense, the vorticity field transports itself, and this is due to Equation 2.50. A given vorticity distribution induces a certain wind field, and this given wind field will in turn generate a new vorticity distribution, etc. Note that the opposite is not true: a velocity field does not necessarily induce vorticity. This is the case for potential flows. While equations on vorticity derive from results on the velocity field, equations on the velocity field can be defined independently of vorticity. This then leads to analyzing the wind velocity field, the governing equation of which is obtained from fluid dynamics. This is the well known Navier-Stokes equation, that comes from three physical principles: the conservation of mass and the applications of Newton's second law of motion and law of viscosity. The Navier-Stokes equation tracks the evolution of the acceleration of a fluid element as function of the pressure force and the viscous forces acting on it. For a fluid element with a kinematic viscosity ν, a mass density ρ in a pressure field p, it is expressed in Equation 2.51 [18].

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}.\nabla)\boldsymbol{v} = -\nabla\Big(\frac{p}{\rho}\Big) + \nu\nabla^2\boldsymbol{v} \tag{2.51}$$

And the vorticity equation for incompressible flows is:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\boldsymbol{v}.\nabla)\boldsymbol{\omega} = (\boldsymbol{\omega}.\nabla)\boldsymbol{v} + \nu\nabla^2\boldsymbol{\omega}$$

The non-linear advective term $(\boldsymbol{v}.\nabla)\boldsymbol{v}$ of Equation 2.51 is the main source of the complexity of fluid mechanics, in particular turbulence, and the reason why simulating a realistic wind field is not a straightforward process. Note that the viscous stresses also have a key role in the complexity of a fluid flow. In particular without viscous stresses (i.e. for an inviscid fluid), the Navier-Stokes equation becomes the Euler equation, which is time-reversible (the equation is the same for t or $-t$). In other words, this implies that the Euler equation can lead to an increase of order (by $\boldsymbol{v}$ returning to its initial state), rather than disorder, which is contrary to what one might think of turbulence [18]. However in the real case, fluids always have a certain viscosity (even if very small), which makes the Navier-Stokes equation ultimately irreversible and turbulence so linked with a greatly disorder.

Until now, a basic introduction on the chaos and the unpredictability of the wind field $\boldsymbol{v}(\boldsymbol{x}, t)$ has been made. However, it still remains possible to simulate theoretically turbulent wind fields, by taking a statistical approach on the properties of $\boldsymbol{v}$. Indeed, although the detailed properties of $\boldsymbol{v}(\boldsymbol{x}, t)$ are highly disorganized and unpredictable, its statistical properties are perfectly reproducible. Great minds have been working on this matter for decades, and the use of "statistical properties" of $\boldsymbol{v}$ to model wind fields shall be addressed below in the report. An example of the Probability Density Function (PDF) for one of the transverse components of velocity is shown in Figure 2.27. The PDF was computed by measuring velocity for a large number of samples in an experiment of grid turbulence, which is wind blown into a grid made of orthogonal bars. If the wind is blowing in the x-direction, then the transverse component is the wind velocity in either the y or z-direction. It was shown that both transverse components will have the same PDF [18]. Remarkably, this distribution is very nearly Gaussian, with a zero mean. This shows that in a statistical sense, the transverse component of the wind velocity in a grid turbulence experiment, is more likely to be null than any other value. Having an idea of the value of this velocity component by solving the Navier-Stokes equation would be much more complicated.

The critical insight here is that instead of solving directly the Navier-Stokes equation to get a deterministic formula for $\boldsymbol{v}$ as a function of time and position, it is possible to consider $\boldsymbol{v}$ as an ensemble of three random variables (the wind velocity vector being three-dimensional) that have a probability distribution and for which it is possible to calculate averages, higher-order moments and correlation functions. The method chosen in this report to represent turbulent wind fields is based on the Mann model, introduced by Jakob Mann, an industry standard (IEC-recommended) [52].

2.5.2. The Mann model

From a physical point of view, the best way to simulate a turbulent wind field would be to solve directly the 3D Navier-Stokes equation through Direct Numerical Simulation. To make this solving less computationally demanding, the approach of Large Eddy Simulation exists, where smallest scale turbulent structures are obtained by modelling assumptions, and not by a direct solving. But still, both

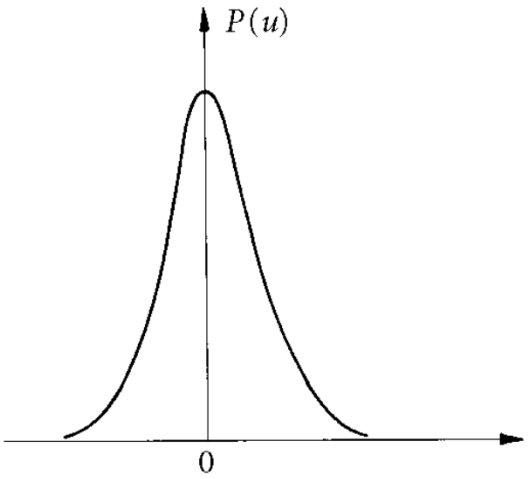


Figure 2.27: Example of one of the statistical properties of the wind field in a particular experimental set-up (grid turbulence). The PDF is shown for one of the transverse components of the wind velocity [18].

methods require an enormous computational cost [50]. The method proposed by Mann, and experimentally verified, is based on a model of the *spectral tensor* of the velocity field, which is the Fourier Transform of the covariance tensor. This covariance tensor, also called the velocity correlation function is the backbone of statistical turbulence theory. It is generally a function of the position $\boldsymbol{x}$, the separation vector $\boldsymbol{r}$ and time t, see Equation 2.52, where i and j can be x, y or z and $< . >$ is the generic ensemble average. Here, the total wind velocity $\tilde{\boldsymbol{v}}(\boldsymbol{x}, t)$ is decomposed as turbulent fluctuations $\boldsymbol{v}(\boldsymbol{x}, t) = (v_x, v_y, v_z) = (u, v, w)$ around a mean wind speed $\boldsymbol{V}$ that is fixed and well-defined for each time and position. This mean wind speed can be for example modelled by the power law, described shortly later.

$$R_{ij}(\boldsymbol{x}, \boldsymbol{r}, t) =< v_i(\boldsymbol{x}, t)\, v_j(\boldsymbol{x} + \boldsymbol{r}, t) > \tag{2.52}$$

The assumptions in the Mann model that reduce the number of variables on which the covariance tensor are the following [50]:

- *Taylor's frozen turbulence hypothesis* (or "linearized transport"). Turbulent eddies are horizontally (i.e. along the stream) transported with a mean speed U. This speed can for example be chosen as the vertical average of the horizontal mean wind speed (the term around which there are turbulent fluctuations). This assumption allows to remove the time-dependency of $\boldsymbol{v}(x, y, z, t)$, which becomes $\boldsymbol{v}(x - U\, t, y, z, 0)$. Time series can then be interpreted as "space series", and the consequence is that the time argument is usually suppressed in turbulent wind field modelling discussions.
- *Homogeneity*. Flow statistics are independent of position. Thus, whether the covariance tensor is determined between $\boldsymbol{x}$ and $\boldsymbol{x} + \boldsymbol{r}$ or between $\boldsymbol{x'}$ and $\boldsymbol{x'} + \boldsymbol{r}$, its value will be the same.

The consequence of these assumptions is that for the covariance tensor, only the dependence on the separation vector $\boldsymbol{r}$ remains, which leaves:

$$R_{ij}(\boldsymbol{r}) =< v_i(\boldsymbol{x})\, v_j(\boldsymbol{x} + \boldsymbol{r}) >$$

The spectral tensor addressed before can then be expressed in the wave vector domain as Equation 2.53. The first component (in the x-direction if x is the stream-wise direction) of the wave vector $\boldsymbol{k}$ is the wave number denoted k_1. It is linked to the corresponding frequency f and the mean wind speed U that was introduced before by $k_1 = 2\pi f / U$.

$$\boldsymbol{\phi}_{ij}(\boldsymbol{k}) = \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{ij}\Big((r_1, r_2, r_3)\Big) \exp\Big(- i\boldsymbol{k} \,.\, (r_1, r_2, r_3) \Big) dr_1 dr_2 dr_3 \tag{2.53}$$

Then, the wind velocity of a field with such a spectral tensor is connected to the stochastic vector field $\boldsymbol{Z}(\boldsymbol{k})$ by Equation 2.54.

$$\boldsymbol{v}(\boldsymbol{x}) = \int \exp(i\boldsymbol{k} \,.\, \boldsymbol{x}) \, d\boldsymbol{Z}(\boldsymbol{k}) \tag{2.54}$$

This vector field is made of three components (Z_x, Z_y and Z_z), which can be considered as complex (in the sense $\in \mathbb{C}$) random variables. It is an orthogonal process, meaning that for two different wave vectors $\boldsymbol{k} \neq \boldsymbol{k'}$, all covariances between one of the incremental components of the first field and the second field are null. Mathematically, this can be written as:

$$< dZ_i(\boldsymbol{k}) \, dZ_j^*(\boldsymbol{k'}) >= 0$$

In other words, the stochastic vector field $\boldsymbol{Z}(\boldsymbol{k})$ has uncorrelated increments across the wave vector domain. This property comes from the assumption of homogeneity of the velocity fluctuations [49]. The connection between the components of $\boldsymbol{Z}$ and the spectral tensor is given by Equation 2.55.

$$< dZ_i^*(\boldsymbol{k}) \, dZ_j(\boldsymbol{k}) >= \boldsymbol{\phi}_{ij}(\boldsymbol{k}) dk_1 dk_2 dk_3 \tag{2.55}$$

Now that all of these elaborate statistical quantities have been defined with their relationship to the wind velocity, how to model and simulate them? In his model, Mann makes uses of an educated guess on the spectral tensor that is based on a linearization of the Navier-Stokes equation and a modelling of turbulent eddy lifetimes.

To understand the concept of eddy lifetime and its modelling as a function of its respective wave number k, it is useful to present the notorious founding of Kolmogorov and Obukhov: the concept of the energy cascade in turbulent flows. As it was introduced before, a turbulent flow is well-modelled by considering it as an ensemble of a lot of turbulent eddies which have a certain vorticity, and on different length scales. The interaction between all these length scales has been studied and has given the crucial concept of energy cascade. This theoretically appealing (and aesthetically pleasing) idea, well verified by experiments, is as follows [18]:

In a flow where turbulence dominates (which is the case for the wind flow in most environments), large eddies are created by instabilities in the mean flow, and these eddies carry the largest amount of kinetic energy. These eddies are themselves subject to inertial instabilities so they ”break-up” into smaller vortices and pass their energy onto eddies with a smaller length scale, which carry less kinetic energy. This phenomenon is repeated until the viscous forces acting on the fluid are no longer negligible compared to the inertial forces, at which point turbulence is ”dissipated”. This effect is depicted in Figure 2.28.

A measure of the length scale of the eddies is $\lambda \sim 1/k$, where k is the corresponding wave number of a turbulent eddy [45]. Thus, low values of k correspond to larger eddies while large values of k correspond to smaller eddies. From dimensional analysis and physical reasoning, it was found that if $E(k)dk$ is the kinetic energy per unit mass of fluid in eddies with a wave number in the range of dk, then in the so-called inertial range (the range of k where the energy cascade described by Kolmogorov happens), the evolution of E goes as Equation 2.56, where ϵ is the mean dissipation rate of energy per unit mass of fluid. It is seen there that larger eddies carry more kinetic energy, and that this cascades down at smaller length scales, according to the -5/3 slope (in log space).

$$E(k) \propto \epsilon^{2/3} \, k^{-5/3} \tag{2.56}$$

The assumption on eddy lifetime used in the Mann model arises from very similar reasoning. The lifetime $\tau(k)$ of an eddy is the typical time an eddy of a given scale (corresponding to the wave number k) will exist before disappearing as it passes down its energy to smaller length scale. The dependence of τ on k is given in Equation 2.57. The physical meaning is that larger eddies will typically exist for longer than smaller eddies.

$$\tau(k) \propto \epsilon^{-1/3} \, k^{-2/3} \tag{2.57}$$

This modelling of the eddy lifetime is used in the Mann model as a way to provide closure to the linearization of the Navier Stokes equation, which finally gives an expression of the spectral tensor as a

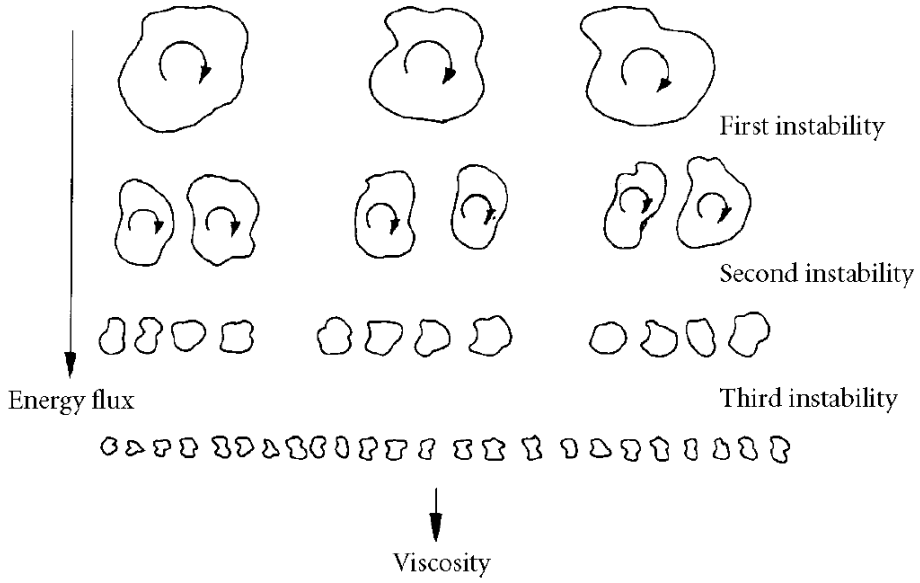


Figure 2.28: Schematic representation of the energy cascade among eddies of turbulent flows [18].

function of some parameters, such as the integral length-scale L, a parameter Γ related to anisotropy, the 3D Kolmogorov constant α and the dissipation of turbulent kinetic energy ϵ, introduced before. As seen with Equations 2.54 and 2.55, this tensor is enough to describe the statistics of the corresponding wind velocity, and thus simulating a turbulent wind field.

In practice, the wind field is simulated by numerically estimating Equation 2.54 inside of a discretized box (a *turbulence box*), with finite dimensions. Note that the generated velocity field will be periodic in all three directions. This periodicity is seen as one of the limitations of the model [50]. Another limitation is that of the size of the turbulence box. It cannot be arbitrarily much larger than the integral length scale of the model L. Anyhow, even if the Mann model uses assumptions that are not valid in all of the atmospheric regions, it remains a standard for research and industry that has shown excellent results in simulating shear-generated turbulence for high wind speeds at wind turbine locations. And if one wants to have a proper modelling of the behavior of a wind turbine, then quite evidently, one needs an accurate modelling of the wind. Accuracy should of course be balanced with computational cost.

Returning to the use of turbulent boxes, one can find an illustration in Figure 2.29.

If the rotor plane is located at a horizontal position $x = 0$ and the simulation starts at time 0, then the wind that will reach the rotor at the time t is located inside of the box in the plane that has the horizontal position $x = -U_{hub}\, t$. This is a perfect example of the powerful Taylor's frozen turbulence hypothesis which allows to transform time domain into space domain. It is also important to note that inside of the box, the turbulent wind is generated at discrete points so interpolation is required to get the wind velocity inside of the whole "vertical slices" and in between each slice, which then allows for the construction of a continuous wind field inside of the turbulence box.

2.5.3. Examples of wind fields

The previous discussion and the introduction to turbulent wind fields can finally be used practically in this subsection. Some examples will be shown of the wind velocity components for different simulated wind fields. The main focus will be to see what shear-generated turbulent wind fields are like and to have an idea of how lidar measurements of theoretical wind fields should behave.

The stream-wise direction of the wind will be assumed to be along the x-direction such that the transverse velocity components of the turbulent wind will be far smaller than the horizontal component. Again, wind velocity can be decomposed as a mean wind speed $\boldsymbol{V}$ around which there are turbulent fluctuations. Given the direction of the stream, the mean wind field will have 0 transverse components and the horizontal wind field is supposed to be purely generated by shear such that:

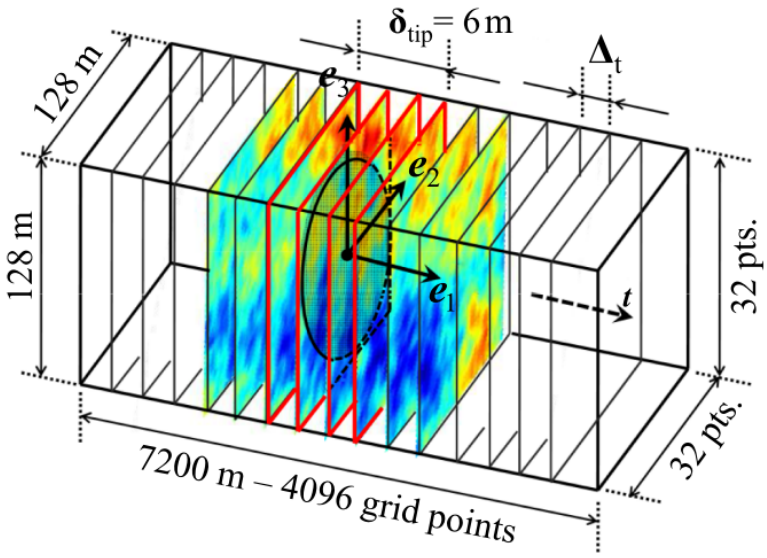


Figure 2.29: 3D illustration of the Mann turbulence grid box [67]. Box made of vertical square slices of dimensions $128\,\mathrm{m}$ x $128\,\mathrm{m}$ (discretized into 32 x 32 points). Total box made of 4096 vertical slices (over a distance of $7200\,\mathrm{m}$). The space interval between each slice is of $d = U_{hub}\,\Delta t$. The wind turbine rotor plane is represented by the black circle and the parameter δ_{tip} represents the blade tip deflection and should be ignored here.

$$\boldsymbol{V} = (U(z), 0, 0)$$

where z is the vertical coordinate. Thus, the transverse components of the total wind velocity are solely generated by turbulence. A global Earth frame with the z-axis pointing upwards and the x-direction along the stream-wise direction should be considered. What has been called *shear* represents the increase of the mean horizontal wind speed as a function of the z-coordinate, in this case $\frac{dU(z)}{dz}$.

The power law introduced in Equation 2.21 is with the Log Law a standard law used in wind energy research and industry to represent the horizontal component of the steady wind as a function of height. Examples of such profiles is shown in Figure 2.30. As expected, a larger shear exponent α induces larger wind speed increment for the same height increment.

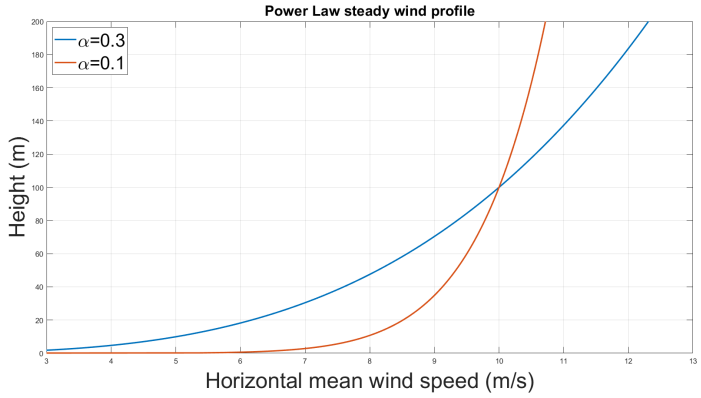


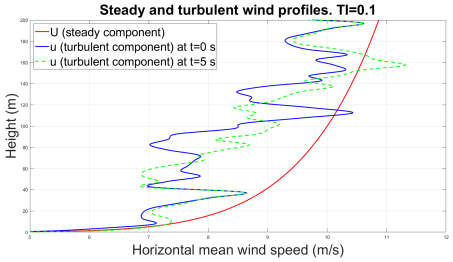
Figure 2.30: Comparison of two profiles of steady horizontal wind speed using power law. The wind speed at the hub height (here $10\,\mathrm{ms}^{-1}$ at $100\,\mathrm{m}$) is the same in both cases, only shear exponent changes.

Now, what becomes more interesting is to add turbulence in the simulations. Turbulence was generated using Mann's spectral tensor. The box generated has the following properties:

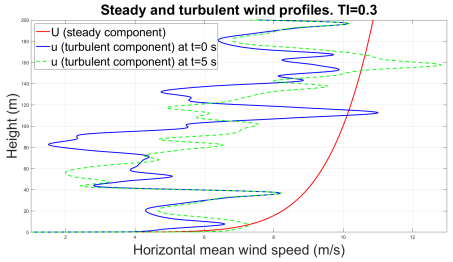
- Dimensions of around $155\,\mathrm{m}$ x $155\,\mathrm{m}$ x $970\,\mathrm{m}$, with 32 x 32 discrete points in vertical slices and 8192 vertical slices in the box.
- Interpolation is performed using Catmull-Rom splines, generating a periodic wind field for the total box (the plane at the start of the box has the same velocities than the one at then end of the box).
- The velocity of the planes' horizontal displacement (used in Taylor's hypothesis) is chosen to be the hub height velocity that is used as a parameter of the power law for the mean profile (again $10\,\mathrm{ms}^{-1}$ at $100\,\mathrm{m}$).
- The parameters of the spectral tensor were set to some standard values used by Siemens-Gamesa. Only one parameter was adjusted: the turbulence intensity TI defined in Equation 2.58, where σ_u is the standard deviation of the turbulent horizontal component of the wind velocity and V is the mean wind velocity. One of the parameters of the Mann spectral tensor, the integral length scale L can be cast as a function of TI [40]. The more turbulence, the more one can expect the total wind velocity to fluctuate, which is why σ_u is the numerator of Equation 2.58.

$$TI = \frac{\sigma_u}{V} \tag{2.58}$$

The turbulent wind profiles for the horizontal wind component at two different times are shown against the mean power law profile in Figure 2.31. The steady profile is chosen to be the same as the red curve of Figure 2.30 ($\alpha = 0.1$ and $U_{hub} = 10\,\mathrm{ms}^{-1}$ at $100\,\mathrm{m}$). Figure 2.31a represents a turbulent wind field with $TI = 0.1$ while Figure 2.31b represents a turbulent wind field with $TI = 0.3$.



(a) Turbulent profiles and steady component of the horizontal wind speed at two different times.



(b) Turbulent profiles and steady component of the horizontal wind speed at two different times. High Turbulence intensity.

Figure 2.31: Turbulent profiles and steady component of the horizontal wind speed for two different fields that differ by TI value.

The profile of the steady component looks different because for $TI = 0.3$, the size of the plot had to be changed to fit the fluctuations of the turbulent wind profiles, but it is the same. It is also interesting that for two wind fields that differ only by their Turbulence Intensity, the turbulent fluctuations are basically the same but scaled to the TI. Now that turbulence has been introduced the wind velocity components in the y and z-directions (respectively v and w) are no longer 0. Their profiles are shown for $TI = 0.1$ and at $t = 0$ in Figure 2.32. Although both components have a magnitude generally lower than the horizontal wind speed, one can see that they are not negligible, especially at lower heights.

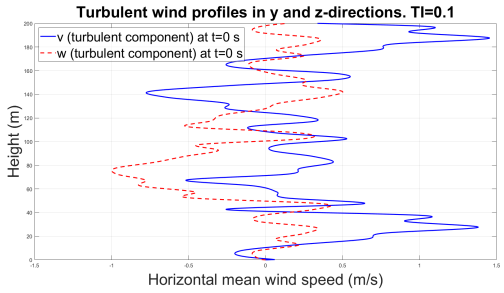


Figure 2.32: Turbulent profiles of the transverse and vertical wind speed for $TI = 0.1$.

2.5.4. Behaviour of the spectrum of the wind measurements from a moving lidar in a turbulent atmosphere

In this subsection, a focus is made on the spectrum of the wind velocity components in a turbulent field and on the theoretical spectrum of the VLOS measurements from a lidar. For this matter, the Power Spectral Density (PSD) of the wind velocity components is calculated by simulating a field similar to that of the previous subsection with $TI = 0.1$. The only difference is that the box is made to be bigger (more simulation time) and the integral length scale is slightly larger. Results in the PSD will mostly be discussed in terms of frequency.

On this matter, if wave numbers were linked with the size of the turbulent eddies, what does frequency in the spectra exactly mean physically? It was introduced before that $k_1 = 2\pi f/U$, with U the mean wind speed, used in Taylor's frozen turbulence hypothesis. Given this relationship, one way to think about this problem is to simply associate high frequencies with large wave numbers, so with small eddies. And the other way around, small frequencies with small wave numbers, that is large eddies. That can also be understood with the fact that small eddies have shorter lifetime so the corresponding phenomena happen on higher frequencies. Thus, high frequencies correspond to smaller eddies, which carry less kinetic energy. Since the relationship between k and f is linear, then the exponent dependence of spectrum on f is the same as on k.

As the Mann model theory predicts, the slope of the PSD for each of the wind velocity components should be -5/3 in log space. The wind field was simulated for $600\,\mathrm{s}$ (10 minutes) at $1\,\mathrm{Hz}$ for a fixed point in space, with a height of around $120\,\mathrm{m}$ in the same global Earth coordinate system as the wind field. The velocities can be normalized. This should just change the magnitude of the PSD by a factor. Indeed, if a signal $x(t)$ has a power spectrum $S_{xx}(f)$, the power spectrum of $a\,x(t)$, where a is a real c onstant is equal to $a^2\,S_{xx}(f)$ [61]. For the u-component of the wind speed, the result is shown in Figure 2.33. Because of the sampling theorem, the PSD is only calculated up until half the sampling frequency, which is $0.5\,\mathrm{Hz}$. In the scope of this project, although it would certainly be better because the inertial subrange generally includes higher frequencies than $0.5\,\mathrm{Hz}$, it is good to show an example of a PSD spectrum with a quite low sampling frequency. In fact, pulsed lidars do not allow for high frequency measurements.

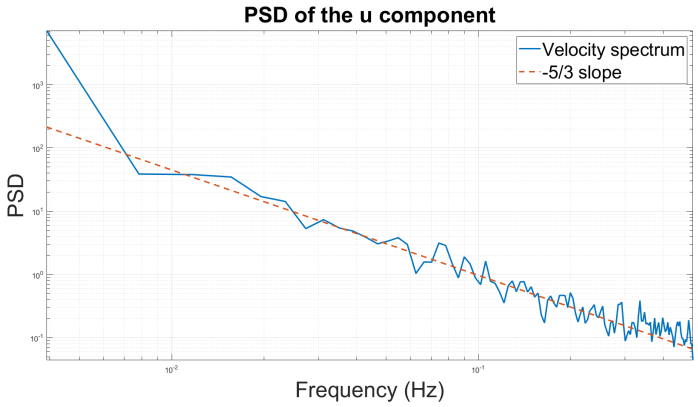


Figure 2.33: Power spectral density of the (non-normalized) horizontal component of the velocity at a given point with a sampling frequency of $1\,\mathrm{Hz}$ for 10 minutes.

The spectrum is not perfectly smooth because the number of samples is relatively low and the numerical methods used to estimate the PSD are discrete so they depend on this number. However, the trend of the spectrum should remain consistent, that is the frequency distribution of the amplitudes. It is possible to see there that the distribution of the spectrum follows the -5/3 slope, and only for a certain domain. For example, for higher frequencies (> $\sim 0.3\,\mathrm{Hz}$), it can be seen the PSD is not so clearly following the slope anymore. From a physical point view, the slope trend of the spectrum is no longer valid after the upper limit of the energy cascade, where viscosity starts to rule over turbulence among smaller eddies. However here, this is most likely due to the dimensions of the box and the grid spacing, causing interpolation artifacts at higher frequency, so a numerical problem.

For visualization purposes, a moving-average of the PSD can also be shown to ensure more smooth-

ness. Some methods like Welch's estimate already use averaging over overlapping segments of the PSD. The result is shown in Figure 2.34. Interestingly, it shows the same trends in slope of the PSD, detaching from the -5/3 slope for high frequencies. For lower frequencies, the moving average is far from the dotted curves because there are less PSD values to calculate the average.

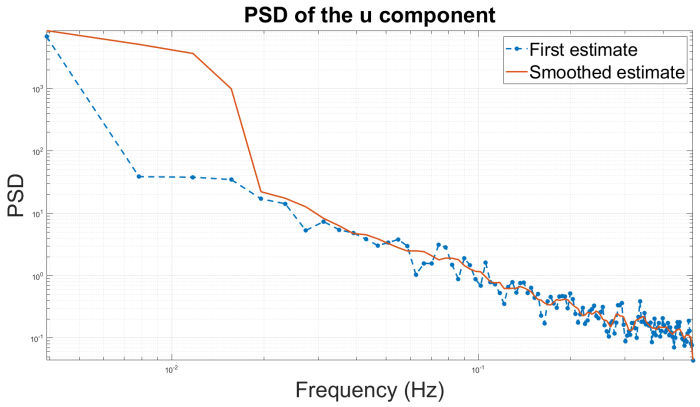
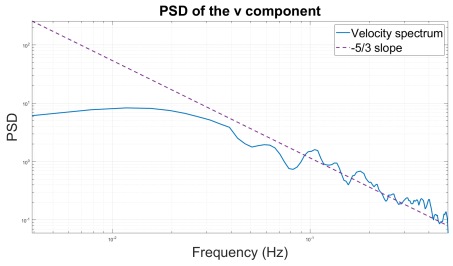
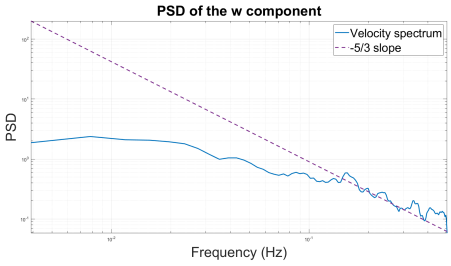


Figure 2.34: PSD estimate of Figure 2.33 against its moving average.

Now what happens for the v and w components of the wind velocity? In Figure 2.35, similar spectra are shown for the same conditions for v in Figure 2.35a and for w in Figure 2.35b . From the Mann model, it is expected that, compared to the horizontal wind speed u, the inertial subrange at which the -5/3 slope can be observed starts at higher frequencies (or wave numbers) for v, and even higher for w [50]. This is because the energy cascade is initiated by the larger eddies in the mean flow, which is essentially is the same direction as u. And the cascade will be passed onto the directions of v and w starting only with smaller eddies. This means that if u is the stream-wise component, larger structures can appear in the inertial subrange of its spectrum, which will not appear in the spectra of v and w. The slope in their respective inertial range of frequencies should however still be the same theoretical -5/3. It is a valuable sanity check to see that this slope is verified. Again, for frequencies higher than $\sim 0.3\,\mathrm{Hz}$, the spectrum drifts away from this slope.



(a) PSD of the transverse component of the velocity.

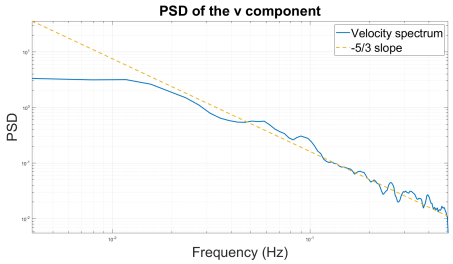


(b) PSD of the vertical component of the velocity.

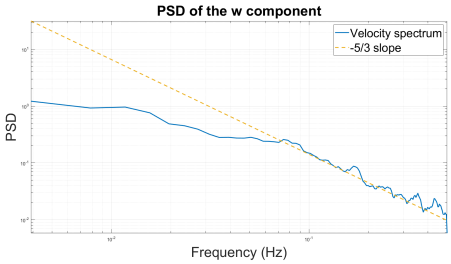
Figure 2.35: Power spectral densities of the (non-normalized) v and w components of the velocity at a given point with a sampling frequency of $1\,\mathrm{Hz}$ for 10 minutes. Shown against the theoretical -5/3 slope.

Nevertheless, given the low sampling frequency, the inertial subrange is much less obvious in the spectra than in Figure 2.33. One way of extending this range would be to change the properties of the wind field. Indeed, the starting frequency of the inertial subrange is roughly determined by the mean wind speed U of the flow divided by the integral length scale L [45]. Thus, this was done in Figure 2.36 for a mean wind speed of $5\,\mathrm{ms}^{-1}$. Figure 2.36a shows the resulting spectrum of v and Figure 2.36b that of w. One important aspect to note is also that because the wind flow is supposed to be mainly in the horizontal direction, one should expect more energy to be carried by the eddies appearing on the u spectrum.

The cross-spectra of the velocity components also matter if one wants to study the spectrum of the VLOS measurements. These cross-spectra depend on the correlation of the wind components between one another, while the spectra shown so far were related to the correlation of the wind components



(a) PSD of the transverse component of the velocity.



(b) PSD of the vertical component of the velocity.

Figure 2.36: Same as Figure 2.35 but with a lower horizontal mean speed of the wind flow.

with themselves (auto-correlation). These cross-spectra should also display an inertial subrange with a slope close to -5/3.
It is now possible to study the spectrum of the VLOS measurements. In reality, to increase the sampling frequency, the measurements of the two upper beams can be simultaneously used to construct one VLOS spectrum, the same for the two lower beams [65].

To simplify first, the point-wise approximation will be used and a lidar without motion will be considered. Up to a sign (whether the lidar coordinate system faces the wind or not), the line-of-sight velocity for one beam with azimuth ϕ and zenith θ (see Figure 4.2) is then simply:

$$\overline{v_r} = \boldsymbol{n} \,.\, \boldsymbol{v}(R) = \cos(\theta)u(R) + \sin(\theta)\cos(\phi)v(R) + \sin(\theta)\sin(\phi)w(R)$$

where $\boldsymbol{v}(R)$ is the wind velocity at the point corresponding to the range-gate center R of the beam. In time, the time series of the VLOS can be seen as the realization of a stochastic process, which is then simply a linear combination of the stochastic processes of u, v and w. The statistics of these 3 quantities are defined by the properties of turbulence that were addressed before. Because the zenith and azimuth are here considered constant, the power spectrum of $\overline{v_r}$ is rather straightforward to calculate. Since the wind components show correlation, the power spectrum of the VLOS is not simply a linear combination of the individual power spectra, but has to take into account the cross-spectra [61]. Writing for instance $S_{uu}(f)$ the power spectrum of the u component of the wind velocity and $S_{uv}(f)$ the cross-power spectrum of u and v, Equation 2.59 comes in the point-wise approximation:

$$\begin{aligned} S_{\overline{v_r}}(f) = \cos(\theta)^2 S_{uu}(f) + \sin(\theta)^2\cos(\phi)^2 S_{vv}(f) + \sin(\theta)^2\sin(\phi)^2 S_{ww}(f) + \\ 2(|\cos(\theta)\sin(\theta)\cos(\phi)|S_{uv}(f) + |\cos(\theta)\sin(\theta)\sin(\phi)|S_{uw}(f) + |\sin(\theta)^2\cos(\phi)\sin(\phi)|S_{vw}(f)) \end{aligned} \tag{2.59}$$

Taking the parameters of the WindCube PPT pulsed lidar for example, it gives:

$$S_{\overline{v_r}}(f) \approx 0.93S_{uu}(f) + 0.067S_{vv}(f) + 0.007S_{ww} + 0.5S_{uv}(f) + 0.16S_{uw}(f) + 0.043S_{vw}(f)$$

It is clear from this that the spectra involving u have the biggest weight. This is understood because this lidar is made to face the wind with low angles deviating from the horizon. Now this should be only considered in a qualitative manner because in reality the spatial averaging matters, the wind field might not exactly fit the modelling assumptions, the lidar will not be perfectly facing the wind and during the lidar motion, the direction vector of its lines-of-sight will change.

It was shown that the spatial averaging effect only affects the slope of the inertial subrange for the spectrum of VLOS measurements at high frequencies [65]. If the lidar is moving, the actual azimuth and zenith angles in a global Earth frame are not constant anymore, but can be seen as stochastic processes that cannot be considered as uncorrelated to the wind velocity (the wind makes the turbine move and thus the lidar). They can also show a peak in their spectrum around the floating platform's eigenfrequencies, which will carry through the VLOS spectrum [34]. Also for a moving lidar, the lidar velocity will interfere with the VLOS measurements, which will also impact the spectrum. In conclusion, the VLOS power spectrum for a moving lidar in a turbulent atmosphere will differ from the theoretical formula of Equation 2.59, and this will be studied. An objective of this project will be to study the impact of taking out the lidar velocity interference of the VLOS measurements and to see if computing the power spectrum of the corrected VLOS can be used as a verification of the motion-correction method.

3

Real data acquisition and algorithm derivation

This chapter will present the physical problem and the equations that will be used in the motion-correction algorithm. The main goal is to define how the lidar motion can be corrected to get the wind velocity with respect to a fixed referential. For this, it is crucial to understand how the motion-related quantities are obtained from the Inertial Navigation System (INS) placed inside of the nacelle, and how they relate, with the necessary transformations, to the lidar velocity. An analysis of some of the signal characteristics given by the INS and by the lidar will be performed in the first subsection, before introducing all relevant referentials. It will then be possible to have the final formula that corrects for the lidar motion, using known parameters.

3.1. Data acquisition and analysis

This section will present and treat the motion outputs from the available measurement devices. Two devices give real-time data on motion-related quantities. First, the lidar has an integrated inclinometer that outputs *roll* and *pitch* angle at the same time as the real-time wind measurements. Then also, the INS outputs orientation (roll/pitch/yaw or quaternion), angular velocity/acceleration as well as position and linear velocity/acceleration. All time and frequency axes are normalised and shown in arbitrary units in the following, for reasons of confidentiality. Now, even if the INS is more complete, the problem with using INS data is that it is misaligned in time and space with the lidar. In fact, the INS motion outputs are typically given at different time stamps than the lidar wind outputs and both are not at the same location in the turbine. The first time aspect has already been introduced in section 2.4 and the position offset is treated in Section 3.1.3. Note that a possible initial orientation offset between the lidar and the INS needs to be explored. Regardless, the INS is a reliable device that uses high-accuracy filters and integration methods for its outputs [23]. One last relevant aspect is that, regarding the lidar roll and pitch measurements, they are directly and internally used to correct for the changing orientation of the lidar [86]. The method used will be introduced in this section too. The performance of that ”low-order” correction method will be compared to that developed in this report.

3.1.1. Lidar outputs

Lidar orientation output and inclinometer principle

As briefly explained, the lidar contains an inclinometer that records pitch and roll angle values at the time of the measurement (so every second). These measurements are based on a different principle than the orientation measurements from the INS, which will be explained later. Fundamentally, inclinometers measure the attitude of orientation with respect to a reference axis, which is defined by gravity (vertical axis in earth reference) [60]. Two different types of inclinometers are shown in Figure 3.1. Figure 3.1a illustrates an accelerometer that serves as an inclinometer. Indeed, consider the frame that the pendulum is suspended to. Due to gravity, the pendulum is always in the direction of

the gravitational force $\boldsymbol{g}$. Suppose that this is the only force applied to the pendulum (well in this case also the opposite force that keeps the pendulum attached to the frame). Then, the acceleration $\boldsymbol{a}(t)$ of the pendulum in the local reference of the frame is directly proportional to $\boldsymbol{g}$ projected onto the local axes of the frame, that projection being a function of the pitch angle θ. Given that the mass of the pendulum is equipped with an accelerometer that gives acceleration along all axes and that the value of $\boldsymbol{g}$ is precisely known, then one can retrieve the pitch angle.

(a) Acceleration inclinometer, represented by a suspended pendulum. θ : angle between the frame and gravity direction

(b) Electrolytic tilt sensor, following the displacement of the bubble inside of an electrolyte-filled tube

Figure 3.1: 2 types of inclinometers [60]

Another example is that given in Figure 3.1b , the electrolytic tilt sensor, or more clearly called the liquid bubble inclinometer. The functioning of this inclinometer is similar to that of the level vial used by carpenters, only using a more sophisticate electrical resistance measurement. It allows for better angle resolution and is based on the following principle. 2 electrodes are placed at the extremities of the cavity (or vial, usually a glass tube) which is filled with an electrolyte and another is placed at the center of the cavity. When the air bubble inside moves due to the inclination of the cavity (in Figure 3.1b only pitch would be possible), the resistance between the center electrode and each of the extremities' electrode varies [27]. The device is able to calculate that resistance and to relate it to the position of the bubble, which in turn relates to the inclination of the tube. In these two examples, only a 2D case for pitch measurement is shown but that generalizes to the roll angle as well. However, inclinometers as such do not function well for yaw angle calculation. This is because gravity is in the same direction as the Z-axis which defines the rotation axis of yaw angular displacement. For instance, imagine the case of the liquid bubble inclinometer, initially aligned with the earth global reference frame (so as in Figure 3.1b If the device undergoes yaw displacement, meaning that is rotated about the axis orthogonal to its upper surface (or global Z-axis), then the bubble would simply not move. However, from a slight displacement in roll or pitch, one can visualize that the bubble would be moving.

One last problem with inclinometers, no matter the type of technology, is that any horizontal acceleration of the frame interferes with the measurement of the angle, introducing an additional erroneous angle. That can be seen in Figure 3.2, the total acceleration of the inclinometer is no longer solely determined by the effect of gravity but also by another acceleration force vector, that could be for example the acceleration of a vehicle in which the device is placed, the nacelle for instance.

Thus, the use of inclinometers is not optimal. In the case of the INS, the difference is that orientation is obtained through error-minimizing integration of angular rates and filtering using aiding sensor such as a Global Navigation Satellite System (GNSS). The process is made so that the error caused by integration drift and relative motion of the measurement unit is minimized, so the signal output is less noisy. Starting with the data from the lidar's inclinometer, one can see in Figure 3.3 a normalized time series example for a part of a day in mid-December 2023.

The mean pitch is shown in red and one can see the oscillation of the pitch angle around that value. A similar time series is shown for another day in Figure 3.4 (around the same hours of the day and for

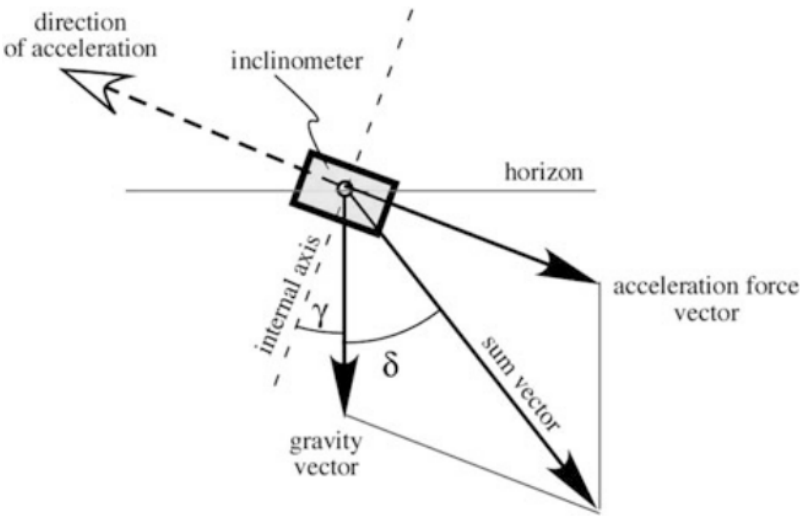


Figure 3.2: Angle error δ in an inclinometer caused by an additional acceleration force vector [27]

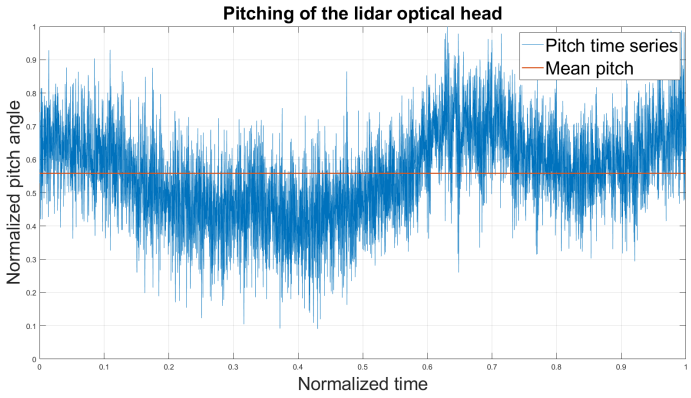


Figure 3.3: Variation of the lidar pitch output during a part of the day in mid-December

the same time-span) and the same trend is found.

Regarding this variation aspect, one can quantify it and link it to the Signal-to-Noise Ration (SNR). One definition of SNR in terms of "closeness to target" is as follows in Equation 3.1 [9], with μ the mean of the signal and σ its standard deviation. Another usual definition of the SNR uses decibels (dB), which is simply 10 times the common logarithm of the quantity defined in Equation 3.1.

$$\text{SNR} = \frac{\mu^2}{\sigma^2} \tag{3.1}$$

The higher that value is, the better the signal quality is supposed to be compared to its noise. For example, for the pitch signal given, in Figure 3.3, the calculated SNR is of around 16.5, or $12.2\,\text{dB}$. And, for the other day displayed in Figure 3.4, the SNR from the lidar pitch signal is of around 25.7 or $14.1\,\text{dB}$. So, one can see that the quality of the signal changes across the days. This can be due to different weather conditions, or to a different configuration of the lidar. For example, if the lidar's optical head is not properly fixed to its tripod, then it will move in the turbine-fixed frame and cause parasitic angle errors in the measurements. Having a technician tightening the optical head bolts will then improve the quality of the signal. This inclinometer has the advantage of being at the same location as the lidar and giving outputs at the same time. Compared to the INS however, it is equipped with less advanced filters and does not offer the full 3D orientation of the device (no yaw angle).

Signal analysis and frequency filtering option

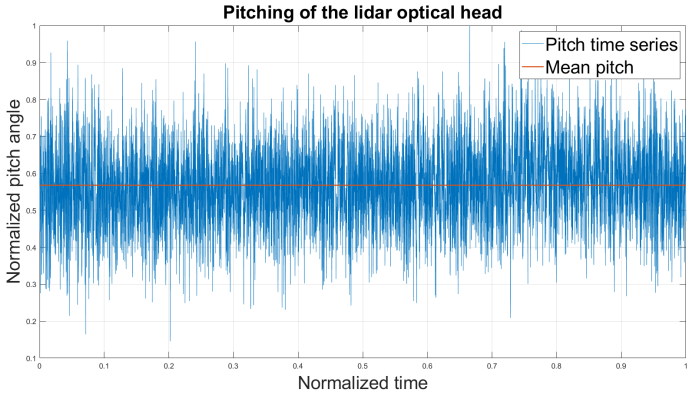


Figure 3.4: Variation of the lidar pitch output during a part of the day in end-January

Next, further study of the signal output from the lidar will be conducted. As it was previously described, the pitching motion of the platform of a FOWT is often characterized by a harmonic oscillator with a given amplitude and frequency, with possibly damping [34],[12],[48]. Visually, particularly the signal from Figure 3.3 seems to have a periodic shape around its mean value but one would need more convincing arguments. Two methods for highlighting the periodic nature of a signal will be used here : the Auto-Correlation Function (ACF) and the Power Spectral Density (PSD) study. The former is a study in the time domain while the latter is a study in the frequency domain. PSD can typically be obtained from the Fourier Transform of the signal, using Fast Fourier Transform (FFT) for example, a powerful engineering tool.

