Low-frequency sea-level variability along the Dutch coast

The relation between non-tidal mechanisms and low-frequency variability in sea-level

Bachelor Thesis

Kareem El Sayed

Delft University of Technology

TU Delft

Low-frequency sea-level variability along the Dutch coast

The relation between non-tidal mechanisms and low-frequency variability in sea-level

by

Kareem El Sayed

Student Name	Student Number
K. El Sayed	5110777

Supervisor:	Prof. Dr. Ir. J.Pietrzak
Supervisor:	Ir. L.Keyzer
Project Duration:	April, 2022 - June, 2022
Faculty:	Faculty of Civil Engineering, Delft

Cover: Photo by Finding Dan | Dan Grinwis on Unsplash

TU Delft

Preface

In front of you lies the bachelor thesis 'Low-frequency sea-level variability along the Dutch coast'. This final study represents my ending phase of my bachelor's degree Civil Engineering at the TU Delft.

During the last weeks of this academic year I have gathered information together with the help of my supervisors to write this thesis. I have gained new knowledge and have improved my research skills.

I would like to thank my supervisors Prof.Dr.Ir. Julie Pietrzak and Ir. Lennart Keyzer for guiding me step by step during this research and providing me the required literature and data. They gave me the trust to carry out this research in an unknown field to me. I hope sincerely that this study will add relevant findings for subsequent studies.

Lastly, I would like to thank my family who have always supported me during my study and my friends with whom I have studied and enjoyed the last three years.

Kareem El Sayed
Delft, June 2022

Summary

In this study the contribution of the low-frequency residuals to the sea-level variability has been examined. This is done using 106-year old sea-level record obtained at Hoek van Holland. Computing the mean sea-level per season each year and the corresponding standard deviation one finds an increase in both these features of the sea-level record. The rise of the mean sea-level implies the effect of climate change. Moreover, it is found that the standard deviation in the sea-level, thus the intensity of variation, is the highest during fall and winter. This implies that the sea-level variability has a seasonal dependency. Furthermore, one finds an increase in the standard deviation on the long term. However, since a big part of the mean sea-level is influenced by tidal events, the increase in standard deviation is possibly linked to climate change in meteorological factors as has been found is in the study carried out by Gerkema and Duran-Matute(2017). With the use of the computational algorithms such as the Fast Fourier Transformation and the Wavelet Transformation one can extract the low-frequency residuals from the sea-level record. Inspired by the study of Gerkema and Duran-Matute(2017) a correlation between the wind speed and the low-frequency residuals have been found. Using the Wavelet Transformation it is found that the low-frequency residuals obtain the most energy during fall and winter, this empowers the finding that these low-frequency residuals are seasonal dependent. Moreover, when studying the low-frequency residuals closely it is found that the frequencies below 0.60 $\frac{1}{day}$ obtain the most energy during fall and winter, especially the frequencies near 0.10 $\frac{1}{day}$. With these findings one can say that the contribution of the low-frequency signals, thus the low-frequency residuals, is correlated to meteorological events, such as the wind. Moreover, it is found that the variation in the low-frequency residuals is much smaller during spring and summer compared to the case during winter and fall. This seems not to be the case for the sea-level variability due to the tidal constituents and the high-frequency waves. Therefore, one may conclude that these low-frequency residuals contribute to a great extent in the standard deviation of the sea-level.

Contents

1

Introduction

The Dutch coast borders to the southern North Sea. This southern North Sea can be seen as a basin. The waves in this basin consist of different type of waves, like seiches and tides. Records of the sea-level along the Dutch coast are available. However, these records does not make a clear separation between the different frequencies occurring in the sea-level. Tidal events are relatively easy to predict