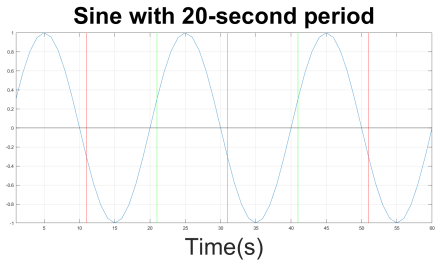
An interesting aspect to note in this project is that the frequency analysis of the lidar motion (in particular pitching motion) has been used to filter out the motion-induced errors in the lidar measurements [34]. The approach is the following:

- Compute the natural frequency of the pitching motion using spectrum analysis tools, which with a simple harmonic oscillator model would correspond to the frequency of that oscillator.
- Use a frequency filter, such as a band-stop filtering function, on the lidar measurements to eliminate all the measurements that match the natural frequency of the FOWT pitching motion.

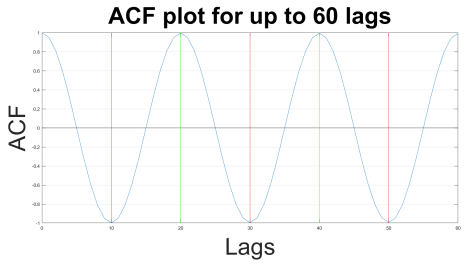
This method was proven to have better results than subtracting directly the lidar velocity to the VLOS measurements, in particular because it is less influenced by the noise of the inertial devices. However, that study was conducted in a case that appears simplified, the platform motion is assumed to have only one significant degree of freedom : pitching, neglecting the other 5. Also, it will be shown next that the apparition of the natural motion frequencies in the lidar measurements is not as obvious as in [34]. Finally, this seems to not be systematic given the different trends of the pitching motion through different days (see Figure 3.3 and 3.4). The advantage is that here, accurate data is available from the INS, being less noisy than a typical inclinometer. The frequency filtering approach will be compared to a 1-by-1 VLOS correction.

To see if there is a repeating pattern in the time-domain lying behind a signal, one tool is the Auto-Correlation Function (ACF), assuming that signal is a stochastic process with a mean and a standard deviation. The ACF shows the correlation between a time series and a delayed version of itself [10]. This can be visualized in a plot where the x-axis represents the lags, i.e. the number of delay indices. For example, if a time series is uniformly distributed in time and each time stamp is separated by $0.1\,\mathrm{s}$, then 3 lags represent a delay of $0.3\,\mathrm{s}$. And the y-axis is the value of the auto-correlation function. The higher that absolute value, the more correlation is found, 1 being the highest positive correlation possible, -1 the highest negative correlation possible and 0 representing uncorrelated time series. Usually, the ACF value for the first lags is to handle with care because there are high chances that a time series is correlated with a very shortly-delayed version of itself.
One example will be given for a purely sinusoidal function with the same time stamps as the time series given in Figures 3.3 and 3.4, bearing in mind that the lidar gives a pitch/roll output every second. The sine time series is made to have a period of 20 seconds, so one should expect positive high values

of the ACF for 20-lag delay (as well as 40, 60 etc.) and negative high values of the ACF for 10-lag delay (as well as 30, 50, 70 etc.). This is shown as a mean of comparison with the real pitch time series' ACF. For the first 60 seconds, the sine time series is shown in Figure 3.5a. The half-periods are highlighted by red lines and the complete periods by green lines. They are not exactly aligned with 10, 20 etc. because the time series does not start exactly at $0\,\mathrm{s}$ (this comes from the lidar measurements time stamps).



(a) Sine function for the corresponding first $60\,\mathrm{s}$ of the lidar pitch time series. Red line : half period. Green line : period



(b) ACF plot for 60 lags. Here, 1 lag = $1\,\mathrm{s}$. Red line : half period. Green line : period

Figure 3.5: Example of the ACF for a sinusoidal function with a period of $20\,\mathrm{s}$

Then, the ACF plot is shown, next to it in Figure 3.5b. Notice how the peaks are indeed situated at the expected locations. At 0 lag, the ACF value is of course 1 because the sine delayed by $0\,\mathrm{s}$ is exactly the same sine. Then, for the half period, the ACF value is close to -1 (visually hardly discernible from -1), because a sine delayed by a half period gives the opposite. And for complete periods similarly, the ACF value is close to 1, because of the periodicity of the sine. Note that the ACF goes to 0 for quarter-periods, for which the delayed sine actually corresponds to the original time series, with sine being replaced by cosine. Then, the same study is made on the end-January time series (see Figure 3.4) for the lidar pitch output. If the pitch is indeed close to a harmonic oscillator, then one should expect a similar ACF plot to that of a sine. Note that in literature, a period of $20\,\mathrm{s}$ was for example used for the pitching motion [34]. The result is shown in Figure 3.6. And the lags are normalized to a certain value such that one lag does not correspond to one second. This normalization also affects the frequency domain.

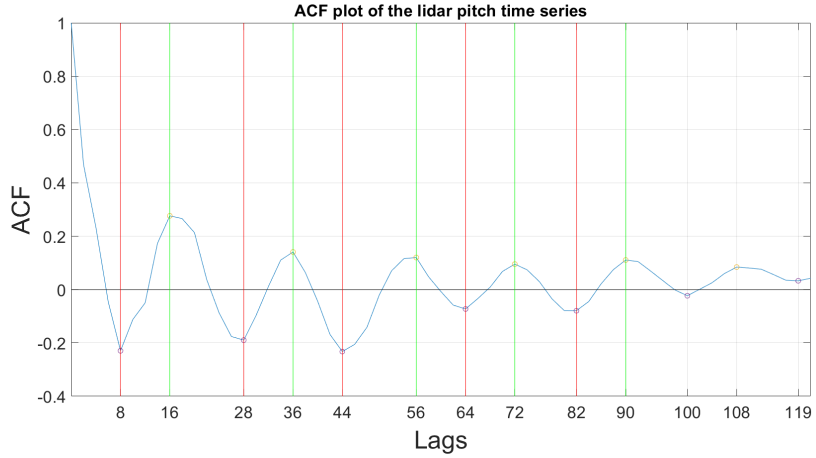


Figure 3.6: ACF plot for the lidar pitch during a part of the day in end-January, corresponding to Figure 3.4, with normalized lags.

There seems to be a similar trend as a sinusoidal function, with periodicity not being as perfect as the ideal harmonic case. What would correspond to the half-periods is shown again in red and what would correspond to the complete periods is shown in green. The peaks where the value of the ACF remains significant are the small dots. The delay between two positive dots and two negative dots is between 16 and 20 lags, which is the period one could look for in that signal. This corresponds to a frequency between 0.05 lags^{-1} and 0.0625 lags^{-1}. This is where the second analysis will be done, now

in the frequency domain. The Power Spectral Density (PSD) of the signal is computed with the help of Matlab, and the frequencies that can be studied without ambiguities are limited by the Nyquist-Shannon sampling theorem. However, this is sufficient because the frequency of interest remains in that range. In the PSD spectrum of the normalized pitch angle from the lidar (see Figure 3.7), one sees what can be expected of a noisy time series, with however a narrow peak amplitude shown by the yellow line, located at a frequency of around 0.0545 lags^{-1}. That would correspond to a period of around 18.3 lags, so in the order of magnitude of what was highlighted previously with the ACF plot. From this spectrum, there can also be harmonics at longer and shorter time-scales.

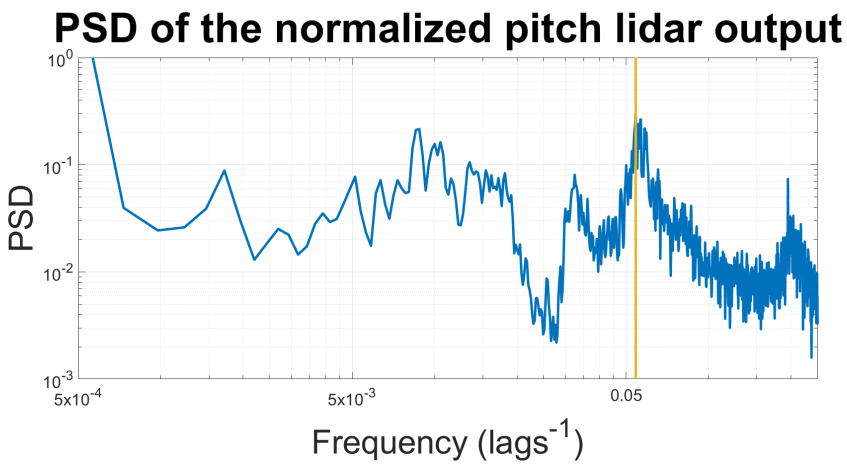


Figure 3.7: PSD spectrum of the pitch angle from the lidar output, for a part of the day in end-January, normalized frequency axis.

This study indeed confirms the harmonic-like property of the pitch angle during the turbine motion. Now can that frequency be found in the VLOS measurement from the lidar? Consider below in Figure 3.8 the PSD spectrum of the floater pitch angle against the VLOS measurements of the lidar, for a given measurement campaign [34]. Values were here not normalized before calculating the PSD. There is also the spectrum of the reconstructed horizontal wind speed, from the line-of-sight projected measurements. What is interesting is the match between the amplitude peak in the spectrum of the pitch angle and that of the VLOS measurements. This suggests that the influence of the floating motion is found for frequencies around this peak, so it could be filtered out.

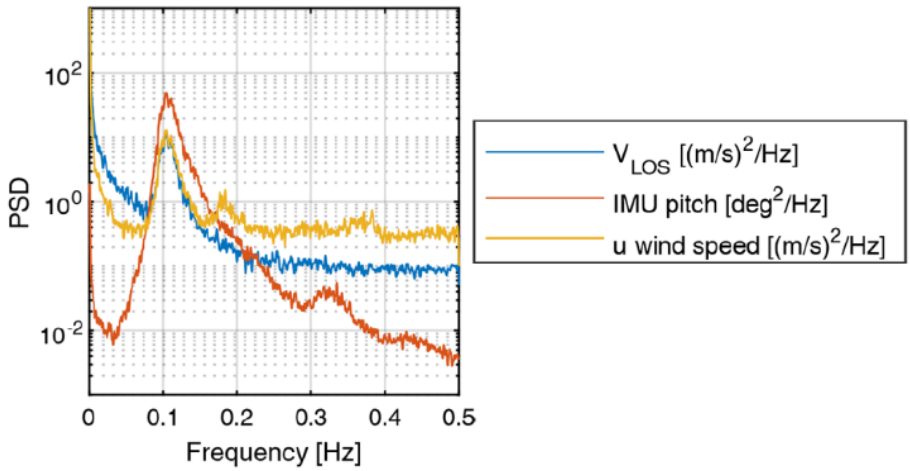
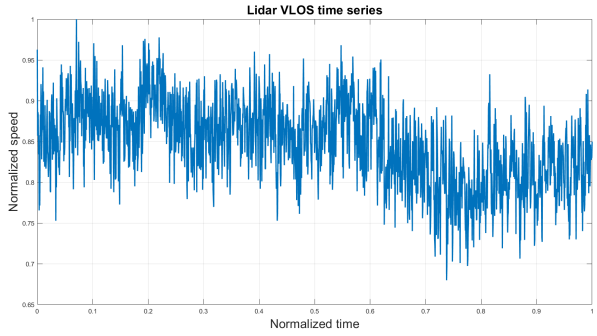


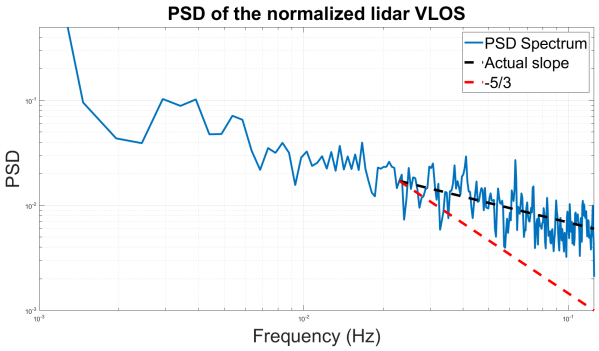
Figure 3.8: Matching natural frequencies of the floater pitch angle and the VLOS measurement time series [34].

And how about the VLOS measurements corresponding to this end-January case? The VLOS measurements are extracted from one of the upper beams at a range of $300\,\mathrm{m}$, one can find the VLOS

measurements for the same time span in Figure 3.9a (normalized wind speed and normalized time axis).



(a) Normalized VLOS measurements.



(b) PSD spectrum of the normalized lidar VLOS output.

Figure 3.9: Analysis of the VLOS measurements for the same part of the end-January day.

The lidar performs alternating 1-second VLOS measurement for each of the four beams, so for one given beam, the actual sampling rate is of $0.25\,\mathrm{Hz}$. So, even if the time span is the same, there are four times less outputs than the pitch output for example. This places the Nyquist frequency at $0.125\,\mathrm{Hz}$, still higher than the sought pitching frequency. Then, the corresponding PSD spectrum is shown in Figure 3.9b. There, no specific frequency stands out the same as it does for the pitch PSD spectrum, especially not as it did on Figure 3.8. One separate relevant comment that could be made is that the PSD spectrum of the VLOS follows the general trend of the wind spectrum having a negative slope in this frequency range. However that slope is remarkably lower than what theory predicts for the wind velocity components [51]. This is shown with the two dotted lines in Figure 3.9b, a short discussion was made on the spectrum of the VLOS measurements in the previous chapter. A lower slope suggests a low Signal-to-Noise Ratio of the measurements or interference of the lidar's motion. This shall be discussed further in the next chapter.

Adding onto the spectra mismatch is the problem that the pitch harmonic properties are not consistent across different days. Even if for a specific day, the spectra of the pitch angle for the same time span lengths show close characteristics, this does not remain true when looking on a longer time scale. In fact, the frequencies showing peak amplitude for PSD differ between some days, or are in some cases too ambiguous. For example, consider Figure 3.10. The time span for calculating the PSD is the same as for Figure 3.7, but the days are separated by around two weeks. Here, the peak amplitude corresponding to the ~ 20 lag-period is located at around 0.041 lags^{-1}, so a period of around 24 lags, and the density is in comparison lower. Nevertheless, there are clear similarities in the trend of the peak locations. These similarities were found also for other days. There is always a high amplitude around 0.05 lags^{-1}, which was highlighted in both figures by a yellow line. Then, amplitude peaks are also found for lower frequencies and for higher frequencies, corresponding to phenomena happening on a longer time scale (~ $50\,\mathrm{s}$) or on a lower time scale (~ $2\,\mathrm{s}$). One can also find a "spectral gap" between the low-frequency harmonics and the one defined by the yellow line. In literature, it is found that the platform pitch of a FOWT typically has a low natural frequency ($0.038\,\mathrm{Hz}$ [48]), that is the

frequency inducing vibrations of the platform. When the FOWT is subject to wind (steady or turbulent) and waves(regular or irregular), another frequency due to its structural response appears for a higher frequency around $0.1\,\mathrm{Hz}$. This does not explain however the peak at the even higher frequency (> $0.4\,\mathrm{Hz}$). This peak could be explained by the bending of the tower, since fore-aft bending of the tower will impact the pitch angle seen by the nacelle-mounted lidar. Or, and that would be more problematic, it can be explained by a pitching motion of the optical head in the nacelle reference frame, for example if the optical head is not properly bolted. But according to the lidar manufacturer, this is not the case.

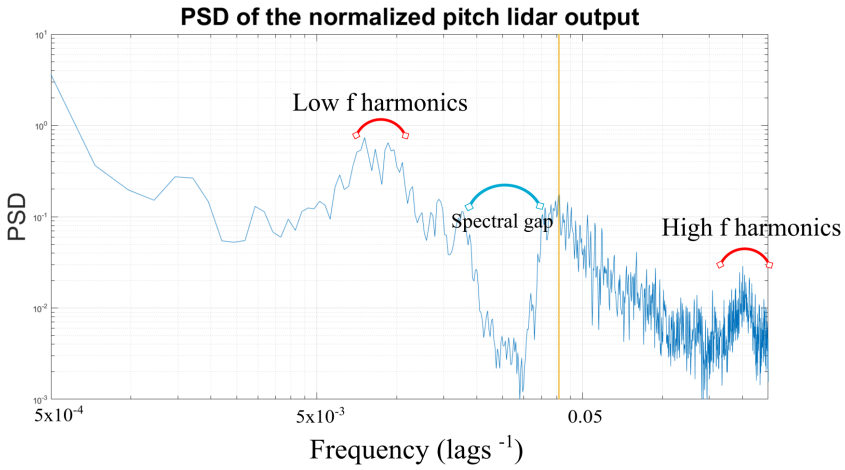


Figure 3.10: Normalized PSD spectrum of the pitch angle from the lidar output, for a part of the day in mid-January.

As a result of these differences, frequency filtering around the pitch frequency is not viable as a generic correction method in this case. The focus will now be made on direct motion-correction, since no shortcut seems possible with frequency filtering. The problem with this kind of correction is that accurate data is needed with a lot of possible sources of error, with a non-exhaustive list being as follows:

- Error coming from the motion and position tracking devices participating in noisy output. For example: nacelle acceleration impacting inclinometer angle error (see Figure 3.2), GPS positioning error, numerical integration/differentiation errors.
- Inaccurate knowledge on the initial position/orientation of the lidar or INS compared to an earth-fixed reference frame, or between the two.
- Error coming from interpolation methods.
- Model error on lidar beam assumption, from point-wise approximation highly simplifying the problem, to spatial averaging methods depending on the assumption of the range-weighting function/numerical integration methods.

All these errors are inevitable and the uncertainty carried by the model requires a further study, which will be started later in this report. In this section, the motion and position tracking devices are treated, as well as their inherent errors. Next, the following subsection will focus on the INS output and show how to use its outputs considering the position, and possibly orientation, offset compared to the lidar.

3.1.2. INS outputs

INS introduction and composition

The Inertial Navigation System is placed inside of the nacelle, so it has a considerable offset from the lidar position that needs to be accounted for. This device is composed of an Inertial Measurement Unit that outputs linear accelerations along all 3 axes of the device and the angular velocities around

all 3 axes [23]. Note that generally angular acceleration is not part of the process. All quantities are given in the device frame that is defined by a North-East-Down (NED) frame. This frame is explained in Figure 3.14. For position tracking, the INS also uses a GNSS antenna. This antenna is able to give the latitude/longitude of the device and the heave (or altitude). Some also have an integrated compass that can give yaw and pitch or roll. The Real-Time Kinematics technology combines the signals from the satellites and from a local base station to yield better positioning accuracy. The GNSS also acts as aiding sensor for the IMU and combines the measurements of the two to reduce errors from the single devices thanks to a filter called the Extended Kalman Filter (EKF), briefly explained later. Linear velocities and orientation attitude are respectively obtained by integrating linear acceleration and angular velocities. As a final output with the aid of the GNSS measurements, the EKF uses mathematical navigation equations to combine the best of the two worlds of the GNSS and of the accelerometers/-gyroscopes. On the one hand, GNSS measurements are accurate but typically have a low sampling rate whereas the IMU has a high sampling rate but suffers from error due to integration. These are for example random walk errors or coning/sculling errors, and will also be addressed. All in all, the output process from the INS is given by Figure 3.11.

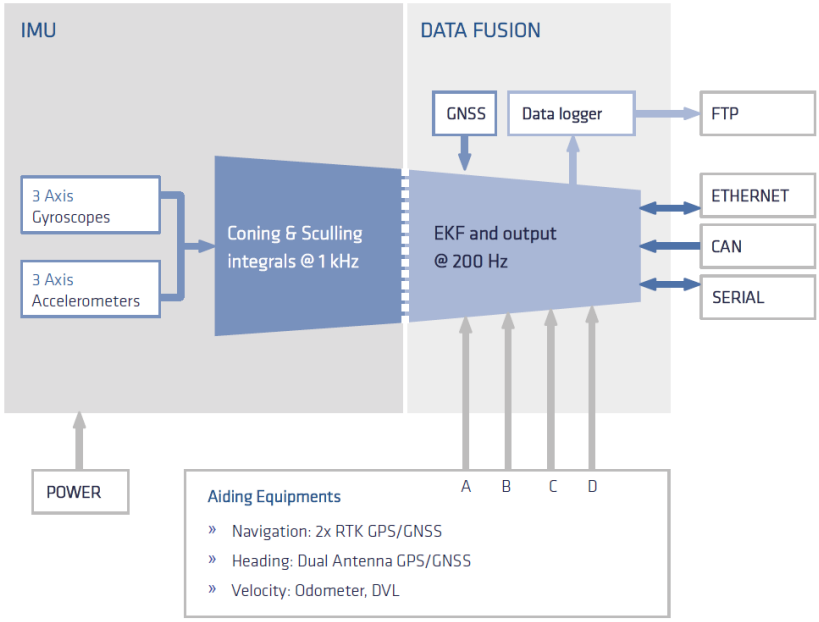


Figure 3.11: INS simplified block diagram [23]

One can see on the right side the connections to the network, the data is usually stored in some online database, and that can also affect the stored output rate, usually being lower than the $200\,\mathrm{Hz}$ of the EKF. Downsampling is also frequently used as a way to filter out some noise frequencies. The diagram refers to ”coning” and ”sculling” integrals. These are integration methods that reduce the error coming from the fact that the device moves while the integration is performed. Before diving into these integrals, it is fundamental to have a comprehension of how the accelerometers and gyroscopes function and give respectively linear acceleration and angular velocity.

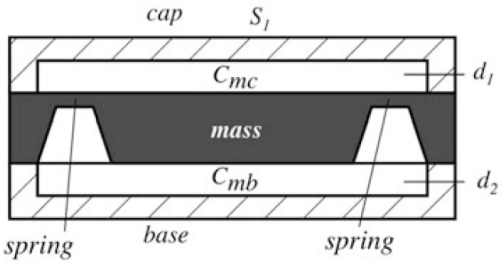
IMU: Accelerometers, gyroscopes and integration

In the INS manual, it is written that the accelerometers and the gyroscopes are Micro-ElectroMechanical Systems (MEMS), meaning that they incorporate both electronic and moving parts. To start with, the accelerometers used here are capacitive. The principle is the same used for the majority of accelerometers that measure linear acceleration. Inside of the accelerometer, a mass is attached to a spring, which moves as the device moves and is subject to the same acceleration. Under some assumptions that generally hold for the accelerometer, for example assuming that electrostatic forces do not affect the mass position, Equation 3.2 links the mass acceleration and the opposite force applied by the spring, which is the product of the spring constant k and the mass displacement Δd. This is by projecting the

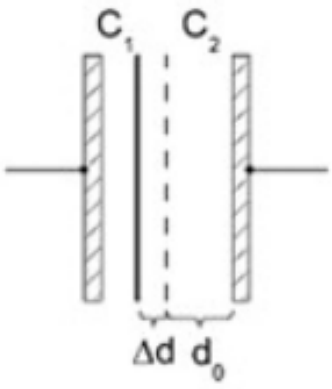
acceleration vector and the spring force onto the axis of the spring. Note that the effect of the gravitational force, which is known with precision, is also taken into account but not shown here for simplicity. Thus, by having springs in 3 different axes, the 3D linear acceleration vector can be reconstructed, this is why there are 3 accelerometers.

$$m\, a_{mass} = F_{spring} = k\, \Delta d \tag{3.2}$$

In capacitive accelerometers, the mass-spring ensemble is fit in between two capacitors, that have a capacitance C_{mc} and C_{mb} in Figure 3.12a (or C_1 and C_2 in Figure 3.12b). When the mass moves, the distance with one of the capacitors reduces by Δd while it increases by Δd with the other. This induces an increase in capacitance for one and a decrease for the other. This change in capacitance depends linearly on the displacement Δd [27]. Because this capacitance change can be measured through the voltage output measurement, it is possible to extract the value of Δd and by knowing the constant of the springs (here usually Silicon springs), the linear acceleration is finally found through Equation 3.2.



(a) Capacitive accelerometer side cross-sectional view [27]



(b) Mass-Spring ensemble displaced by Δd from the initial distance d_0 [81]

Figure 3.12: Functional principle of capacitive accelerometers

Next, the gyroscopes are utilizing the principle of vibratory gyroscopes. To keep the section as concise as possible, this discussion is kept short. The basic idea is to use Coriolis force to estimate angular velocity. Consider a rotating mass that vibrates in the radial direction of the rotation. For example, if the motion of the mass describes a circle, then the vibration is in the same direction as the radius of this circle. In this case, Coriolis force works perpendicular to the vibration direction, which modifies the motion of the vibrating element, typically inducing vibration in the direction of Coriolis force. The amplitude of this new vibration can be measured and happens to be a function of angular velocity. This is how vibratory gyroscopes estimate angular velocity.

Accelerometers give linear acceleration and gyroscopes give angular velocity. Thus, to find linear velocity and orientation angles, it requires respective integration of the two previous quantities. This comes with errors, due to : numerical integration errors, coning and sculling errors (defined below) and impact of the initial noise of the acceleration and angular velocity. Noise can be defined as a random process, with a mean and standard deviation, that alters the expected output of the sensor. In particular, there exists Gaussian noise with a 0 mean. This noise can for example come from the non-ideal properties of the material, or from interfering forces/acceleration on the sensors and they can result either in a higher signal value than expected or in a lower signal value.
First, one can look into the integration errors that come from the noise of the integrated quantity. This noise error is accumulated over time and is called drift. To understand this drift, consider Equations 3.3 and 3.4 [11] for the integration of the angular velocity $\boldsymbol{\omega_i^{IMU}}$ and acceleration $\boldsymbol{a_i^{IMU}}$, $\boldsymbol{i}$ being a subscript that implies these quantities are given with respect to an inertial frame (so for example in the Earth-fixed reference frame). The angular velocities and accelerations of the IMU are given in the non-inertial IMU local frame, so they typically have to be applied a rotation matrix or quaternion that

transforms the local coordinates into global Earth coordinates. The integration is performed between a time t_{m-1} and t_m. The angular quantity described by $\Delta\boldsymbol{\theta}^{\boldsymbol{IMU}}$, called the incremental angle vector, is different than Euler angles. It is rather linked to the quaternion representing the rotation of the IMU frame between t_{m-1} and t_m (axis-angle representation). Equations 3.3 and 3.4 are vector integrals so they in fact imply 3 different integrals, one for each space component.

$$\Delta\boldsymbol{\theta}_{\boldsymbol{m}}^{\boldsymbol{IMU}} = \int_{t_{m-1}}^{t_m} \boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{IMU}}(\tau)d\tau \tag{3.3}$$

$$\Delta\boldsymbol{v}_{\boldsymbol{m}}^{\boldsymbol{IMU}} = \int_{t_{m-1}}^{t_m} \boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{IMU}}(\tau)d\tau \tag{3.4}$$

Notice the symbol Δ. This insinuates that, for example, Equation 3.4 does not give directly the velocity $\boldsymbol{v}^{\boldsymbol{IMU}}$ at time t_m but the results is $\boldsymbol{v}^{\boldsymbol{IMU}}(t_m) - \boldsymbol{v}^{\boldsymbol{IMU}}(t_{m-1})$. Thus,

$$\boldsymbol{v}^{\boldsymbol{IMU}}(t_m) = \boldsymbol{v}^{\boldsymbol{IMU}}(t_{m-1}) + \Delta\boldsymbol{v}_{\boldsymbol{m}}^{\boldsymbol{IMU}}$$

And one can write the same for $\boldsymbol{v}^{\boldsymbol{IMU}}(t_{m-1})$, and so on, iteratively, until reaching $\boldsymbol{v}^{\boldsymbol{IMU}}(t_1)$, with t_1 the start of the time interval. If the time interval has a length N, and considering that the process is initialized by $\boldsymbol{v}^{\boldsymbol{IMU}}(t_0) = \mathbf{0}$, the final measurement $\boldsymbol{v}^{\boldsymbol{IMU}}(t_N)$ is the following accumulation:

$$\boldsymbol{v}^{\boldsymbol{IMU}}(t_N) = \sum_{i=1}^{N} \Delta\boldsymbol{v}_{\boldsymbol{i}}^{\boldsymbol{IMU}}$$

So, the integral errors from each step are carried through the process. One of the error sources is the inherent noise contained within the quantities being integrated (angular velocities and accelerations). For each time step, the noise error can either be positive or negative, then either compensating each other or accumulating. This can be modelled by a stochastic process called *random walk* (angle random walk or velocity random walk), an analogy to randomly walking forwards (increasing cumulative error) or backwards (reducing cumulative error). This process has a standard deviation called the Allan deviation. The unit of the random walk errors is typically given in unit per $\sqrt{\text{hour}}$. For example, for the IMU used in this TetraSpar nacelle, the angular random walk is 0.14 $/\sqrt{\text{hr}}$. So, say the integrations are performed over a period of a minute. One should then expect an error coming from noise drift of $0.018°$. To counter this drift, the INS uses other sources of velocity and angle measurements, for example following the position through GNSS data can help give another estimate of velocity. This was the study of the error coming from noise present before integration. There exists another kind of significant errors, called coning and sculling errors, that occur especially for specific motions of the IMU.

Coning error relates to the integral of Equation 3.3 while sculling error relates to the integral of Equation 3.4. Coning error is due to the fact that the orientation vector is obtained not only by the integration of the angular velocity but also by the addition of another term that accounts for the change of the angular velocity itself during the integration between t_{m-1} and t_m [11]. This term is usually negligible but it can become prevalent when the axis of rotation describes a cone with non-zero angular acceleration. Then, sculling error also happens because of the change of the acceleration vector during the integration time. Thus, both integrals require specific algorithms, which is why Figure 3.11 specifies it. Because the motion of the wind turbine is non-trivial with constantly changing linear acceleration and angular acceleration due to the dynamics of the system, these kinds of errors should be accounted for. Consequently, as sources of error are legion, the use of the Extended Kalman Filter as a way to make outputs more reliable is common.

Extended Kalman Filter

The Kalman filter is a recursive data processing algorithm that estimates efficiently the state of a process using a set of mathematical equations. So-called states can be for example position or velocity. The algorithm is informed by the measurements from sensors (accelerometer or gyroscope for example), and also by a mathematical model of the system that describes how different states depend on each other and how measurements depend on states [32]. In the case of an Inertial Navigation System,

the equations of this mathematical model are called *navigation equations*. They can for example be defined by kinematics principles. The term *extended* refers to the use of non-linear equations, whereas the simple Kalman filter uses linear equations.

The EKF works in a repeating recursive cycle. Based on an initial estimate of given states (for example position, velocity and acceleration) and their uncertainties, the filter projects (or **predicts**) this estimate at the next time step using navigation equations. This step is called ”prediction”, or time update. With the initial uncertainty of the estimate and the accuracy of the modelling equations, the uncertainty (as a covariance matrix P) of this predicted estimate is calculated. Then, at the time step of the predicted state, the measurements of the sensors give new information on the value of the state. The measurements also have an uncertainty, which is treated as a covariance matrix R. Depending on the values of P and R, the predicted estimate is adjusted (or **corrected**).
As a general rule and simply explained, if the uncertainty of the measurements is far lower than the uncertainty of the predicted estimate, the corrected state value will drift towards the value of the measurement. If the opposite happens, then the correction is almost zero and the value of the predicted estimate is kept [85]. Most of the time, both uncertainties are in a comparable order of magnitude. Therefore, at each time step, the output state value is a weighting of both the predicted estimate (based on the previous state value and navigation equations) and the measurement value. The basic principle of the algorithm of an EKF is shown in Figure 3.13.

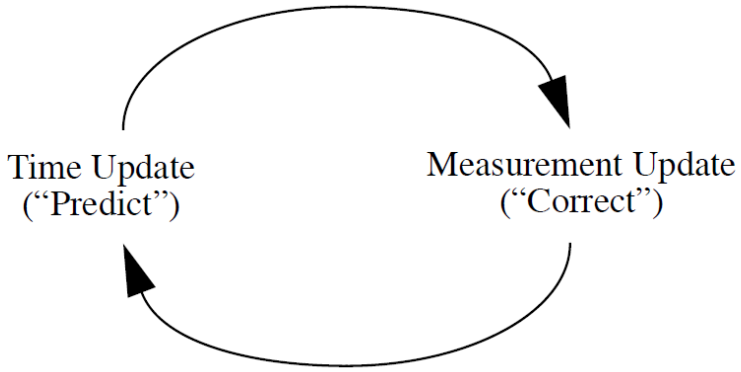


Figure 3.13: Discrete Kalman filter cycle [85].

The output of the EKF at each time step is then the value of the state optimized such that it obeys both physical principles (navigation equations) and real measurements. It is generally accepted that outputs from such a filter are far more reliable than raw measurements.

3.1.3. Definition and positioning of all reference frames

There have been allusions to the Earth-fixed global frame and to the local frame, but how they behave exactly remains to be defined. This is a fundamental part of the project so it is worth defining and showing them clearly. All the relevant reference frames are defined below and shown in the following figures.

- Global Earth-fixed frame, $\mathcal{R}_e$, defined by a fixed point in space, for example at the center of the Earth or close to the turbine location, with a fixed orientation. It is assumed to be inertial. Preferably, all motion and position quantities will be expressed in that frame after necessary transformations.
- Local body-fixed North-East-Down frame, $\mathcal{R}_{NED}$, attached to the INS' center of mass, with negligibly changing orientation. Its X-axis is aligned with true north (on top of the Earth ellipsoid), its Z-axis is pointing down and the Y-axis follows the right-hand rule, pointing East. This frame is introduced because by convention, the INS' outputs are expressed in it [38].

- Local body-fixed INS frame, $\mathcal{R}_{INS}$, attached to the INS' center of mass and following the orientation of the INS.
- Local body-fixed lidar frame $\mathcal{R}_{lidar}$, attached to the lidar's center of mass and following the orientation of the lidar. By assumption, it has a constant orientation offset compared to the INS and undergo the same rotations during the turbine motion. The position of its center's offset to that of the lidar also follows the same rotations.

First, to understand the NED frame, consider Figure 3.14. The cross represents the position of the body considered, for example the INS or the lidar. In an Earth-Centered Earth-Fixed referential, this body has a latitude ϕ and longitude λ. The X-axis of the frame is tangent to the corresponding meridian towards North, the Z-axis is defined in the direction of the Earth radius, "down", so in the same direction as gravity. Finally, the Y-axis is given by the right-hand rule, and is directed towards East. The INS gives all quantities, for example its orientation, compared to this NED frame, the origin of which follows the center of mass of the INS.

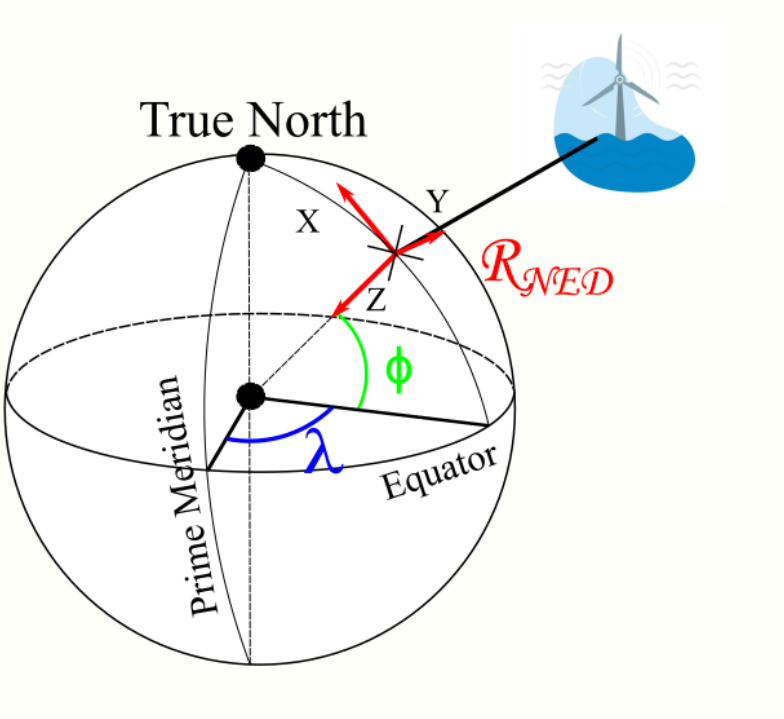


Figure 3.14: Earth and local North-East-Down frame. ϕ : latitude, λ : longitude

Now that $\mathcal{R}_{NED}$ has been defined, it is necessary to define the global frame. For this, a point is arbitrarily fixed. For example, it can be the point at which the INS is at $t = 0$. So, at that time, the origin of the global frame and that of the NED local frame are the same. However, they can be defined to have a different orientation. The choice of this orientation is also arbitrary. It is more practical to have a fixed frame with the Z-axis pointing up (so the opposite of the NED's Z-axis) and a X-axis facing the general wind direction. One can find a circle defining the location of the turbine in Figure 3.15. This is defined broadly due to the sensitivity of this information.

Around the location of the turbine, dominant wind is found to be southeasterly, with a range of $115°$ - $145°$. That can however change to a northeasterly direction [3] [26]. The global Earth-fixed reference frame $\mathcal{R}_e$ is thus made to face a typical wind direction, for example $125°$. Then, the Z-axis is made to be upwards, exactly the opposite of the initial NED frame's Z-axis and the Y-axis simply follows the right hand rule. One can see this in Figure 3.16. The green global frame $\mathcal{R}_e$ is fixed. Once again, this fixed frame could also be oriented North-East-Down, but this reference change is made because it is more intuitive to have a frame that is close to the initial orientation of the turbine. In this Figure, the Earth curvature is exaggerated such that the orientation of the NED frame between $\mathcal{R}_{NED,0}$ and $\mathcal{R}_{NED'}$ seems to be quite different. However, since the turbine moves within small boundaries compared to the

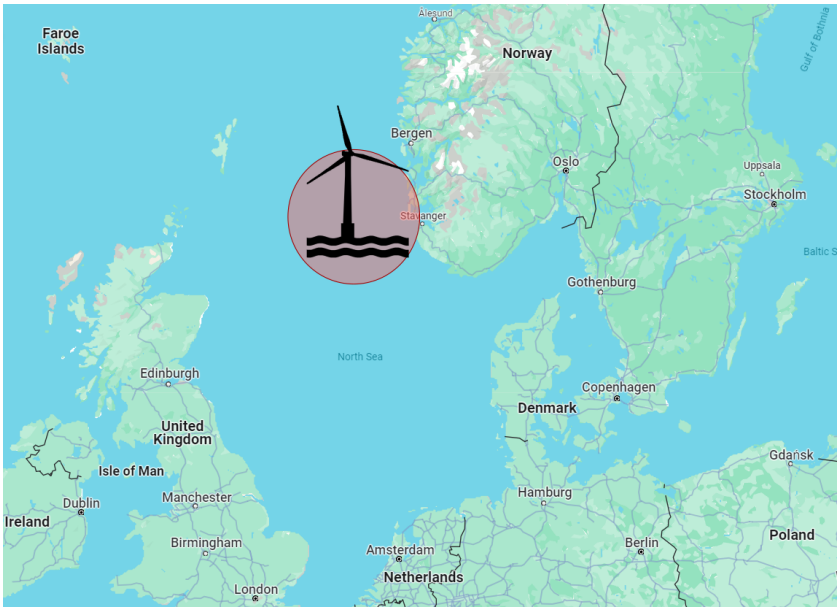


Figure 3.15: Approximate turbine location in the North sea close to Norway coast, defined by the circle

Earth's dimension, the actual orientation of the NED frame can be considered constant. Consequently, the quaternion that rotates the local NED frame's orientation to the global frame $\mathcal{R}_e$, q_e^{NED}, is assumed constant. And if one makes the choice to keep a global frame oriented NED, then that quaternion is in fact the identity quaternion. There is another rotation that should be considered. It is the one that rotates the INS' body frame to that of the lidar, in case of an orientation offset. This rotation is always the same and corresponds to the initial orientation offset between the two, represented by q_{INS}^{lidar}. Thus, finding the final orientation of the lidar in the global frame can recapitulated as such, using quaternions to represent rotation:

- Find the orientation of the lidar in the INS local frame, by applying q_{INS}^{lidar}. If both devices are perfectly aligned in the rotor axis, then that is the identity quaternion. Not time-dependent.
- Find the orientation of the INS in the NED local frame, given as an output of the INS, by applying $q_{NED}^{INS}(t)$. Time-dependent.
- Finally, find the orientation of the lidar in the global frame, by applying q_e^{NED}. If the global frame is oriented NED, then that is the identity quaternion. Not time-dependent.

For example, take the vector representing the position of one beam waist in the lidar reference frame, $\boldsymbol{x}_{\boldsymbol{lidar}}^{\boldsymbol{beam}}$. This vector is not time-dependent, it is fixed in the lidar reference frame for a pulsed lidar. For a CW lidar, because the beam follows a scanning pattern, this vector has a relative motion compared to the lidar optical head so this additional effect would need to be considered. Then, to have the position of the beam waist in the INS reference frame, one should apply the first rotation introduced, and account for the position offset between the two in the INS referential, written as $\boldsymbol{x}_{\boldsymbol{INS}}^{\boldsymbol{offset}}$.

$$\boldsymbol{x}_{\boldsymbol{INS}}^{\boldsymbol{beam}} = \boldsymbol{x}_{\boldsymbol{INS}}^{\boldsymbol{offset}} + q_{INS}^{lidar}\, \boldsymbol{x}_{\boldsymbol{lidar}}^{\boldsymbol{beam}}\, q_{INS}^{lidar^{-1}}$$

Here again, this position is not time-dependent because by assumption, both devices belong to the same rigid body, being the nacelle of the turbine. The next step is to get the beam position vector in the local NED frame. Now, this is time-dependent because the turbine is moving so its orientation compared to North-East-Down is constantly changing. No additional vector is needed because the origin of the two frames match. The order of rotation matters.

$$\boldsymbol{x}_{\boldsymbol{NED}}^{\boldsymbol{beam}}(t) = q_{NED}^{INS}(t)\, \boldsymbol{x}_{\boldsymbol{INS}}^{\boldsymbol{beam}}\, q_{NED}^{INS^{-1}}(t) = q_{NED}^{INS}(t)\, (\boldsymbol{x}_{\boldsymbol{INS}}^{\boldsymbol{offset}} + q_{INS}^{lidar}\, \boldsymbol{x}_{\boldsymbol{lidar}}^{\boldsymbol{beam}}\, q_{INS}^{lidar^{-1}})\, q_{NED}^{INS^{-1}}(t)$$

Finally, because the NED frame is not the most practical, one might want to perform a final rotation. This rotation is also considered constant, the Earth being locally flat in the turbine environment. And the position of the origin of the local NED frame needs to be added, because the FOWT can not only rotate but also translate. This gives the final expression for the beam position vector in the Earth frame in Equation 3.5. Note that $\boldsymbol{x_e^{NED}}(t) = \boldsymbol{x_e^{INS}}(t)$. This position can be given directly from the IMU with Surge, Sway and Heave representing the displacement of the INS with respect to a reference point (see next subsection), which would be the origin of $\mathcal{R}_e$. Alternatively, it can be obtained in the Earth-Centered, Earth-Fixed (ECEF) reference frame from GPS data, with longitude, latitude and altitude. By knowing the ECEF position of the origin of $\mathcal{R}_e$, a simple subtraction gives $\boldsymbol{x_e^{INS}}(t)$.

$$\begin{aligned}\boldsymbol{x_e^{beam}}(t) = \boldsymbol{x_e^{NED}}(t) + q_e^{NED}\, \boldsymbol{x_{NED}^{beam}}(t)\, q_e^{NED^{-1}} = \boldsymbol{x_e^{NED}}(t) + \\ q_e^{NED}\left(q_{NED}^{INS}(t)\left(\boldsymbol{x_{INS}^{offset}} + q_{INS}^{lidar}\, \boldsymbol{x_{lidar}^{beam}}\, q_{INS}^{lidar^{-1}}\right) q_{NED}^{INS^{-1}}(t)\right) q_e^{NED^{-1}}\end{aligned} \tag{3.5}$$

In the eventuality that the lidar is perfectly aligned with the INS, and if the global Earth frame is defined in the NED convention too, then the only rotation that needs to be applied is the one from the INS to the NED.
All that is left in this case is $\boldsymbol{x_e^{beam}}(t) = \boldsymbol{x_e^{NED}}(t) + q_{NED}^{INS}(t)\left(\boldsymbol{x_{INS}^{offset}} + \boldsymbol{x_{lidar}^{beam}}\right) q_{NED}^{INS^{-1}}(t)$. This corresponds to replacing the exterior and interior quaternions of Equation 3.5 by identity quaternions. Similar operations apply to all quantities. They will be better defined in the next subsection.

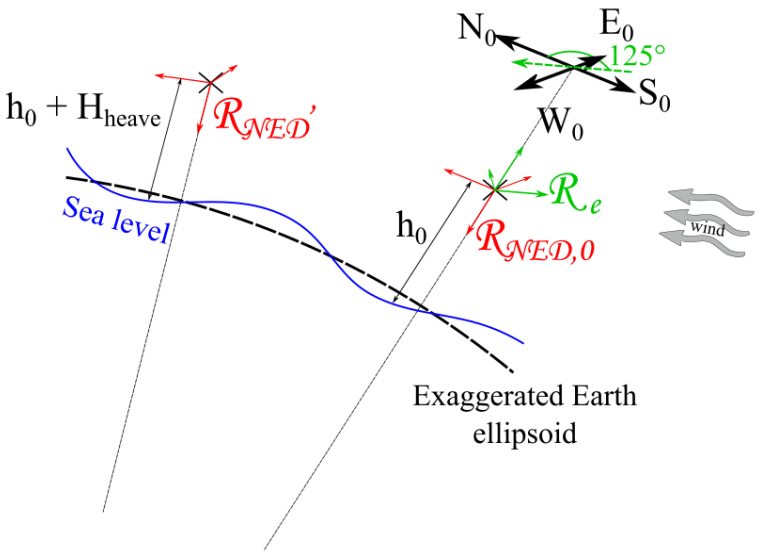
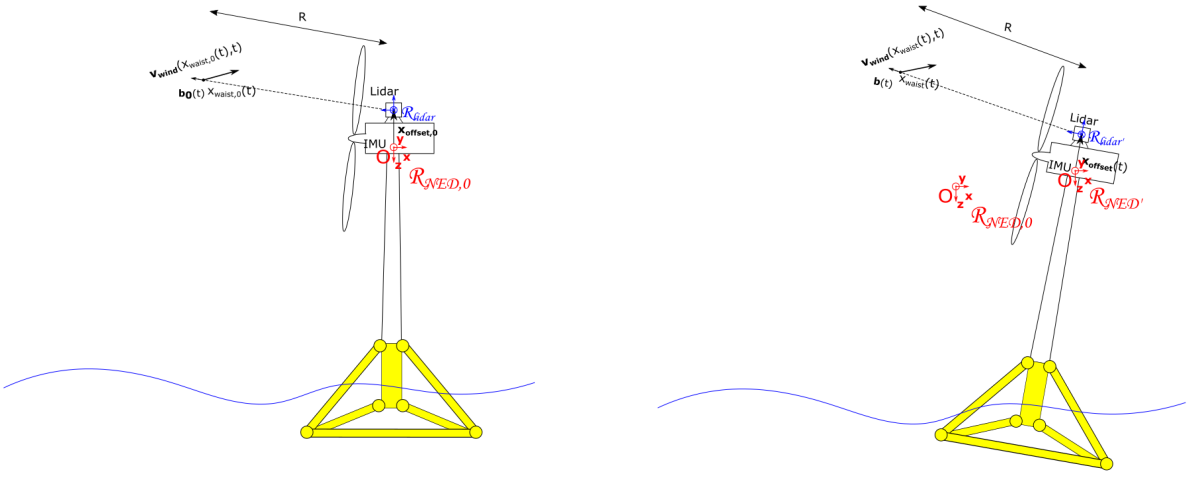


Figure 3.16: NED local frame and global Earth-fixed reference frame, as the turbine moves

3.2. Problem definition and point-wise approximation

In this part, it will be assumed that the laser beam is focused at one point in space and that the corresponding line-of-sight velocity is related straightforwardly to the wind velocity. The lidar used is mounted to the nacelle of the floating wind turbine, as shown in Figure 3.17 and it is a pulsed lidar. Figure 3.17a and Figure 3.17b attempt to illustrate clearly the problem.

The frames are defined correspondingly as what was done in the previous subsection. Figure 3.17a would represent the cross-section of an initial situation where the turbine is straight and Figure 3.17a is the position of the turbine at a later time where the turbine has translated and rotated. The local NED referential and the lidar referential follow the motion of the turbine. In Figure 3.17b, because it is a cross-section, the lidar reference frame's X-axis seems to be in the exact opposite direction as the local NED's X-axis (which follows North) but it is not precisely the case since the turbine would not be exactly facing South. The initial NED frame $\mathcal{R}_{NED,0}$ is shown also in Figure 3.17b, since it can serve as the



(a) Initial position

(b) Different position at time t. The beam $\boldsymbol{b}(t)$ can change from the motion of the turbine or from a change of beam.

Figure 3.17: 2D view of the floating turbine and the lidar in motion

Earth-fixed global frame. Consider a certain beam with unit vector $\boldsymbol{b}(t)$ measuring the wind velocity at a range R. It measures the velocity of the wind in the lidar referential $\boldsymbol{v}_{\boldsymbol{lidar}}^{\boldsymbol{wind}}$ at the beam waist $\boldsymbol{x}_{\boldsymbol{waist}}(t)$. The beam waist position in the Earth-fixed frame is defined in Equation 3.7, taking into account the rotation of the lidar reference frame with respect to the Earth-fixed frame, by the quaternion $q_e^{lidar}(t)$. In the lidar referential, the beam unit vector is constant. But in the Earth frame, it is given as Equation 3.6.

$$\boldsymbol{b_e}(t) = q_e^{lidar}(t)\, \boldsymbol{b_{lidar}}\, q_e^{lidar^{-1}}(t) \tag{3.6}$$

$$\boldsymbol{x_{waist}}(t) = \boldsymbol{x_e^{lidar}}(t) + R\boldsymbol{b_e}(t) \tag{3.7}$$

By changing referential, the absolute wind velocity can be expressed as Equation 3.8.

$$\boldsymbol{v_e^{wind}}(t) = \boldsymbol{v_e^{lidar}}(t) + \boldsymbol{v_{lidar}^{wind}}(t) + \boldsymbol{\omega_e^{lidar}}(t) \times R\boldsymbol{b_e}(t) \tag{3.8}$$

In the point-wise approximation, the velocity measured by the lidar is simply:

$$\overline{v_r}(t) = \boldsymbol{v_{lidar}^{wind}}(t) \cdot \boldsymbol{b_e}(t)$$

Thus, by taking the projection of Equation 3.8 onto the beam vector, the term with the angular velocity of the lidar to Earth disappears because of the cross product with the beam vector. It then comes that:

$$\boldsymbol{v_e^{wind}}(t) \cdot \boldsymbol{b_e}(t) = \boldsymbol{v_e^{lidar}}(t) \cdot \boldsymbol{b_e}(t) + \overline{v_r}(t)$$

The output of the lidar is $\overline{v_r}(t)$ so it only remains to get $\boldsymbol{v_e^{lidar}}(t)$. Because the INS and the lidar belong to the same rigid body, their relative velocity is Equation 3.9, where $\boldsymbol{\omega_e^{INS}}(t)$ is the angular velocity of the whole nacelle with respect to Earth.

$$\boldsymbol{v_e^{lidar}}(t) = \boldsymbol{v_e^{INS}}(t) + \boldsymbol{\omega_e^{INS}}(t) \times \boldsymbol{x_e^{offset}}(t) \tag{3.9}$$

Here, $\boldsymbol{v_e^{INS}}(t)$ and $\boldsymbol{\omega_e^{INS}}(t)$ are outputs of the INS. The offset between the lidar and the INS $\boldsymbol{x_e^{offset}}(t)$ can be determined by applying the quaternion $q_e^{INS}(t)$ to the constant offset in the INS referential. That quaternion is also the only remaining unknown to get $q_e^{lidar}(t)$. Thus, all quantities defined are available from the INS and the lidar output. Doing the necessary interpolations, the algorithm can then perform these chain of calculations to retrieve the wind velocity with respect to the Earth-fixed referential.

4

Model and uncertainties

This chapter will focus on presenting the results obtained from the algorithm with simulations of the lidar measurement scheme by adding prescribed motion to it. The input wind field is made synthetically and should be the same when reconstructed from lidar measurements. To be able to run such simulations, the lidar model framework needs to be defined. Some simple examples on how the algorithm can simulate lidar measurements and correct an interfering velocity will be shown. A separate part will also focus on uncertainty quantification of the model and its parameters. Only primary results on uncertainty will be discussed since obtaining the final uncertainty in wind components estimates is outside the scope of this project

4.1. Lidar model and first model verifications

4.1.1. Lidar model

In this section, the lidar modelling algorithm will be presented. The algorithm used to simulate numerically the functioning of a pulsed lidar will be done using foundations set by M. Pellé and N. Warncke for a CW lidar [63]. An overview of this model was given in the former's report, see Figure 4.1. Among the steps of this algorithm, the Doppler spectra will not be a part of this project.

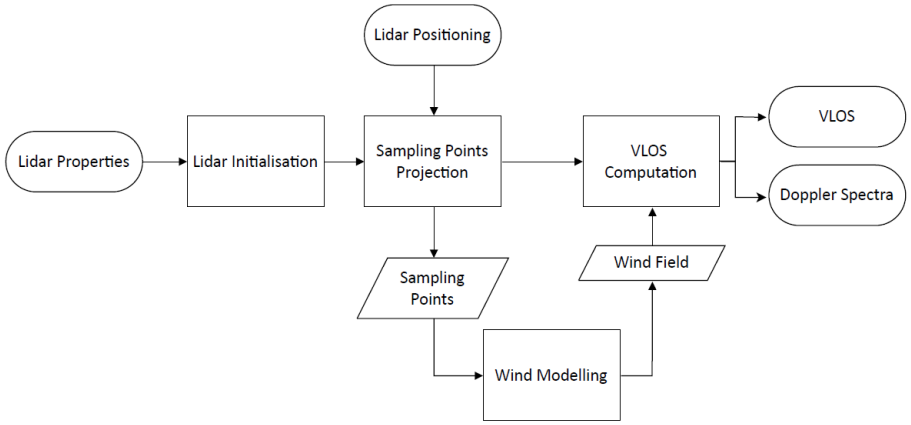


Figure 4.1: Lidar model overview [63].

At first, all of the steps will be briefly explained in the case of a CW lidar. The content that was added in the scope of this report will then be detailed.

Lidar properties

The lidar properties should be defined. That includes the optical properties of the laser beams used by the lidar and the positioning of the beams in the lidar frame because the beams are generally not pointing straight. For the optical properties of a CW lidar, a simple model only needs:

- The *focal length*. This is determined by the transmitter telescope optics. For a CW lidar, given that the optical lens is situated at the aperture of the lidar, then that also corresponds to the range where the beam is focused. In other words, that is the range where the beam waist is minimal, and the intensity maximal.
- The *Rayleigh length*. This was defined in 2.2.1. It defines the axial envelope of the beams and the beam waist.
- The *wavelength* of the beams.

The lidar uses a certain number of beams that have in principle all the same optical properties. The beams only vary by their orientation in the lidar frame, that can be defined by two angles: *azimuth* angle and *zenith* angle. These are the two last parameters of the lidar properties. The two angles are shown in Figure 4.2. For current industry lidars, the number of beams ranges for example from 4 to 50.

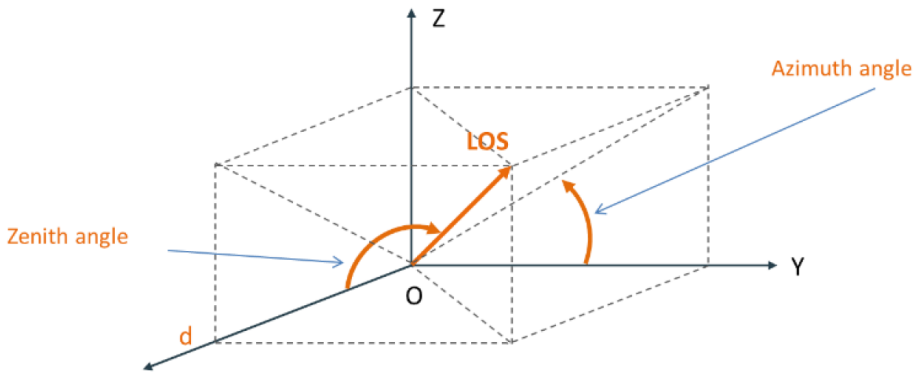


Figure 4.2: Azimuth/Zenith angle for the definition of a Line Of Sight in the lidar coordinate system [86].

Lidar initialisation

Now that the lidar properties were defined, one can initialize one type of lidar, the properties being given by the manufacturer. That lidar will have a certain number of beams, usually their zenith angle is the same, only the azimuth angle varies somewhere between -180° and +180°. For example, the ZephIR lidar sends 50 beams, one by one in a time window of around $1\,\mathrm{s}$ [64], forming a conical scanning pattern that is repeated second after second. Now, for each beam, how exactly to perform the simulation of a lidar line-of-sight velocity, taking into account the spatial averaging? As a reminder, for a beam that has a unit direction vector $\boldsymbol{n}$, the line-of-sight velocity measurement of the lidar is given by the convolution of the radial wind speed and a range-weighting function W, that is maximal at the beam waist and that integrates to 1:

$$\overline{v_r} = \int_{-\infty}^{+\infty} W(s)\, \boldsymbol{n} \,.\, \boldsymbol{v}(s) ds$$

In contrary to a pulsed lidar, there is no ambiguity on the range of this measurement, since a CW lidar can only measure wind at the range corresponding to its focal length.
Numerically, the previous integral is well approximated, for a large enough number of ”sample points” distributed uniformly, by the sum in Equation 4.1:

$$\overline{v_r} \approx \sum_{i=1}^{N} W\,(\boldsymbol{s_i})\, \boldsymbol{n} \,.\, \boldsymbol{v}(\boldsymbol{s_i}) \tag{4.1}$$

with the normalization constraint on the weighting function, expressed in Equation 4.2.

$$\sum_{i=1}^{N} W(\boldsymbol{s_i}) = 1 \tag{4.2}$$

Here, it is important that the velocity of each sample point at the position $\boldsymbol{s_i}$ (that can be seen as the position of a backscattering particle), i.e. the wind velocity at the position $\boldsymbol{s_i}$, is expressed in the same coordinate system as the unit line-of-sight direction vector $\boldsymbol{n}$. Because the dot product is rotation conserving, they can both be expressed in the lidar frame, or in the local NED frame for example. As it was discussed before, since the Gaussian beam has a negligible radial spread, the points $\boldsymbol{s_i}$ can be defined in 1 dimension, only in the axial direction. However, to have a better modelling of the spatial averaging effect, the points will also have radial components, so 3 coordinates.

From sampling points to final VLOS computation

From this expression, the lidar VLOS measurement simulation appears to only have four components. The first component is to define the unit vector of the beam. If the chosen frame is the lidar frame, then this vector is purely defined by the azimuth and zenith angles. Up to a sign (depending on if the X-axis is defined towards or away from the lidar), if θ is the zenith angle and ϕ the azimuth angle, then the coordinates of the unit vector are:

$$\boldsymbol{n} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta)\ \cos(\phi) \\ \sin(\theta)\ \sin(\phi) \end{bmatrix}$$

If the frame chosen has a different orientation (NED frame for example), then this vector should be rotated (with a given rotation matrix or quaternion). Thus, this line-of-sight vector is directly known from the lidar properties and, if applicable, its orientation. There remains to determine three other parameters:

- The coordinates of the sample points $\boldsymbol{s_i}$, meant to represent the position of the particles that will backscatter light to the lidar.
- The value of the range-weighting function W at each of these positions, meant to represent how intense the backscattered light is for each particle. It can for example be a normalized version of the Gaussian beam intensity in Equation 2.12. Regardless of the modelling choice in this function, it has to integrate to 1.
- The 3D wind velocity $\boldsymbol{v}$ at each of the positions. For simulations, such velocities can be obtained from numerically synthetic wind fields.

Consequently, the definition of the sample points $\boldsymbol{s_i}$ is crucial, and there exists several modelling choices to create their distribution in space. For a 3D Gaussian beam of a CW lidar, a study was conducted to find the optimal distribution of points to approximate the VLOS integral. This distribution of points, or "nodes", in the integral approximation context is called a quadrature and different quadratures were compared [63]. It was finally found that in the axial direction of the beam, the optimal quadrature follows the so-called *Tan-Midpoint rule* while in the radial direction it is the *Gauss-Hermite* quadrature. Further details are beyond the scope of this report but one should bear in mind that these methods allow for an accurate evaluation of the VLOS integral with a reasonably low number of discrete points (here, the $\boldsymbol{s_i}$). Note that in this case, the function to represent weights in Equation 4.1 is not exactly W, because of the quadrature method. As an illustration, an example of this optimal point cloud for a certain beam is shown in Figure 4.3.

This example shows one beam of a lidar placed at a certain origin : height $z = 120\,\mathrm{m}$ and $x = y = 0$. The X-axis was chosen to be pointing in an opposite direction to the lidar, so the coordinates in x are negative. To illustrate what was written before, the lidar properties were here set to:

- Number of beams = 1
- Focal length $F = 100\,\mathrm{m}$

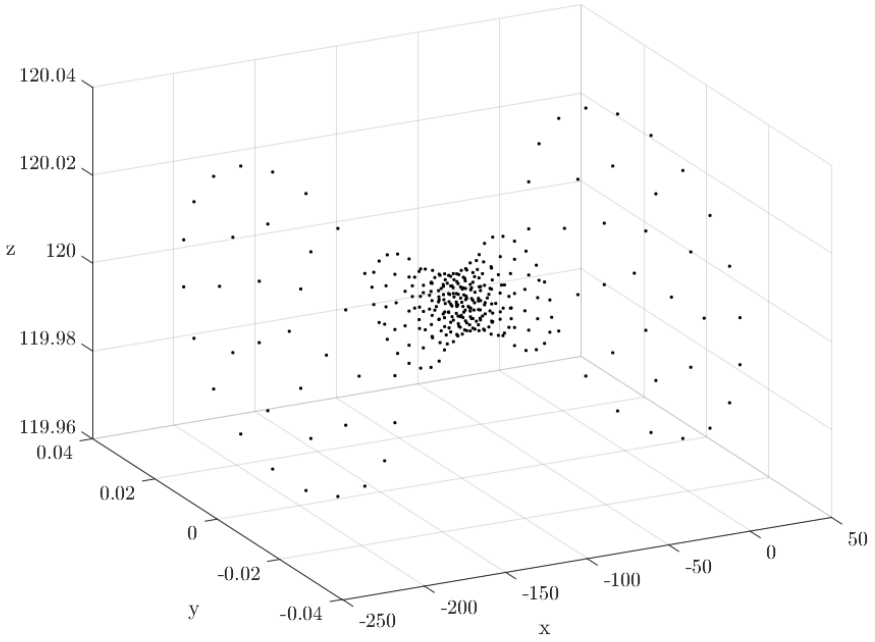


Figure 4.3: Example of an optimal point cloud for the Gaussian beam of a CW lidar, with a focal length of $100\,\mathrm{m}$.

- Rayleigh length $z_R \approx 19\,\mathrm{m}$
- Wavelength $\lambda \approx 1.53\,\mu\mathrm{m}$
- Azimuth/Zenith angle $\phi = \theta = 0$

One can see that with this optimal distribution the point cloud is more tightened around the beam waist and spreads more and more the further from the waist. This highlights well the envelope of the Gaussian beam, as it was shown in Figure 2.4 for example. Remarkably, one can also note the different axis scales. Indeed, because of the low radial spread of the Gaussian beam compared to its axial spread, the X-axis that is along the line-of-sight is far more stretched. Finally, if the lidar moves, then the position of the sampling points in a global frame is calculated internally by changing the origin and orientation of the lidar. This corresponds to the step ”Sampling Points Projection” in Figure 4.1. Now that the distribution of the sample points $\boldsymbol{s_i}$ was treated, the next steps are to determine the value of the range-weighting function and the wind velocity at these points.

For the wind field, this is done separately by generating a synthetic field with fixed properties, for example it can be a simple power law: shear exponent and wind speed at a reference height. Then, once the exact positions of the sampling points are calculated from the zenith/azimuth angles of the beam and the lidar origin/orientation, the wind velocity at each point can be computed. This step corresponds to the arrow between ”Wind Field” and ”VLOS Computation” in Figure 4.1. The final step is then to calculate $W(\boldsymbol{s_i})$.

For this calculation, what matters is the position of the sample points **in the lidar frame**, and especially compared to beam waist. If the sample points have a position $\boldsymbol{s_i}$ in the lidar frame and the beam waist has a position $\boldsymbol{x_{waist}}$ in the lidar frame, let the waist-relative position of the particle be $[x, y, z] = \boldsymbol{x_{waist}} - \boldsymbol{s_i}$. A good modelling of the backscattered intensity by the particle is to use the normalized version of Equation 2.12 as the range-weighting function :

$$W(\boldsymbol{s_i}) \;=\; \frac{w_0^2}{w(x)^2}\,\exp\Big(-2\frac{y^2+z^2}{w(x)^2}\Big)$$

where as previously mentioned,

$$w(x) = w_0\sqrt{1+(\frac{x}{z_R})^2}$$
$$w_0 = \sqrt{\frac{z_R\,\lambda}{\pi}}$$

If the sum of the weights of all the discrete points is not equal to 1, then each weight should be updated by being divided by the weight sum to ensure the normalization constraint. The steps are now complete to make the VLOS computation. How should this method be changed to simulate measurements with a *pulsed* lidar, instead of CW?

From CW to Pulsed lidar

First, regarding the point cloud generation, it was chosen to use uniformly distributed points along the beams, and simply calculate the corresponding weights with the ”real” range-weighting function W. The optimal distribution of Figure 4.3 is only viable for the CW range-weighting function. The pulsed lidar being modelled with a different range-weighting function (basically the product of the same function with the convoluted pulse shape), the optimal quadrature would be different. Using an uniform quadrature of points allows to bypass this optimization but will require far more points for a good approximation of the integral. This means that there is room for improvement in this point cloud generation in the pulsed lidar simulation.
In this case, pulses are sent into the atmosphere and travel with the speed of light. A way of generating short pulses of laser light is for example Q-Switching. What is important there is that these pulses originate from laser light, and the beam sent by the lidar can still be modelled by the Gaussian beam theory, with Rayleigh length, focal length, and beam waist. The Gaussian intensity profile was defined previously and the expression of the pulse shape as a function of time was given in Equation 2.17. The power of the returned signal then has the same shape, arriving from a range R at a time $t = \frac{2R}{c}$. This signal is post-processed into range gates depending on the time they arrive (which translates to a distance in space). This involves Fourier transforming the voltage output of the photo-detector to generate averaged Doppler Spectra. In terms of modelling, it is well modelled by the convolution of the pulse shape with a windowing function. Expression 2.23 gave this expression for example for the case a rectangular windowing function, with the Error function.

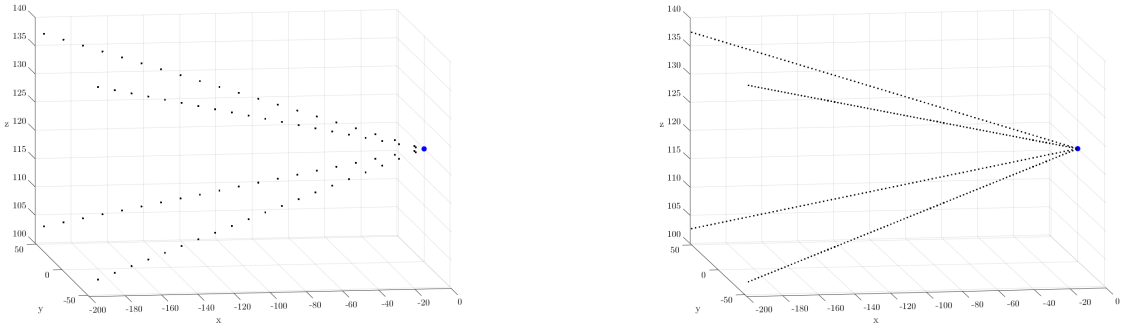
What is generally done is to only use this convoluted pulse shape as the final range-weighting function in the VLOS integral. However, this neglects the fact that the ranges that are close to the beam waist are more illuminated than that further away from it, due to the intensity profile of the beam. Thus, the pulses that are backscattered to the detector coming from a range close to the beam waist are associated with higher intensity. In this regard, the final range-weighting function W_{pulsed} is modelled by Equation 4.3, with A a normalization constant, and $W(\boldsymbol{s_i})$ the previously defined Gaussian intensity.

$$W_{pulsed}(\boldsymbol{s_i}) = A\,W(\boldsymbol{s_i})\left(Erf\Big(\frac{r+\Delta p/2}{\sqrt{2}r_p}\Big) - Erf\Big(\frac{r-\Delta p/2}{\sqrt{2}r_p}\Big)\right) \tag{4.3}$$

Some terms remain to be defined in Equation 4.3. The distance r is the distance between the point $\boldsymbol{s_i}$ and the considered range-gate center R, along the beam axis. In other words, if in the beam reference frame (centered at the waist at the focal length F), the particle has along the main axis the coordinate s, then $r = F + s - R/\cos(\theta)$. As defined earlier, Δp is the range-gate length and r_p depends on the pulse duration. The effect of this modelling will be studied shortly later.

Examples and verification

It is now possible to create a lidar object that will simulate an existing lidar type. For example, consider the WindCube nacelle-mounted lidar using 4 beams. Only the sampling points corresponding to one range gate of $100\,\mathrm{m}$. In the model, it is possible to create an arbitrary number of points, distributed axially and radially. The points of the 4 beams are visible in Figure 4.4, for two different number of axial points. The first lidar object, in Figure 4.4a has the exact same properties as the one in Figure 4.4b, but they will use a different number of sampling points to perform the VLOS computation. The lidar origin is the same in both cases, at a height of $120\,\mathrm{m}$. The point distribution is always centered around the range of $100\,\mathrm{m}$, but having more points allows to take into account the simulation of more particles.



(a) First point cloud made of 20 axial points

(b) Second point cloud made of 100 axial points

Figure 4.4: 3D view of two point clouds of a nacelle-mounted WindCube lidar model. The lidar origin is in blue.

In the figures, because of the low radial spread of the beams, it is not possible to see the radial points but they exist, there are a total of 100 radial points, distributed in a ”grid”, as can be seen by zooming in, see Figure 4.5. Having more points, axially and radially will ensure a better approximation of the VLOS measurement, but it has a computational cost. With regard to the point cloud in Figure 4.4a, there is a total of 2000 points and 10000 in Figure 4.4b.

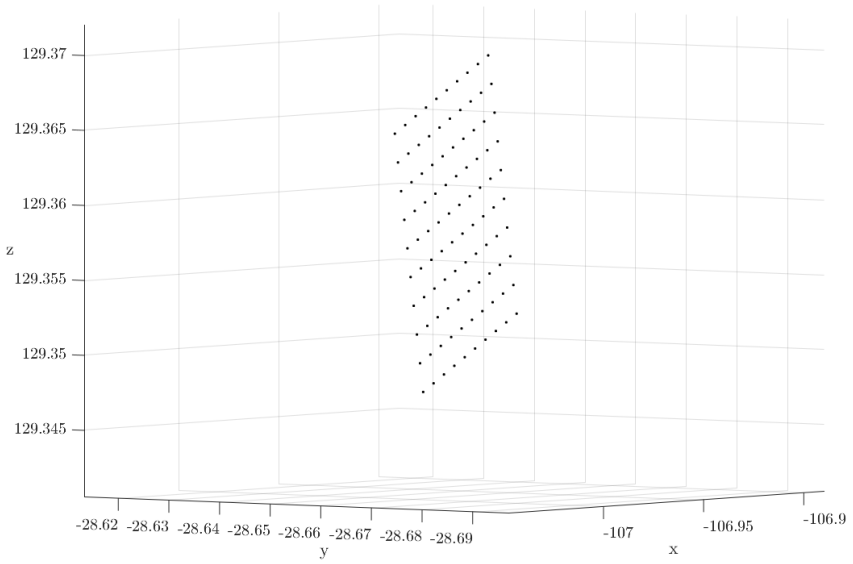


Figure 4.5: Zoom on the point cloud to see the radial distribution.

Apart from the point cloud distribution, it is also possible to see how the weighting function of the simulated pulsed beam behaves. Its shape will be compared to that of a CW lidar. At first, to simplify, only one dimension in the points distribution will be considered, meaning that the sample point are only along the line-of-sight axes. For a CW lidar, the Rayleigh length depends linearly on the focal length, which is the probing distance [55]. On the other hand, in the pulsed case, the focal length, and thus the

Rayleigh length are set constant at whatever the probing distance is. The Range-Weighting function however changes for each range gate considered, as predicts Equation 4.3.

The first lidar object created is a CW lidar with the same beam positioning parameters as the Wind-Cube model that was shown in Figure 4.4. The difference is in the Range-Weighting function. The focal length of this CW lidar was set to be corresponding to a horizontal distance of $100\,\mathrm{m}$ from the lidar. Due to the beams not being horizontal but with a certain angle, the corresponding focal length of each beam is $F_1 = \frac{100}{\cos(\theta)} \approx 103.9\,\mathrm{m}$, where θ is the zenith angle of the beams.
The objective is to visualize the Range-Weighting function with respect to the axial distance along the beam. For this, a number of 2000 points were generated uniformly along all of the 4 beams (in total 8000 points). Because all points are along the beam axis, the exponential factor of the intensity profile is equal to $\exp(0) = 1$. The normalized Range-Weighting function for one beam can be seen in Figure 4.6. For all other beams, the profile is the same. As expected, it reaches its maximum at F_1. Note that, in this figure, the values of the Y-axis are of minor importance because they are normalized to the sum of all weights. For instance, if there are more or loss points, so more or less weights, this value is bound to change by a certain factor. What matters is the shape of the curve, which should be conserved no matter the distribution. This is why for visualization, it is more adapted to normalize the weights to the function maximum, so it will reach 1 at the focal length. This is shown below on Figure 4.7.

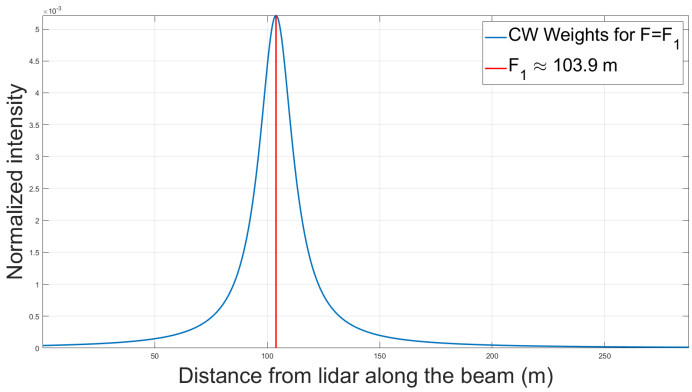


Figure 4.6: Weighting function for one beam of the CW lidar object at one focal length. The function is normalized so that the sum of the weights equals 1.

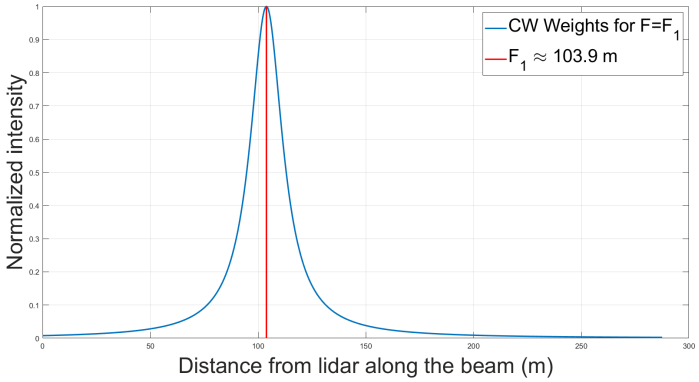


Figure 4.7: Weighting function for one beam of the CW lidar object at one focal length. The function is normalized so that the value at the peak is 1.

Then, a second CW lidar object was created with a different focal length. If the beam is focused instead at a further distance, for example at a horizontal distance of $150\,\mathrm{m}$ (that is a focal distance of $F_2 \approx 155.8\,\mathrm{m}$), the Rayleigh length changes so the intensity profile spreads more. This is shown in Figure 4.8, with the previous beam focused at F_1 for comparison. The higher the focal length, the more the beam will spread. In other words, the beam will illuminate more and more the particles that are

away from the point of focus, meaning a poorer ”focusing”.

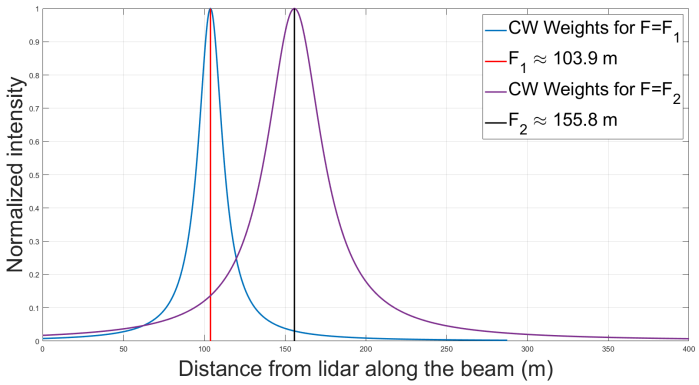


Figure 4.8: Weighting functions for two beams of the CW lidar object with two different focal lengths.

Now, the weights for a pulsed lidar can also be studied. The object created is a pulsed lidar using the same wavelength and beam directions as the CW object. The focal length of the beam was also set to a horizontal distance of $100\,\mathrm{m}$, and the pulse duration at $150\,\mathrm{ns}$. The range-gate length will be set to $\Delta p = 30\,\mathrm{m}$. The results for different ranges are shown in Figure 4.9. Again, only axial points are considered.

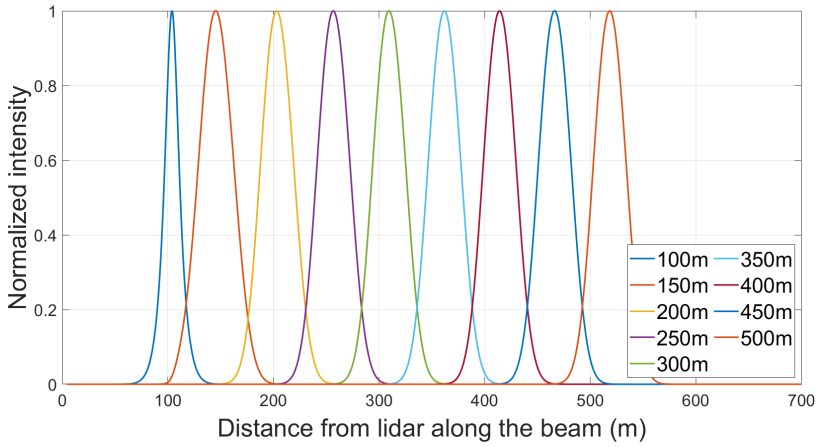


Figure 4.9: Weighting functions for a pulsed lidar corresponding to different ranges.

When the range-gate center matches the focal length, the convoluted pulse shape and the Gaussian intensity profile reach their maximum at the same location: at the beam waist. When considering particles at a range far from the focal length, the pulses still reach them but the intensity of the Gaussian beam is far lower at that point. At some point, the intensity profile of the Gaussian beam does not contribute anymore to the value of the range-weighting function, which is why only the convoluted pulse shape is often considered. As can be seen, the shape of the weighting function is more conserved as the probing distance increases than in the CW case.

4.1.2. Simple test cases and verification

This Subsection has been conducted with a range-weighting function that corresponds to the case of a CW lidar. It should be seen as a verification of the simulator, that can accept any type of range-weighting function. Results are still relevant in the case of a pulsed lidar, especially on the filtering effect in the spectrum. What will be used in the simulations however is based on the range-weighting function defined in the pulsed case. The pulsed range-weighting function typically filters out less of the turbulence, as one can see it reaches 0 more quickly than the Lorentzian profile.

Power law, no turbulence, no motion

Now that the lidar modelling has been introduced, it is possible to run some simple test cases. The first will be to generate a simple wind field without turbulence, determined by the power law with a shear exponent of 0.12, and a wind speed of $10\,\mathrm{ms}^{-1}$ at $100\,\mathrm{m}$. The objective is to model the VLOS measurements of the lidar and to see how the number of uniformly distributed axial and radial points along the beam impacts the measurements. The wind velocity only has stream-wise component and is shown as a function of height in Figure 4.10.

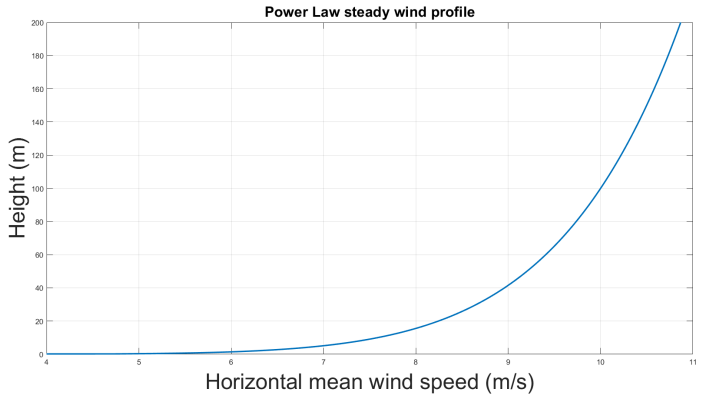


Figure 4.10: Example of a simple wind field with only horizontal component depending on height.

The lidar is simulated as facing the wind, at a height of $120\,\mathrm{m}$ and with the configuration stated earlier. The VLOS for one of the upper beams at a distance of $100\,\mathrm{m}$ along the beam axis is studied. The corresponding height of the range-gate center is then around $128\,\mathrm{m}$. The VLOS measurements were calculated for an increasing number of axial points to average around (according to Equation 4.1), compared to the value of the point-wise approximation. In this case, no radial points are considered, only particles along the beam. The result is given in Figure 4.11. As expected, with only one axial point, the value matches that of the point-wise approximation. Adding more points, the value then gets closer to real value of the weighted integral, where it seems that using more than 50 axial points only changes marginally the value of the simulated VLOS measurement. Note that in this case, even the point-wise approximation yields a close result to the converging value of the sum, with a difference of less than $1 \times 10^{-3}\,\mathrm{ms}^{-1}$.

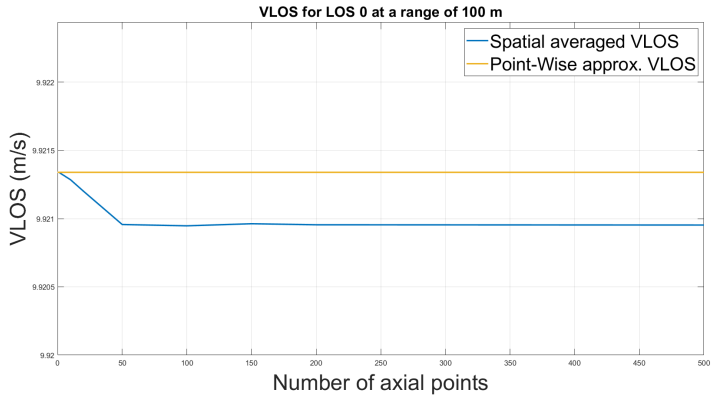


Figure 4.11: Evolution of the VLOS approximation as the number of axial points increase.

Then, fixing the number of axial points to 100, the impact of increasing the number of radial points was considered. The number of radial points is always set to be the square of an integer, see Figure 4.12. As can be seen in this case, the oscillations of the measured value stop after using more than 49 radial points (7×7). It is clear that there is trade-off between accuracy and computational time. Indeed, ignoring the radial spread of the beam causes an error of less than $1 \times 10^{-3}\,\mathrm{ms}^{-1}$ compared

to using 225 radial points, which will drastically increase the computational time. Note that this is just an example and that in real wind fields, because of turbulence the wind velocity fluctuations will be larger and more information will be lost by taking into account less particles. From now on, a typical number of sampling points used will be 100 axially and 25 radially for each range gate of each beam. For computational purposes, the number of radial points will possibly be reduced to 9.

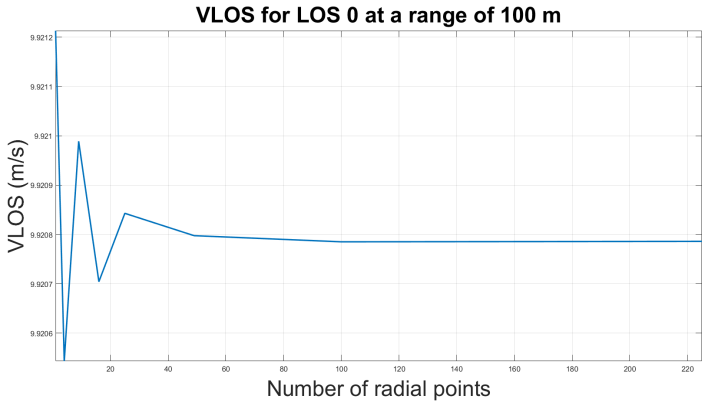


Figure 4.12: Evolution of the VLOS approximation as the number of radial points increase. Number of axial points fixed to 100.

Power law, turbulence, no motion

Then, still without motion of the lidar, the VLOS in a turbulent wind field will be calculated. In this case, spectral study will also be conducted. The wind field parameters are defined as follows :

- Shear exponent $\alpha = 0.1$
- Reference wind speed $U_{ref} = 10\,\mathrm{ms}^{-1}$ at $120\,\mathrm{m}$
- Turbulence intensity $TI = 5\%$
- Integral length scale $L = 35.4\,\mathrm{m}$, scaling factor $\alpha_\epsilon = 1.0$, spectral tensor parameter $\Gamma = 3.9$ and advection speed of U_{ref}
- Turbulent box dimensions: $\sim 1600\,\mathrm{m}$ x $290\,\mathrm{m}$ x $290\,\mathrm{m}$ (8192 x 64 x 64 grid points)

The simulation was ran for 30 minutes at $1\,\mathrm{Hz}$ for a lidar placed at the height of the reference wind speed ($120\,\mathrm{m}$), with the parameters already mentioned in the previous section. The box repeats in that time so periodic behaviours are expected to be seen, in the time series and in the spectrum for low frequencies. The resulting VLOS measurements for an upper beam with spatial averaging at a focus of $200\,\mathrm{m}$ along the beam axis are shown in Figure 4.13 in red. They are shown against point-wise approximated measurements in dotted blue, that is if the laser would only illuminate one particle. The difference is clear, and the spatially averaged VLOS shows less fluctuations, which is what is meant by the fact that lidar measurements "average out" turbulence. For indicators such as turbulence intensity, this causes an artificially lower value if it is computed using the fluctuations in the raw VLOS measurements. For these 30 minutes, the Mean Error between the point-wise approximation and the spatially-averaged quantity is of around $4 \times 10^{-3}\,\mathrm{ms}^{-1}$, which brought back to the mean is less than 0.1%. However, the Root Mean Square Error (positive/negative errors do not cancel out) is of around $0.3\,\mathrm{ms}^{-1}$, which is not as evidently negligible. As far as linear correlation goes, the relationship between the two signals (point-wise vs. spatially averaged VLOS) is reasonably high but not evident, due to the complexity brought by the convolution with the weighting function. Indeed the two time series show a linear correlation coefficient of around 0.833.

The result of spatially averaging the VLOS measurement with 100 x 5 x 5 points was compared to using more axial points (200 x 5 x 5). Then it was also compared to generating only axial points (100 x 1 x 1), with a different number (10 x 1 x 1 / 250 x 1 x 1 / 500 x 1 x 1) and to using the same number of radial points but more or less axial points (200 x 5 x 5 / 10 x 5 x 5 / 1 x 5 x 5). The results are

summarized in Table 4.1. This table gives insights on how to distribute the sampling points. Typically, more axial points are better, but as it was addressed earlier, after a certain limit, using more points does not help in accuracy. And, although it is a common assumption in lidar theory, the contribution of particles that are not along the beam axis in a turbulent atmosphere is not negligible. Given that the result is better for 10 axial points and 25 radial points than for 250 axial points and 1 radial point, it is reasonable to state that radial particles can not be evidently neglected in real simulations. Given that the simulated VLOS is anyway an approximation of reality, one can still argue that generating axial particles is enough.

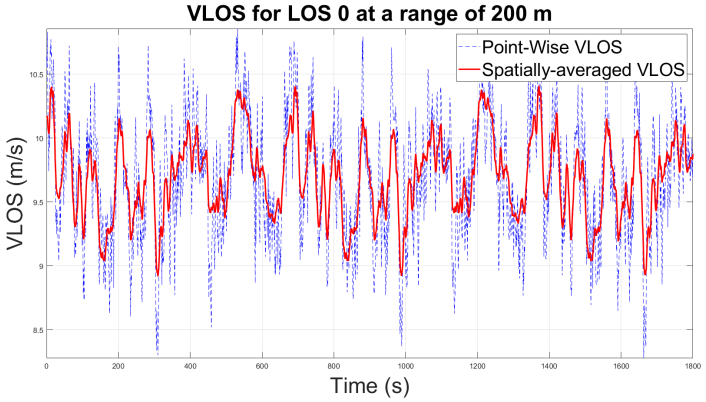


Figure 4.13: Effect of the spatial averaging of the lidar (100 axial, 25 radial sampling points) compared to the point-wise approximation for an example of turbulent wind field.

Point cloud config. (Axial x radial x radial)	Mean Error	RMSE	Linear correlation coeff.
1 x 1 x 1	3.8×10^{-3}	0.282	0.833
10 x 1 x 1	1.1×10^{-3}	0.146	0.933
100 x 1 x 1	1.8×10^{-3}	0.156	0.937
250 x 1 x 1	1.9×10^{-3}	0.154	0.938
500 x 1 x 1	1.9×10^{-3}	0.154	0.938
1 x 5 x 5	3.8×10^{-3}	0.282	0.833
10 x 5 x 5	5.0×10^{-4}	0.068	0.976
200 x 5 x 5	7.8×10^{-5}	0.006	0.999

Table 4.1: Comparison of the VLOS time series with different point cloud configurations using different indicators to a reference point cloud. This reference point cloud is made of 100 axial points, and 5^2 radial points. This reference VLOS time series is that in red in Figure 4.13. Mean Errors and RMSE are expressed in ms^{-1}

Next, the spectrum was studied. From the discussion in Subsection 2.5.4, an inertial subrange should appear in the spectra, and it should be clearer in the point-wise approximation given that the spatial averaging is not taken into account. The spectra in Figure 4.14 show well how the lidar measurements ”filter out” the capture of turbulence at higher frequencies. The -5/3 slope appears for the point-wise approximation (see Equation 2.59), whereas for the spatial average estimate (closer to the reality of the measurements), the -5/3 slope can only be considered for a short frequency range, after which it gets steeper. This is coherent will literature findings [65].

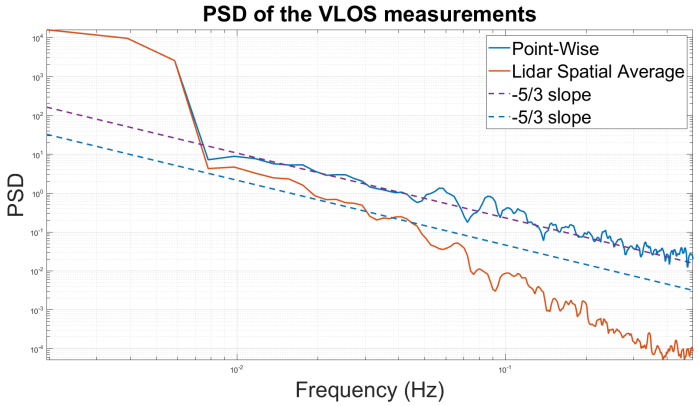


Figure 4.14: Comparison of the Power Spectral Densities of point-wise lidar measurements and realistic lidar measurements with a point cloud distribution of 100 x 5 x 5.

To highlight in another way the effect of the spatial averaging on the lidar VLOS spectrum, it will be investigated if it possible to find the ”real” VLOS spectrum from the artificial point-wise spectrum and the weighting function. To simplify this study, only one dimension will be considered. Modelling-wise, this means generating points only along the axis of the beam. Then, Equation 2.13 will be rewritten to create a convolution in time of the signal represented by the projected wind velocity with a filter, i.e. the weighting function. Ignoring the radial spread of the beam, only points along the beam will contribute to the VLOS measurements, and with the speed of light being known, each point in space can be mapped into a point in time. Thus, the VLOS equation can be written as:

$$\overline{v_r}(R) = \overline{v_r}\Big(t = \frac{2R}{c}\Big) = \int_{-\infty}^{+\infty} W\Big(\frac{2R}{c} - t\Big)\, \boldsymbol{n} \,.\, \boldsymbol{v}(t)\, dt$$

where the weighting function W reaches its maximum at $t = \frac{2R}{c}$. This is clearly a time convolution, and in signal processing terms, the VLOS measurement corresponding to the range R is approximately an output of a Linear Time-Invariant System, where the impulse response is the range-weighting function. The power spectrum of such an output is then simply the product of the power spectrum of the signal being convoluted and the square of the absolute value of the Fourier Transform (or periodogram) of the impulse response function [61] [58]. The corresponding formula is:

$$S_{\overline{v_r}}(f) = S_{PW}(f)\, |\widehat{W}(f)|^2 \tag{4.4}$$

where $\widehat{W}(f)$ is the Fourier transform of the weighting function and S_{PW} is the power spectrum of the point-wise VLOS. In that sense, the convolution of the signal of the projected wind velocity with the range weighting function has, in the frequency domain, the effect of a filter with respect to the point-wise measurement signal. This principle is the basis of some lidar deconvolution algorithms [35]. Because the range-weighting function can be known, then the term $|\widehat{W}(f)|^2$ can be estimated. This is also possible straightforwardly because only points along the beam are considered, thus the range-weighting function is a function of only one variable. For the rest of this section, this approximation will be considered. Even if it was demonstrated that it does not show exactly the real results of spatial averaging, it remains an acceptable approximation that is very often used in literature.

Note that this is dependent on the type of lidar and the range-weighting function that best models the measurement. For example, this filter function is dependent on the Rayleigh length of the Gaussian

beam. Also, because the integral in the VLOS measurements is estimated by a sum, the mapping between the spectra introduced by Equation 4.4 will not be perfect. Now what is needed is the Fourier transform of the range-weighting function. For this study and to simplify, again only the dimension along the axis will be considered.

Then, one needs a correspondence between the measurement time of the point-wise measurement and that of the weighting function to get the corresponding spectra. Luckily, a mapping exists from space to time domain for the velocity measurements ($t = \frac{x}{U}$) where x is a distance in the stream-wise direction of the wind and U is the advection speed of the turbulent vertical slices (Taylor's hypothesis). And the value of the weighting function can be calculated in space, keeping this space-time correspondence. Thus, since the VLOS are sampled every second and U was chosen to be $10\,\mathrm{ms}^{-1}$, that corresponds to a displacement of $10\,\mathrm{m}$. Along the beam that is a distance of $10/\cos(\theta)$. For the reference WindCube Nacelle, this is around $10.39\,\mathrm{m}$. This is the spatial step (that also corresponds to a time step) at which the value of the range-weighting function can be sampled to get its Fourier transform using a FFT algorithm. This gives an idea of how the frequency domain of the spectrum of the range-weighting function can relate to that of the VLOS measurements. The result of the periodogram of W is shown in Figure 4.15. Remarkably, this has a similar effect to a low-pass filter, which again demonstrates how a lidar filters out the capture of some turbulence effects.

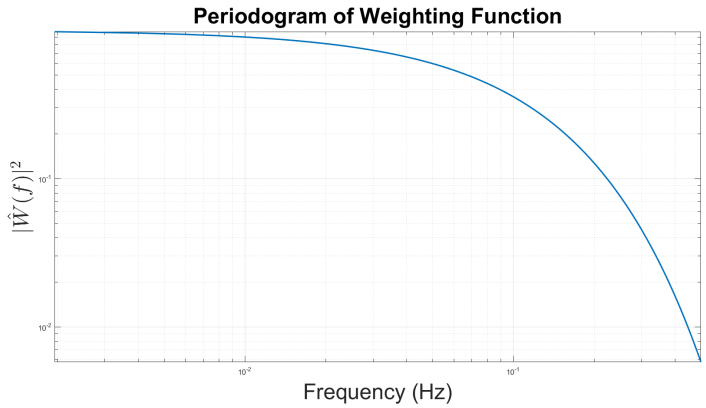


Figure 4.15: Square of the absolute value of the Fourier Transform of W with the current beam parameters.

What this means is that because of Equation 2.59, the point-wise spectrum should be obtainable from that of the spatially averaged measurement, by simply dividing frequency-wise the spectrum of the spatially averaged VLOS by the periodogram of the weighting function. The simulated lidar VLOS spectrum with only axial points was divided by $\widehat{W}(f)$ and the result is shown in Figure 4.16. As can be seen, the -5/3 slope appears again clearly in the "reverse-filtered" spectrum. Since the periodogram of the range-weighting function was determined for points only in the beam direction axis, this study can only be made simulating particles solely along the beam.

To conclude, this subsection is meant to show the results of the pulsed lidar simulator's measurements in a modelled turbulent atmosphere and to show that it has expected properties. It also serves as a baseline comparison (non-moving lidar). Again, this is only valid when neglecting the contribution of the particles that are not along the axis of the beam. Thus, for simulations that will take into account radial particles, even without any motion, one cannot expect to find the same behaviour in the spectra.

Power law, turbulence, lidar velocity interference

It was shown in the previous subsection that even if the spectrum of the real lidar VLOS measurements does not follow the theoretical slope of the wind velocity components' spectra, it is still linked to it by a filter that depends on the range-weighting function when only axial particles are considered. Now, the following study will be conducted : a small (relative to the mean wind speed) velocity interference of the lidar is introduced. The goal is to see how that can affect the VLOS measurements and their spectrum. The change in the beam direction (by the lidar rotating) in formula 2.15 is for now ignored. The idea behind this study is to see what magnitude of lidar velocity interference will impact the lidar measurements up to the point where not accounting for the lidar velocity will cause a too large error in

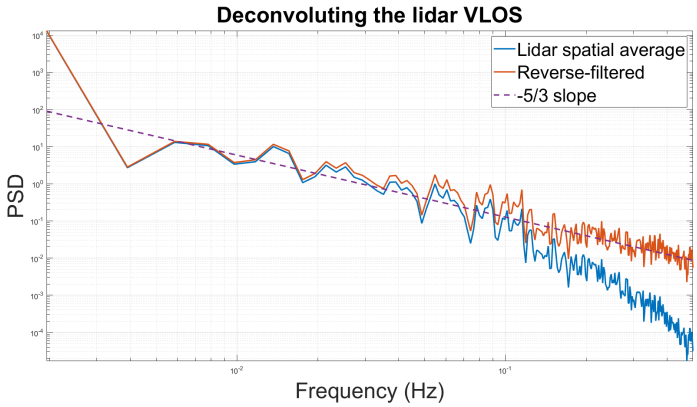


Figure 4.16: Dividing the spectrum of the spatially-averaged by $\widehat{W}(f)$ as a way to get back to the point-wise spectrum. The spectra are not smoothed and again, only points along the beam are considered (500 x 1 x 1).

wind field reconstruction. This will also give an insight of how accurate the velocity of the lidar must be known (i.e. their uncertainty) for different levels of accuracy in the wind field reconstruction. For the wind field reconstruction, different levels of fidelity can be considered:

- Low level : The goal is to reconstruct only the mean wind speed $U(z)$ for a given time period (i.e. 10 minutes).
- Mid level : The goal is to reconstruct the mean wind speed $U(z)$ and to have a reliable estimate of the turbulence intensity TI within the averaging time period.
- High level : Here, one tries to reconstruct up to the lowest scales of turbulence. In other words, it is assumed that the lidar measurements come from a turbulent atmosphere with its known statistical properties. Typically, this involves considerations on the velocity correlation function and the Doppler spectra of the lidar measurements. Such a reconstruction respects the fact that lidar measurements come from a wind field with velocity components that obey Kolmogorov-Obukhov law in the inertial range [2].

The measurements with interference of the lidar velocity will be compared to that with no interference (fixed lidar). For the mean wind speed, the criterion will be simply be that of the mean of the VLOS. For turbulence intensity, the criterion will be the standard deviation of the VLOS measurements divided by its mean for 10-min periods. Even if moving lidars tend to overestimate Turbulence Intensity, algorithms exist to correct this [33]. Finally, for most developed turbulence reconstruction, the criterion will be the spectrum of the VLOS measurements, and similar studies as were done in the previous subsection.
Unlike the rotation of the lidar that can be, as a first approximation, considered with harmonic oscillators, no similar assumption can easily be made on the behaviour of the lidar's interfering velocity. To remain general and as a similarity to what was found in the real velocity time series, the lidar's velocity is made to resemble Gaussian white noise for all three components. In this case, the velocity components are all normally distributed with a zero mean. Depending on the standard deviation of the distribution, the fluctuations are more or less wide. Figure 4.17 shows the difference in measured VLOS between a lidar with no interference and one with an interfering velocity, the distributions of which have two different standard deviations. The distributions are also truncated such that the norm of each velocity component does not go over a certain threshold.

For three 10-min periods and for velocity interference cases with different standard deviations (and truncature), the results for the mean VLOS are shown in Table 4.2 and for estimated Turbulence Intensities in Table 4.3.

As it is displayed, the impact of such lidar's velocities on the mean of the VLOS is negligible, even for the largest interference. Regarding Turbulence Intensity, as it was expected, for large interference (i.e. more motion), it is overestimated, but that remains a slight increase. Next, the impact on the spectrum was studied. The PSD spectra were reconstructed for time series with and without interference of the lidar's velocity, and are displayed in Figure 4.18.

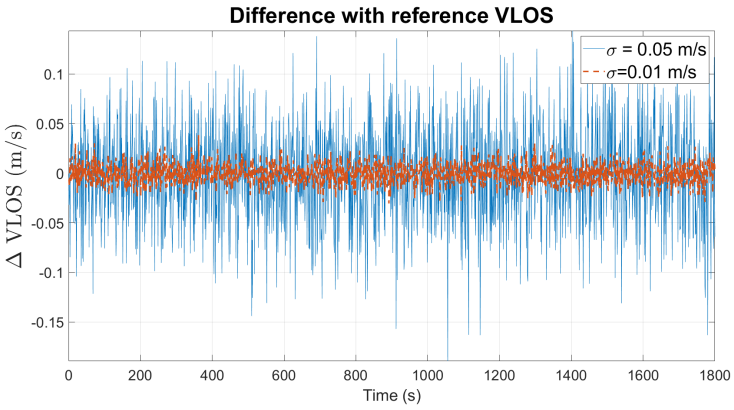


Figure 4.17: Interference of the lidar's velocity for a VLOS time series (LOS 0 at a range of $200\,\mathrm{m}$). Components follow normal distributions.

	0-10	10-20	20-30
No motion	9.71	9.65	9.72
$\sigma = 0.01$ ($\|.\| < 0.3$)	9.71	9.65	9.72
$\sigma = 0.05$ ($\|.\| < 0.3$)	9.71	9.65	9.71
$\sigma = 0.1$ ($\|.\| < 0.3$)	9.71	9.64	9.71
$\sigma = 0.5$ ($\|.\| < 1$)	9.69	9.68	9.71

Table 4.2: Mean VLOS for different velocity interference cases.

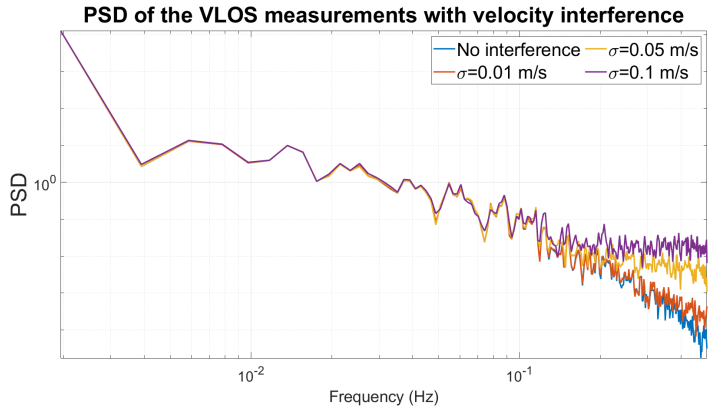


Figure 4.18: Interference of the lidar's velocity on the spectrum (for LOS 0 at a range of $200\,\mathrm{m}$).

The conclusion drawn from this figure is important. It shows that even if an interfering velocity has barely any impact on the mean measured wind speed and Turbulence Intensity (consider for example $\sigma = 0.1\,\mathrm{ms}^{-1}$), it can heavily disrupt the spectrum of the VLOS measurements. This shows that the information on turbulence contained in the VLOS time series (especially for small scales) can be destroyed by the velocity interference of the lidar, even though this influence does not appear on mean

	0-10	10-20	20-30
No motion	0.04	0.04	0.04
$\sigma = 0.01$ ($\|.\| < 0.3$)	0.04	0.04	0.04
$\sigma = 0.05$ ($\|.\| < 0.3$)	0.04	0.04	0.04
$\sigma = 0.1$ ($\|.\| < 0.3$)	0.04	0.04	0.04
$\sigma = 0.5$ ($\|.\| < 1$)	0.06	0.06	0.06

Table 4.3: Fluctuations of the measured VLOS (estimated Turbulence Intensity) for different velocity interference cases.

quantities.
For the highest physical level of reconstruction, given how the measurements' spectrum can be disturbed even with small interference of the lidar's velocity, it seems impossible to reconstruct a theoretically accurate description of turbulence. As the interfering velocity here is akin to noise, this also demonstrates that when reconstructing turbulence from VLOS measurements, one should also be particularly careful of the inherent noise contained in the VLOS measurements (Signal-to-Noise Ratio or Carrier-to-Noise Ratio). This noise can come for example from parasite scatterers that do not move with the speed of the wind. Next, the effect of the lidar's rotation in a simple case will be studied.

Power law, turbulence, pitch motion

The objective of this part is to see how a theoretically simple rotation motion of the lidar affects the measurements, and to verify how a frequency-filtering method can be used, in different configurations of wind. Only the pitch angle of the lidar will be modified, and this angle is considered a harmonic oscillator, as it was done in literature. An example of this pitch angle is given in Figure 4.19. Since the lidar measurements are made to be sampled at $1\,\mathrm{Hz}$, the pitch angle that the lidar "sees" at the time of the measurement is also shown, close to the continuous angle. This is a modelling hypothesis, assuming that the lidar probes a "frozen" wind field, neglecting the fact that one measurement comes from different backscattered intensities, received at different times.

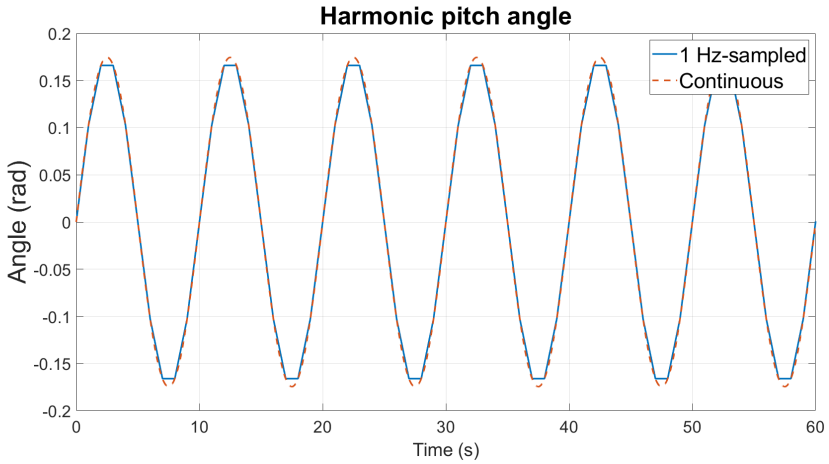
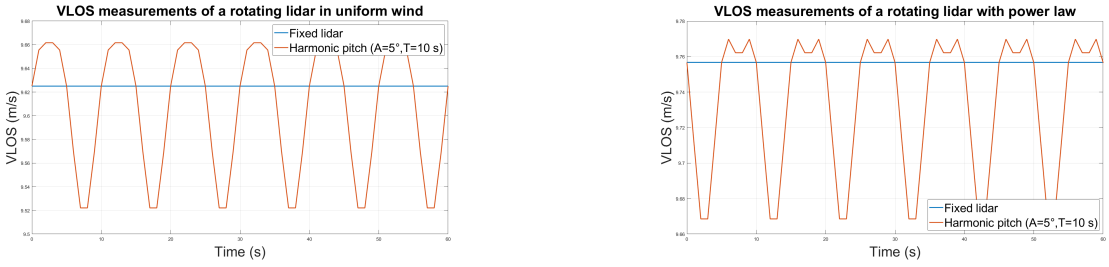


Figure 4.19: Time series of harmonic pitch angle with a period of $10\,\mathrm{s}$.

To start with, simpler wind fields (without turbulence so constant in time) will be considered: uniform wind and power law profile. The impact for $60\,\mathrm{s}$ of harmonic pitch of the lidar with an amplitude of $5°$ and a period of $10\,\mathrm{s}$ is shown in Figure 4.20. For uniform wind (Figure 4.20a), the VLOS measurements have a clear periodic behaviour, the change in value is purely due to the change of the beam direction. The maximal VLOS is attained when the beam direction is best aligned with the wind direction. For a power law profile (Figure 4.20b), the shape of the VLOS time series with harmonic pitch of the lidar is more skewed. This comportment can be explained by the fact that even if the line-of-sight is less aligned with the direction of the wind velocity, the value of the wind speed can be higher due to the power law profile, thus compensating the value of the dot product.



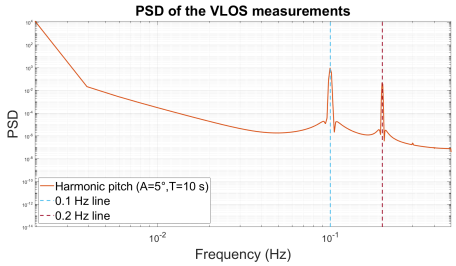
(a) Uniform wind.

(b) Power law profile.

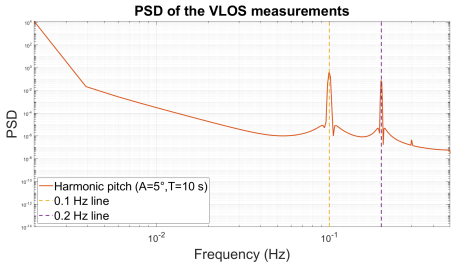
Figure 4.20: Effect of harmonic pitch motion of the lidar on the VLOS measurements in two simple cases.

Such periodic behaviours have been highlighted in literature for real cases of floating lidars [34]. As it was done there, one can then look into the spectrum of the VLOS measurements. The resulting spectra are shown in Figure 4.21. Whether that is for uniform wind or a power law profile, the frequency of the harmonic oscillator is clearly visible at $0.1\,\mathrm{Hz}$, and even at twice this frequency at $0.2\,\mathrm{Hz}$.

Applying a filter is tempting to attempt to remove the periodic effect in the VLOS measurement caused by the pitch motion. Here, a simple low-pass filter can be used and the resulting filtered spectra



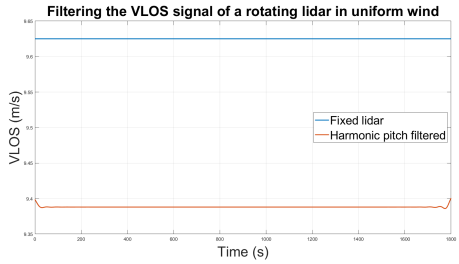
(a) Uniform wind.



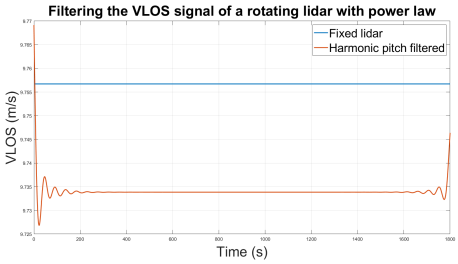
(b) Power law profile.

Figure 4.21: PSD spectra of the VLOS measurements with harmonic pitch motion of the lidar of the measurements in two simple cases.

are shown in Appendix in Figure A.14.



(a) Uniform wind.



(b) Power law profile.

Figure 4.22: Filtering out the effect of harmonic pitch motion of the lidar on the VLOS measurements in two simple cases.

The time series of VLOS after the filter is applied are then displayed in Figure 4.22. As shown, albeit transient effects of the filter at the beginning and end of the signal, the periodic behaviour is removed and the VLOS is closer to what it should be. There is still an offset, which is most likely due to the filter being inappropriate. Nevertheless, this project does not aim at diving into signals and filter theory. The aim is simply to show how frequency filtering can correct for the rotation of the lidar.

Now that the simple cases have been addressed, one can see what happens when a turbulent field is introduced. The same field as used previously is considered. The same time series for the VLOS measurements is generated as for the two previous cases, in Figure 4.23 The periodic effect is here much less obvious, due to turbulence causing fluctuations as well.

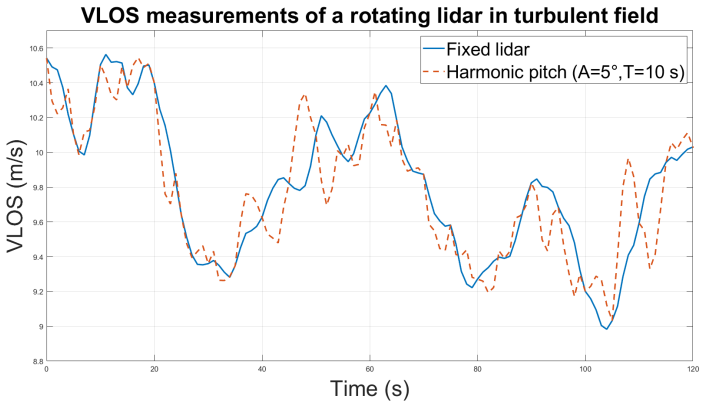
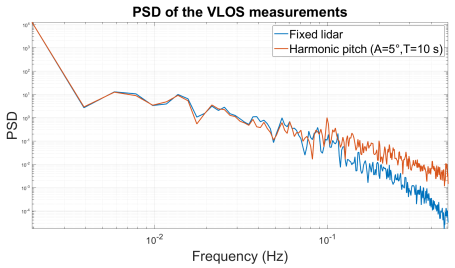


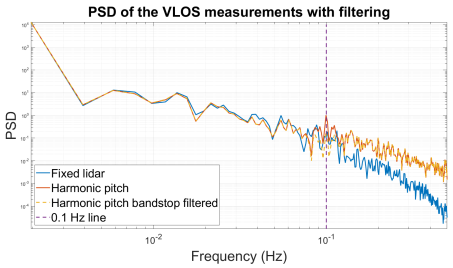
Figure 4.23: Effect of harmonic pitch motion of the lidar on the VLOS measurements in a case of turbulent wind field.

A small peak at $0.1\,\mathrm{Hz}$ still appears in the spectrum of the measurements in Figure 4.24a. However, this seems to get lost in turbulence. Also, for lower amplitudes of harmonic pitch, this peak does not even show up. An example of spectrum with a higher amplitude of pitch ($10°$) is shown in Appendix, Figure A.15. It would be dangerous to apply a low-pass filter since the effects of turbulence at higher

frequencies might be lost. Thus, a bandstop frequency filter was applied around $0.1\,\mathrm{Hz}$, giving the spectra shown in Figure 4.24b.



(a) PSD spectrum of the VLOS measurements with harmonic pitch motion of the lidar of the measurements in a case of turbulent wind field.



(b) Bandstop filtered PSD spectrum of the VLOS measurements with harmonic pitch motion of the lidar of the measurements in a case of turbulent wind field.

Figure 4.24: Effect of the bandstop filter on the spectrum of the VLOS measurements with turbulence.

A zoom on the resulting filtered VLOS time series is shown in Figure 4.25.

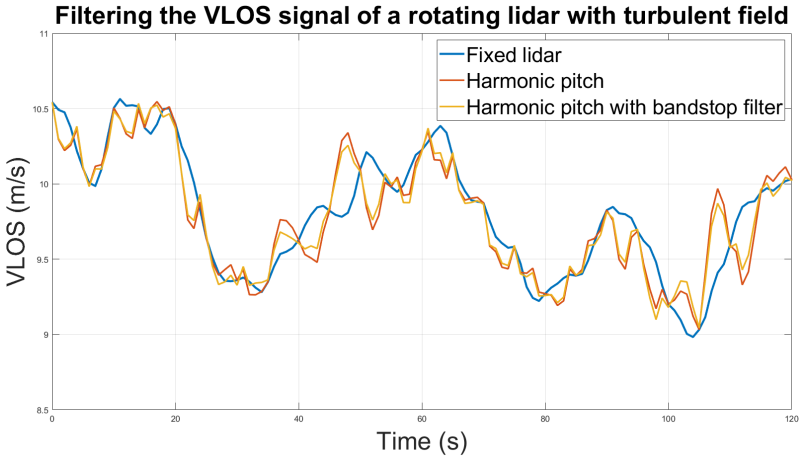


Figure 4.25: Filtering out the effect of harmonic pitch motion of the lidar on the VLOS measurements in a case of turbulent wind field.

For this turbulent wind field, the results of the filter on the mean VLOS and on an estimate of Turbulence Intensity are given respectively in Table 4.4 and Table 4.5.

	0-10	10-20	20-30
No motion	9.71	9.65	9.72
Harmonic pitch	9.69	9.62	9.69
Harmonic pitch filtered	9.69	9.62	9.69

Table 4.4: Mean VLOS with and without bandstop filter around the pitch natural frequency, in a turbulent wind field.

	0-10	10-20	20-30
No motion	0.04	0.04	0.04
Harmonic pitch	0.04	0.04	0.04
Harmonic pitch filtered	0.04	0.04	0.04

Table 4.5: Fluctuations of the measured VLOS (estimated Turbulence Intensity) with and without bandstop filter around the pitch natural frequency, in a turbulent wind field.

It appears that with turbulence, such a rotation has a negligible impact on the mean VLOS and even on its standard deviation, even though this amplitude of pitch was considered an extreme case in literature [34]. Moreover, the application of the filter seems to have no effect on both quantities.

In conclusion, this part gives some limitations on the frequency filtering method, which are as follows:

- The choice of the filter is not straightforward. Initial conditions can be required and precaution is needed when filtering measurements with turbulence, since the real turbulent fluctuations might be lost.
- For a peak to appear on the spectrum, the amplitude of the rotation angle needs to be large enough. And that is when only considering a simple harmonic pitch.
- Even after filtering, the corrected VLOS time series can still be too far off from the desired one.

Power law, turbulence, full DOF

Here, the goal is to simulate the full motion of a lidar in a turbulent atmosphere and its resulting measurements if it is placed on the nacelle of a floating turbine. For isolated and simple motion like what was discussed before with only translational velocity or only pitching motion, it seemed acceptable to assume some kind of motion, based on simplifying assumptions. This was meant as a proof of concept to show how impactful the lidar motion can be on the measurements. Now that the requirement is to have full degrees of freedom activated, the best would be to have the motion at the nacelle coupled with the wind field (and waves for a FOWT). This is far from straightforward considering that requires simulating the dynamic response of the turbine.

Softwares exist to do so, and the one used in this project is OpenFAST, an industry and research standard. Then, the conducted plan is the following. First, a turbulent wind field is generated. With OpenFAST, this wind field is used as an input in the InflowWind file, together with wave conditions and structural properties of the simulated FOWT, so that this incoming wind results in some motion at the nacelle of the turbine. The calculation is made internally with OpenFAST, and no attempt will be made in diving in the functioning of this aero-hydro-servo-elastic simulation software. Then, this same wind field is used as an input of the lidar simulator, with which it is possible to simulate the measurements at several range gates for all beams. The aim is then to see how the motion at the nacelle would theoretically affect the lidar VLOS measurements, and how the correction method implemented here (compensating for the lidar velocity interference using interpolation methods) can perform with noisy motion signals.

When doing so, one significant challenge appears: replicating the same wind field for the lidar simulator (internal Siemens Gamesa module) as in OpenFAST. To ensure that the motion outputs can indeed be considered as coming from the same field in both cases, a study was made on how to make the fit. It is rather straightforward to test this since OpenFAST and the Siemens Gamesa wind module can take as input the same binary file for the Mann turbulent box (3 binary files for u, v and w components if the wind). These files can be scaled (to get to a certain Turbulence Intensity for example) with Matlab using the wind module and then fed to OpenFAST as inputs for the wind field. Now the difference can be in the interpolation method between the grid points of the turbulence box. If one uses the default methods for interpolation in this internal wind module and in OpenFAST, the result is as follows in Figure 4.26. The wind speed shown is the U component (mean wind + turbulence) at the center of the box, only for a part of simulation, to highlight the difference in the two time series.

The origin of the box was set at a height of $150\,\mathrm{m}$. The size of this box used for testing is around $6800\,\mathrm{m}$ in the x-direction (wind direction), $290\,\mathrm{m}$ in the y and z-directions, with 8192 points in the wind direction and 64 in both the transverse and vertical direction. The steady wind profile is chosen to be represented with a power law ($12\,\mathrm{ms}^{-1}$ at $150\,\mathrm{m}$ with $\alpha = 0.2$). Since the advection speed inside of the box is also at $12\,\mathrm{ms}^{-1}$, this means that the actual simulation lasts slightly less than 10 minutes, after which the box repeats itself.

By default, OpenFAST uses a quadratic method in interpolation while the internal wind module uses Catmull-Rom splines (cubic). Can that difference explain the offset of the curves from Figure 4.26 ? For this, the simplest way of interpolation will be used in both cases: linear interpolation. In this case, accuracy is lost due to simplicity of interpolation but there is no ambiguity on the method. The result is shown for 10 minutes in Figure 4.27. The difference is now barely visible, and the root mean square difference between the two time series is of only $0.0033\,\mathrm{ms}^{-1}$ over this 10-minute period. Now it is reasonable to think that this error most likely comes from rounding errors for example, but that the wind field that is input to OpenFAST for the simulation of the turbine motion is the same that can be measured

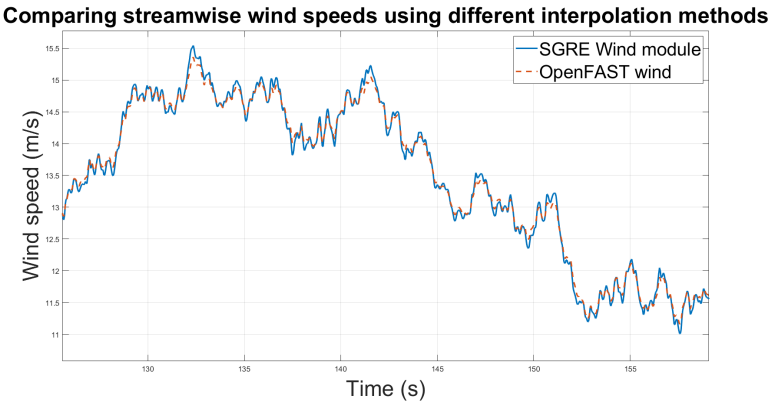


Figure 4.26: Time series of stream-wise wind components at the origin of the turbulence box for OpenFAST and SGRE (Siemens Gamesa Renewable Energy) internal module. The interpolation methods are kept to default and the difference is visible.

with the lidar simulator. Now that the foundation for comparison is established, it is possible to run these simulations with OpenFAST and use the motion outputs for corresponding wind measurements with a lidar in Matlab. This is done in the Results Section.

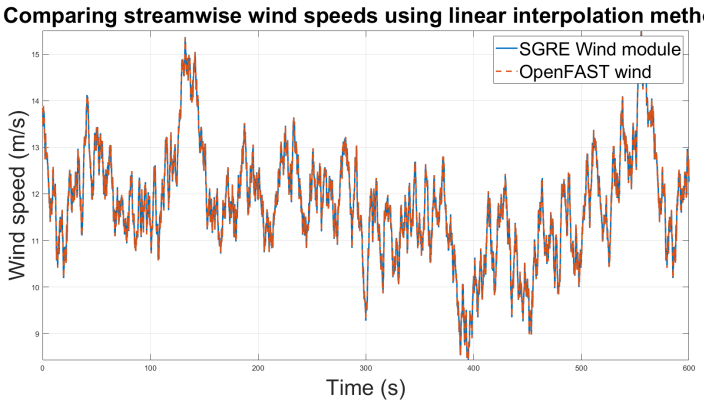


Figure 4.27: 10-min time series of stream-wise wind components at the origin of the turbulence box for OpenFAST and SGRE (Siemens Gamesa Renewable Energy) internal module. The interpolation method is made linear and the two curves cannot be distinguished anymore.

4.2. Uncertainty quantification

In this section, the goal is to introduce the first results on uncertainty, that is knowing the uncertainty of the different models that characterize motion characteristics. For instance, it is necessary to know the standard error of the parameters in the interpolated curves of the acceleration components. One also wants these estimates for velocity, position, orientation and angular velocity. Again, no real discussion is possible on angular acceleration since this is not an output of the IMU. These errors are modelling errors, meaning they quantify the expected error when using a certain model to fit a certain quantity. On top of these errors, there are also measurement errors, or noise in signal processing, which come from the fact that data are not generally not exact. These measurements can for example be given by the manufacturer of the measuring device. Uncertainties should be seen as ”errors” that are inevitable. These errors concern data from the measurement but also model parameters for the fitting of the data, for example that could be the uncertainty in the control points of the Quintic Hermite interpolation. The goal is ”to ensure that these errors are as small as reasonably possible, and to have some reliable estimate of how large they are” [78].

4.2.1. Example of model parameters' uncertainty for acceleration

For interpolation of acceleration, data has to be fit to some kind of model, linking the time t to an observation $y(t)$. For position, it was argued that using Quintic Hermite interpolation is a viable option since it allows for low error and for continuity of the second derivative (i.e. acceleration). Now any model has parameters, and these parameters have to be determined from the available data by some kind of fitting. This is usually done using least-squares fit. In the case of position, the model can be expressed as a linear combination between terms depending on t and constant terms that depend where the nodes are located, namely *control points*.

Then, as it was discussed in subsection 2.4.1, the problem can be written as a matrix product $A\,X = B$, where B is the vector of the observations, or real values measured by the IMU. B could be containing in this case values of position, linear velocity of linear acceleration. A is called the *design matrix*. For position, A is made of the values of the Quintic Hermite basis functions at the times corresponding to the observation. For linear position and acceleration, A is composed of the first, and second respectively, derivative values of the basis functions. Finally, X is the vector made of so-called control points. After choosing the right nodes, X can be found by least squares fit for example and it will be the same vector for position, velocity and acceleration. What is interesting for this subsection is that the components of the X vector are subject to uncertainties. Indeed, it is not always possible to find the exact solution to the fitting problem. Thus, depending on the positioning of the nodes, which changes the design matrix so the problem, the problem can be solved with more or less uncertainty. The uncertainty in the estimated control points can be stored in what is called a **covariance matrix**, that will be written P in what follows. What is particularly useful is the diagonal elements of P, which are the variances (squared uncertainties) of the fitted parameters (control points) in X [66]. The vector with the square root of these diagonal elements will be written Σ:

$$\Sigma = \begin{bmatrix} \sigma_{0,0} \\ \sigma_{0,1} \\ \sigma_{0,2} \\ \sigma_{1,0} \\ \vdots \\ \sigma_{n-1,2} \end{bmatrix}$$

where, with n nodes, $\sigma_{i,j}$ is the uncertainty of the control point corresponding to the $(j+1)$-th point at the $(i+1)$-th node. From Quintic Hermite interpolation theory, one should bear in mind that there are 3 control points per node so j can be 0, 1 or 2. It is fundamental to know the uncertainty of these parameters since they will propagate in the whole model of wind speed reconstruction. How to determine them from the fit?

The matrix P is easily obtained with Matlab when X is fitted using the standard optimal rule for the least squares method [77]. It is typically computed from the inversion of the product of the transpose of the design matrix with the design matrix itself. To illustrate this better, some examples will be shown for acceleration. The first example is the one where the interpolation time nodes are placed at the exact

same time as the real acceleration measurement times. Because of this and due to Quintic Hermite interpolation theory, for acceleration it means that the control points corresponding to the third point of each node will be exactly the measured acceleration. In this case, in practice there is not even a need to do the fitting, the control points are exactly known. For the sake of the exercise, it is however possible to calculate the uncertainties from the least squares fit, with the design matrix having in this case a very simple form with only 0 and 1 values. This is because the interpolation nodes are placed at the exact same time as the measurements. The reconstructed fitted curve in this case is a polynomial curve of third order, continuous but non-differentiable at the nodes. Position would be of fifth order while velocity would be of fourth order, both being continuous and differentiable everywhere. It is shown for a certain time window in Figure 4.28. Time and acceleration values were normalized, they do not matter much to understand.

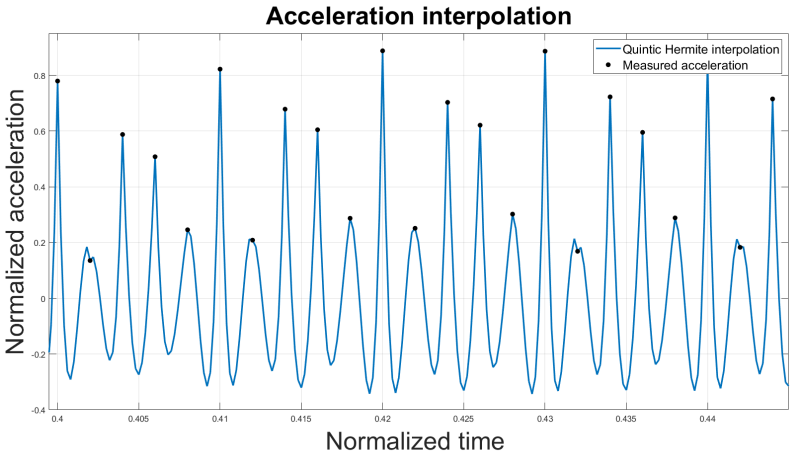


Figure 4.28: Example of interpolation with the nodes at the measurement times. No uncertainty on the control points.

It should not be surprising that for this first case, the uncertainties of each control points are 0 when calculated with Matlab:

$$\Sigma_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

It is good that in this case, the model is determined with no uncertainty. However, there remains a problem. It is the fact that interpolating as such conserves the noise/uncertainty that are intrinsic to the measurements. This can be seen by a high standard deviation of the reconstructed signal (even higher than the measured signal), leading to an unlikely behavior of acceleration with extremely rapid changes between high positive values and high negative values. Another extreme example that can be shown simply is where there are only two nodes, with lots of measurement in between. In normalized time, that would mean one node at $t = 0$ and another at $t = 1$. In this case only 6 control points exist, 3 at the start and 3 at then end. An example of a resulting fitted curve is shown in Figure 4.29.

One can see that the problem of the high standard deviation of the interpolated acceleration is fixed but two problems appear. The first is obvious, it is the error between the interpolated quantity and what one should expect of the actual values. Indeed, the blue curves completely ”misses” the measured values for acceleration. It can be acceptable to allow some error in that regard to lower the noise in the reconstructed curve but there appears to be far too much error in this case. The second problem is that some of the 6 control points now have some uncertainty, due to the least squares fit. From Matlab, it comes that, for this second case, the normalized uncertainties of the control points are:

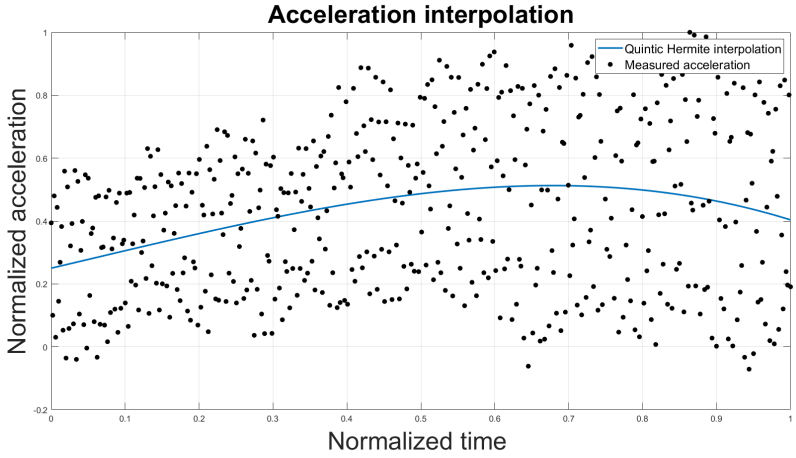


Figure 4.29: Example of interpolation with only 2 nodes. Low standard deviation of the fitted curve.

$$\Sigma_1 = \begin{bmatrix} 0.0066 \\ 0.0115 \\ 0.0455 \\ 0 \\ 0 \\ 0.0455 \end{bmatrix}$$

There is an optimal in-between to find. A trade-off needs to exist between three criteria:

- A low error between the real measured accelerations and the values of the fitted Quintic Hermite interpolation at the measurement times
- A low uncertainty of the control points
- A low standard deviation of the interpolated acceleration data

The problem of finding the control points using least-squares fitting only with acceleration data lies in the left-hand side matrix A of the least squares fitting equation. With Quintic Hermite interpolation, acceleration is calculated using the second derivative of the basis functions. And the values of the second derivative of the basis functions are typically larger than that of the basis functions themselves and their first derivative. This leads to large numbers present in A, increasing the condition number of the matrix and thus the numerical instability of the solving.
To relax this condition number, using also the values of velocity and position is better. In this case, for example the vector B can contain the real values of first position, then of velocity and finally of acceleration. Consequently, the matrix A will contain first the basis functions evaluated at the nodes, then their first derivative and finally their second derivative. Compared to fitting with only acceleration, instead of A having a size $M \times N$, it will have a size $3M \times N$. The same applies to B, with a size $3M \times 1$ instead of $M \times 1$.

A last problem appears to remain in this case, it is that data channels with different inherent uncertainties are used. The noise in acceleration is typically higher than for velocity and even higher than for position. As a rule of thumb, this can be understood by analyzing the spectral characteristics of noise. Consider for this the following result on the power spectrum of a signal: if a signal $x(t)$ has a power spectrum in the frequency domain (or angular frequency $\omega = 2\pi f$) given by $S(\omega)$, then the power spectrum of its n-th derivative is $\omega^{2n}\, S(\omega)$ [61].
For example, consider that the position signal contains Gaussian White noise, the spectrum of which is constant in the frequency domain. Then, the power spectrum of acceleration will be a quartic function in the frequency domain. For most frequencies, this increases the power spectral density of noise for acceleration compared to position. It thus comes that in general, the position channels are more reliable than the acceleration channels in terms of uncertainty caused by noise. The weights of each

signal used for least-squares fitting can then be adapted such that position has the larger weight, then velocity and then acceleration. This is done using weighted least-squares fitting.

4.2.2. Model parameters' uncertainty using position, velocity and acceleration

An example on how to estimate the uncertainty of the control points in the model to interpolate position, velocity and acceleration will be discussed in this subsection. It also shows general guidelines on how to interpolate these three channels from an inertial device.
The signals used here for position, velocity and acceleration of the INS originate from the Ship Motion reference frame, that is introduced in Appendix A.10. As discussed in that section, these outputs are not directly usable as local measurements in the body frame of the inertial device. Another difference with the outputs in the body frame is that position, velocity and acceleration in the ship motion reference frame are not results of the EFK. Velocity is obtained by simple integration of acceleration and position by double integration. The drifts due to integration are filtered out using a high-pass filter with Infinite Impulse Response. This is another reason why it is better to use the channels coming from the EKF in the local frame (or in the NED frame). At the time of this project, only acceleration was available in the body frame. Nevertheless, the results on the quantities from the ship motion reference frame should generally hold for any modelling of position/velocity/acceleration.
For a certain period of time in the month of May, surge, velocity and acceleration in the x-direction are given in Figure 4.30. Time axis and vertical axes are normalized.

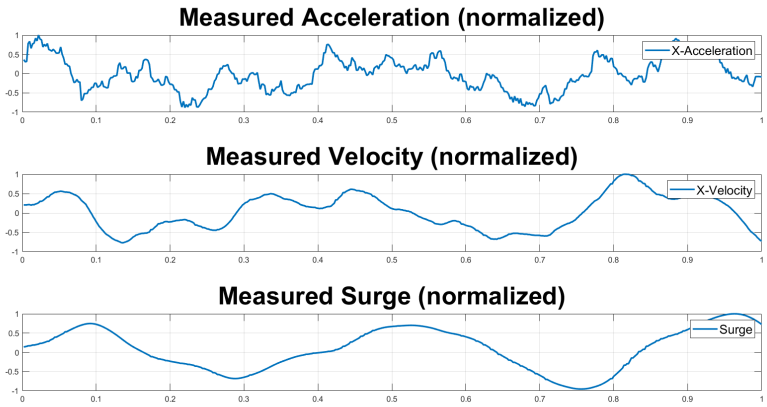


Figure 4.30: Position, velocity and acceleration in the x-direction measured by the INS.

First, one can study the effect of giving different weights to position, velocity and acceleration on the standard error (or uncertainty) of the control points. This discussion is qualitative, using the basic argument that acceleration is typically noisier than velocity and position. Deeper signal study is needed if one wants to get to the exact weights that each channel should be given. In this case, the nodes are arbitrarily placed every $0.5\,\mathrm{s}$. Starting with basic least-squares (non-weighted) fitting, the results are shown in Figure 4.31. The fitting is first made with the real data but then normalized for the plots. Next, the standard error of the control points can be determined. From theory, quintic Hermite basis functions are defined piece-wise between each node. The number of control points is 3 times the number of nodes. For one given node, the value of the first control point corresponds to the fitted value of position, the second to the fitted value of velocity and the third to acceleration. As it was discussed in the previous subsection, the least-squares fit gives the standard error of each control point. The mean standard error for each control point corresponding to a reconstructed value of position at the nodes is first calculated. Then, this standard error is divided by the maximum value of position to get an order of magnitude of the significance of the error. This quantity will be denoted as $\overline{\sigma_x}$. The same is done for velocity and acceleration. In this case, the results are:

- For position: $\overline{\sigma_x} \approx 4.4\%$.
- For velocity: $\overline{\sigma_v} \approx 4.6\%$.

- For acceleration: $\overline{\sigma_a} \approx 17.9\%$.

It comes that the fitted values of acceleration at the nodes are far more uncertain than for velocity and position. This also gives an idea of the uncertainty of the measurements. Thus, trying to fit by all means the acceleration data will lead to larger errors. This reinforces the idea of giving a different weight to acceleration, velocity and position in the fitting because they are all initially not as reliable. Moreover, what matters most for the motion-correction algorithm is not acceleration but position and velocity.

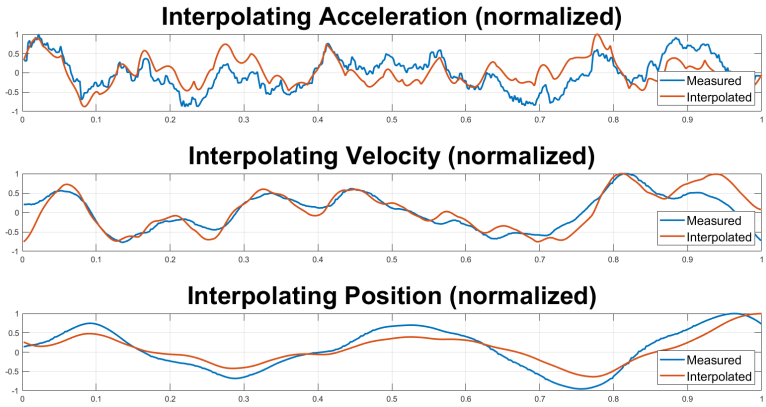


Figure 4.31: Interpolating position, velocity and acceleration in the x-direction using Quintic Hermite interpolation. The nodes are placed every $0.5\,\mathrm{s}$ and the weights for the fitting are uniform.

Now using weighted least-squares fitting with Surge being given the highest weight and acceleration the lowest, the result is given in Figure 4.32. The fit is much better visually for position and velocity. For velocity particularly, there still remains some non-negligible errors, see for example at the start. The result on the mean normalized standard errors are now:

- For position: $\overline{\sigma_x} \approx 1.9\%$.
- For velocity: $\overline{\sigma_v} \approx 3.3\%$.
- For acceleration: $\overline{\sigma_a} \approx 19.7\%$.

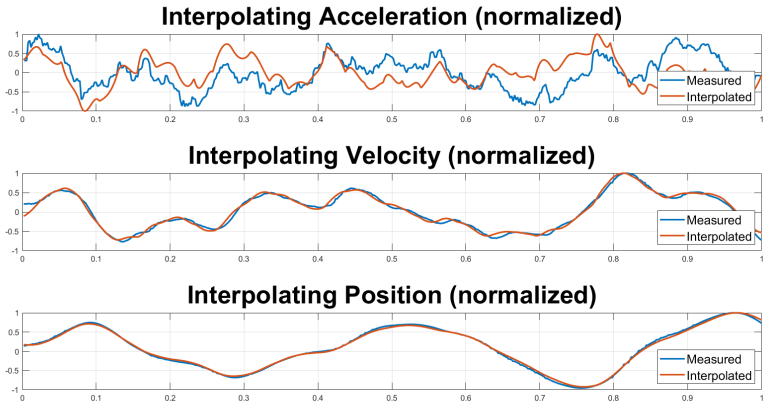
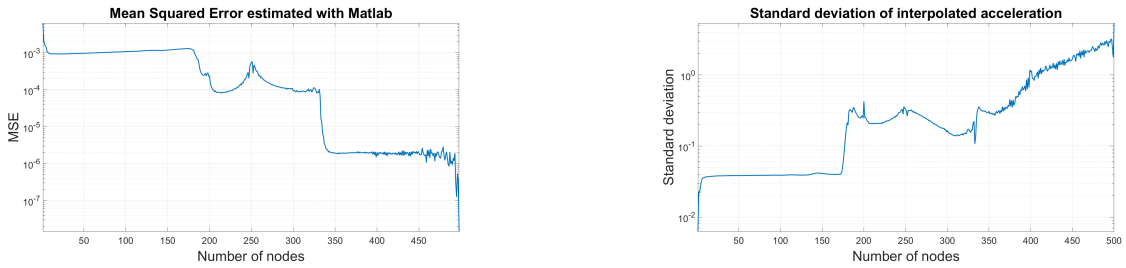


Figure 4.32: Interpolating position, velocity and acceleration in the x-direction using Quintic Hermite interpolation. The nodes are placed every $0.5\,\mathrm{s}$ and weighted least-squares fit is used.

This brief study shows the appeal of least-squares fitting coupled with the Quintic-Hermite modelling of position, velocity and acceleration. It can first give an order of magnitude of the measurement's uncertainties with the standard error of the control points. The second conclusion is that interpolation should be performed by using weighted least squares fitting for the calculation of the control points at the nodes, with a higher weight on position than on velocity, and higher on velocity than on acceleration.
Finally, the previous investigation on the positioning of the control points can be done in this case, using all three data channels and the same weighting. For the given time period, the number of nodes is gradually increased. Each time the fitting is done, the standard error (or uncertainty) of the control points, the Mean Square Error (MSE), and the standard deviation of the interpolated signal are determined. The standard error of the control points simply corresponds to what was shown earlier. The MSE is an output of the weighted least-squares fitting in Matlab. It gives the weighted error of the interpolation at the time of the measurements, compared to the real measurements. It is divided by the difference in number of rows and columns in A. On top of this, it does not take into account that all data may have different units so its physical meaning is lessened. Thus, its absolute value is not so significant, but its relative value compared to the previous numbers of nodes is. Then, the standard deviation of the signal is calculated by interpolating the acceleration over a large number of points (twice as many as the number of measurements), and then taking the standard deviation of this interpolated signal. It is done with acceleration because it is the less numerical stable. So it could be that position and velocity have relevant behaviour but acceleration on the other hand shows wild and non-physical fluctuations.
The results for MSE and standard deviation of interpolated acceleration are shown in Figure 4.33.



(a) Matlab estimated MSE of the weighted least-squares fitting with increasing number of nodes.

(b) Standard deviation of interpolated acceleration signal from the weighted least-squares fitting with increasing number of nodes.

Figure 4.33: Comparison of the performance (MSE) of the fitting vs. fluctuations in the interpolated acceleration.

The results are that generally, the more nodes there are, the lower the MSE is (see Figure 4.33a), but the higher the standard deviation in the interpolated acceleration (see 4.33b). This is consistent with the discussion in the subsection on acceleration. The trade-off is not so straightforward because a third criterion also appears: uncertainty of the fitted control points. This is shown in Figure 4.34,

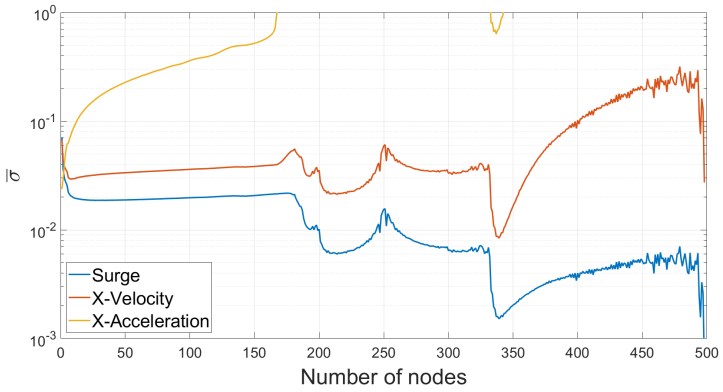


Figure 4.34: Standard error in the fitted control points for position, velocity and acceleration.

To measure the importance of the trade-off between all three criteria, some examples can be studied. First, one can start by prioritizing minimizing the standard deviation of interpolated acceleration and the

uncertainty in the control points. That corresponds to taking a low number of nodes. For example with 10 nodes, the mean normalized standard error of the acceleration control points stay below 10%. For both velocity and position, they are below 3%. And the standard deviation of interpolated acceleration remains in the low part. The result is shown in Figure 4.35. Note that this corresponds to placing nodes approximately every $2\,\mathrm{s}$.

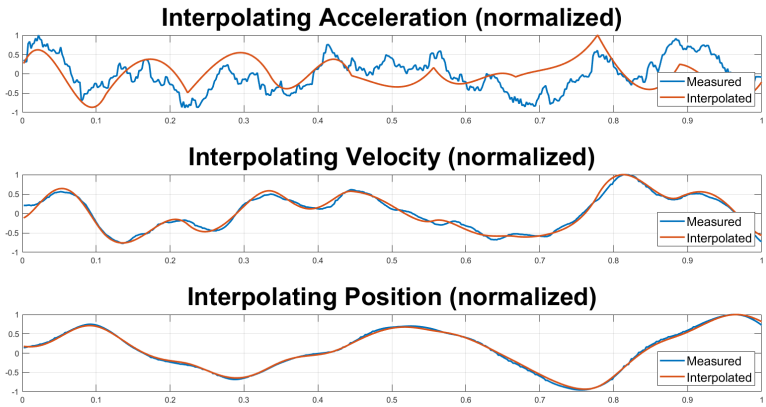


Figure 4.35: Interpolating position, velocity and acceleration in the x-direction using Quintic Hermite interpolation. Weighted least-squares fit is used and a low number of nodes (10) is considered.

Depending on the arguments, this can already be accepted. However, the curve for acceleration still seems too coarse. As it was discussed earlier, using too many nodes leads to undesired behaviours in the interpolation. This is why the nodes should be chosen such that for example the standard deviation of interpolation acceleration remains in the low part. Also one does not want to have too high standard error in the fitted control points, otherwise it cannot be considered fitting anymore. With a higher number of nodes, for example 150, the interpolated curves are shown in Figure 4.36.

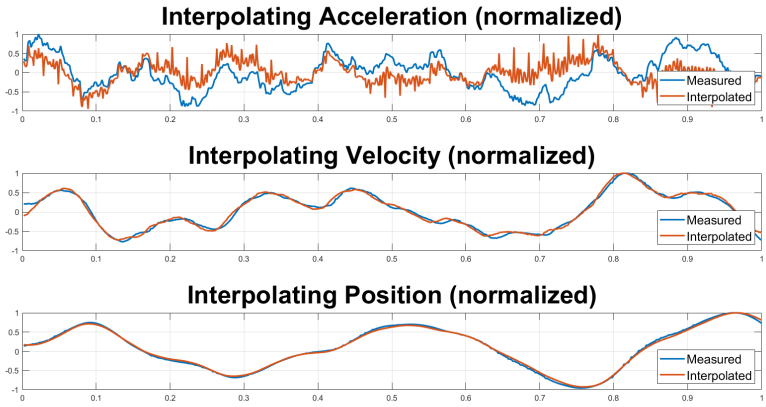


Figure 4.36: Interpolating position, velocity and acceleration in the x-direction using Quintic Hermite interpolation. Weighted least-squares fit is used and a higher number of nodes (150) is considered.

At this point, acceleration starts to show even more fluctuations than the measured acceleration, which is not desirable. The measured standard deviation of the interpolated signal is still low but the mean normalized standard error of the control points is over 50%. This shows that too many nodes will ruin the quality of the interpolation. The objective is to reconstruct something closer to what is seen with nodes approximately every $0.5\,\mathrm{s}$ (see Figure 4.32) and every $2\,\mathrm{s}$ (see Figure 4.35). With this kind of initial signal of measurements, the interpolations of position and velocity remain fundamentally unaffected, although it starts to be the case for velocity with higher number of nodes (see the orange curve in Figure 4.34).

In conclusion, another essential point of this subsection is that the nodes should be placed in the order of magnitude of every second for interpolation, depending on the result that one wants on the reconstructed acceleration. Unfortunately, similar study cannot be done for the interpolation of orientation and angular velocity. The developed quaternion method is powerful but it involves using a mathematical space that is not equipped with many numerical tools.

5
Results

This chapter will be focused on showing simulation results. The simulations are based on a coupling of the motion of a modelled FOWT with the corresponding measurements from a nacelle-mounted lidar in the turbulent atmosphere that caused the motion. Given that all data channels for the TetraSpar prototype were not fully available during the time frame of this project, results from real measurements will not be presented. The treatment of real data consists of what was already presented: insights on interpolation and uncertainties of the motion data from the INS, discussion on the position/orientation of the lidar compared to the INS and definition of the reference frames used by the INS. Regarding the simulations, they were made using one model of FOWT and several cases of wind field and waves, defined shortly after. The results of the simulations should help answer the research questions.

5.1. Presentation of the turbine and of the problem

5.1.1. WindCrete model

The FOWT modelled is a spar floater called *WindCrete*, available freely online [14] and used in research for simulating floating lidars [34]. The sketch of the design of the WindCrete floater is found in Figure 5.1. As shown, the tower height is $129.495\,\mathrm{m}$ with respect to Mean Sea Level (MSL). This was set in order to have a hub height of $135\,\mathrm{m}$. Notably, the red part at the bottom is the ballast that is used to balance the structure and the cylindrical spar starts at $10\,\mathrm{m}$ below MSL, with a diameter of $18.6\,\mathrm{m}$.

From OpenFAST, the outputs for position and orientation are Surge/Sway/Heave and Yaw/Pitch/Roll of the platform, for example at the foundation as defined in Figure 5.2. In the rigid body assumption, the rotation and the translational displacements are the same at the nacelle for example. So that can be extrapolated at the position of the lidar for example. Note that the x-axis is defined in the same direction as the wind. So if the lidar is facing the wind, the coordinates of the particles in the global frame will be negative. It is the same in the lidar simulator. Apart from neglecting structural deformations in the turbine, this also has the disadvantage of not taking into account the nacelle yaw (disabled degree of freedom) and pre-tilt.

Additionally, OpenFAST can also output inertial data at the nacelle (the same as an IMU would do). This is given in a local frame that moves with the tower and the nacelle. The mapping to a global inertial frame that has the same orientation as the one for which Yaw/Pitch/Roll angles are given is then simply made using the rotation matrix defined by these angles (in the ”Z-Y-X” convention for OpenFAST). The global frame is chosen to be at the Mean Sea Level with the same orientation as the one in which the Euler angles and translational displacements are given. In this referential, the initial coordinates of relevant components are shown in Table 5.1. The placement of the lidar was made arbitrarily such that it is somewhere above the IMU. This is meant to resemble the real configuration where the IMU is in the nacelle floor and the lidar on the roof.

5.1.2. Wind Field input

Generating a matching wind field between OpenFAST and the lidar simulator is crucial to get relevant and usable results. This was done as explained in the previous Section by writing the scaled boxes

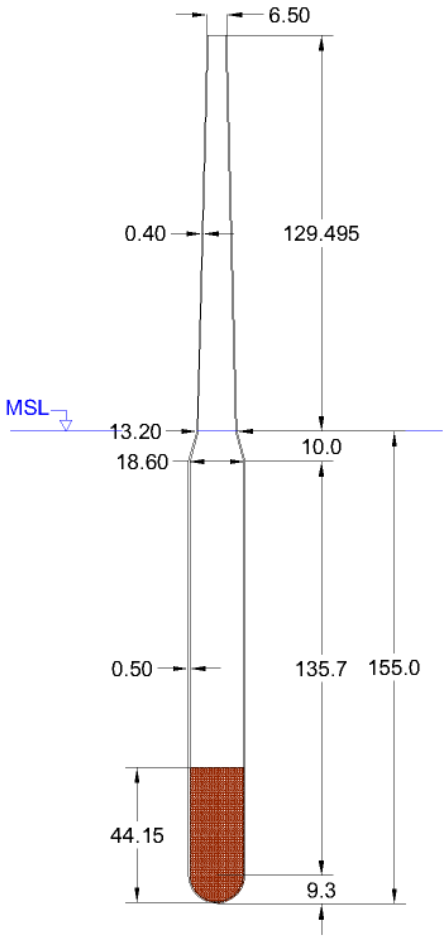


Figure 5.1: Cross-sectional sketch of the WindCrete model (values in meters) [14].

first in Matlab using a program internal to the company, and then using the exact same in OpenFAST. The wind fields used have the following properties:

- Mean wind field : Power law with $U_{ref} = 12\,\mathrm{ms}^{-1}$ at a reference height of $150\,\mathrm{m}$ and a shear exponent of $\alpha = 0.2$
- Turbulence intensity $TI = 0.1$, $L = 35.4\,\mathrm{m}$, $\alpha_\epsilon = 1.0$ and $\Gamma = 3.9$
- The grid inside of the box is made of 64 points both transversely and vertically, and 32768 (2^{15}) points in the wind direction. The width/height of the box is $296.1\,\mathrm{m}$ and its length is around $27\,000\,\mathrm{m}$. With an advection speed of $12\,\mathrm{ms}^{-1}$, this allows for more than 30 minutes of simulated statistically independent turbulence.

A time series of 30 minutes for the U component of the wind at the box origin (height of $150\,\mathrm{m}$) is shown in Figure 5.3. This is for a chosen random seed. With the exact same parameters, it is possible to generate a different field by changing the seed for the randomness of turbulence.

Name of component	X-coordinate	Y-coordinate	Z-coordinate
Tower top	$0\,\mathrm{m}$	$0\,\mathrm{m}$	$129.495\,\mathrm{m}$
IMU	$-4.985\,\mathrm{m}$	$0\,\mathrm{m}$	$133.853\,\mathrm{m}$
Lidar	$-3.985\,\mathrm{m}$	$0.5\,\mathrm{m}$	137.853

Table 5.1: Coordinates of the IMU and of the lidar in the inertial fixed global frame, the X-axis of which is in the downwind direction and Z-axis pointing upwards. Its origin is fixed at the Mean Sea Level aligned with the turbine.

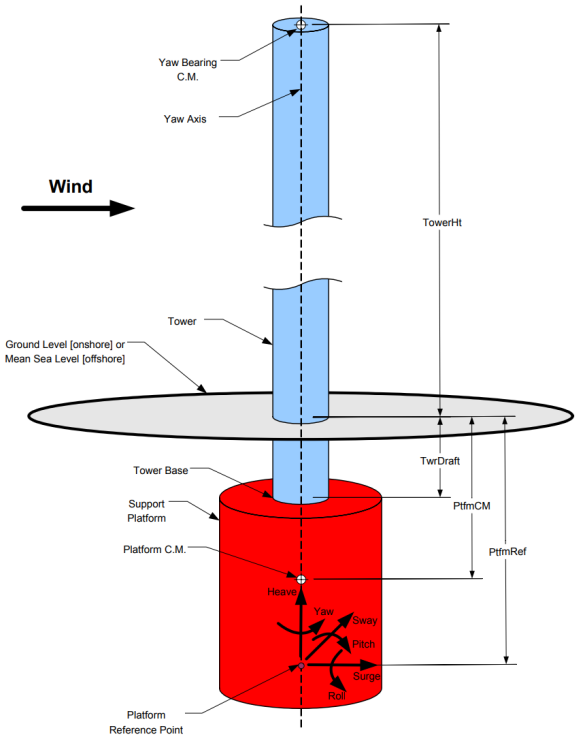


Figure 5.2: Support platform / foundation layout in FAST [57].

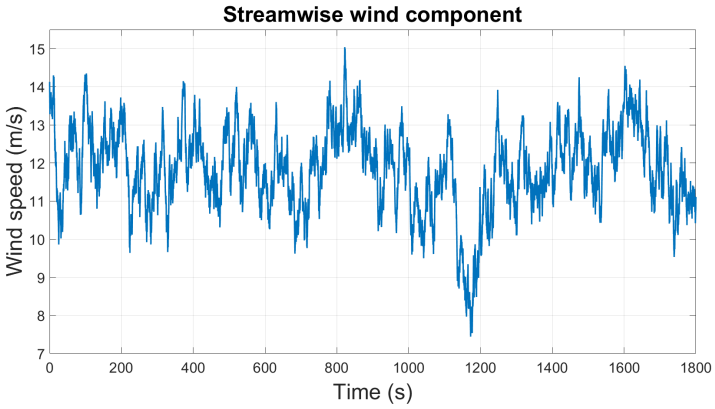


Figure 5.3: 30-minute time series of U at the origin of the turbulence box ($150\,\mathrm{m}$) for the first random seed.

5.1.3. Waves input

To model more realistic motion of the turbine, wave conditions are also set in the model. It will be specified for a given simulation what these conditions are. The generic type of waves generated is set to the third type in OpenFAST. This corresponds to irregular waves generated from a white noise spectrum. For example, the significant wave height can be set to $H_s = 2\,\mathrm{m}$. An example time series of wave elevation is shown in Figure 5.4.

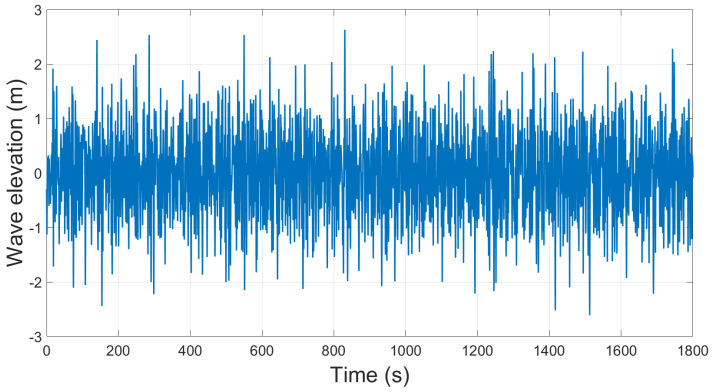
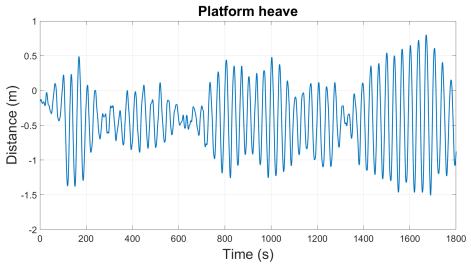


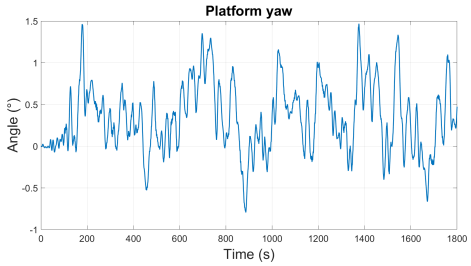
Figure 5.4: 30-minute time series of the wave elevation at the turbine location. White noise spectrum with $H_s = 2\,\mathrm{m}$.

5.1.4. Examples of motion outputs

Some examples of motion time series will be shown here for one wind field as defined in the previous subsection and irregular waves with $H_s = 2\,\mathrm{m}$. The heave and yaw of the platform are shown in Figure 5.5. The initial condition on heave is left by default at $-0.16\,\mathrm{m}$ and Figure 5.5a shows the clear oscillations and periodic nature of heave. The platform yaw in Figure 5.5b also shows oscillations, although less obvious than for heave, around an angle close to 0.



(a) Time series of the platform heave.



(b) Time series of the platform yaw.

Figure 5.5: Platform yaw and heave with initial random seed for the wind field and $H_s = 2\,\mathrm{m}$.

To conduct some more tests on the response of the turbine, the PSD spectrum of the platform heave was computed. It is shown in Figure 5.6. In the manual for the WindCrete model, it is written that the **natural frequency** of the platform in **heave** is $0.030\,52\,\mathrm{Hz}$ [14]. The corresponding line is shown in the PSD plot, with a peak amplitude close to that frequency.

Next, some more surprising results are found in Figure 5.7. Figure 5.7a represents the platform surge and Figure 5.7b represents the platform pitch. What stands out is an event of extreme pitch and surge after around $100\,\mathrm{s}$ of simulation. One explanation could be that this comes from an extreme forcing applied to the turbine. Indeed, a high force in the x-direction will result in large surge and pitch response since these degrees of freedom evolve in the same plane. However, the wind and wave conditions do not appear to reach extreme values around this time. Another explanation could be resonance. Indeed, concentrating on the first $100\,\mathrm{s}$ of the simulation, it seems that the pitch angle shows a periodic behaviour with a period of around $40\,\mathrm{s}$ (see Figure 5.7b). As a first approximation, this is close to the **natural period** of the platform in **pitch**, which is $40.97\,\mathrm{s}$ or $0.024\,41\,\mathrm{Hz}$ in terms of **natural frequency** [14].

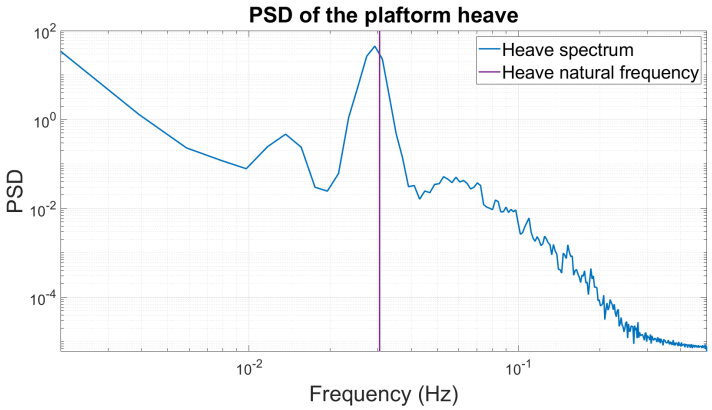
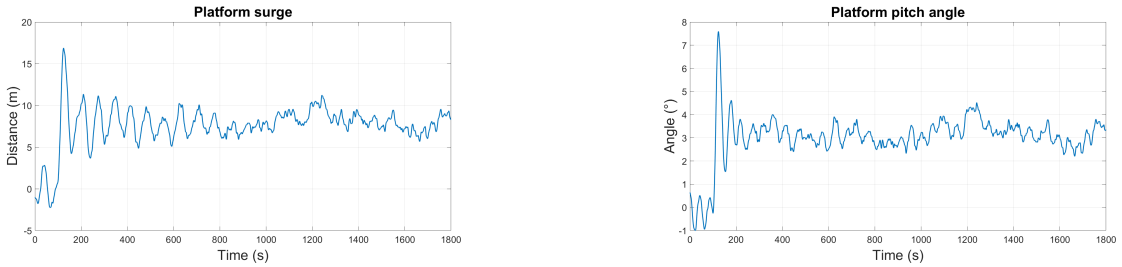


Figure 5.6: PSD spectrum for the platform heave in the given simulation conditions, computed for the 30 minutes of the simulation with a sampling frequency of $1\,\mathrm{Hz}$.



(a) Time series of the platform surge.

(b) Time series of the platform pitch.

Figure 5.7: Platform pitch and surge with initial random seed for the wind field and $H_s = 2\,\mathrm{m}$.

To highlight this in a quantitative way, the PSD spectrum of the pitch angle was calculated for the first $100\,\mathrm{s}$ of the simulation, to verify if the platform natural frequency shows an amplitude peak. This was computed using the *periodogram* function of Matlab and one can find the result in Figure 5.8. It seems clear that the natural frequency of the platform in pitch was excited, which can explain the considerable escalation in pitch. As for surge, it turns out that the natural frequency of the platform in surge is exactly half of that in pitch [14], explaining the excitation too. After this peak, both quantities keep fluctuating but stabilize over a mean. The PSD spectrum of the platform pitch for the whole simulation is also shown in Figure 5.9. The excitation at the natural frequency does not appear as clearly there.

This extreme event at the start of the simulation is in fact valuable for the results since it enables to show how this impacts nacelle-mounted lidar measurements, and how useful the correction can be in such cases.

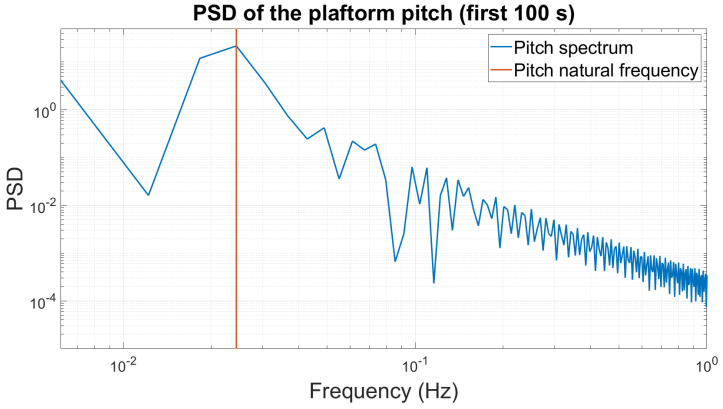


Figure 5.8: PSD spectrum of the platform pitch at the start of the simulation. This is studied as a way to explain the extreme pitch response. For better resolution, the sampling frequency was increased and a simple rectangular window was used.

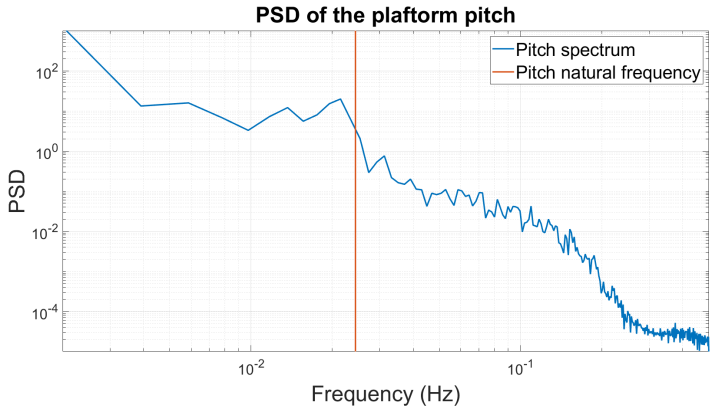


Figure 5.9: PSD spectrum for the platform pitch in the given simulation conditions, computed for the 30 minutes of the simulation with a sampling frequency of $1\,\mathrm{Hz}$.

5.2. Results

The results will be shown for a simulated lidar modelled with each beam and range gate with 100 axial particles and 9 radial particles. The lidar parameters were set to the same as in Section 4. The resulting VLOS will be studied for three different range gates (horizontal distance from the lidar) : $200\,\mathrm{m}$, $300\,\mathrm{m}$ and $400\,\mathrm{m}$. To mimic the functioning of the WindCube Nacelle used on the TetraSpar prototype, the VLOS used for wind field reconstruction are sampled every second alternating between beam. LOS 0 and LOS 1 are the upper beams, and LOS 2 and LOS 3 are the lower beams. Additionally, the inertial quantities are sampled at $25\,\mathrm{Hz}$, again to resemble the properties of the IMU in the TetraSpar prototype.

5.2.1. Resulting VLOS and impact of lidar interfering velocity

To start with, for more clarity, the VLOS measurements of a single beam will be shown. For the first upper beam, the time series of the measurements at a range of $200\,\mathrm{m}$ and $400\,\mathrm{m}$ are displayed in Figure 5.10. Figure 5.10 shows the VLOS for a moving lidar that is measuring at the correct locations, but without interference of its own velocity. In other words, this is the measurements expected if the lidar's velocity would be perfectly compensated for in the VLOS. Given the inclination of the beam, a longer range means a higher measurement height for the upper beams and the opposite for the lower beams. Thus, it is expected that the measurements at a $400\,\mathrm{m}$ range have a higher mean. This is verified in this case: mean of $11.7\,\mathrm{ms}^{-1}$ at the first range and $12.1\,\mathrm{ms}^{-1}$ at the last range.

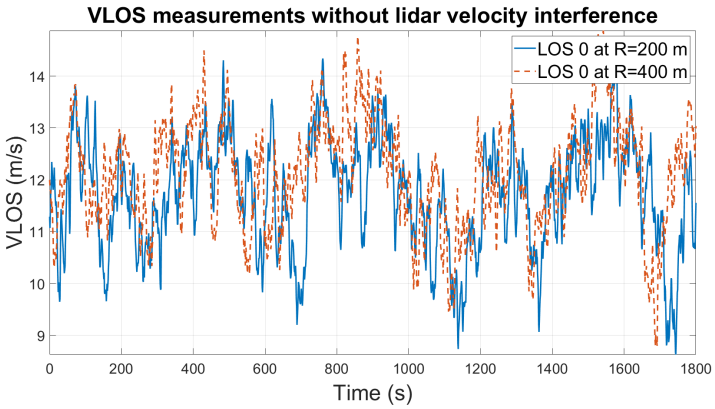


Figure 5.10: VLOS time series for one of the upper beams in a wind-motion coupled simulation. No lidar velocity interference

Then, more interestingly, the algorithm developed allows to model the impact of the pulsed lidar's translational and angular velocity on the particle's velocities that the moving lidar "sees". The result is shown in Figure 5.11. As expected, the lidar velocity interference does not have much impact on the time series on average but the particular extreme event highlighted previously (around $100\,\mathrm{s}$) stands out, especially for the red dotted curve that falls under $10\,\mathrm{ms}^{-1}$ in the first $200\,\mathrm{s}$.

To align with the correction methods discussed before, it is possible to look into the spectrum of the VLOS, to see if the platform's natural frequency in pitch stands out. As a control sample, a fixed lidar was simulated. This lidar has the same initial coordinates as the moving lidar modelled here but does not move with the nacelle. It represents a case where the wind turbine would not be moving at all due to wind/waves and the only contribution to the lidar measurements would be the turbulent wind, no effect of rotation or velocity. This will serve as a reference for the spectrum. To start with, studying the effect of only rotation, the spectrum of the theoretical lidar VLOS measurements without interfering velocity is shown against that of a fixed lidar in Figure 5.12. The spectra of the moving lidar without velocity interference shows a close slope to that of the fixed lidar. Note that the spectrum of the fixed lidar measurements does not have the perfect low-pass filter-like behaviour shown before because the convolution of the projected velocities does not happen only in one dimension, but takes into account radial particles.

As can be seen, there is no amplitude peak showing a clear periodic phenomenon due to the pitching of the platform. This is not a good sign for the frequency-filtering method in this case, which would not have a satisfying performance. Apart from this comparison, the spectrum of the measurements with full motion taken into account (so also the velocity of the lidar) was calculated and is shown in Figure

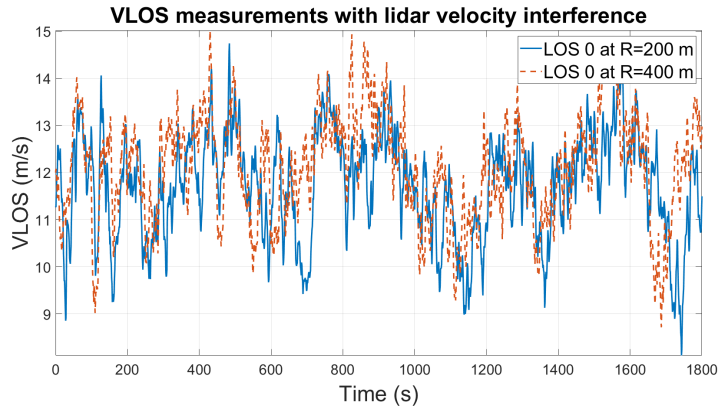


Figure 5.11: VLOS time series for one of the upper beams in a wind-motion coupled simulation. This takes into account lidar velocity interference

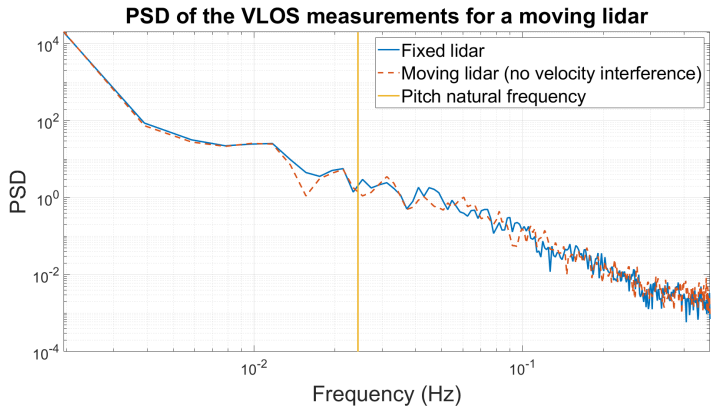


Figure 5.12: PSD of the VLOS measurements for a fixed lidar vs. a moving lidar, with only changing orientation/position but no velocity interference. The spectrum is computed with a $1\,\mathrm{Hz}$ sampling rate using the 30 minutes of the simulation, for an upper beam at a range of $400\,\mathrm{m}$.

5.13. As it was seen before, the spectrum of the measurements with velocity interference ”detaches” from the slope of a fixed lidar spectrum.

Next, the logical step is to verify the performance of the part of motion-correction algorithm that subtracts the lidar velocity to the measurements. The algorithm does this by calculating the part of integral 2.15 that corresponds to the lidar velocity. By adding this to the ”raw” measurements, then the theoretical VLOS is obtained, without the lidar interference. By passing as inputs the lidar’s translational and angular velocity, the algorithm calculates the velocity of the lidar with respect to each modelled particle, and then approximates the integral by a weighted sum as it is done for the VLOS measurements. It turns out that in this case, to calculate the VLOS time series with respect to a moving lidar, the exact same reverse process was applied. Thus, verifying that the motion-correction method is functional comes down to verifying that it is possible to alternate correctly between corrected and uncorrected VLOS measurements by using the same inputs for the lidar’s velocity.

This is not as straightforward as it appears because the methods used for the lidar simulator are coded in C++, with a function that enables translation to Matlab. Thus, some small numerical errors such as rounding errors are expected. There are cases when more than rounding errors will arise, as for example:

- Interpolation and noise in the motion data of the lidar is simulated. In this case, the lidar’s velocity used for correction will not be exactly that used to calculate the ”real” measurements. This is valid for the lidar’s velocity but also for its position and orientation.
- The velocity is corrected with the point-wise approximation, in this case the correction term is not an approximated integral but just one value.

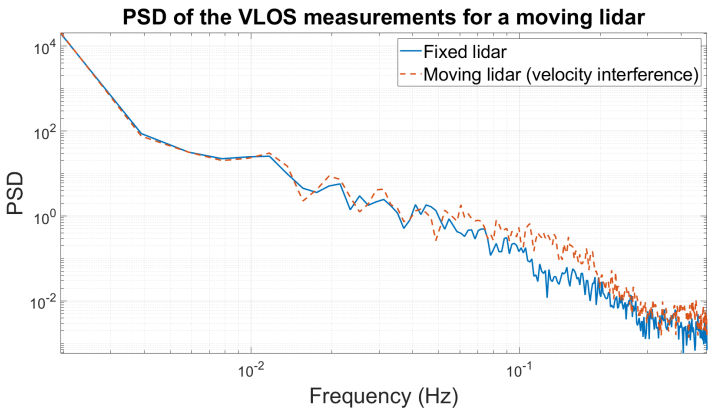
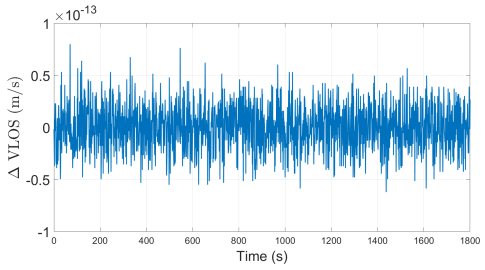
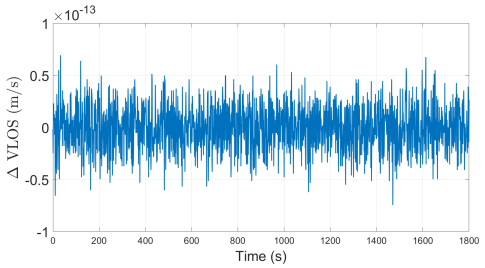


Figure 5.13: PSD of the VLOS measurements for a fixed lidar vs. a moving lidar, with changing orientation/position and velocity interference. The spectrum is computed with a $1\,\mathrm{Hz}$ sampling rate using the 30 minutes of the simulation, for an upper beam at a range of $300\,\mathrm{m}$.

The same point cloud distribution for the averaging of the lidar's velocity is kept as for the calculation of the VLOS. Again for one of the upper beams, the difference between the simulated real lidar measurements (with its velocity interfering) and the corrected ones (subtracting this interference) is shown in Figure 5.14a. For the beam-alternating measurements, the result is shown in Figure 5.14b. In summary, with errors below $1 \times 10^{-13}\,\mathrm{ms}^{-1}$, Figure 5.14 shows that the algorithm suffers some numerical errors switching between corrected and uncorrected measurements, that should mainly be caused by a mapping between C++ and Matlab. However, this seems robust enough since these errors are largely bounded.



(a) Reversing the interference of the lidar's velocity in the VLOS measurements to check if the correction has intrinsic error. This is done for an upper beam at a range of $200\,\mathrm{m}$.



(b) Reversing the interference of the lidar's velocity in the VLOS measurements to check if the correction has intrinsic error. This is done with alternating each second between beam at a range of $200\,\mathrm{m}$.

Figure 5.14: Checking for numerical errors in the calculation of the lidar's interference in the VLOS measurements.

Another way of verifying that the influence of the lidar velocity is correctly calculated is to determine the difference between the corrected and uncorrected time series. As it was discussed before, on average, the interference of the lidar velocity should cancel out in the VLOS measurements. If the difference between the two time series is significant, then it is highly likely that the correction suffers a bias. This difference is displayed in Figure 5.15 for the beam-alternating time series at a range of $100\,\mathrm{m}$. The range does not matter here because the difference should only be the spatial average of the lidar's velocity projected onto the beam direction. And, along the same beam, the direction unit vector is the same.

The average difference between corrected and uncorrected lidar time series is as follows:

- Between 0 and 10 minutes: $0.019\,\mathrm{ms}^{-1}$ difference.
- Between 10 and 20 minutes: $0.013\,\mathrm{ms}^{-1}$ difference.
- Between 20 and 30 minutes: $0.006\,\mathrm{ms}^{-1}$ difference.

Thus, even for the first 10 minutes when the lidar's velocity reaches higher values and thus creates larger interference, the difference on average is negligible. Now that the first results on the impact of

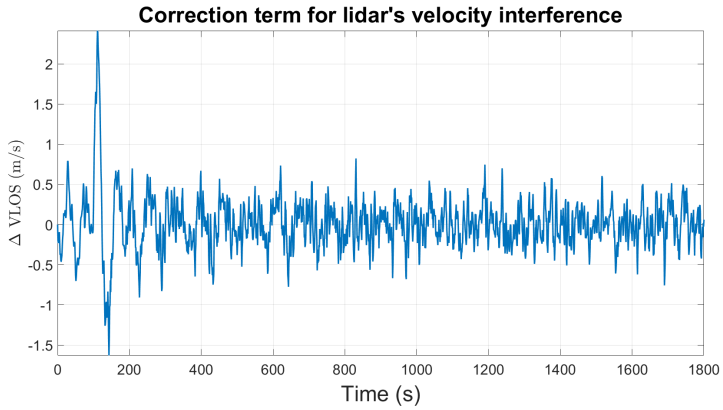


Figure 5.15: Correction term to add to the ”real” lidar measurements to cancel the influence of the lidar's velocity on the measurements. The measurements alternate between beam, as for the real WindCube model.

the lidar's motion on the measurements were introduced, it is possible to quantify this impact on the results of some wind field reconstruction methods.

5.2.2. Methods for reconstructing mean wind field

In this subsection, three methods are introduced to get to some wind field parameters, using more or less information.

Direct beam-average reconstruction

This method is the easiest to implement and was used in literature to reconstruct the horizontal wind speed [34]. It uses an average of the four beams' measurements, reversing the projection of the wind velocity onto the beam direction by assuming that the wind velocity is initially purely horizontal. The expression is simply found in Equation 5.1. Again, θ is the zenith angle of the beams. This does not take into account the real-time position and orientation of the lidar. This can only be used as an estimation of the horizontal wind speed at the height of the lidar. Turbulence intensity can be estimated using the standard deviation of this reconstructed time series to its mean.

$$U_{rec} = \frac{1}{4\ \cos(\theta)} \left(\mathrm{VLOS}_0 + \mathrm{VLOS}_1 + \mathrm{VLOS}_2 + \mathrm{VLOS}_3\right) \tag{5.1}$$

2D Upper-Lower tilt-corrected reconstruction

This method is proposed by the lidar manufacturer [86]. The name given corresponds to the fact that the transversal component is also reconstructed on top of the horizontal component (2D). The reconstructed quantities are given separately at the upper and the lower beams (Upper-Lower). It makes sense to separate both given that the two upper beams will have a close measurement height, which is significantly different however than that at the lower beams. Finally, the formulas take into account the tilt of the lidar, but only on a 10-min average. The formulas for 10-min averaged VLOS ($\overline{\mathrm{VLOS}}$) and Turbulence Intensity (TI) are given for the first beam (LOS 0) in Equation 5.2. N_0 is the number of valid measurements coming from LOS 0 for 10-minutes. For instance, for 10 minutes where measurements are fully available (no blade blockage for example), $N_0 = 600/4 = 150$. Exactly the same formulas hold for the 3 other beams, with just the index changing.

$$\begin{aligned} \overline{\mathrm{VLOS}_0} &= \frac{\sum_{i=1}^{N_0} \mathrm{VLOS}_{0,i}}{N_0} \\ TI_0 &= \frac{\sqrt{\frac{\sum_{i=1}^{N_0} \left(\mathrm{VLOS}_{0,i} - \overline{\mathrm{VLOS}_0}\right)^2}{N_0}}}{\overline{\mathrm{VLOS}_0}} \end{aligned} \tag{5.2}$$

Then, the reconstruction of the mean horizontal wind speed (HWS) at the upper and lower beams is done separately. It uses the mean tilt of the lidar τ over the 10-minute period. The reconstructed quantities for the upper beams are given in Equation 5.3. The same principle applies for the two lower beams, with a change of index. Note that in this case, the contribution of the transversal reconstructed component to the mean horizontal wind speed will always be negligible.

$$\begin{aligned} U_{rec}^{+} &= \frac{\overline{\mathsf{VLOS}_0} + \overline{\mathsf{VLOS}_1}}{2(\cos(\theta)\cos(\tau) - \sin(\theta)\sin(\phi^{+})\sin(\tau))} \\ V_{rec}^{+} &= \frac{\overline{\mathsf{VLOS}_0} - \overline{\mathsf{VLOS}_1}}{2\sin(\theta)\cos(\phi^{+})} \\ HWS^{+} &= \sqrt{U_{rec}^{+} + V_{rec}^{+}} \\ TI^{+} &= \frac{TI_0 + TI_1}{2} \end{aligned} \tag{5.3}$$

From this, the manufacturer also proposes a way to estimate the shear exponent of the mean wind profile. This uses an estimation of the measurement height corresponding to the given range R of the measurements. It is found in Equation 5.4.

$$\begin{aligned} H^{+} &= Z_{\mathsf{lidar}} + \frac{R}{\cos(\theta)}\left(\cos(\theta)\sin(\tau) + \sin(\theta)\sin(\phi^{+})\cos(\tau)\right)) \\ H^{-} &= Z_{\mathsf{lidar}} + \frac{R}{\cos(\theta)}\left(\cos(\theta)\sin(\tau) + \sin(\theta)\sin(\phi^{-})\cos(\tau)\right)) \\ \alpha &= \frac{\ln(\frac{HWS^{+}}{HWS^{-}})}{\ln(\frac{H^{+}}{H^{-}})} \end{aligned} \tag{5.4}$$

Power law fitting with full correction

The first two methods are initially based on raw VLOS measurements, so without velocity correction. Additionally, the first and simplest method does not take into account the orientation change of the lidar. The second method, proposed by Vaisala, does take it into account but only on average. This is mainly meant to account for a voluntary tilt of the nacelle, that would change the actual direction of the beam in a global Earth frame.

Now this method was designed to use VLOS measurements that correct for the lidar's velocity in the global frame, and to take into account the actual orientation of the lidar at each time step of measurement. It relies on a non-linear least-squares fitting of the measurements' to a power law. In that case again, only the mean profile can be reconstructed. The defaults of this method are that it needs an initial guess on the parameters of the power law (reference wind speed and height, shear exponent), and that it is only able to reconstruct reliably an input wind profile that is close to a power law. For real profiles that are more skewed, similar methods exist to fit the measurements to a set of Laguerre basis functions, but it is out of the scope of the project. Turbulence reconstruction needs a separate algorithm that relies for example on Kriging method, this is also out of scope.

In conclusion, because the mean profile is a power law and because the initial guess can be given close to the real parameters, this method should have the best performance on the wind field reconstruction.

Interpreting the results of the wind field reconstruction

The result of each method needs to be compared to a reference, to what *should be reconstructed*. The mean wind profile and the turbulence intensity are input in advance and well known so that is a solid reference. As a reminder, TI is set to 10%, and for the mean profile, the reference wind speed is set to $12\,\mathrm{ms}^{-1}$ at $150\,\mathrm{m}$ with a shear exponent of 0.2.

However, for example because the interpolation inside of the turbulence box is set to only linear and that the box is used for a lower transport velocity than it is made for, the mean wind speed generated at $150\,\mathrm{m}$ will not be exactly $12\,\mathrm{ms}^{-1}$ when averaged at $1\,\mathrm{Hz}$ over the 30-minute period. In this case, the 30-min average of the horizontal wind speed component at the reference height from the simulations is of

$11.84\,\mathrm{ms}^{-1}$. This is between 1% and 2% of difference compared to the reference value of $12\,\mathrm{ms}^{-1}$. This mean is also different for the three different 10-minute periods. A threshold has to be set somewhere, and that can be for example accepting around a 2% difference, such as between the theoretical and the actual 30-min average at $150\,\mathrm{m}$. This gives a range of acceptable values for the mean horizontal wind speed at $150\,\mathrm{m}$ between $11.8\,\mathrm{ms}^{-1}$ and $12.2\,\mathrm{ms}^{-1}$. If the reconstructed profiles are over this threshold, then it can be concluded that the method used is not satisfactory. At the height of the lidar, the theoretical mean wind speed is supposed to be $11.8\,\mathrm{ms}^{-1}$. Again accepting around 2% of difference, that is a range between $11.6\,\mathrm{ms}^{-1}$ and $12.0\,\mathrm{ms}^{-1}$.

Next, what is interesting is to see the sensitivity of the reconstructed parameters depending on the motion of the lidar and on the correction (or not) of the lidar's velocity.

5.2.3. Results: no correction, orientation correction, full correction

To build a basis for comparison, a fixed lidar configuration is first considered. This lidar is placed at the initial position of the lidar (see Table 5.1, always facing the wind and does not suffer any interfering velocity. For the three 10-minute periods and ranges, the output of each reconstruction method is shown in Table 5.2.

	Direct beam average		Vaisala method				Power law fitting		
	U_{rec}	TI	HWS^+	HWS^-	TI^+	α_0	U_{ref}	z_{ref}	α_1
0-10 ($R = 200\,\mathrm{m}$)	11.9	0.05	12.2	11.6	0.09	0.19	12.1	150.0	0.20
0-10 ($R = 300\,\mathrm{m}$)	11.9	0.05	12.2	11.5	0.08	0.18	12.1	150.0	0.17
0-10 ($R = 400\,\mathrm{m}$)	11.8	0.06	12.4	11.2	0.08	0.19	12.1	150.1	0.19
10-20 ($R = 200\,\mathrm{m}$)	11.6	0.07	11.9	11.3	0.1	0.20	11.8	150.0	0.21
10-20 ($R = 300\,\mathrm{m}$)	11.7	0.07	12.2	11.2	0.1	0.23	12.0	150.1	0.23
10-20 ($R = 400\,\mathrm{m}$)	11.8	0.06	12.5	11.2	0.1	0.22	12.1	150.1	0.21
20-30 ($R = 200\,\mathrm{m}$)	11.9	0.07	12.2	11.7	0.09	0.19	12.1	150.0	0.19
20-30 ($R = 300\,\mathrm{m}$)	11.9	0.07	12.3	11.5	0.09	0.17	12.1	150.0	0.26
20-30 ($R = 400\,\mathrm{m}$)	11.8	0.07	12.3	11.3	0.1	0.17	12.1	150.1	0.34

Table 5.2: Results from different methods with a fixed lidar, for all 10-minute periods and ranges.

As can be seen, for this fixed lidar the Vaisala method seems to perform well in terms of shear reconstruction (at most an error of 0.03 in the shear exponent α_0). α_0 refers to the shear obtained with Vaisala method while α_1 refers to the shear obtained with a full correction method. It also appears that even though the wind velocity comes from a field with consistent properties throughout the 30-minute period, the reconstruction parameters can be quite different for distinct 10-minute periods, due to the effect of turbulence. In particular, this stands out for the power law-fitting method, where for the period between 20 and 30 minutes of simulation, the shear reconstruction is not usable for the ranges of $200\,\mathrm{m}$ and $300\,\mathrm{m}$ (see values of α_1). Regarding this method, in general using a larger range causes poorer results, since the measurement heights from the upper beams and the lower beams are more far apart. It can be valuable however to include measurements from larger ranges because with more skewed profiles than a perfect power law, the trend in the profile shape will not be the same at a height close to that of the lidar than further apart from it.

An advantage of this method is that it is also possible to include the measurements from all ranges into the fitting. The results of merging all ranges for the fitting is:

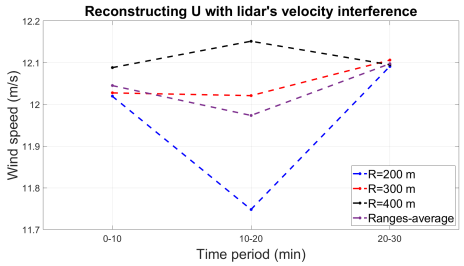
- Between 0-10 minutes: $U_{\text{ref}} = 12.1\,\mathrm{ms}^{-1}$, $z_{\text{ref}} = 150.1\,\mathrm{m}$ and $\alpha_1 = 0.19$
- Between 10-20 minutes: $U_{\text{ref}} = 12.0\,\mathrm{ms}^{-1}$, $z_{\text{ref}} = 149.8\,\mathrm{m}$ and $\alpha_1 = 0.22$
- Between 20-30 minutes: $U_{\text{ref}} = 12.1\,\mathrm{ms}^{-1}$, $z_{\text{ref}} = 149.9\,\mathrm{m}$ and $\alpha_1 = 0.17$

Doing so helps the fitting get closer to the real parameters of the field, especially for the 20 to 30 minutes period, where significant deviation was found by using isolated ranges. Regarding this fitting, another point is particularly important to take into consideration. For a fixed lidar, the measurement heights are fixed as well. Thus, information is only available for, typically here, 12 measurement

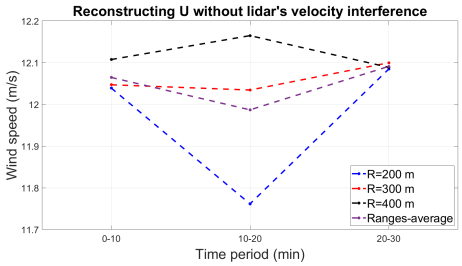
heights (4 beams and 3 ranges). When the lidar is moving, the measurement heights change as well, thus providing more richness for the fitting in terms of height-dependence. Thus, it could be that if the motion of the lidar is correctly compensated/taken into account, the performance of the power-law fitting method is better for a moving lidar than for a fixed one. The counterpart is that it can also increase the uncertainty of the reconstruction. The best performance is of course obtained when the measurement heights and beams' orientation can be accurately known. This will be discussed in what follows. But first, the results of the direct beam-average reconstruction method will be addressed for a moving lidar. To compare to, it was seen that in a fixed lidar configuration, the method ensured reliable output for the mean wind speed at the lidar height, remaining between $11.6\,\mathrm{ms}^{-1}$ and $11.8\,\mathrm{ms}^{-1}$. In terms of Turbulence Intensity however, the estimation did not appear satisfying, see Table 5.2.

Direct beam-average

To make the results easier to look at, plots were made for the resulting reconstructed horizontal mean wind speed at the lidar height and Turbulence Intensity. The results of the method are compared for a VLOS time series that does not correct for the lidar's velocity and one that does. For mean wind speed, results are found in Figure 5.16, and for turbulence intensity in Figure 5.17. As it was written before, the direct beam-average reconstruction does not take into account any change of orientation/-position of the lidar.



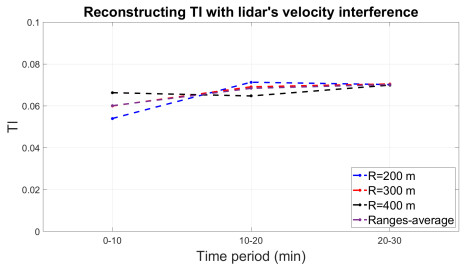
(a) Direct beam-average method on the reconstruction of the mean horizontal wind speed U. The lidar's velocity interferes on the VLOS.



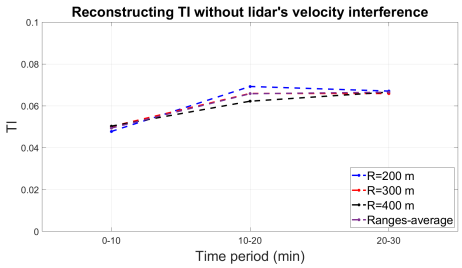
(b) Direct beam-average method on the reconstruction of the mean horizontal wind speed U. The lidar's velocity interference on the VLOS is corrected.

Figure 5.16: Assessing the impact of the lidar's velocity interference on the results of the direct beam-average method : U.

It seems clear from Figure 5.16a and Figure 5.16b that on the mean wind speed calculation, the lidar's velocity has low impact. The difference for all ranges (and for the average of the ranges) between the result with lidar's velocity interference and with correction of this interference are always lower than 1%, at any 10-minute period. What is seen however is that, compared to the fixed lidar case, the reconstructed wind speed is systematically greater, with most values exceeding $12\,\mathrm{ms}^{-1}$. This suggests that the lidar's velocity interference does not have a large impact, but the main challenge is the change in position and orientation of the lidar. Now, how does the lidar's velocity alter the reconstructed Turbulence Intensity from this method?



(a) Direct beam-average method on the reconstruction of the Turbulence Intensity TI. The lidar's velocity interferes on the VLOS.



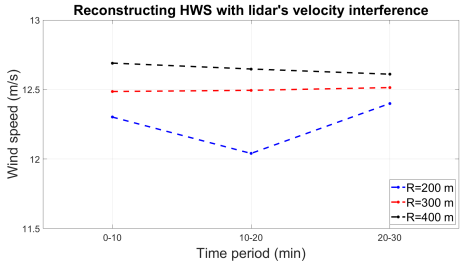
(b) Direct beam-average method on the reconstruction of the Turbulence Intensity TI. The lidar's velocity interference on the VLOS is corrected.

Figure 5.17: Assessing the impact of the lidar's velocity interference on the results of the direct beam-average method : TI.

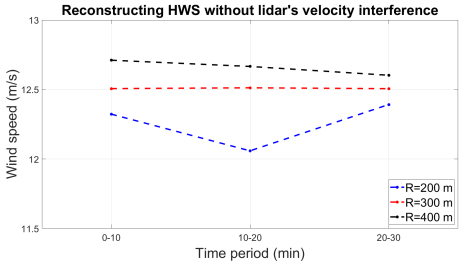
The difference between Figure 5.17a and Figure 5.17b is more visible than for mean wind speed. Generally, the Turbulence Intensity calculated from the uncorrected VLOS time series is higher. This can be explained by the fact that some of the fluctuation of the lidar's velocity is artificially taken into account in the VLOS fluctuations. However, this remains negligible, and whether or not the lidar's velocity is corrected, the turbulence intensity of the field is underestimated with this method. Now the results from the Vaisala method will be studied.

2D Upper-Lower tilt-corrected Vaisala method

This method uses more sophisticated trigonometric formulas that take into account the mean tilt of the lidar and the fact that the wind speed has some transversal component. It also separates the upper beams and the lower beams for horizontal wind speed reconstruction and turbulence intensity. The results are shown for HWS and TI only for the two upper beams for conciseness, in Figure 5.18 and 5.19 . The reconstruction of the shear exponent is also included, see Figure 5.20.



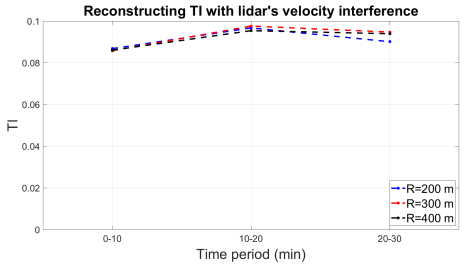
(a) Lidar manufacturer's method on the reconstruction of the mean horizontal wind speed HWS for the upper beams. The lidar's velocity interferes on the VLOS.



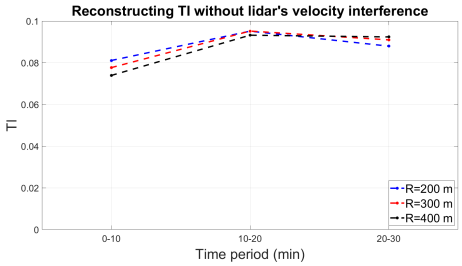
(b) Lidar manufacturer's method on the reconstruction of the mean horizontal wind speed HWS for the upper beams. The lidar's velocity interference on the VLOS is corrected.

Figure 5.18: Assessing the impact of the lidar's velocity interference on the results of the Vaisala method : HWS.

There, the conclusions are similar to that made previously on the beam-average method, the lidar's velocity has little impact on the reconstruction of the mean horizontal wind speed. Regarding Turbulence Intensity (Figure 5.19), again the difference between with lidar's velocity interference and without is more visible, see Figure 5.19a against Figure 5.19b. Once more, the Turbulence Intensity calculated is slightly larger when not accounting for the lidar's velocity. The result is closer to the real Turbulence Intensity of the field than the previous method, with typically lower values than in the fixed case (see Table 5.2). It is accepted that generally, lidars perform poorly for turbulence intensity calculations and need a correction. However here, the reconstructed TI remains close to the actual value of 0.1. This can be attributed to the fact that the lidar simulator only takes into account the spatial averaging of the projected wind velocity, but not the time averaging of the received signal ($\sim 4\,\mathrm{s}$ for some pulsed lidar technologies).



(a) Lidar manufacturer's method on the reconstruction of the Turbulence Intensity TI for the upper beams. The lidar's velocity interferes on the VLOS.

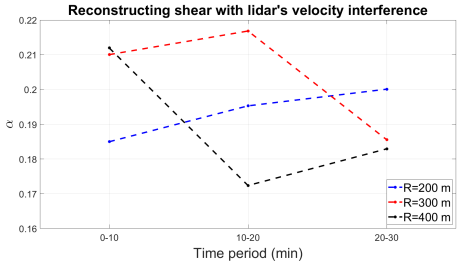


(b) Lidar manufacturer's method on the reconstruction of the Turbulence Intensity TI for the upper beams. The lidar's velocity interference on the VLOS is corrected.

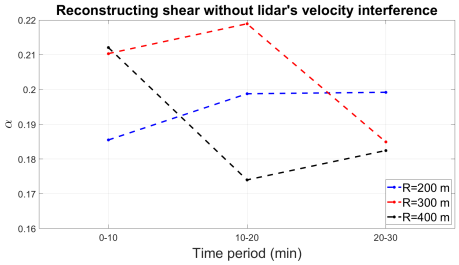
Figure 5.19: Assessing the impact of the lidar's velocity interference on the results of the Vaisala method : TI.

Finally, one can look into the effect of the lidar's velocity on shear. There, both curves are again similar, with close values and trends. This shows that the performance of the method is only slightly affected by the lidar's velocity on shear reconstruction. The shorter range shows better results (closer

to 0.2). This can be again attributed to the measurement heights of the upper and lower beams being closer at short ranges.



(a) Lidar manufacturer's method on the reconstruction of shear α. The lidar's velocity interferes on the VLOS.



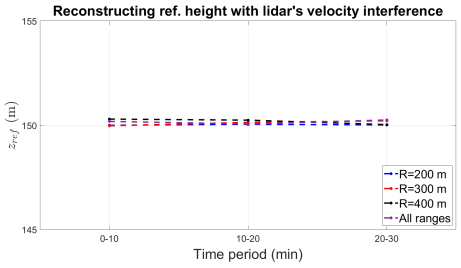
(b) Lidar manufacturer's method on the reconstruction of shear α. The lidar's velocity interference on the VLOS is corrected.

Figure 5.20: Assessing the impact of the lidar's velocity interference on the results of the Vaisala method : α.

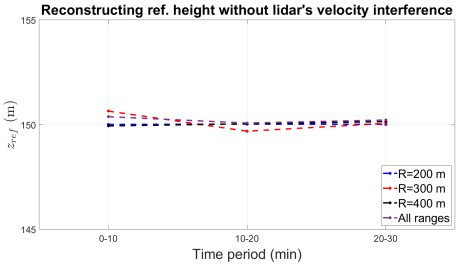
In conclusion, the lidar's manufacturer method seems robust in the case of a FOWT's nacelle-mounted lidar, even by default when not accounting for the lidar's velocity interference. Notably, it performs better than the first method for Turbulence Intensity and gives a reasonable shear reconstruction. The advantage of this method is that it only needs one information on the displacement of the moving lidar: the mean tilt. It is also important to note that for the first 10-minute period, where the lidar's velocity reaches extremes, the Turbulence Intensity is more overestimated than for the rest due to its interference. Finally, the most complete method will be studied. It also allows for what the previous method lacks: a measure of the uncertainty of the reconstruction.

Power law fitting with full correction

This method takes into account the full real position and orientation of the lidar in the reconstruction of the wind field parameters. As it was done previously, the performance of this method will be considered with lidar's velocity interference and by correcting it. Starting with the fitting of the reference height in Figure 5.21, it is harder to draw conclusions here because the non-linear fitting is more sensitive than the simpler averaging methods used before. However, it seems that with and without lidar's velocity interference, the reference height is close to the real value. At most, the largest error is obtained when using the measurements from $R = 300\,\mathrm{m}$ during the 0-10 minute period, and represents less than 1% compared to $150\,\mathrm{m}$. Again, Figure 5.21a and Figure 5.21b have differences but are still extremely close.



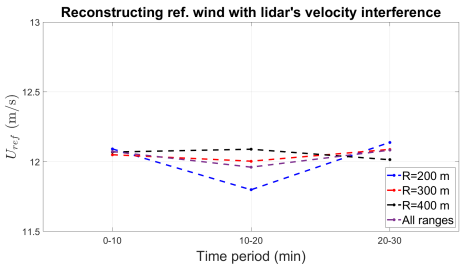
(a) Full correction method on the reconstruction of the reference height z_{ref}. The lidar's velocity interferes on the VLOS.



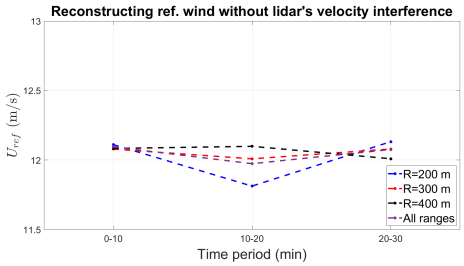
(b) Full correction method on the reconstruction of the reference height z_{ref}. The lidar's velocity interference on the VLOS is corrected.

Figure 5.21: Assessing the impact of the lidar's velocity interference on the results of the full correction method : z_{ref}.

Next, moving onto the reference wind speed, the results are found in Figure 5.22. It is not new that, from comparing Figure 5.22a and 5.22b, the lidar's velocity has a marginal role in the result of reconstructing the mean horizontal wind speed. Additionally, the results of the method remain coherent with the fixed case. This is because this method takes into account real position and orientation of the lidar, and thus the reconstructed wind speed is less artificially increased by measurements coming from different positions than expected. Finally, the last reconstructed parameter α is studied. Results are displayed in Figure 5.23.



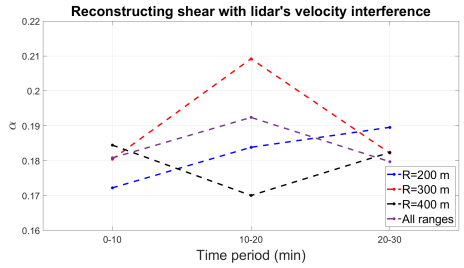
(a) Full correction method on the reconstruction of the reference wind speed U_{ref}. The lidar's velocity interferes on the VLOS.



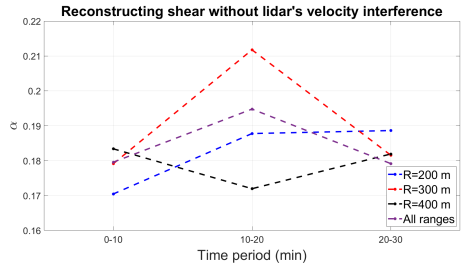
(b) Full correction method on the reconstruction of the reference wind speed U_{ref}. The lidar's velocity interference on the VLOS is corrected.

Figure 5.22: Assessing the impact of the lidar's velocity interference on the results of the full correction method : U_{ref}.

The outcome regarding shear and the influence of the lidar's velocity (Figure 5.23a and Figure 5.23b) on shear reconstruction is comparable to what was written for the Vaisala method: it has negligible contribution. It is interesting to note that this method does not significantly outperform the previous method for shear reconstruction. Moreover, the assumptions stated previously seem to be verified: the power law-fitting method gives better results for a moving lidar for which the measurement heights are more varied, at least in terms of shear.



(a) Full correction method on the reconstruction of the shear α. The lidar's velocity interferes on the VLOS.



(b) Full correction method on the reconstruction of the shear α. The lidar's velocity interference on the VLOS is corrected.

Figure 5.23: Assessing the impact of the lidar's velocity interference on the results of the full correction method : α.

As a way to evaluate the functionality of this method, one can study the quality of the fitted power laws when using all three ranges, first correcting for the lidar's velocity. The resulting lines will be shown against the original ones and the two tolerance ones (using the $\sim$ 2% difference initially accepted on the reference wind speed at $150\,\text{m}$). The outcome is shown in Figure 5.24.

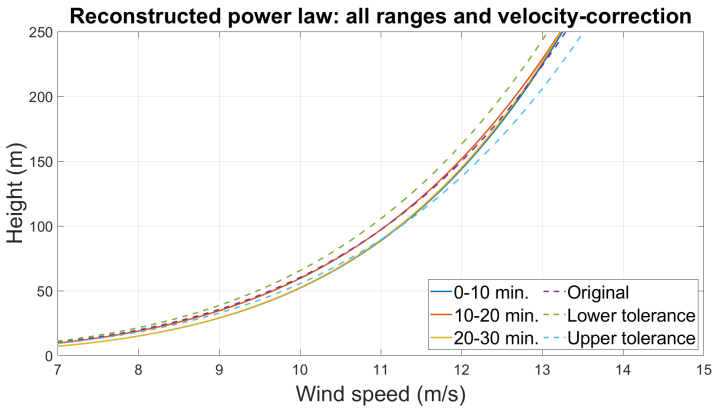


Figure 5.24: Full reconstructed power law between $40\,\text{m}$ and $180\,\text{m}$ using the velocity-corrected VLOS time series and the real position/orientation of the lidar.

The profiles reconstructed between 0-10 minutes and 20-30 minutes are almost indiscernible. 2 of the 3 reconstructed profiles exceed the tolerance bounds for the lower heights, because of a lower

shear. This is understood since the lidar probes mainly around hub height, and not at lower heights. Then, this result is also shown for uncorrected VLOS in Figure 5.25. It verifies that the results with and without velocity-correction are also close in terms of full mean profile, which is argument to the fact that accounting for the lidar's velocity introduces close to no bias. This kind of mean profile reconstruction is extremely valuable at the turbine location and this method is the only one enabling a reconstruction of the power law informed on the real-time height of the measurements. An important discussion on how the fitting performs in different cases and an evaluation of the uncertainty of the mean profile reconstruction is found in Appendix A.11.

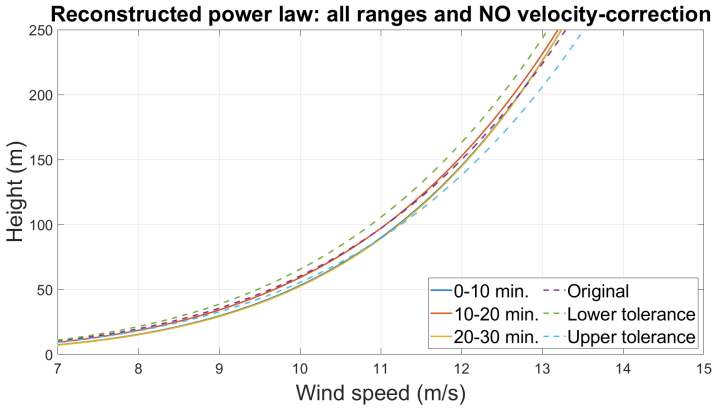


Figure 5.25: Full reconstructed power law between $40\,\mathrm{m}$ and $180\,\mathrm{m}$ using the raw VLOS time series and the real position/orientation of the lidar.

5.2.4. Point-wise vs. Spatial-average correction

The aim of this subsection is to assess the importance of considering the lidar's velocity interference inside of the VLOS integral or simply as a point-wise correction. Since the lidar is moving relative to the global Earth frame, the velocity of the particles with respect to the lidar (what the device measures) and the velocity of the particles with respect to Earth is not exactly the same. Due to this, Equation 4.1 can be rewritten in terms of the position of the particles $\boldsymbol{s_{i,e}}$ in the global Earth frame, the beam direction in the global frame $\boldsymbol{n}$, the velocity of the particles in the global frame $\boldsymbol{v_e}(\boldsymbol{s_{i,e}})$, the lidar's translational velocity $\boldsymbol{v_e^{lidar}}$ and angular velocity $\boldsymbol{\omega_e^{lidar}}$ in the global Earth frame. This gives Equation 5.5.

$$\overline{v_r} \approx \sum_{i=1}^{N} W\left(\boldsymbol{s_{i,e}}\right) \boldsymbol{n} \,.\, \left(\boldsymbol{v_e}(\boldsymbol{s_{i,e}}) - \boldsymbol{v_e^{lidar}} - \boldsymbol{\omega_e^{lidar}} \times \boldsymbol{s_{i,e}}\right) \tag{5.5}$$

To understand better, consider that only points along the axis are modelled, thus $\boldsymbol{s_{i,e}} = R_i\,\boldsymbol{n}$. The term in angular velocity cancels out due to a projection along $\boldsymbol{n}$ of a vector that is orthogonal to $\boldsymbol{n}$ due to the cross-product. This leaves out:

$$\sum_{i=1}^{N} W\left(\boldsymbol{s_{i,e}}\right) \boldsymbol{n} \,.\, \boldsymbol{v_e}(\boldsymbol{s_{i,e}}) \approx \overline{v_r} + \sum_{i=1}^{N} W\left(\boldsymbol{s_{i,e}}\right) \boldsymbol{n} \,.\, \boldsymbol{v_e^{lidar}}$$

This is in fact the formula for the correction of the lidar's velocity interference in the VLOS measurement. The beam direction and the lidar's translational velocity do not depend on the summation index, thus can be left as a factor of the sum. Because the weighting function integrates to 1, it comes that:

$$\sum_{i=1}^{N} W\left(\boldsymbol{s_{i,e}}\right) \boldsymbol{n} \,.\, \boldsymbol{v_e}(\boldsymbol{s_{i,e}}) \approx \overline{v_r} + \boldsymbol{n} \,.\, \boldsymbol{v_e^{lidar}}$$

Thus, in this case, a point-wise correction of the projected lidar's velocity is sufficient. In reality, non-axial particles also contribute to the VLOS measurement. The position of all points can be decomposed

into a component along $\boldsymbol{n}$ and along a component orthogonal to it $\boldsymbol{n}_\perp$ such that $\boldsymbol{s_{i,e}} = R_i\,\boldsymbol{n} + r_i\ \boldsymbol{n}_\perp$. The variation in the previous equation is then only an addition of the following term on the right hand side:

$$\sum_{i=1}^{N} W\left(\boldsymbol{s_{i,e}}\right) \boldsymbol{n}\,.\,(r_i\,\boldsymbol{\omega_e^{lidar}} \times \boldsymbol{n}_\perp)$$

This term increases when r_i increases. However, when r_i increases, particles are further away from the beam axis, thus the corresponding weight gets (literally) exponentially smaller. The contribution of this term is then expected to be negligible. This was verified in this given simulation case, where as a reminder the lidar simulator uses 100 axial points and 9 radial points per beam and per range. For the simulation at the last range and for an upper beam, the difference between the point-wise correction and the correction taking into account the averaging of this angular velocity term is shown in Figure 5.26. The peaks at the start corresponds to the same extreme event that caused surge and pitch angle to reach severe values. Indeed, angular velocity in the y-direction also peaks at the same time. Note that this is just an example with a quite low number of radial points but this result should be widely expected. In conclusion, correcting for the lidar's velocity in a point-wise manner and by neglecting the impact of its angular velocity in the VLOS seems largely acceptable (error of less than $1 \times 10^{-14}\,\mathrm{ms}^{-1}$, at the limit of machine precision).

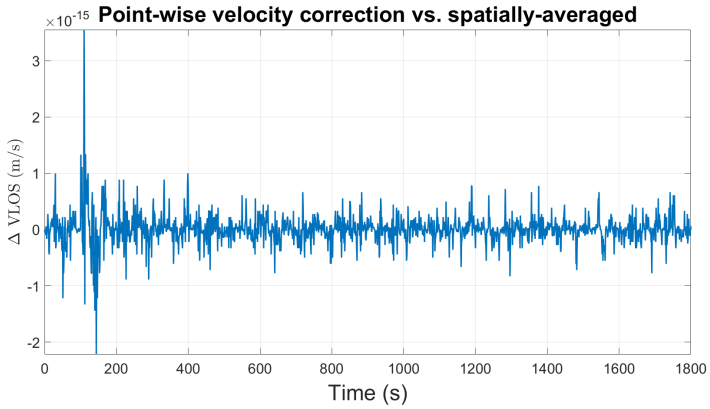


Figure 5.26: Assessing the impact of taking into account the interference of the lidar's angular velocity into the correction.

5.3. Introducing interpolation and noise

A disclaimer is needed before starting with this section. The lidar simulator used here corresponds to the Range-Weighting function of a CW lidar, which is inherently different than that of a pulsed lidar. The lidar still retrieves the wind speed at different ranges at a time, which does not correspond to the functioning of a CW lidar. The adaptation to the case of a pulsed lidar as it was done in the previous section was not possible due to a lack of time. However, given that the Range-Weighting functions have similar properties in both cases, the results of this section are still valuable and should be treated qualitatively.

The two first methods discussed in Section 5.2.2 have the advantage of simplicity. The first method uses no additional data on the lidar motion so it cannot suffer from interpolation and noise in the motion data. The second method uses only the mean pitch of the lidar. The lidar's pitch is given by the lidar itself so there is no time offset with the VLOS measurement. Regarding noise, Vaisala uses an average so that has the advantage of being less impacted by noisy data. This is why it does not matter much for the manufacturer to have a high-performance inclinometer, with less advanced filtering than in the INS for example.

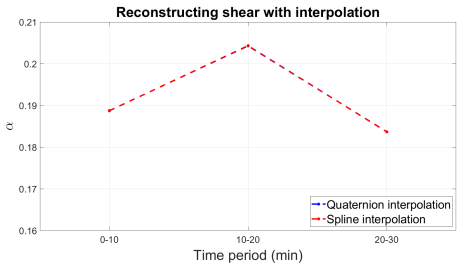
Finally, the last method however uses real-time position, orientation and velocity of the lidar, at a different time than the samples of the lidar VLOS. Errors can come from the initial inertial data, from

the fact that this data is not exactly at the same location as the lidar, and from interpolation. The impact of interpolation and noise on the reconstructed quantities will be studied in this section. The main objective of the section is to compare the quaternion interpolation method developed in the course of this project and another interpolation method. The latter method relies on piece-wise spline interpolation of Euler angles and angular velocities separately. As it was addressed in the implementation of quaternion interpolation, this is far from optimal but there are chances that this is sufficient due to the high sampling rate of the INS. Then, noise will be added to the motion data and again the two methods will be compared.

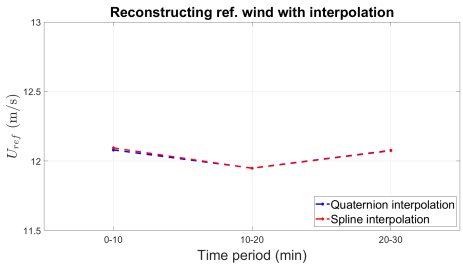
Finally, the output time from the IMU was set to every $0.04\,\mathrm{s}$, as in the TetraSpar prototype and the output time of the lidar every $1\,\mathrm{s}$, but falling right in between two sampling times of the IMU. Thus, there is always a $0.02\,\mathrm{s}$ time offset between the lidar VLOS output and the closest (in time) IMU output.

5.3.1. Performance of interpolation methods

Position and translational velocities are interpolated using the Quintic Hermite interpolation described previously. Orientation and angular velocities are interpolated either using the quaternion method or the spline method. The results of both interpolation methods are shown in Figure 5.27, for shear in Figure 5.27a and reference wind speed in Figure 5.27b.



(a) Full correction method on the reconstruction of the shear α with interpolation. The lidar's velocity interference on the VLOS is corrected.



(b) Full correction method on the reconstruction of the reference wind speed U_{ref} with interpolation. The lidar's velocity interference on the VLOS is corrected.

Figure 5.27: Assessing the difference in reconstruction when using two different methods on the interpolation of orientation and angular velocity: α and U_{ref}.

No clear difference between the two methods appear, and the behaviour is extremely similar to what was seen without interpolation. Now for reference height the difference is highlighted in Figure 5.28. The results of spline and quaternion interpolation are now distinct, with quaternion interpolation performing slightly better for the first 10-minute period. The difference remains however quantitatively low and it would be inadequate to state that a given interpolation method is better than the other. Next, one can check what happens with noisy data.

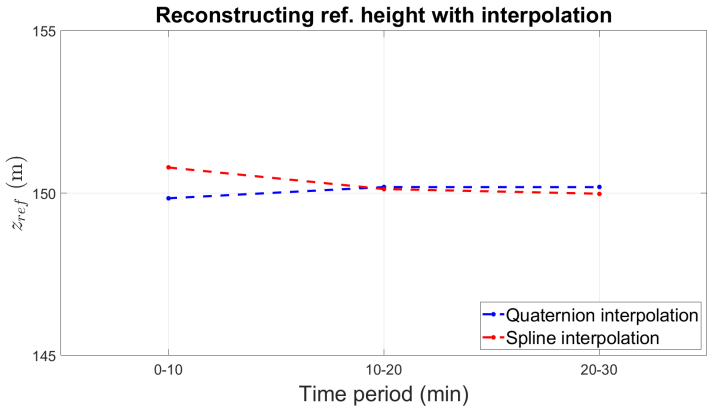


Figure 5.28: Assessing the difference in reconstruction when using two different methods on the interpolation of orientation and angular velocity: z_{ref}.

5.3.2. Performance of algorithm and methods with noise

All kinds of noise types and sources exist that correspond to different phenomena. To remain as general as possible, in this case Gaussian white noise was added to the signals. The resulting curve for the pitch angle for example is depicted in Figure 5.29. As it was seen in the real measurements from the TetraSpar prototype's INS, the SNR for acceleration was made lower than for velocity and the SNR for the velocity signal lower than that of position.

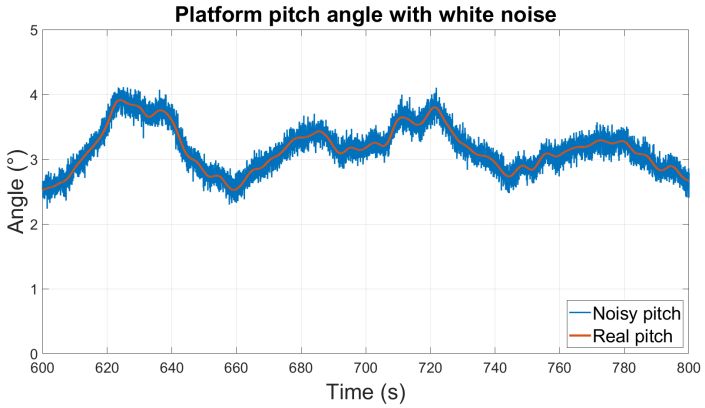


Figure 5.29: Introducing noise in the signal for pitch angle.

With an appropriate choice of nodes and by using the data from position, velocity and acceleration for the fitting of the control points, Quintic-Hermite interpolation shows to be a reliable method to interpolate noisy signals. For a noisy surge for example, its interpolated curve is seen in Figure 5.30.

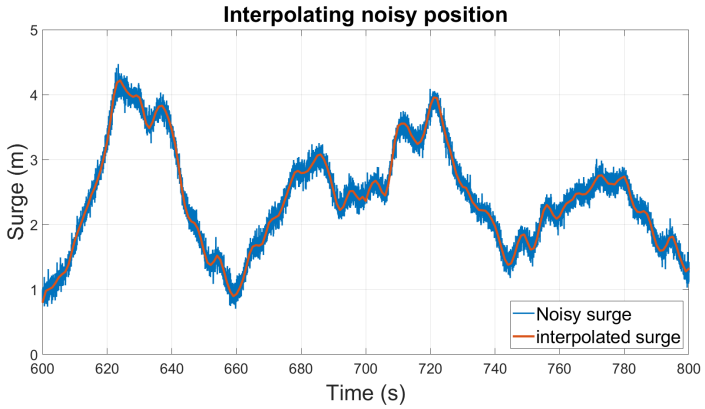
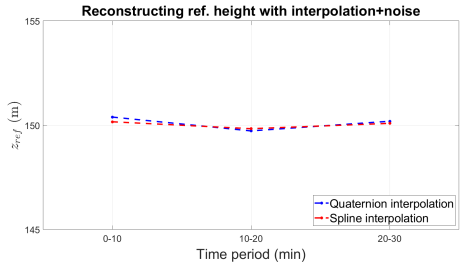


Figure 5.30: Interpolating a noisy position signal.

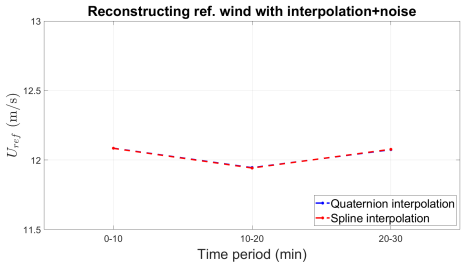
Now that the noisy signals have been introduced, it is possible to see how this affects the reconstruction results of the power-law fitting method. For reference wind speed and reference height, the outputs are shown in Figure 5.31.

Again for reference wind speed, the two curves seem to merge. For the reconstruction of the reference height, the spline interpolation method now has better results than the quaternion interpolation method, even though the difference is extremely low. Finally, the real difference comes in the shear reconstruction, see Figure 5.32.

With the spline interpolation method, the results are almost exactly the same than without noise. For the quaternion interpolation method however, the reconstructed shear is higher than without noise. This artificial increase in shear with quaternion interpolation can be purely attributed to the presence of noise in the angles and angular velocities. Thus, if the results are better it is merely a coincidence. It is unfortunate for this project, since a simpler (and less computationally costly) method using Matlab built-in splines seems to perform better than the more mathematically advanced and theoretically better quaternion interpolation method. The high performance of the measuring device (high sampling rate of the INS making the interpolation period short) enables the use of simpler interpolation methods.



(a) Full correction method on the reconstruction of the reference height z_{ref} with interpolation of noisy motion signals. The lidar's velocity interference on the VLOS is corrected.



(b) Full correction method on the reconstruction of the reference wind speed U_{ref} with interpolation of noisy motion signals. The lidar's velocity interference on the VLOS is corrected.

Figure 5.31: Assessing the difference in reconstruction when using two different methods on the interpolation of noisy orientation and angular velocity: z_{ref} and U_{ref}.

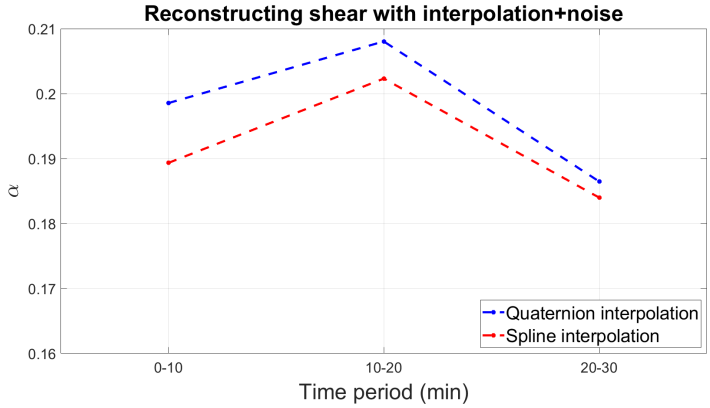


Figure 5.32: Assessing the difference in reconstruction when using two different methods on the interpolation of noisy orientation and angular velocity: α.

Returning to the core issue, this means that the quaternion interpolation method is sensitive to inaccuracies in the orientation and angular velocity data. Directly, two reasons for this can be thought of:

- As it was addressed in the second Section, the derivation of the first derivative of the quaternionic Bézier curve can be faulty. This would cause errors in the interpolation of the *angular velocity*. However, as it was shown before, the interference of the lidar's velocity has low impact and therefore could not explain such a significant upwards shifting of the reconstructed shear.
- The Bézier quaternion interpolation method interpolates between two time stamps using control quaternions that depend on the real values of quaternion and angular velocities at the nodes. Due to noise, the derivatives of the quaternions at the nodes are not exactly linked to angular velocity by Equation 2.27. This causes faulty control points and thus an interpolated quaternion that does not satisfy the precision of interpolation defined by theory. This impacts the beam directions and measurement heights used for the fitting. It is apparent that this is the best explanation on the sensitivity of quaternion interpolation to noise.

Nevertheless, it is reasonable to conclude that the shown reconstruction of wind field parameters is reliable in the context of interpolation and moderate noise.

5.4. Economical impact and system integration

These results shall be used in consideration of other criteria than pure performance. If the aim is only to obtain the best theoretical reconstruction, then one could argue that an IMU should be installed as close as possible to the lidar and sampled at matching times to avoid interpolation errors. Then, this will be used as an input of sophisticated algorithms for mean profile and turbulence reconstruction. Now, one can also decide that this is not necessary and to neglect some of the lidar motion.

Of course, that will decrease the reliability of the reconstructed parameters, but requires less equipment and time. This also simplifies the whole system for wind field reconstruction using lidar output. In this case, the system is only made of the lidar and the software functionalities that allow for these outputs to be stored in a database and accessed by engineers for example. Ramping up in complexity, one can include inclinometer data inside of the lidar, and even further full inertial data from an advanced INS. It is understood that each step of complexity adds a layer of equipment, more parts to assemble to form the final system, and thus more money and time. On the matter of system integration, it can be defined as **"the process of deliberate assembly of the parts of the system into a functioning whole"** : physical assembly of parts, connecting electronics to power sources and uploading of test and operational software for example [83].

5.4.1. Economical study

To start with, one can wonder why all this interest on correcting lidar measurements. As for most of industry-motivated research, there is an economical interest. The first interest is in terms of prototype certification: power curve and loads validation. The second one is to determine what devices and methods should be used to correct the measurements. Different equipment and methods for correction will have a different cost and this will be addressed.

To start with the first point, having a reliable reconstruction of incoming wind field is essential to validate the power curve of a turbine and the predicted loads. Although this is a well-defined process in general, in the case of floating wind turbines, it remains a problem due to the complexity of installing reliable wind measurement devices. First, one can look into the importance of having well-defined power curves for specific turbines. It essential to measure the power curve in real site conditions. In fact, it is an IEC standard (61400-12-1) to determine the power curve by simultaneous measurements of wind speed at hub height and power output at a test site. The guidelines prescribe the use of anemometers at the relevant sectors [74] but, as it was treated in the report, this is far more challenging for FOWTs. In this case, lidars are more attractive for several reasons. First, they are cheaper and easier to install than a met mast equipped with anemometers offshore. But also, being at the height of the nacelle, they can sense the wind speed close to the hub height. Met masts would typically require extrapolation at much higher heights than the measurements, due to the hub being particularly large offshore. However, using lidar measurements as a reliable way to reconstruct the incoming wind field is still at the stage of research. This is why this project is fully in line with the development of Floating Offshore Wind Turbines.

Back to the economical side, having an accurate power curve and reliable wind measurements at a specific site of installation is crucial. Indeed, errors in the wind speed estimation or in the power curve can lead for example to over-sizing of the wind energy conversion system, which directly causes money losses. Moreover, power curves can also be the base for sizing large energy systems, made of batteries and solar panels for example [19]. If the power curve is not well determined, or if the wind characteristics are not well estimated, this will also cause inevitable cost issues. In conclusion, the proper estimation of the power curve in real conditions will have a significant impact on the cost of a project. If the power produced is overestimated, the reliability of the system is endangered and it will under-perform. On the other hand, if it is underestimated, the sizing of the system will be too large, leading to unnecessary expenses.

To put this in numbers, an example will be considered where the importance of the nacelle's motion will appear. It was shown in literature that the reconstruction of the power curve for FOWTs using nacelle-based lidars is sensitive to the motion of the nacelle, in particular to the standard deviation of its acceleration [59]. It is shown that for a quarter of the time, when the standard deviation of the nacelle's acceleration is highest, the difference in power on the power curve around the transition

to rated wind speed is of maximum 4.5% compared to the times when the standard deviation of the nacelle's acceleration is the lowest. This means that the motion of the nacelle can affect the lidar measurements such that the measured power curve is in fact lower above rated wind speed than the "real" one, which is closest to times when the nacelle has less motion, thus interferes less with the lidar measurements.
Consider that no interest is payed to the motion of the nacelle-based lidar and that the reconstructed power curve is measured for a prototype in an environment that corresponds to significant motion of the nacelle. One will then conclude that at rated wind speed, the turbine is going to produce a certain power P_{meas}, which is 4.5% below the real value P_{real}. The total energy production of a farm made with this turbine for a year is aimed to be E_{aim}. One can then decide, with a capacity factor C, to place N turbines such that $E_{aim} = N\,C\,P_{meas}\,T$. That is $N = \frac{E_{aim}}{C\,P_{meas}\,T}$.

However, the real production is going to be closer to $E_{real} = N\,C\,P_{real}\,T$. By replacing N, it comes that $E_{real} = \frac{E_{aim}}{C\,P_{meas}\,T}\,C\,P_{real}\,T = E_{aim}\,\frac{P_{real}}{P_{meas}}$.

In other words, the farm will produce 4.5% more energy than it is made to. In consequence, the farm can be sold for less than what it is worth. Consider an example of a rated $100\,\mathrm{MW}$ farm with a capacity factor of around 40%, that is an annual energy production of around $350\,\mathrm{GW\,h}$. Taking NREL's calculation on the capital cost of floating offshore wind turbine ($\sim$ 93\$/$\mathrm{MW\,h}$) [76], that represents a unnecessary expense of millions of dollars. In reality, the difference in production will not be as much, because even if the wind speed is overestimated by the lidar, at a certain point in wind speed, the rated power will be reached anyway. This difference happens mostly when the wind speed is close to rated. These are gross calculations but it shows that given the size of wind energy projects, accuracy in power curves are immensely valuable.

Next, the importance of loads validation can be addressed. Again, the interest is to have a correspondence between a wind field and a turbine response, but this time with loads instead of power production. It was shown that nacelle-mounted lidar measurements (for non-floating turbines) are completely valid to estimate loads due to incoming wind field characteristics [65], with even less uncertainty than cup anemometers. This has obviously an economical interest, with lidars being cheaper to install. Additionally, accurate wind speed measurements are needed. Indeed, if the wind speed or the turbulence intensity are overestimated due to the lidar motion, it will give an estimation of loads that actually correspond to less extreme conditions than predicted. The loads prediction will then not be correct, which is a risk to the turbine lifetime. Approximately for floating offshore turbines, the production per year is around $3.3\,\mathrm{GW\,h}$ / MW installed [76]. Any loss in lifetime of the turbine will thus cause huge financial losses.

Finally, another economical aspect is that of the cost of the devices needed for the motion-correction. The two devices and their cost (estimated with the help of people involved in the TetraSpar project) are as follows:

- Nacelle-mounted lidar: $\sim$ €200 K.
- SBG Inertial Navigation System + GPS dual-antenna: $\sim$ €20 K.

The cost of the INS device is around 10% of that of the lidar. Compared the cost of the turbine, this is even more negligible. It appears that on a financial aspect, it is not an issue to include motion from such an inertial device. However, if one only wants to reconstruct mean wind speed and Turbulence Intensity, it was shown that the lidar's velocity has a low impact on the reconstruction and that the cost of such a device can be saved. If it is decided to use the INS device, the challenge appears to be more in terms of system integration than in finances. This will be seen in the next Subsection.

5.4.2. System integration

This subsection will address the motion-correction algorithm itself, considering the lidar and motion-correction assembly as a whole system that outputs wind field characteristics. According to NASA's *Systems Engineering Handbook*, a system is **the combination of elements that function together to produce the capability required to meet a need**. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose; that is, all things required to produce system-level results. In this case, the system's results are reconstructed wind field characteristics.

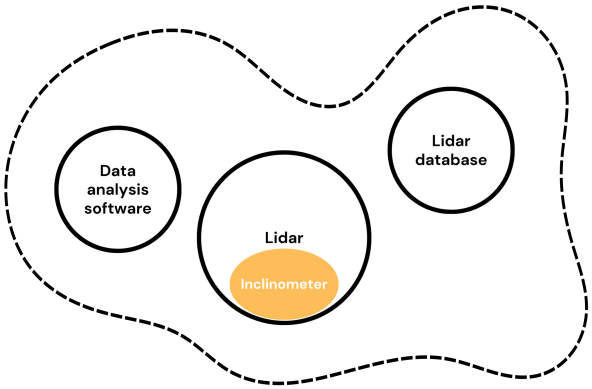


Figure 5.33: Elements of a wind field reconstruction system using only the data from the lidar.

To start with, the initial setup of the lidar is examined. The elements of such a system are found in Figure 5.33.
As shown there, it is a rather simple system. The mode of operation of this system to reconstruct wind field characteristics for 10-min periods is also rather simple, and it is found in Figure 5.34'. The calculations of Horizontal Wind Speed, Turbulence Intensity and shear are made internally by the lidar, using the tilt from the inclinometer so on a system level, there is not much to do. The availability check for the data is also given by the lidar. It is set to 0 if there is blade blockage or if the Carrier-to-Noise Ratio is too low, otherwise 1.

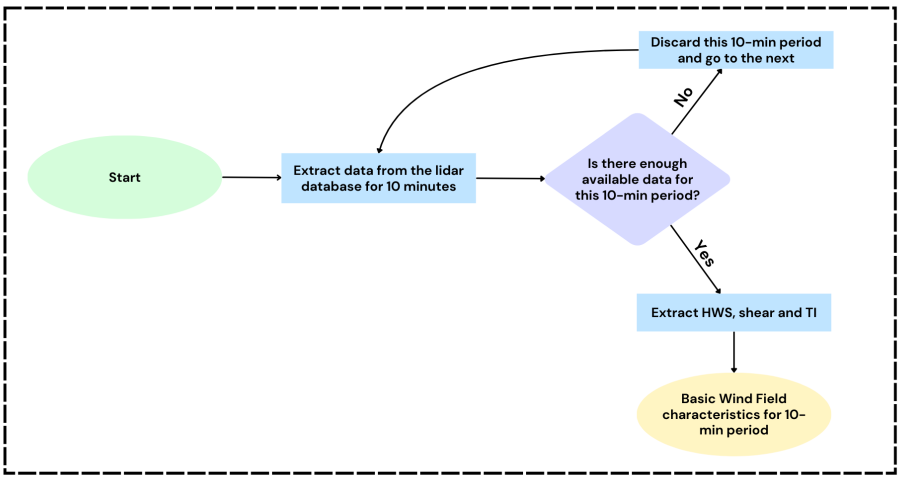


Figure 5.34: Process of this first simple system to result to Wind Field Reconstruction.

Now, one can consider a more complex system that uses the data from the INS device. The aim of adding this device is to have a more accurate, in principle, tracking of the orientation of the nacelle and to know the lidar's velocity and height. The components of this new system are shown in Figure 5.35.

Some complexity is brought to the system by the addition of two new components, one being the INS device itself (hardware) and the second being the database where the data is extracted and stored (software). For the TetraSpar prototype, the data from the INS and that from the lidar are stored on two different databases, which do not facilitate their integration.

Now the process is slightly different since the data from the INS is included and used in the Wind Field Reconstruction. It is shown in Figure A.20. One can see that the process gets significantly larger in terms of step. The additional complexity compared to the previous system can be decomposed in several points:

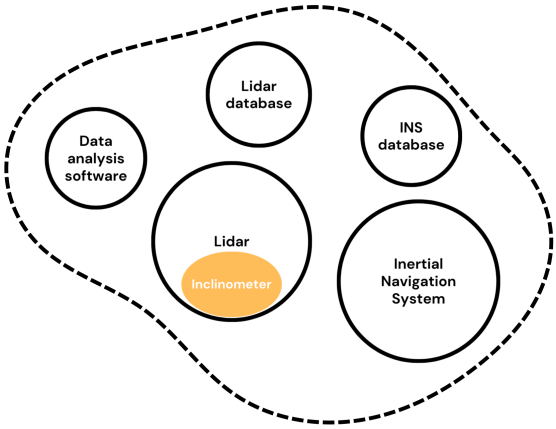


Figure 5.35: Elements of a wind field reconstruction system using the data from the lidar and from the INS.

- The data from the lidar and the INS are stored on two separate databases and servers (at least for the TetraSpar prototype) that use different conventions for their time protocol. Therefore, one has to check that both data are aligned in time such that the timestamps of the lidar's measurements can be exactly mapped to a timestamp in the INS' reference.
- The unavailability of the data can come from the lidar side (as it was written before), but also from the INS device. Problems can come for example from software updates, maintenance, or from unavailability of GPS data.
- Data have to be interpolated to match the timestamp of the lidar's measurements and, in the rigid-body assumption, extrapolated at the lidar's position, which is above the INS.
- Reconstructions of the mean wind profile and of turbulence have to be done with separate algorithms (for example in Matlab), which increase the computational time.

On top of this, one can also add the fact that technicians have to intervene on site, not only for the lidar, but also to calibrate the INS, to measure its position in the nacelle, and for maintenance/repair, which complicates even more the process in long-term and increases its cost.
Although it is clear that using the INS causes system integration difficulties, both systems presented are used in post-processing of the data. So for example after a few months of data collection, the presented processes can be applied for all desired 10-min periods. Because at the same time, loads and power production are tracked, one can reconstruct real power curves and validate loads prediction. There is no hard time constraint on these processes, whether the reconstruction takes a minute or a day should not have a large impact.

However, if the corrected lidar measurements are also used for control, computational time and efficiency of the process play a larger role, given that this system is now merged into a much larger system: the wind turbine and its controller. Consider for example the block diagram in Figure 5.36 representing how a lidar can be used in a preview based combined feedforward/feedback control routine [72].

If only the lidar is used (first simple system of Figure 5.33 for example), the block corresponding to the lidar measurements in Figure 5.36 is not expanded and remains as it is. On the other hand, if the INS is included in the Wind Field Reconstruction system (as in Figure 5.35), the part of the control between the lidar measurement and the wind evolution model undergoes consequent changes, where the part between the wind and the controller will resemble the process shown in Figure A.20. In this case, one is no longer dealing with a post-processing task that can be thoroughly checked and spread-out in time but with an application of the lidar measurements that will affect the real-time behavior of the turbine.
Thus, one should be extremely careful in the errors that can happen during the wind field reconstruction process and make sure that all tasks are completed within a reasonable time window. It is for now clear that lidar used on the TetraSpar prototype is not aimed at control, but in case it is in the future or for other prototypes, this is a key aspect.

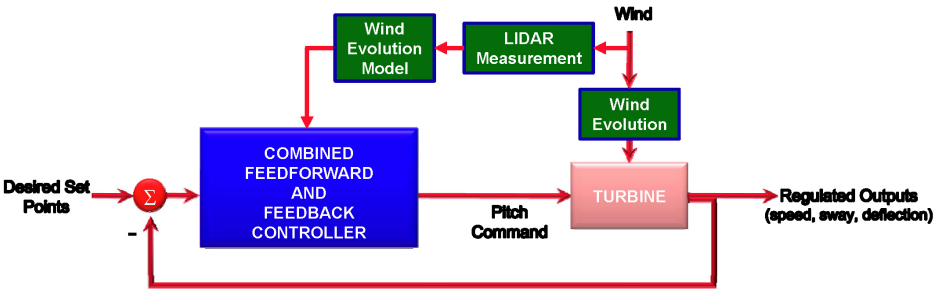


Figure 5.36: Block diagram of lidar-assisted wind turbine control [72].

In conclusion, the argument made here is that the main challenge of using an INS to correct for lidar measurements and reconstruct the wind field on FOWTs is not so much in the price of the equipment but more in its integration to the wind field reconstruction system. This becomes even more crucial when considering lidar-assisted control.

6

Conclusions and Recommendations

Throughout this project, many demanding steps had to be completed before being able to get to the final product: reconstructing a wind field using data from a moving lidar. Among these steps, here are the most important.

The existing lidar model at Siemens-Gamesa had to be extended to include a specific type of pulsed lidar model and to accept quaternions everywhere rotation matrices can be used. The mathematical and algorithmic framework had to be developed for the interpolation of position, velocity and acceleration using Quintic Hermite basis functions. Moreover, this also had to be done for orientation and angular velocity, with a whole new challenging branch of mathematics using quaternions. Then, a study was conducted on the use of real lidar and IMU data on the TetraSpar prototype, which unfortunately did not come to fruition given the unavailability of entire data channels. Finally, focusing on simulations, a coupling had to be found between the wind field used in OpenFAST to extract FOWT dynamics and that used with the internal lidar module.

The completion of these steps is detailed in the report, for one can judge the quality of the work. The final results presented will help answer the initial research questions formulated in Introduction. Generally, the aim of this Master's thesis project was to develop an approach to take into account the motion of a nacelle-mounted lidar, to reconstruct incoming wind field. It was also to start looking into the limitations of such an algorithm, since reality is made of imperfect measuring devices, as well as non-ideal setups, containing spatial and temporal issues. In conclusion, the answers obtained from this research are found in Section 6.1. Since so much is left to be done, from uncertainty quantification, to upgrading of the lidar model and real data validation, recommendations on further work and on the use of this project's results are stated in Section 6.2, including flaws that one should consider after reading the report.

6.1. Conclusions

- **Does the lidar simulator constitute a good model of the measurements of a moving lidar in a turbulent atmosphere?**

Considering first a fixed lidar, it was shown that the simulator has expected properties in terms of pure VLOS measurements in the sense that it gives an accurate measure of the mean wind speed at the measurement positions, but an underestimation of turbulent fluctuations. The spectrum of the measurements is affected and a link exists with the Fourier Transform of the Weighting Function.

Regarding the integration of motion, the simulator is able to take into account the own velocity of the lidar and changes in orientation/position. As expected, the interference from the velocity of the lidar is negligible for mean quantities. The change in orientation can be seen as a peak in the spectrum of the VLOS with simple harmonic pitch, as it was shown in research.

- **Through simulations and prescribed wind fields, do the lidar measurements and the motion-correction algorithm enable to reconstruct the wind field? How do other methods perform?**

Simulations have shown the results of different methods for wind field reconstruction when considering a simulated nacelle-mounted lidar on a FOWT. Before diving into the method that corrects for the full motion of the lidar, other methods and their performance will be addressed. For a fixed lidar, a simple method on averaging the measurements from all beams gives the correct order of magnitude in terms of wind speed at the lidar height. Even when considering a moving lidar, this method remains in the same order of magnitude, although the motion of the lidar causes an overestimation of the wind speed at the lidar height (between 1% and 3% increase compared to fixed case). In both cases, the Turbulence Intensity estimated is not usable.

A more complete method proposed by the lidar manufacturer has better results on the estimation of Turbulence Intensity. With motion of the lidar, the estimated Turbulence Intensity is lower than in the fixed case, degrading the reliability of this method for accurate Turbulence Intensity reconstruction when the lidar has such motion. The reconstructed mean Horizontal Wind Speed is also overestimated at the measurement positions when some motion is considered, but that is again only by less than 3%. Regarding shear reconstruction, the Vaisala method has a good performance overall, with better results for a short range.

Finally, the last method corrects for the lidar's velocity and takes into account the change of coordinates of the measurement points of the lidar, due to its motion. This information is then passed to a fitting algorithm that reconstructs the mean horizontal wind profile. This method, intially developed by fellow students (R. Tiphaigne and M. Hitzerd) and used here, is made to reconstruct the parameters of a power law for the mean wind speed, using non-linear least squares fitting of the VLOS measurements. For simulation, where the mean wind speed is an *actual power law*, this method has shown encouraging results, enabling to reconstruct the full mean profile within tolerance boundaries. It also allows for an estimation of the uncertainty of the fitted profile. It is remarkable that this method performs as well in the fixed case as in the moving case. A limitation is that this requires accurate knowledge on the lidar's orientation and height in time. Another limitation seems to be that even in this simulated case using precise data (without interpolation and noise) and the same incoming wind field parameters, the reconstruction results are not exactly consistent between different 10-minute periods. This also highlights the complexity of turbulence modelling. Indeed, even by proper setting of the parameters, the mean horizontal wind speed can be different than the desired value. Moreover, the mean transverse and vertical components of the wind are not exactly zero, which the power-law fitting method assumes to be the case. Nevertheless, the results of such a method that is informed on the position and orientation of the lidar are promising. The reconstruction of turbulence is not discussed in this report.

- **What criteria should be used to draw conclusions on the right method to use (performance, cost of having an inertial device in the nacelle, cost of storing the data, cost of having this algorithm in place...)?**

Not accounting for the lidar's motion is clearly the easiest and most straightforward method. If the aim is to only get an order of magnitude of the horizontal wind speed at the hub height, then simulation results show that this method performs well enough. Using an average of the VLOS measurements at all beams is however not reliable enough to estimate Turbulence Intensity.
The method proposed by the lidar manufacturer then has the advantage that the mean tilt can be estimated without adding any components to the system, since an inclinometer is already included in the device. It also has an advantage over the previous method, which is that it accounts for a mean tilt of the nacelle. If the tilt angle oscillates around a non-zero mean (which has shown to be the case in simulations and reality), then this method adjusts the mean increase in height of the measurements. If the aim is to have an order of magnitude of the wind shear and of the mean horizontal wind speed at different heights, a better representation of Turbulence Intensity (expecting an error of no less than ± 0.02 on TI), then this method can be exploited. However, for more precise reconstruction of the mean wind speed at all heights, accounting for a more complete real-time motion is preferred. For this, two options can be separated.
The first option would be to account for the change in position, orientation of the lidar and the interference of its velocity in the VLOS. Even if the lidar's velocity has negligible impact on the reconstruction of mean quantities, it was shown that small (compared to the mean wind speed) interference can affect significantly the VLOS spectrum and thus perturb the turbulent properties contained in the VLOS signal. If one hopes to model the small-scales turbulent behaviour, then accounting for the lidar's velocity inter-

ference is necessary. This requires a device similar to the INS in the TetraSpar prototype. This device has a cost, which remains about 10 times lower than the cost of the lidar, so that does not appear to be an obstacle.

What is more challenging is the integration of the data of this INS to complete the lidar data. The INS is not placed at the same location as the lidar, and the outputs are not given at the same time. This requires interpolation, which adds up to the errors of the final results, and to computational time. If the objectives are such as wind field reconstruction, loads validation or energy prediction, then the data can be post-processed without *real* time limits. Having the motion data and the lidar data on two different databases is, in this case, also not a problem. On the other hand, if an objective is to use corrected lidar data for real-time control, then the integration of all steps to the controller in real time will be an obstacle.

The other option would be to neglect the interference of the lidar's velocity, and to only use the real height and orientation of the lidar for mean wind field reconstruction. Such outputs can come directly from the lidar, such as tilt and roll already do. This would lower the cost (less advanced and expensive device than the current INS) as well as steps to system integration, if this data can be aligned with the VLOS in time.

- **What is the performance of the discussed interpolation methods and motion-correction methods?**

For interpolation of position and translational velocity, the Quintic Hermite interpolation method has shown to be an excellent candidate, being adjustable to follow more or less the fluctuations of the real data. Regarding orientation and angular velocity, this is far more challenging, as this report attempted to describe. Quaternion interpolation methods are theoretically more attractive, since they better encapsulate the properties of rotation and enable smoother interpolation. It seems however that the method developed in this project, using the de Casteljau algorithm, has a high sensitivity on possible errors in the real data. Given the high sampling rate of the INS, simpler methods on the interpolation of Euler angles can also be used.
To simulate the VLOS measurements of a lidar, including the spatial-averaging effect has a significant impact on the result. However, for what consists of correcting the lidar velocity, that in principle affects the velocity of each particle with respect to the lidar, the point-wise approximation is enough. Moreover, if the reconstruction of the mean wind profile is required, this velocity interference can be neglected with low impact on the performance of the reconstruction.It even affects only by little the estimation of Turbulence Intensity.
Besides, it appeared in the course of this project that the frequency filtering method is not necessarily exploitable, for the simulated FOWT and for the real TetraSpar prototype case. Finally, it should be reminded that the performance of the shown motion-correction approach is dependent on the uncertainty of the motion data, which has been introduced in the report, of the lidar data itself (impact of the SNR), and of the uncertainty produced by the mean wind speed and turbulence fitting methods to reconstruct desired parameters.

6.2. Recommendations

The recommendations for future work to follow can be divided into different sections.

- **Lidar model**

To make the numerical model of the lidar simulator better and computationally efficient, the calculation of the integral modelling the measurement of the radial velocity can be improved. Knowing the expression of the Range-Weighting function, the quadrature optimizing the positioning of the sampling points for the approximation of VLOS integral (in one dimension) should be determined. This will improve the accuracy of the integral and give the possibility of using less points for a better result, thus reducing computational time.

Then, regarding the Range-Weighting function, its expression should be better determined to make a good fit with the reality of the WindCube Nacelle lidar model used in the TetraSpar prototype. In this

report, the matching was made roughly, using the available information. To simulate this type of lidar and to apply a better wind field reconstruction with real lidar data, this function should be made as close as possible. Also, as it was mentioned in the report, time was ultimately lacking due to some carried-through mistakes. Thus, Sections 4.1.2 and 5.3 can be re-performed with the correct Range-Weighting function that corresponds to the pulsed case.

Finally, another improvement of the lidar model could be to include the temporal variability of the wind field in the measurement of the VLOS, and to go through the actual end-to-end measurement process using Doppler Spectra. Indeed, for now, the VLOS is modelled by the Range-Weighting integral equation. The integral is calculated using the spatial distribution of the wind velocity at a given and fixed time. However in reality, as the backscattered signals coming from a specific range gate are integrated, time passes so the wind field also changes. Also in practice, the line-of-sight velocity at a certain range is calculated by averaging Doppler spectra during the corresponding observation time. This is done on the lidar post-processing side. Thus, the use of simulated Doppler spectra as a way to retrieve the VLOS gives room for improvement. The results of this method can be compared to the results of the current method.

- **Further simulations**

Further simulations can be ran to verify the validity of the current results in different configurations, and also to verify the assumptions.

First, the rigid-body assumption in the nacelle should be verified. This can be done using the simulation software BHawC-OrcaFlex and choosing to output angular velocities and Euler angles at different locations in the nacelle to compare. Also as a comparison, these quantities can be computed at the tower bottom for example. The values in the nacelle should be reasonably close. They can however be quite different compared to tower bottom due to tower deformation.

Continuing with OpenFAST (whilst making sure there is a correct match with the lidar simulator), different sets of wind and waves can be tested. It would be valuable to see how the different reconstruction methods behave with for example different Turbulent Intensities, random seeds for turbulence boxes and significant wave heights. Additionally, the dynamics of other platforms can be tested. For instance, it was found in literature that, with the same inflow wind and wave conditions, the WindCrete and Floatgen platforms impact differently the VLOS time series of a nacelle-mounted lidar. Then, since cross-checks are always valuable, the same can be done using dynamics outputs from BHawC-OrcaFlex instead of OpenFAST, using available models.

- **Wind Field Reconstruction**

The wind field reconstruction aspect was not the focus of this project, it only came as a last step. Thus, lots of improvement can still be made in this regard.

To start with, the performance of the reconstruction of the mean wind speed profile using Laguerre basis functions can be assessed when the profile is more skewed than a simple power law. This already consists a challenge in itself.

Then, for turbulence reconstruction, it would be valuable the see the difference in results when accounting for the lidar's velocity interference or not. From the study shown in the report, not correcting for the lidar's velocity should give erroneous results on the reconstruction of turbulence. Another important point is the impact of spatial averaging. In fact, for mean wind speed reconstruction, the fitting is made using the point-wise assumption on the VLOS measurements. In other words, it is assumed that the VLOS is only the projection of the velocity of one particle at the center of the range gate onto the beam direction. This is sufficient because for mean quantities, taking into account the spatial averaging of the measurements has extremely low impact. This is however not the case for turbulence, as it was shown. Thus, for turbulence reconstruction, an algorithm that considers each VLOS as an average of different projections should outperform an algorithm that uses the point-wise approximation like the mean profile reconstruction does. If possible, one should also assess the impact of including radial particles into the turbulence reconstruction, compared to using only points along the beam axis. Note that such a level of precision will only matter if, at this level of the reconstruction, the carried-through and expected uncertainties are low enough.

- **Interpolation**

As of position and translational velocity, their interpolation appear to be robust enough using Quintic Hermite interpolation. The main challenge and interrogation remain on the interpolation of orientation and angular velocity. The first step would be to verify the derivation of the algorithm that leads to the estimation of the first derivative of the quaternion Bézier curve.

Then, the impact of noisy orientation/angular velocity on the interpolation should be further quantified. The main problem is that there exists no known least-squares fitting routine such as in the real number case, which makes the calculation of the control points so efficient for the Quintic Hermite interpolation method. One might have to accept performing the interpolation of Euler angles, since in this practical case quaternion interpolation may remain idealistic. This is only possible with a device that has a high sampling rate, condition without which the interpolated Euler angles will behave unexpectedly.

- **Use of real data**

This whole project only has value if somehow applied to real circumstances. To do so, assuming that all the data becomes available, the wind field should be reconstructed from the VLOS measurements using the same methods but instead of simulated values of position, orientation and velocity, real data coming from the INS.

To do so, there remains to make sure that the lidar is correctly fixed on the nacelle sunroof, and that the lidar's optical head rotates only with the nacelle and not also due to a "bad" fixation. On top of this, thorough uncertainty quantification is immensely valuable, since the uncertainty of the real data together with the uncertainties caused by modelling assumptions will impact the final errors in wind field reconstruction.

Bibliography

[1] G. P. Agrawal and D. N. Pattanayak. “Gaussian beam propagation beyond the paraxial approximation”. In: *Optica* (1979).

[2] V. Banakh and I. Smalikho. *Coherent Doppler Wind Lidars in a Turbulent Atmosphere*. Artech House, 2013.

[3] I. BARSTAD and S. GRØNÅS. “Southwesterly flows over southern Norway—mesoscalesensitivity to large-scale wind direction and speed”. In: *Tellus* (2004).

[4] P. Betsch and R. Siebert. “Rigid body dynamics in terms of quaternions: Hamiltonian formulation and conserving numerical integration”. In: *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING* (2009).

[5] F. Bilendo et al. “Applications and Modeling Techniques of Wind Turbine Power Curve for Wind Farms—A Review”. In: *Energies. MDPI* (2023).

[6] W. Boehm and A Müller. “On de Casteljau’s algorithm”. In: *Computer Aided Geometric Design* (1999).

[7] C.F. Bohren and D. R. Huffman. *Absorption and Scattering of Light by Small Particles*. John Wiley & Sons, 1983.

[8] D. Borvarán, A. Peña, and R. Gandoin. “Characterization of offshore vertical wind shear conditions inSouthern New England”. In: *Wind Energy* (2021).

[9] G. Box. “Signal-to-Noise Ratios, Performance Criteria and Transformations”. In: *Technometrics* (1988).

[10] G. E. P. Box, M. J. Gwilym, and C. R. Gregory. *Time Series Analysis: Forecasting and Control. 3rd ed.* Prentice Hall, 1994.

[11] J. D. Brouk and K. J. DeMars. “Uncertainty Propagation for Inertial Navigation with Coning, Sculling, and Scrolling Corrections”. In: *Sensors. MDPI* (2021).

[12] Z. Chen, X. Wang, and S. Kang. “Effect of the Coupled Pitch–Yaw Motion on the Unsteady Aerodynamic Performance and Structural Response of a Floating Offshore Wind Turbine”. In: *Processes. MDPI* (2021).

[13] *Comparison of Lifecycle Greenhouse Gas Emissions of Various Electricity Generation Sources*. Tech. rep. World Nuclear Association, 2011.

[14] CoreWind. *Public design and FAST models of the two 15MW floater-turbine concepts*. WindCrete model available at: https://zenodo.org/records/4322446. 2020.

[15] M. Courtney, R. Wagner, and P. Lindeöw. “COMMERCIAL LIDAR PROFILERS FOR WIND ENERGY. A COMPARATIVE GUIDE”. In: 2008. URL: `https://api.semanticscholar.org/CorpusID:208653543`.

[16] E. B. Dam, M. Koch, and M. Lillholm. *Quaternions, Interpolation and Animation*. Tech. rep. University of Copenhagen, 1998.

[17] R. Danielian et al. “SurfaceLayer Wind and Turbulence Profiling from LIDAR: Theory and Measurements”. In: *European Wind Energy Association (EWEA)* (2006). URL: `http://www.risoe.dtu.dk/rispubl/art/2007_86_paper.pdf`.

[18] P. A. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. Oxford university Press, 2004.

[19] M. Diafa S. amd Belhamel, M. Haddadic, and A. Louche. “Technical and economic assessment of hybrid photovoltaic/wind system with battery storage in Corsica island”. In: *Energy Policy* (2008).

[20] N. Dimitrov et al. “Wind turbine load validation using lidar-based wind retrievals”. In: *Wind Energy* (2019).

[21] M. E. Dreier. *Introduction to Helicopter and Tiltrotor Simulation*. American Institute of Aeronautics and Astronomics, 2007.

[22] F. Dunn and I. Parberry. *3D Math Primer for Graphics and Game Development. Second Edition*. CRC Press, 2011.

[23] *EKINOX Series. Tactical Grade MEMS Inertial Sensors. Hardware Manual*. SBG Systems. 2022.

[24] R. Ferguson, W. Wilkinson, and R. Hill. “Electricity use and economic development”. In: *Energy Policy* (2000).

[25] *Final Report Cost of Energy (LCOE)*. Tech. rep. Trinomics for DG ENER, 2020.

[26] M. Flügge et al. “Wind Stress in the Coastal Zone: Observations from a Buoy in Southwestern Norway”. In: *Atmosphere. MDPI* (2019).

[27] J. Fraden. *Handbook of Modern Sensors. Physics, Designs, and Applications. Fifth edition*. Springer, 2016.

[28] R. Frehlich. “COHERENT DOPPLER LIDAR MEASUREMENTS OF WIND FIELD STATISTICS”. In: *Boundary-Layer Meteorology* (1997).

[29] R. Frehlich and M. Yadlowsky. “Performance of Mean-Frequency Estimators for Doppler Radar and Lidar”. In: *Journal of Atmospheric and Oceanic Technology* (1994).

[30] R. Frehlich et al. “Measurements of Boundary Layer Profiles in an Urban Environment”. In: *Journal of Applied Meteorology and Climatology* (2006).

[31] T. Fujii and T. Fukuchi. *Laser remote sensing*. CRC Press, 2005.

[32] K. Gade. *Introduction to Inertial Navigation and Kalman Filtering. Tutorial for IAIN World Congress, Stockholm, Sweden.* 2009.

[33] J. Gottschall et al. “Results and Conclusions of a Floating-lidar Offshore Test”. In: *Energy Procedia* (2014).

[34] M. Gräfe et al. “Quantification and correction of motion influence for nacelle-based lidar systems on floating wind turbines”. In: *Wind Energy Science* (2023). DOI: 10.1088/1742-6596/2265/4/042022.

[35] S. Harsdorf and R. Reuter. “STABLE DECONVOLUTION OF NOISY LIDAR SIGNALS”. In: *European Association of Remote Sensing Laboratories (EARSel) Proceedings* (2000).

[36] R. Hersh. *18 Unconventional Essays on the Nature of Mathematics*. Springer, 2006, pp. 250, 288.

[37] C. Hill. “Coherent Focused Lidars for Doppler Sensing of Aeorosols and Wind”. In: *remote sensing* (2018). DOI: doi:10.3390/rs10030466.

[38] *Inertial Sensor Interface. Firmware Reference Manual*. SBG Systems. 2023.

[39] H. Jiang et al. “Accurate evaluation of a polynomial and its derivative in Bernstein form”. In: *Computers and Mathematics with Applications* (2010).

[40] M. C. Kelly. “From standard wind measurements to spectral characterization: turbulence length scale and distribution”. In: *Wind Energy Science* (2018).

[41] I. Khan. *Renewable Energy and Sustainability*. Elsevier, 2022, pp. 135, 178.

[42] M. J. Kim, M. S. Kim, and S. Y. Shin. “A General Construction for Unit Quaternion Curves with Simple High Order Derivatives”. In: *SIGGRAPH* (2014).

[43] M.J. Kim and M.S. Kim. “A Compact Differential Formula for the First Derivative of a Unit Quaternion Curve”. In: *Journal of Visualization and Computer Animation* (1995).

[44] R. Krishnamurthy et al. “Offshore Wind Turbine Wake characteristics using Scanning Doppler Lidar”. In: *Energy Procedia* (2017).

[45] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics. Second Edition.* Pergamon Press, 1987.

[46] M. Leimeister. “Reliability-Based Optimization of Floating Wind Turbine Support Structures”. PhD thesis. University of Strathclyde, 2021.

[47] P. J. P. Lindelöw. “Fiber Based Coherent Lidars for Remote Wind Sensing”. PhD thesis. 2008.

[48] Y. Luo et al. “Rigid-flexible coupling multi-body dynamics modeling of a semi-submersible floating offshore wind turbine”. In: *Ocean Engineering* (2023).

[49] J. Mann. “The spatial structure of neutral atmospheric surface-layer turbulence”. In: *Journal of Fluid Mechanics* (1994).

[50] J. Mann. “Wind field simulation”. In: *Probabilistic Engineering Mechanics* (1998).

[51] J. Mann, M. Kelly, and G. C. Larsen. “Modeling Atmospheric Turbulence via Rapid Distortion Theory: Spectral Tensor of Velocity and Buoyancy”. In: *JOURNAL OF THE ATMOSPHERIC SCIENCES* (2017).

[52] J. F. Manwell, J. G. McGowan, and A. L. Rogers. *Wind Energy Explained. Theory, Design and Application. Second Edition.* John Wiley & Sons, 2009.

[53] M.J. McGill and J. D. Spinhirne. “Comparison of two direct-detection Doppler lidar techniques”. In: *SPIE* (1998).

[54] P. McManamon. *LiDAR Technologies and Systems*. SPIE, 2019.

[55] F. Meyer et al. “Modelling lidar volume-averaging and its significance to wind turbine wake measurements”. In: *Wake Conference* (2017).

[56] *Mitigation of Climate Change*. Tech. rep. IPCC, 2022.

[57] NREL. *FAST User’s Guide*. 2005.

[58] A. V. Oppenheim and G. C. Verghese. “Introduction To Communication, Control, And Signal Processing”. From MIT Course.

[59] U. Özinan et al. “Power curve measurement of a floating offshore wind turbine with a nacelle-based lidar”. In: *Journal of Physics: Conference Series* (2022).

[60] R. Pallàs-Areny and J. G. Webster. *SENSORS AND SIGNAL CONDITIONING - Second Edition*. John Wiley & Sons, 2001.

[61] A. Papoulis. *Probability, Random Variables, and Stochastic Processes. First edition.* McGraw-Hill, Inc., 1965.

[62] R. Paschotta. *Field guide to lasers*. SPIE, 2008.

[63] M. Pellé. “Flow structure detection using a numerical lidar measurement model”. MA thesis. TU Delft, 2022.

[64] A. Peña et al. *Remote sensing for Wind Energy*. Tech. rep. DTU Wind Energy, 2015.

[65] A. et al. Peña. “Impact of floating turbine motion on nacelle lidar turbulence measurements”. In: *Journal of Physics : Conference Series* (2024).

[66] W. H. Press et al. *Numerical Recipes : The Art of Scientific Computing*. Cambridge Press, 2007.

[67] A. K. Sabale and N. K. V. Gopal. “Nonlinear Aeroelastic Analysis of Large Wind Turbines Under Turbulent Wind Conditions”. In: *American Institute of Aeronautics and Astronautics* (2019).

[68] B. E. A. Saleh and M. C. Teich. *Fundamentals of Photonics*. John Wiley & Sons, 1991.

[69] A. R. Sathe. “Influence of wind conditions on wind turbine loads and measurement of turbulence using lidars”. PhD thesis. TU Delft, 2012.

[70] *SBG Knowledge Base*. `https://support.sbg-systems.com/sc/kb/latest/integrated-motion-navigation-sensors/ship-motion-measurements` and `https://support.sbg-systems.com/sc/kb/latest/underlying-maths-conventions/vehicle-body-coordinate-frame`. SBG Systems.

[71] K. Shoemake. “Animating Rotation with Quaternion curves”. In: (1985).

[72] E. Simley et al. “Analysis of Wind Speed Measurements using Continuous-Wave LIDAR for Wind Turbine Control”. In: *AIAA Aerospace Sciences Meeting* (2011).

[73] D. Smith, M. Harris, and A. Coffey. “Wind Lidar Evaluation at the Danish Wind Test Site in Høvsøre”. In: *Wiley Interscience* (2006). DOI: `10.1002/we.193`.

[74] V. Sohoni, S. C. Gupta, and R. K. Nema. “A Critical Review on Wind Turbine Power Curve Modelling Techniques and Their Applications in Wind Based Energy Systems”. In: *Journal of Energy* (2016).

[75] *State of the Energy Union 2023: EU responds effectively to crisis, looks to the future, and accelerates the green transition*. Tech. rep. European Union, 2023.

[76] T. Stehly, P. Beiter, and P. Duffy. *Cost of Wind Energy Review*. Tech. rep. NREL, 2019.

[77] G. Strang. *Introduction to applied Mathematics*. Cambridge Press, 1986.

[78] J. R. Taylor. *Introduction To Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, 1997.

[79] G. Terzakis et al. “On quaternion based parameterization of orientation in computer vision and robotics”. In: *Journal of Engineering Science and Technology Review 7* (2014).

[80] *The Sustainable Development Goals Report*. Tech. rep. United Nations, 2023.

[81] A. Venkatanarayanan and E. Spain. “Review of Recent Developments in Sensing Materials”. In: *Material Science and Materials Engineering* (2014).

[82] N. Warncke. *Ainslie model*. Tech. rep. 2024.

[83] O. L. de Weck. “Fundamentals of Systems Engineering”. From MIT Course.

[84] C. Weitkamp. *Lidar : Range-resolved Optical Remote Sensing of the Atmosphere*. Springer, 2005.

[85] G. Welch and G. Bishop. “An Introduction to Kalman Filter”. In: *International Conference on Computer Graphics and Interactive Techniques*. 1995. URL: https://api.semanticscholar.org/CorpusID:215767582.

[86] *WindCube Nacelle - User Manual*. Leosphere, a Vaisala company. 2020.

[87] G. Wolken-Möhlmann, H. Lilov, and B. Lange. “Simulation of motion induced measurement errors for wind measurements using LIDAR on floating platforms”. In: *Fraunhofer IWES* (2010).

[88] Y. Wu and D. Hu. “Strapdown inertial navigation system algorithms based on dual quaternions”. In: *IEEE Transactions on Aerospace and Electronic Systems* (2005).

[89] *ZX TM Turbine Mounted Wind Lidar*. ZX Lidars. 2022. URL: https://www.zxlidars.com/wind-lidars/zx-tm/.

A

Appendix

A.1. Detection techniques

Once the backscattered signal returns to the lidar, this return light is isolated and transmitted to a detector by optical means. The aim is to extract information about the velocities of the backscattering particles that is carried by the received signal. As depicted in Figure A.1, it is the circulator's function to direct the backscattered light from the telescope to the detector. Most circulators use polarization for efficient transmission and isolation between the laser and the detector ports [64]. From the polarization of the light wave (s for emitted beam and p for received beam), the circulator transmits it only in the desired direction. Then, the detector uses one of two techniques to estimate the frequency shift: *direct* detection or *coherent* detection.

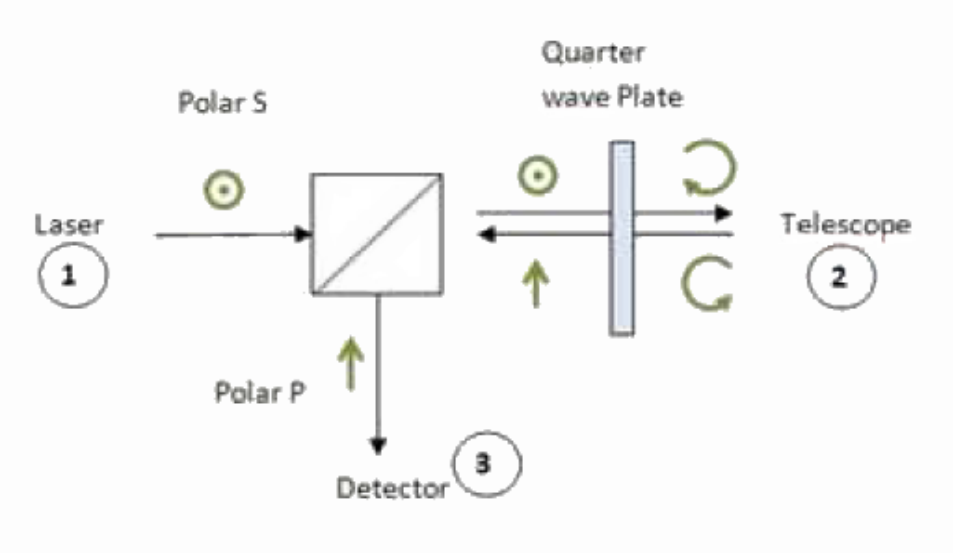


Figure A.1: Circulator [64]

A.1.1. Direct detection

Direct detection uses only the frequency of the return signal to determine the line-of-sight velocity, so it is referred to as *homodyne*. To understand the principle of direct detection, it is necessary to introduce the two different types of elastic scattering contributing to backscattering of the light: *Mie* scattering and *Rayleigh* scattering, the latter being a particular case of the former. Rayleigh scattering is associated with particles small compared with the incident light's wavelength whereas Mie scattering refers to scattering by particles with larger or similar diameter to the wavelength [7]. The differences between the two lie mainly in the light-scattering angular distribution and the scattered irradiance. While

generally, particles scatter light differently at every angle in space (Mie theory), smaller particles show a symmetrical scattering distribution. Additionally, for an incident wavelength λ, Rayleigh scattering yields an irradiance proportional to λ^{-4}, whereas no such simple law exists in the general case [84].

Lidars typically use electromagnetic waves with a wavelength in the order of $1.5\,\mu\mathrm{m}$ [64]. Thus, small molecules in the air (composed of mainly Nitrogen and Oxygen) contribute to Rayleigh scattering while aerosols with a size between 1 and $10\,\mu\mathrm{m}$ such as dust (oxides of iron for example) and sea spray (salt and organic particles) contribute to Mie scattering [7]. Both the signal intensity spectra of molecular scattering and aerosol scattering are centered around the frequency of the backscattered wave, corresponding to the wind speed through Equation 2.1. This can be visualized in Figure A.2. The frequency distribution of the Rayleigh scattering signal intensity is far wider than that of the Mie scattering because of the random thermal motion of the molecules, hence the Gaussian shape [53]. Aerosol particles being heavier, their velocity is much closer to that of the wind so they are more amenable to measurement of the wind speed [84].

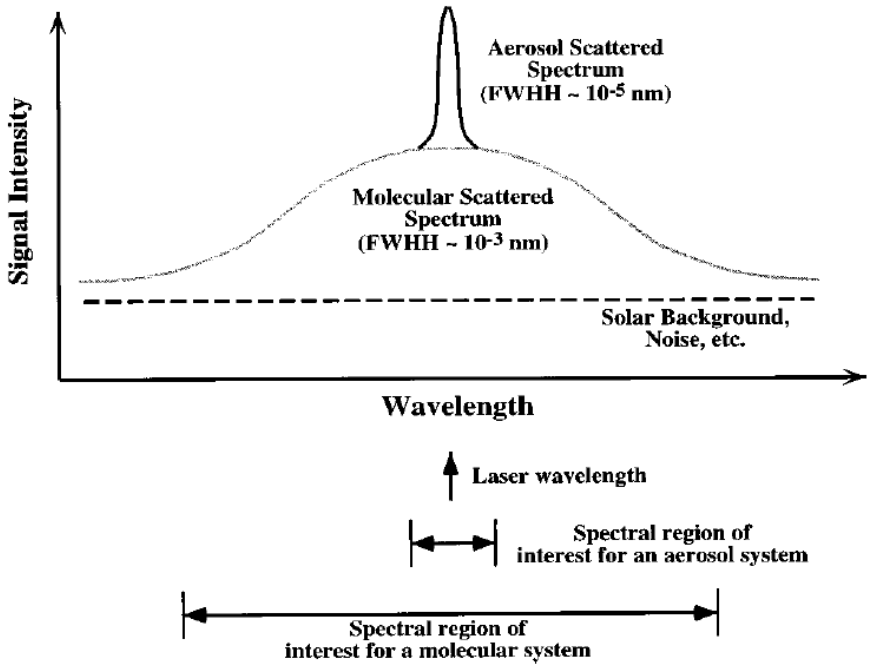


Figure A.2: Typical molecular and aerosol scattered spectra in the atmosphere [53]

Nevertheless, direct detection lidars mainly make use of molecular scattering and calculate the center of the molecular distribution to get the radial velocity of the wind. To do so, the backscattered light is filtered using etalons or Fabry-Perot interferometers [84]. Using two different filters and calculating the light power after transmission through filter 1 and filter 2, the frequency corresponding to the center of the Rayleigh distribution can be retrieved, and thus the line-of-sight velocity. However, direct detection Doppler lidars are not ideal for remote wind sensing in the atmosphere because a high concentration of aerosols, corresponding to a high intensity Mie peak, complicates the measurement of the Rayleigh signal. Also, the frequency range that needs to be measured is much larger than in the case of aerosol scattering. This is why coherent Doppler lidars are more widely used.

A.1.2. Coherent detection

In coherent lidars, the incoming Doppler-shifted radiation is optically mixed with a reference Local Oscillator (LO) beam [64]. Since a coherent lidar mixes the frequency of two waves, the detection is called *heterodyne* detection. With a sensing beam wavelength of $1.5\,\mu\mathrm{m}$, the carrier frequency is typically around $200\,\mathrm{THz}$. The returning beam is also in the same order of frequency, since wind velocities of up to $30\,\mathrm{m/s}$ cause a frequency shift of up to $40\,\mathrm{MHz}$. Mixing the LO wave with the return wave in a square law detector, for example a photodiode, will result in a detectable *beat frequency* in the order of

magnitude of their frequency difference (less than $100\,\mathrm{MHz}$) [47]. This is more manageable to measure than a frequency around $200\,\mathrm{THz}$ [64]. The resulting mixed electric field is converted into a measurable electrical current by the photodetector. This method is illustrated in Figure A.3 .

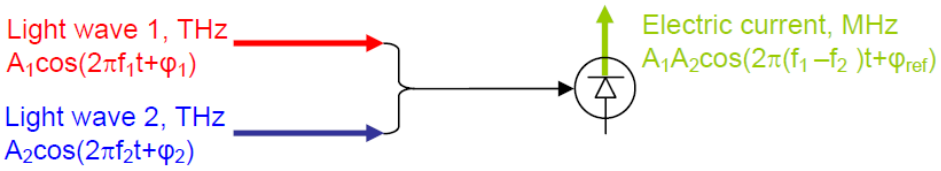


Figure A.3: Heterodyne detection mixing principle [47]

Now, signal processing is necessary to output the Doppler shift from the photocurrent. As discussed in the previous paragraph, the photodiode mixes the LO field with the received backscattered field that can be expressed as follow in Equations A.1 and A.2 [47]:

$$E_{LO}(t) = A_{LO}\ \cos(2\pi f_{LO}\, t + \phi_{LO}) \tag{A.1}$$

$$E_{rec}(t) = \sum_{V} A_i\ \cos(2\pi f_i\, t + \phi_i) \tag{A.2}$$

where A_{LO} is the amplitude, f_{LO} the frequency and ϕ_{LO} the phase of the LO electric field. For the received electric field, the backscattered wave is generated from a sensing volume V of i particles. Consequently, the receiver field is described by a superposition of many particles' contributions, where the field scattered from particle i as it arrives on the photodetector has an amplitude A_i, frequency f_i and phase ϕ_i. Note that $f_i = f_{ref} + f_{Doppler,i}$ where f_{ref} is the frequency of the reference emitted beam and $f_{Doppler,i}$ is the Doppler shift caused by the velocity of the backscattering particle i (close to the wind speed for aerosols). The LO has a frequency offset compared to the reference beam, introduced by an Acousto-Optic Modulator. Thus, $f_{LO} = f_{ref} + f_{offset}$. The signal heterodyne current generated by the photodiode that also acts as a low-pass filter can then be expressed as a function of:

- the diode responsivity R_{diode}
- a receiver loss η
- the collection efficiency of the scatter from particle i F_i
- the backscattered power P_i
- the LO power P_{LO}
- the phase difference between the scattered field and the LO $\phi_{diff,i}$

This gives Equation A.3 [47]. One can see that, by choosing an appropriate f_{offset}, the Doppler frequency shift can be steered to a desirable detectable range. The wind signal can then be deduced since f_{offset} is known. This is done by performing a Discrete Fourier Transform (DFT) of the sampled signal current, to obtain the desired wind velocity distribution in the sensing volume through the power spectrum.

$$i(t) = 2R_{diode} \sum_{V} \sqrt{\eta F_i P_i P_{LO}}\ \cos(2\pi(f_{Doppler,i}\ -\ f_{offset})t + \phi_{diff,i})\ +\ R_{diode}P_{LO} \tag{A.3}$$

A.2. Quaternion properties

Quaternions were invented by Sir William Rowan Hamilton in 1843, when he was trying to generalize complex numbers to three dimensions. Ordinary two-dimensional complex numbers can be used to link algebra and geometry, as for example the addition of two complex numbers can be visualized as the sum of the two corresponding vectors in the complex plane. Since the motion of most objects (and in this project, of a FOWT) is best described by a 3D geometry rather than a 2D geometry, it

seems naturally interesting to develop an algebraic system appropriate in a 3D space, the same as complex numbers are appropriate for 2D spaces. In Hamilton's words, the hope was *'to connect, in some new and useful (or at least interesting way) calculation with geometry, through some extension [of the association achieved in two dimensions], to space of three dimensions'* [36]. However, the problem of three-dimensional algebra is that this set cannot be closed under multiplication [16]. Hamilton then found that it is only possible to define closed multiplication with four-dimensional numbers instead. This gave birth to quaternion mathematics, which includes addition, non-commutative multiplication (akin to matrices), subtraction and multiplication with a scalar. Quaternion mathematics is significantly advantageous in some four-dimensional problems, including rigid-body 3D rotation.

A quaternion is a quadruple that consists of a so-called scalar part, $s \in \mathbb{R}$ and a so-called vector part, $\boldsymbol{v} \in \mathbb{R}^3$ such that the quaternion q can be written $q \equiv [s, \boldsymbol{v}] \in H$, with H the set of quaternions. The parallel with ordinary complex numbers comes from the vector part, which can be compared to the imaginary part of a complex number. In fact, in a similar fashion to complex numbers where $i^2 = -1$, one can define $\boldsymbol{i}, \boldsymbol{j}$ and $\boldsymbol{k}$ where $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$ and write:

$$\begin{aligned} q &\equiv [s, \boldsymbol{v}] && , \ s \in \mathbb{R}, \boldsymbol{v} \in \mathbb{R}^3 \\ &\equiv [s, (x, y, z)] && , \ s, x, y, z \in \mathbb{R} \\ &\equiv s + \boldsymbol{i}x + \boldsymbol{j}y + \boldsymbol{k}z && , \ s, x, y, z \in \mathbb{R} \end{aligned}$$

By convention, the set of quaternions $\{[s, \mathbf{0}] | s \in \mathbb{R}\}$ will be associated with $\mathbb{R}$ and the set $\{[0, \boldsymbol{v}] | \boldsymbol{v} \in \mathbb{R}^3\}$ with $\mathbb{R}^3$. In what follows, useful properties of quaternion mathematics will be presented for they will be used in the report. Let us define the main operations between quaternions/scalar :

- Quaternion addition : Let $q, \ q' \in H$, with $q = [s, \boldsymbol{v}]$ and $q' = [s', \boldsymbol{v'}]$. Then, $q + q' = [s + s', \boldsymbol{v} + \boldsymbol{v'}]$ where inside of the brackets, one can find the standard additions in $\mathbb{R}$ and $\mathbb{R}^3$.
- Quaternion multiplication with a scalar : Let $q = [s, \boldsymbol{v}] \in H$ and $r \in \mathbb{R}$. Then, $rq = qr = [rs, r\boldsymbol{v}]$ where inside of the brackets, one can find standard scalar multiplication and multiplication of a $\mathbb{R}^3$ vector with a scalar. This multiplication is commutative whereas multiplication between two quaternions is generally not. By convention, $\frac{q}{r}$ represents $\frac{1}{r} \, q$
- Quaternion multiplication with another quaternion : Let $q, \ q' \in H$, with $q = [s, \boldsymbol{v}]$ and $q' = [s', \boldsymbol{v'}]$. Then, $q\,q' = [ss' - \boldsymbol{v} \,.\, \boldsymbol{v'}, \boldsymbol{v} \times \boldsymbol{v'} + s\boldsymbol{v'} + s'\boldsymbol{v}]$, where . and $\times$ are the standard dot and cross products in $\mathbb{R}^3$. Since this is generally not commutative, attention is needed when manipulating quaternion multiplication.

Then, as for complex numbers, the conjugate of a quaternion can be defined, but also its norm and its inverse. Let $q = [s, \boldsymbol{v}] \in H$ and $q' = [s', \boldsymbol{v'}] \in H$:

- Quaternion conjugate : The conjugate q^* of q is defined as $q^* = [s, -\boldsymbol{v}]$. Directly, it comes that $(q^*)^* = q$. Also, q commutes with its conjugate for multiplication, which results in a scalar. Then, $(q' \, q)^* = q^* \, q'^*$. And, $(q' + q)^* = q'^* + q^*$
- Quaternion inner product : The inner product $\cdot$ is defined as $q \cdot q' = ss' + \boldsymbol{v} \,.\, \boldsymbol{v'}$
- Quaternion norm : The norm of q is defined as $\|q\| = \sqrt{q \, q^*} = \sqrt{s^2 + \boldsymbol{v} \,.\, \boldsymbol{v}}$. Also, $\|q\| = \sqrt{q \cdot q}$, which makes it a norm in the mathematical sense.
- Quaternion inverse : The inverse of q is defined as $q^{-1} = \frac{q^*}{\|q\|^2}$. And $qq^{-1} = I$ where $I = [1, 0, 0, 0]$ is the identity quaternion (or neutral element of the quaternion group).

To represent rotations, it is extremely useful to define *unit quaternions*, since they form the only subgroup of H that is needed to represent any rotation, the subgroup of unit quaternion will be referred to as H_1. This set is simply defined by all quaternions q for which $\|q\| = 1$. Their properties will now be presented.
Let $q, q' \in H_1$. Then, $\|qq'\| = 1$ and $q^{-1} = q^*$. It can be shown that for any unit quaternion q, there exists $\theta \in]-\pi; \pi]$ and $\boldsymbol{n} \in \mathbb{R}^3$ with $\|\boldsymbol{n}\| = 1$ such that $q = [\cos(\theta), \sin(\theta)\boldsymbol{n}]$. This property will be highly practical.

From this, one can also define the $\exp$ and $\log$ functions of quaternions. Let $q = [\cos(\theta), \sin(\theta)\,\boldsymbol{n}] \in H_1$. Then,

$$\log(q) = [0, \theta\boldsymbol{n}]$$

It can be considered that $\log$ maps from H_1 to $\mathbb{R}^3$. Next, let $q' = [0, \theta'\,\boldsymbol{n'}]$ with $\theta' \in]-\pi;\pi]$ and $\boldsymbol{n} \in \mathbb{R}^3$, $\|\boldsymbol{n}\| = 1$. Then,

$$\exp(q') = [\cos(\theta'), sin(\theta')\boldsymbol{n'}]$$

It can be considered that $\exp$ maps from $\mathbb{R}^3$ to H_1 and one should note that $\log$ and $\exp$ are mutually inverse functions. Another way of defining exp and log function can be seen below, with assimilating quaternions with null scalar part to $\mathbb{R}^3$.

$$\begin{aligned} \exp : \mathbb{R}^3 &\longrightarrow H_1 \\ \boldsymbol{v} &\longmapsto [cos(\|\boldsymbol{v}\|), sin(\|\boldsymbol{v}\|)\ \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}] \end{aligned} \qquad \begin{aligned} \log : H_1 &\longrightarrow \mathbb{R}^3 \\ [s, \boldsymbol{v}] &\longmapsto [\arccos(s)\ \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}] \end{aligned}$$

Finally, exponentiation is defined as Equation A.4.

$$\forall\,(q,t) \in H_1 \times \mathbb{R},\ q^t \equiv \exp(t\log(q)) \tag{A.4}$$

Regarding differential calculus, some general results will also be used, in particular the fact that the general product rule and chain rule also apply to quaternionic functions, where the order of the terms matter because of the non-commutativity of quaternions.

A.3. Quaternion-Euler angles correspondence

A.3.1. From Euler angles to quaternion

Given Euler angles γ, β and α in ZYX convention, the corresponding quaternions q and q^{-1} are given by Equation A.5 (they are unit quaternions) [21]:

$$\begin{aligned} q &= \begin{bmatrix} \cos(\frac{\alpha}{2})\cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}) + \sin(\frac{\alpha}{2})\sin(\frac{\beta}{2})\sin(\frac{\gamma}{2}) \\ \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}) - \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2})\sin(\frac{\gamma}{2}) \\ \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}) + \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}) \\ \cos(\frac{\alpha}{2})\cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}) - \sin(\frac{\alpha}{2})\sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}) \end{bmatrix} \\ q^{-1} &= \begin{bmatrix} \cos(\frac{\alpha}{2})\cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}) + \sin(\frac{\alpha}{2})\sin(\frac{\beta}{2})\sin(\frac{\gamma}{2}) \\ -\sin(\frac{\alpha}{2})\cos(\frac{\beta}{2})\cos(\frac{\gamma}{2}) + \cos(\frac{\alpha}{2})\sin(\frac{\beta}{2})\sin(\frac{\gamma}{2}) \\ -\cos(\frac{\alpha}{2})\sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}) - \sin(\frac{\alpha}{2})\cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}) \\ -\cos(\frac{\alpha}{2})\cos(\frac{\beta}{2})\sin(\frac{\gamma}{2}) + \sin(\frac{\alpha}{2})\sin(\frac{\beta}{2})\cos(\frac{\gamma}{2}) \end{bmatrix} \end{aligned} \tag{A.5}$$

A.3.2. From quaternion to Euler angles

Given the unit quaternion $q = [s, x, y, z]$, the corresponding rotation matrix in ZYX convention is given by Equation A.6 [21]:

$$R = \begin{bmatrix} s^2 + x^2 - y^2 - z^2 & 2(xy + sz) & 2(xz - sy) \\ 2(xy - sz) & s^2 - x^2 + y^2 - z^2 & 2(yz + sx) \\ 2(xz + sy) & 2(yz - sx) & s^2 - x^2 - y^2 + z^2 \end{bmatrix} \tag{A.6}$$

And the set of Euler angles follows (ZYX convention) as Equation A.7. See that there are some forbidden values, showing

$$\begin{aligned} \alpha &= \tan^{-1}\Big(\frac{2(yz + sx)}{s^2 - x^2 - y^2 + z^2}\Big) \\ \beta &= \sin^{-1}\Big(2(sy - xz)\Big) \\ \gamma &= \tan^{-1}\Big(\frac{2(xy + sz)}{s^2 + x^2 - y^2 - z^2}\Big) \end{aligned} \tag{A.7}$$

A.4. Visualizing quaternions

The principle of quaternion visualization relies on *stereographic projection*, that is the projection of an hypersphere in a space of dimension N onto a space of dimension N-1. For quaternions, the objective is to have a way to visualize them since they exist in a 4-dimensional space. In lower dimensions, stereographic projection is for example used to represent the globe on a planisphere. To understand better, consider the case of the unit sphere (so in $\mathbb{R}^3$). The aim is to be able to represent it onto a plane, so from coordinates $[x, y, z]$ to some coordinates $[x', y']$. This can be done because the unit sphere is actually a surface, so a 2-dimensional object. Indeed, one of its coordinate is constrained by the two others since the relation $x^2 + y^2 + z^2 = 1$ holds.

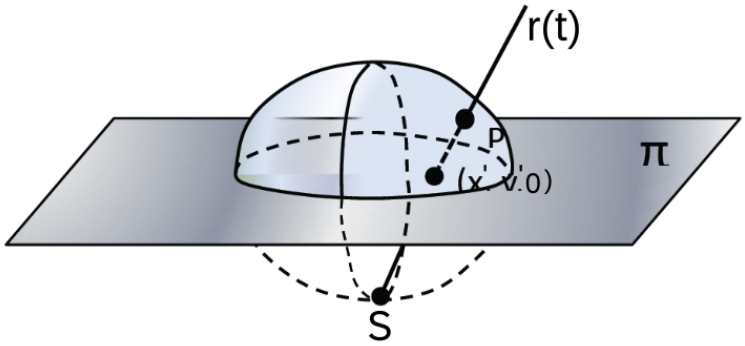


Figure A.4: Ray tracing and stereographic projection of the unit sphere onto the plane. This can be generalized to any hypersphere in a N - dimensional space and a N-1 - dimensional hyperplane. Adapted from [79].

To project the unit sphere onto the plane, stereographic projection uses the position of all the sphere points with respect to a pole, for example the south pole $S = [0, 0, -1]$. The projection of a point $P[x, y, z]$ of the sphere is obtained by the intersection of the "ray" that goes from the south pole S to the P and the 2D plane π, for which $z = 0$, see Figure A.4. The ray $r(t)$ is a straight line between $[x, y, z]$ and $[0, 0, -1]$ so its equation is given by:

$$r(t) = [x, y, z]\, t + (1 - t)\, [0, 0, -1] = [tx, ty, tz - 1 + t] \tag{A.8}$$

For the last coordinate to be null, there needs to be $tz - 1 + t = 0$ so $t = \frac{1}{1+z}$. By replacing t in the expression of the ray, it comes that $[x', y'] = [\frac{x}{1+z}, \frac{y}{1+z}]$.

Now that the principle has been explained, it can be generalized to quaternions, this time using as a south pole $[0, 0, 0, -1]$. The stereographic projection of a unit quaternion $[s, x, y, z]$ in $\mathbb{R}^3$ with coordinates $[x', y', z']$ is given by Equation A.9 [79] :

$$\begin{cases} x' = \frac{s}{1+z} \\ y' = \frac{x}{1+z} \\ z' = \frac{y}{1+z} \end{cases} \tag{A.9}$$

Note that the south pole itself cannot be shown on the new map (division by 0 in Equation A.9). It is mapped to infinity, whereas for all other points, the correspondence is 1-to-1 : to one unit quaternion corresponds only one 3D point, this is because the ray intersects one and only one point of the hypersphere. Now, another transformation will be made so that unit quaternions can be seen on the surface of a 3D sphere. For now, because $[s, x, y, z]$ is an unit quaternion, $\|[x', y', z']\| = \sqrt{\frac{s^2+x^2+y^2}{(1+z)^2}} = \sqrt{\frac{1-z^2}{(1+z)^2}} = \sqrt{\frac{1-z}{1+z}}$. One can see that the more z approaches -1, the further the projected points are from the origin (norm goes to infinity).

Because this distribution of the points is not practical to visualize, it is preferable to do another projection of these points. For example, it was decided to project the first projected 3D points as, once again, the intersection between the ray passing through the south pole $[0, 0, -1]$ and the first projected point

and the unit sphere in $\mathbb{R}^3$, it now comes some second projected points that belong to the unit sphere. Some information is lost because all the points belonging to the same ray will be represented by the same point (loss of 1-to-1 correspondence) on the sphere but for this case, it is sufficient. In Figure A.5, the representation of several quaternions is shown. They have the same angle of rotation $[1, 1, 1]$ (normalized) and angles of rotation range between 0 and 2π. The blue dot is the first stereographic projection and the red dot is the second projection onto the unit sphere of $\mathbb{R}^3$. Notice that all rays, depicted in green, originate from the south pole $[0, 0, -1]$. There is now a way to visualize quaternions, which will be useful to visualize interpolation.

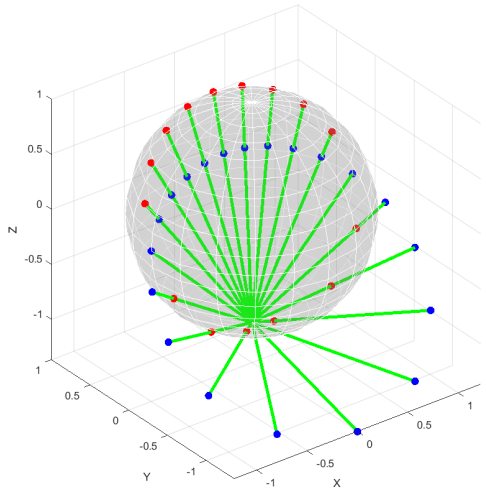


Figure A.5: Non-bijective projection of some quaternions onto the unit sphere of $\mathbb{R}^3$.

Additionally, a zoom and a part of the plane created by two of the rays is shown in Figure A.6. There are 2 unit quaternions involved there. The blue dots are simply the stereographic projection points of the unit quaternions by applying Equation A.9 to find the coordinates in $\mathbb{R}^3$. Notice how the blue dots *do not* belong to the surface of the unit sphere of $\mathbb{R}^3$. This is because the surface of the sphere is in fact a 2D space and because a single stereographic projection only allows to reduce the dimension of the visualizing space by 1 (so here from 4 dimensions to 3). This is why a second projection is made because it is much simpler to have all visualized points on the surface of a sphere. This projection is made by considering the lines (in green) that go from the blue dots to an arbitrary point (here the ”south pole”, black dot). Then, the final projections (red dots) are the intersection points between these lines and the unit sphere (gray sphere). The problem discussed before is that if the first projection of two unit quaternions, represented here by blue dots, happen to be on the same line tracing to the south pole, represented here in green, then they will have the same final projection, here the red dots. This is why this projection is *non-bijective*. In other words, while one blue dot can only correspond to one red dot on the sphere, one red dot could correspond to an infinity of blue dots, along the same green line. Nevertheless, because this case is rare, it is acceptable to lose the bijective property of the projection, to win in simplicity and aesthetic of visualization.

A.5. Results of Quintic Hermite translation interpolation

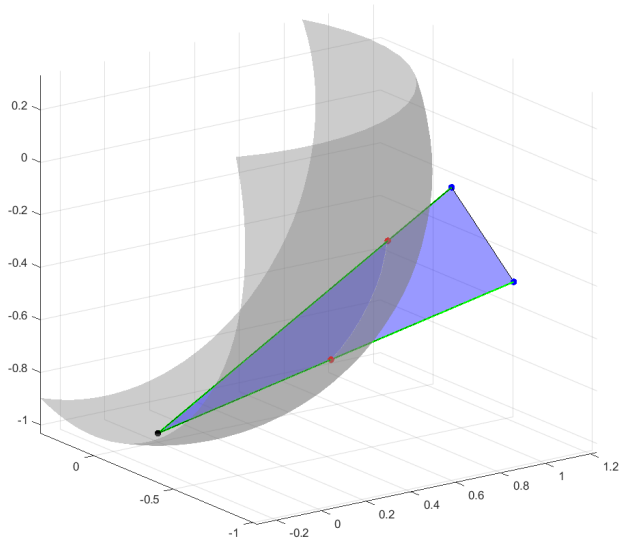


Figure A.6: Zoom-in for two of the rays and corresponding projections.

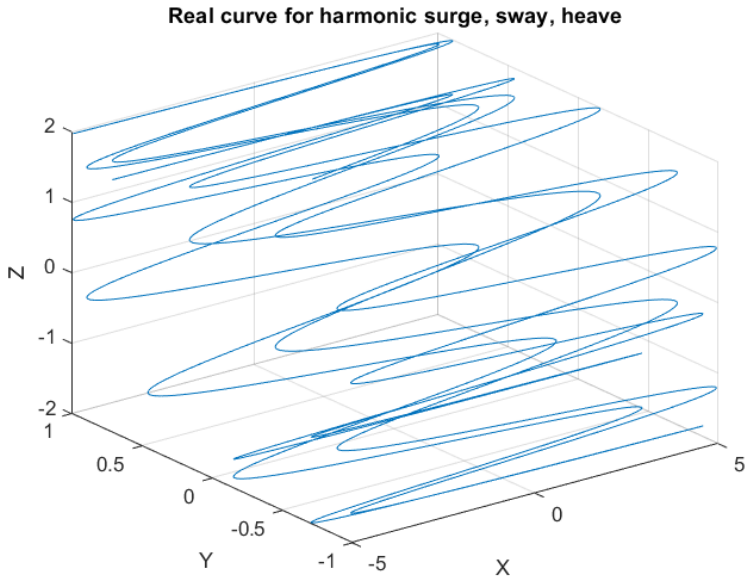


Figure A.7: Position for harmonic test case

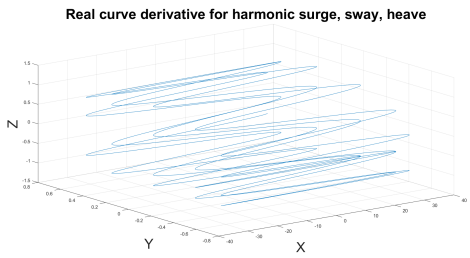


Figure A.8: Linear velocity for harmonic test case

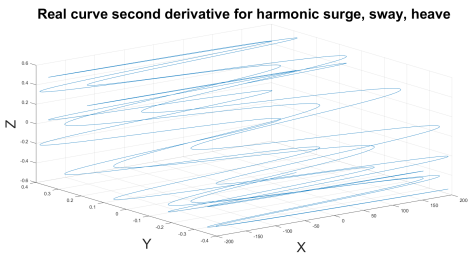


Figure A.9: Linear acceleration for harmonic test case

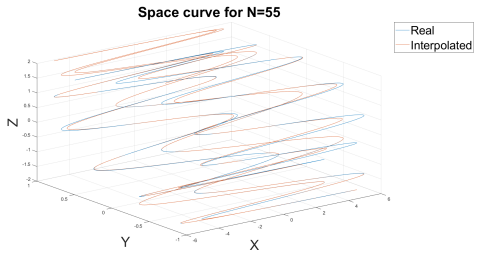


Figure A.10: Display of the interpolated curve, N=55. Slight differences visible

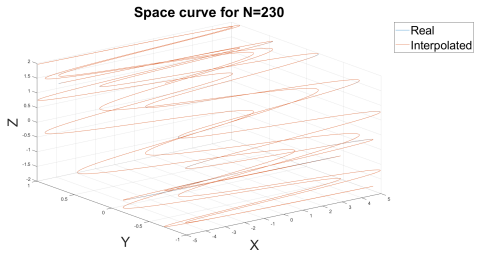


Figure A.11: Display of the interpolated curve, N=230. The difference cannot be seen anymore

A.6. Cumulative basis quaternion curve

A.6.1. Expression of the first derivative of the curve

$$\begin{aligned}\frac{\partial \hat{q}}{\partial t}(t) = q_0\Big(&\exp(\log(q_0^{-1}\, q_1)\, \tilde{\beta}_1(h_i))\ \log(q_0^{-1}\, q_1)\ \frac{\partial \tilde{\beta}_1(h_i)}{\partial t}\ \exp(\log(q_1^{-1}\, q_2)\, \tilde{\beta}_2(h_i))\\ &\exp(\log(q_2^{-1}\, q_3)\, \tilde{\beta}_3(h_i))\ \exp(\log(q_3^{-1}\, q_4)\, \tilde{\beta}_4(h_i))\ \exp(\log(q_4^{-1}\, q_5)\, \tilde{\beta}_5(h_i)) +\\ &\exp(\log(q_0^{-1}\, q_1)\, \tilde{\beta}_1(h_i))\ \exp(\log(q_1^{-1}\, q_2)\, \tilde{\beta}_2(h_i))\ \log(q_1^{-1}\, q_2)\ \frac{\partial \tilde{\beta}_2(h_i)}{\partial t}\\ &\exp(\log(q_2^{-1}\, q_3)\, \tilde{\beta}_3(h_i))\ \exp(\log(q_3^{-1}\, q_4)\, \tilde{\beta}_4(h_i))\ \exp(\log(q_4^{-1}\, q_5)\, \tilde{\beta}_5(h_i)) +\\ &\exp(\log(q_0^{-1}\, q_1)\, \tilde{\beta}_1(h_i))\ \exp(\log(q_1^{-1}\, q_2)\, \tilde{\beta}_2(h_i))\ \exp(\log(q_2^{-1}\, q_3)\, \tilde{\beta}_3(h_i))\\ &\log(q_2^{-1}\, q_3)\ \frac{\partial \tilde{\beta}_3(h_i)}{\partial t}\ \exp(\log(q_3^{-1}\, q_4)\, \tilde{\beta}_4(h_i))\ \exp(\log(q_4^{-1}\, q_5)\, \tilde{\beta}_5(h_i)) +\\ &\exp(\log(q_0^{-1}\, q_1)\, \tilde{\beta}_1(h_i))\ \exp(\log(q_1^{-1}\, q_2)\, \tilde{\beta}_2(h_i))\ \exp(\log(q_2^{-1}\, q_3)\, \tilde{\beta}_3(h_i))\\ &\exp(\log(q_3^{-1}\, q_4)\, \tilde{\beta}_4(h_i))\ \log(q_3^{-1}\, q_4)\ \frac{\partial \tilde{\beta}_4(h_i)}{\partial t}\ \exp(\log(q_4^{-1}\, q_5)\, \tilde{\beta}_5(h_i)) +\\ &\exp(\log(q_0^{-1}\, q_1)\, \tilde{\beta}_1(h_i))\ \exp(\log(q_1^{-1}\, q_2)\, \tilde{\beta}_2(h_i))\ \exp(\log(q_2^{-1}\, q_3)\, \tilde{\beta}_3(h_i))\\ &\exp(\log(q_3^{-1}\, q_4)\, \tilde{\beta}_4(h_i))\ \exp(\log(q_4^{-1}\, q_5)\, \tilde{\beta}_5(h_i))\ \log(q_4^{-1}\, q_5)\ \frac{\partial \tilde{\beta}_5(h_i)}{\partial t}\Big)\end{aligned} \tag{A.10}$$

A.6.2. Derivation of control points

For this task, the values of the cumulative basis function and their first/second derivatives at the nodes are needed. For the function values, it is given in Table A.1. The angular velocities used are in the local frame.

	$\tilde{\beta}_1$	$\tilde{\beta}_2$	$\tilde{\beta}_3$	$\tilde{\beta}_4$	$\tilde{\beta}_5$
t_i	0	0	0	0	0
t_{i+1}	1	1	1	1	1

Table A.1: Values of the cumulative basis functions

For the first derivative of the basis functions, it is given in Table A.2.

	$\dot{\tilde{\beta}}_1$	$\dot{\tilde{\beta}}_2$	$\dot{\tilde{\beta}}_3$	$\dot{\tilde{\beta}}_4$	$\dot{\tilde{\beta}}_5$
t_i	$\frac{5}{\Delta t_i}$	0	0	0	0
t_{i+1}	0	0	0	0	$\frac{5}{\Delta t_i}$

Table A.2: Values of the cumulative basis functions first derivative

Finally, for the second derivative of the basis functions, it is given in Table A.3.

	$\ddot{\tilde{\beta}}_1$	$\ddot{\tilde{\beta}}_2$	$\ddot{\tilde{\beta}}_3$	$\ddot{\tilde{\beta}}_4$	$\ddot{\tilde{\beta}}_5$
t_i	$\frac{-20}{\Delta t_i^2}$	$\frac{20}{\Delta t_i^2}$	0	0	0
t_{i+1}	0	0	0	$\frac{-20}{\Delta t_i^2}$	$\frac{20}{\Delta t_i^2}$

Table A.3: Values of the cumulative basis functions second derivative

Using the expression of Equation A.10 and the values of the first derivative of the cumulative basis functions, it comes that:

$$\dot{\hat{q}}(t_i) = q_0\,\Omega_1\,\frac{5}{\Delta t_i}$$
$$\frac{1}{2}\,q_0\,\boldsymbol{\omega_b}(t_i) = q_0\,\frac{5\,\Omega_1}{\Delta t_i}$$
$$\text{So, } \Omega_1 = \frac{\Delta t_i\,\boldsymbol{\omega_b}(t_i)}{10}$$

On the right node, the same principle applied gives:

$$\dot{\hat{q}}(t_{i+1}) = q_0\,q_0^{-1}\,q_1\,q_1^{-1}\,q_2\,q_2^{-1}\,q_3\,q_3^{-1}\,q_4^{-1}\,q_5\,\Omega_5\,\frac{5}{\Delta t_i}$$
$$\frac{1}{2}\,q_5\,\boldsymbol{\omega_b}(t_{i+1}) = q_5\,\frac{5\,\Omega_5}{\Delta t_i}$$
$$\text{So, } \Omega_5 = \frac{\Delta t_i\,\boldsymbol{\omega_b}(t_{i+1})}{10}$$

Then, for the two last parameters, one needs that

$$\ddot{\hat{q}} = \frac{1}{2}\,(\dot{\hat{q}}\,\boldsymbol{\omega_b} + \hat{q}\,\dot{\boldsymbol{\omega_b}})$$
$$\ddot{\hat{q}} = \frac{1}{2}\,\hat{q}\,(\frac{\boldsymbol{\omega_b}^2}{2} + \dot{\boldsymbol{\omega_b}})$$

By using the second derivative of the basis functions at the left node, it comes that:

$$\frac{q_0}{2}\,(\frac{\boldsymbol{\omega_b}(t_i)^2}{2} + \dot{\boldsymbol{\omega_b}}(t_i)) = \frac{q_0}{\Delta t_i^2}\,\Big(\Omega_1(25\Omega_1 - 20) + 20\Omega_2\Big)$$

Using the results on Ω_1 found previously, further derivation shows that

$$\Omega_2 = \frac{\Delta t_i}{10}\Big(\boldsymbol{\omega_b}(t_i) + \frac{\Delta t_i\,\dot{\boldsymbol{\omega_b}}(t_i)}{4}\Big)$$

And similarly at the right node,

$$\Omega_4 = \frac{\Delta t_i}{10}\Big(\boldsymbol{\omega_b}(t_{i+1}) - \frac{\Delta t_i\,\dot{\boldsymbol{\omega_b}}(t_{i+1})}{4}\Big)$$

Since $\Omega_k = \log(q_{k-1}^{-1}\,q_k)$, the results of the control points are then obtained from these four expressions.

A.7. Results for the derivative of the cubic Bézier curve

The function $d\exp$ is necessary to get the derivative of the cubic Bézier curve. Applying $d\exp$ to a 3D vector gives a 4-by-3 matrix. And when multiplying this matrix with another 3D vector, one gets a 4D vector, that will be assimilated to a quaternion. Find below in Equation A.11 the definition of this matrix, with as argument $\boldsymbol{v} = (x, y, z) \in \mathbb{R}^3$ [43]. For lighter notation, the parameters A and B will be defined such that:

$$A = \frac{\sin(\|\boldsymbol{v}\|)}{\|\boldsymbol{v}\|}$$
$$B = \Big(\frac{\cos(\|\boldsymbol{v}\|)}{\|\boldsymbol{v}\|^2} - \frac{\sin(\|\boldsymbol{v}\|)}{\|\boldsymbol{v}\|^3}\Big)$$

$$d\exp(x,y,z) = \begin{bmatrix} -A\,.\,x & -A\,.\,y & -A\,.\,z \\ B\,.\,x^2 + A & B\,.\,xy & B\,.\,xz \\ B\,.\,xy & B\,.\,y^2 + A & B\,.\,yz \\ B\,.\,xz & B\,.\,yz & B\,.\,z^2 + A \end{bmatrix} \tag{A.11}$$

Some proof is also needed for the formula of the derivative of the logarithm of a quaternion function. Let $g : t \longmapsto \log(q(t))$ where $q(t)$ is a parametrized unit quaternion curve.
First, one has that $\exp(\log(q(t)) = q(t)$. So, $\frac{d}{dt}\,(\exp(\log(q(t))) = q'(t)$, where $q'(t)$ is the derivative of the parametrized quaternion curve.

Also, because of the chain rule and the property of the exp function in this case, one has that $\frac{d}{dt}(\exp(\log(q(t))) = (\exp(\log(q(t)))\,g'(t) = q(t)\,g'(t)$.
Thus finally, $g'(t) = q^{-1}(t)\,q'(t)$

A.8. Examples for quaternion interpolation

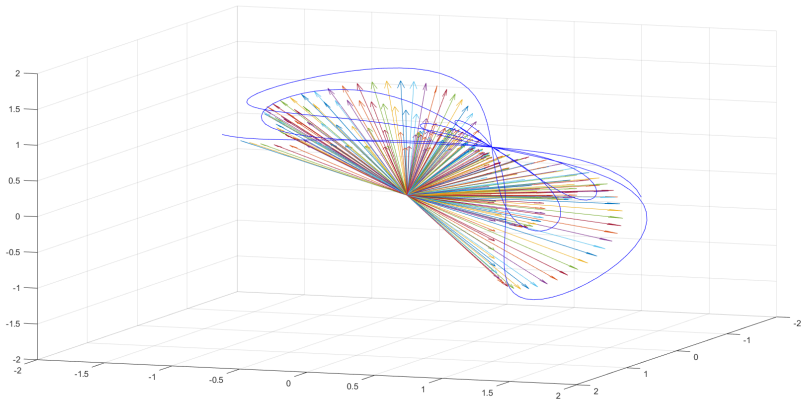


Figure A.12: Path of the vector $[1, 1, 1]$ being rotated by the harmonic quaternion sequence defined

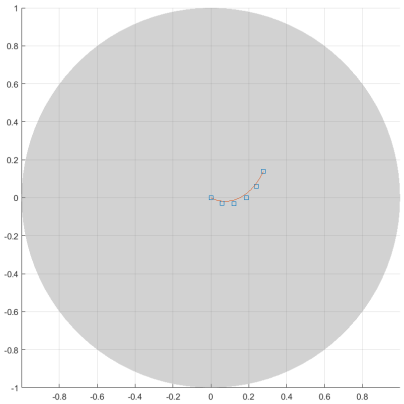
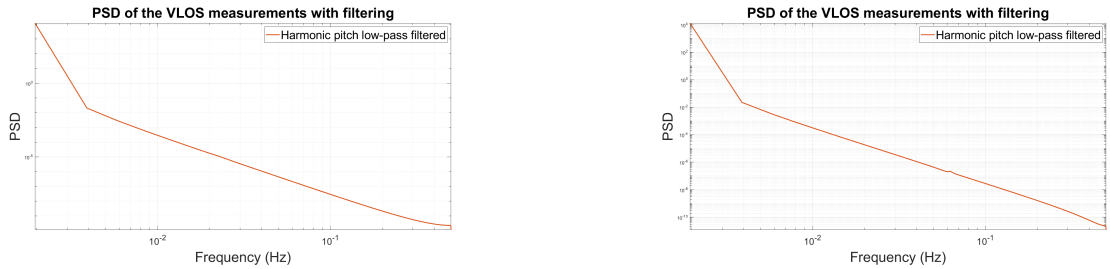


Figure A.13: Example of the 6 control quaternions for the quintic cumulative basis. Notice how they follow the curve

A.9. Simulated rotation of the lidar: pitch motion



(a) Uniform wind.

(b) Power law profile.

Figure A.14: Low-pass filtered PSD spectra of the VLOS measurements with harmonic pitch motion of the lidar on the measurements in two simple cases.

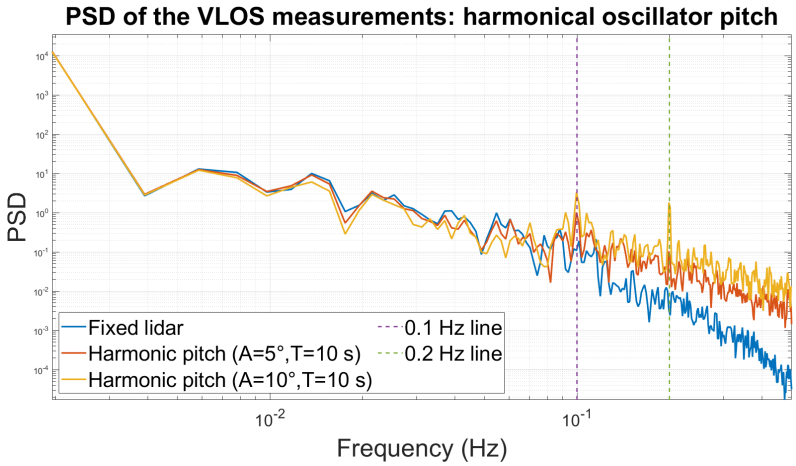
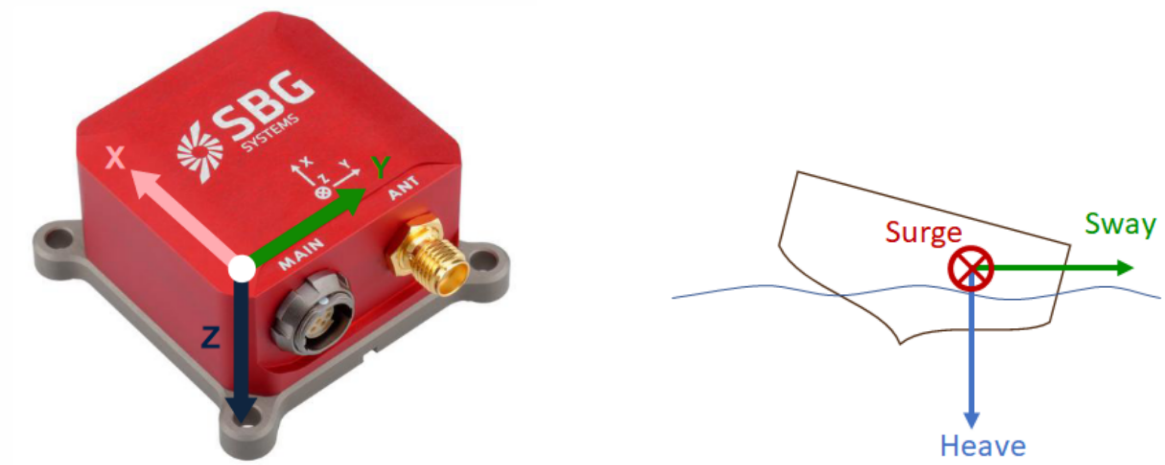


Figure A.15: PSD spectra in case of two different amplitudes of the harmonic pitch motion. Higher amplitude induces higher amplitude of the peaks.

A.10. Reference frames of the Inertial Navigation System

The Inertial Navigation System device uses numbers of different coordinates frames to express the motion data. Here, two different reference frames are introduced, see Figure A.16. Figures come from the INS' manufacturer online knowledge base [70]. The first is the sensor body frame in Figure A.16a. This reference frame is used extensively in this report, it is the local reference frame that is attached to the INS device and that moves exactly the same as the device does.
The second reference frame shown is the Ship Motion reference frame in Figure A.16b. The difference is that the axes are not exactly aligned with the orientation of the device but aligned with the "vessel" the sensor is placed in. This device is mostly used for ships, hence the maritime lexicon. The z-axis (heave) is always pointing downwards (same direction as gravity). The x-axis (surge) is pointing towards what is called the "vessel bow" and the y-axis (sway) is pointing towards what is called the "vessel starboard". As can be seen in the figure, the x and y axis do not follow exactly the orientation of a device that would be placed on the ship. This is why outputs coming from this reference frame are not suited for a mapping to a global reference frame.



(a) Sensor body frame [70].

(b) Ship motion reference frame [70].

Figure A.16: Two different existing coordinate frames for the INS device.

A.11. Power law fitting and uncertainty

This section will address the performance of the power-law fitting method in the case of data coming from a fixed lidar, from a moving lidar without correction of the lidar's velocity, and from a moving lidar with application of the correction algorithm. It serves as a study of what really happens when one reconstructs the mean wind speed profile from data of a floating lidar. It is also useful to assess the impact of the lidar's velocity interference on the uncertainty of the reconstruction.

There are different kinds of interval discussed here that relate to the concept of uncertainty. The first is the *confidence interval* of a model. It is a measure of how confident one can be that an estimate falls in between two values. Here, the confidence interval at 95% of the modelled mean wind speed will be calculated, as well as the confidence intervals at 95% of the parameters of this model (power law). In this case, the horizontal wind speed at a given height is seen as a stochastic process, that has a mean and standard deviation. In fact, turbulence not being normally distributed, these two quantities are not enough to determine entirely how the wind speed is distributed. It is however standard to represent wind speed with its mean and standard deviation (or Turbulence Intensity). This mean is precisely what is being estimated with the power law model, by fitting the measurements to it. Having a 95% confidence interval on the mean wind speed at a given height amounts to having great confidence (up to the 95%-level), that the true mean wind speed at this height lies within the interval. Evidently, the narrower this interval, the more precise (and thus better) the estimated value is. The same reasoning applies for the parameters of the fitted power law.

On the other hand, a prediction interval is an estimate of the interval in which a future observation will fall. In this case, the often used 68% prediction interval will be considered. In this case, for a given height, one can ensure that 68% of the time, the next measured wind speed (not the mean) will remain in this interval. Again, the narrower this interval is, the more precisely one can predict the wind speed. Now, how does one calculate the confidence interval (or the uncertainty) of the fitted power law model?

Consider a non-linear function that depends on parameters and that gives an output based on the value of a given input. For example here, the power law's input is the height, its output is the wind speed and its expression depends on the parameters: reference height, reference wind speed and shear exponent. Then, the standard deviation of this function (seen as the uncertainty) at a given height is a function of the variance-covariance matrix of the parameters (or the uncertainty of the parameters) and the Jacobian matrix of the non-linear function at this height. The Jacobian basically contains the values of the partial derivatives of the function with respect to its parameters. The Jacobian matrix is easily obtained by knowing the expression of the modelling function (here the power law). To relax the condition number of this Jacobian matrix, the reference height is fixed to $150\,\mathrm{m}$, thus reducing the number of the parameters from 3 to 2.

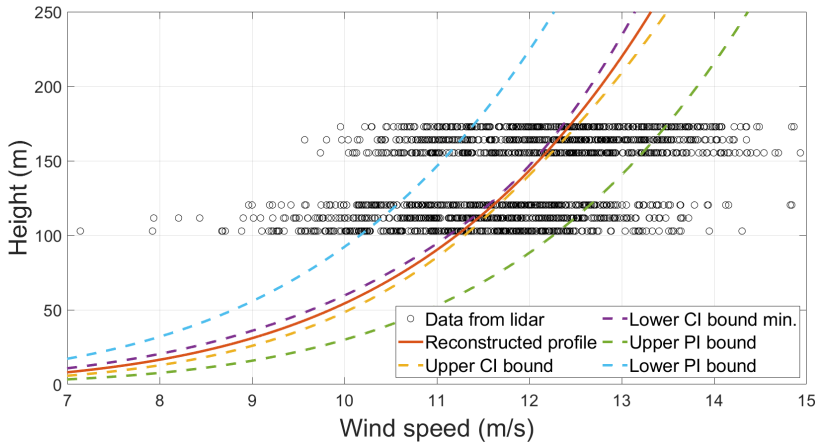


Figure A.17: Uncertainty when reconstructing mean profile with a fixed lidar. The period considered is the first 10 minutes of the simulation.

Starting with the fixed lidar case, the data used for the fitting, the reconstructed power law, the

confidence interval for this mean profile and the prediction interval are shown in Figure A.17. This is for the period between 0 and 10 minutes. The data used for the fit are simply:

$$u_{\text{fit}} = \frac{\text{VLOS}}{\cos(\theta)}$$

This has two main problems: it neglects the spatial averaging effect of the VLOS measurements and it neglects the contribution of the transverse and vertical components of the wind velocity. The fit is then only as good as these assumptions, which are sufficient when reconstructing the mean profile. The bounds of the confidence interval (CI) for the reconstructed profile and that of the prediction interval (PI) are visible in Figure A.17, calculated with Matlab function *nlinfit*. As expected, the bounds for the confidence interval are narrower given that it is easier to predict the true value of the mean wind speed than that of the non-mean (i.e. turbulent) wind speed. One can also see that roughly 2/3 of the data are inside of the prediction interval and 1/3 outside, which comes from the fact that the 68% prediction interval is considered. The fact that this interval seems quite large is not really a defect of the model and of the fitting, but more an intrinsic property of turbulence. It corresponds to what can be expected when fixed measurement devices are used to probe the wind speed. What would be interesting

Regarding the shown confidence interval, the tighter regions correspond to low values of the Jacobian. The mean uncertainty of this reconstructed mean profile (averaged in terms of heights) is of around $0.15\,\mathrm{ms}^{-1}$. As a first estimation, one can then consider that the mean wind profile is reconstructed with an accuracy of $\pm 0.15\,\mathrm{ms}^{-1}$. For the prediction interval, its mean half-width (averaged in terms of heights) is of around $1.1\,\mathrm{ms}^{-1}$. Uncertainties will be **rounded up**. Both these two values are useful for comparison with the floating case. Then, one can also compute the standard errors of the parameters of the model, that is shear and reference wind speed at $150\,\mathrm{m}$ in the power law. These uncertainties are linked to the uncertainty of the mean profile by the Jacobian matrix. In that case, the fitting gives that:

- $U_{\text{ref}} = 12.10 \pm 0.06\,\mathrm{ms}^{-1}$
- $\alpha = 0.19 \pm 0.02$

Now the focus is made on the simulation when considering a moving lidar in the simulations for the same period. The first case will be considering VLOS measurements for which the lidar's velocity interference is corrected. The 6 lines corresponding to all the measurement heights defined by the 3 ranges of the upper beams and the 3 ranges of the lower beams are not visible anymore, due to the motion of the lidar. The result is shown in Figure A.18.

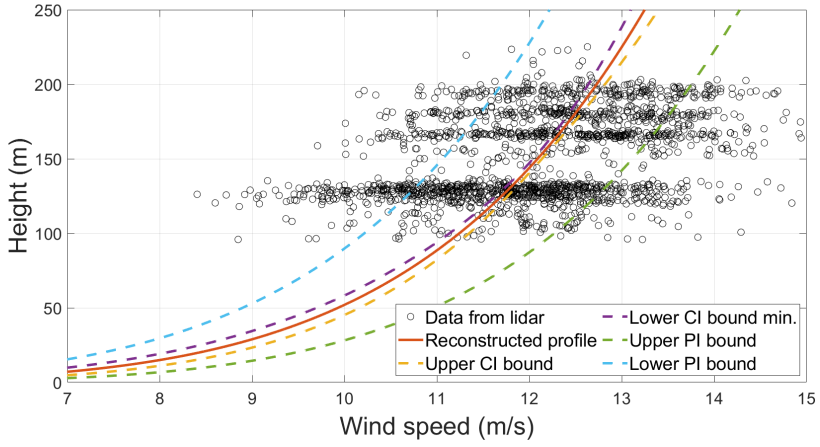


Figure A.18: Uncertainty when reconstructing mean profile with a moving lidar, using corrected VLOS. The period considered is the first 10 minutes of the simulation.

Now, the vertically-averaged uncertainty of the mean profile is $0.16\,\mathrm{ms}^{-1}$, and the mean half-width of the prediction interval is $1.1\,\mathrm{m}$. Regarding the parameters, it is found that:

- $U_{\text{ref}} = 12.09 \pm 0.05\,\text{ms}^{-1}$
- $\alpha = 0.18 \pm 0.02$

Moving onto the case with uncorrected VLOS, the result is found in Figure A.19

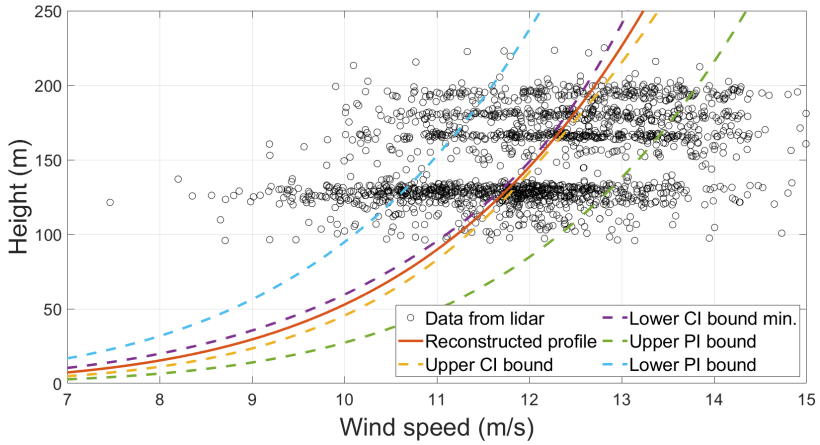


Figure A.19: Uncertainty when reconstructing mean profile with a moving lidar, using uncorrected VLOS. The period considered is the first 10 minutes of the simulation.

The scatter of the point cloud is more expanded than with corrected VLOS, with more outliers. However, the points still remain largely clustered around some mean value. Again, the vertically-averaged uncertainty of the mean profile without velocity correction is $0.17\,\text{ms}^{-1}$, and the mean half-width of the prediction interval is $1.2\,\text{m}$. In terms of parameters, the fit gives:

- $U_{\text{ref}} = 12.07 \pm 0.06\,\text{ms}^{-1}$
- $\alpha = 0.18 \pm 0.03$

In conclusion, the lidar's velocity interference creates outliers in the data used for the fitting of the mean wind profile. Nevertheless, from this study it is not apparent that it causes significantly larger uncertainties on the wind field reconstruction. One should bear in mind that this is only an example for one time period of one simulation (which is still quite extreme in terms of velocity interference).

What remains to be evaluated is the errors that can be carried through neglecting the non-horizontal wind components and neglecting spatial averaging effect of the beam. These uncertainty calculations come from numerical methods that only consider the raw data given, without considering previously propagated uncertainties. Also, results are expected to be different when accounting for the time-averaging of the measurements.

Finally, it is also important to assess how much the uncertainty of the fitting is affected when using noisy data for orientation, position and for the lidar's velocity.

A.12. System Integration

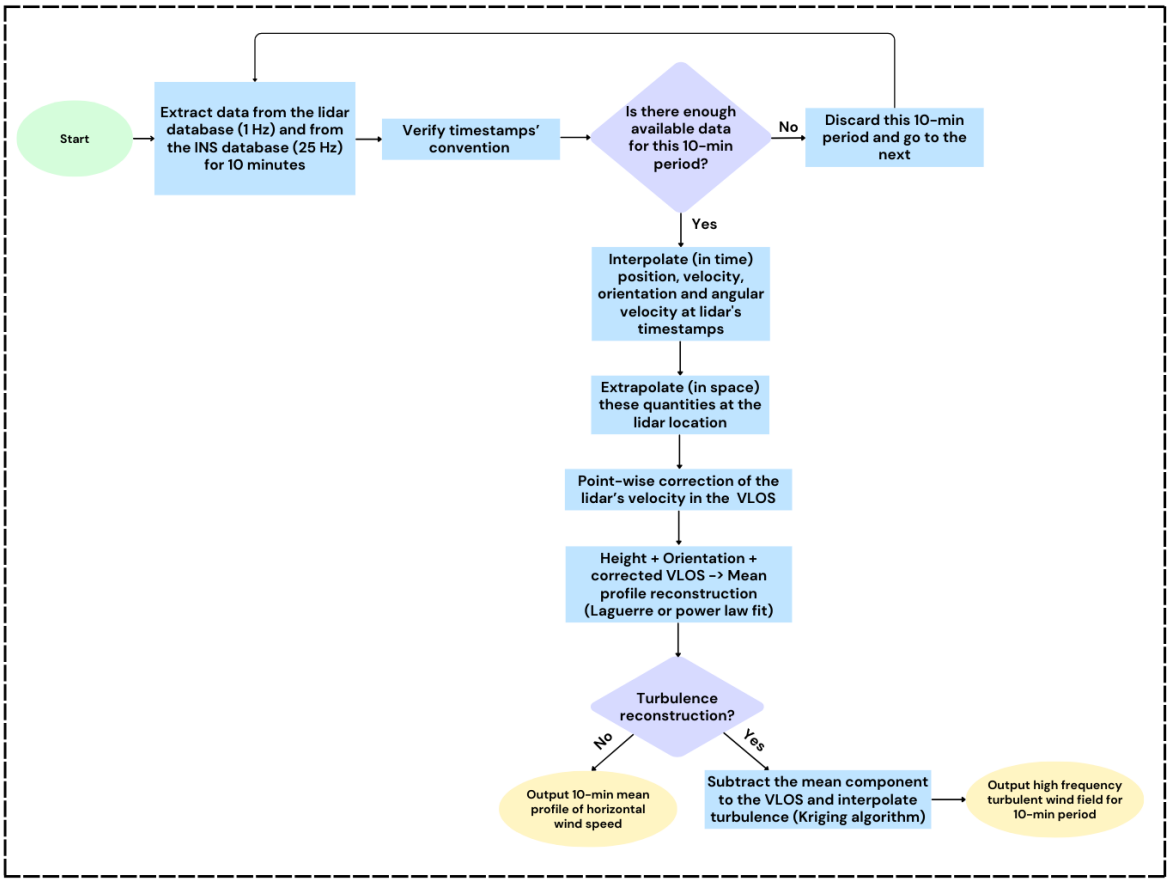


Figure A.20: Process of a more complex system that includes the INS data for the Wind Field Reconstruction.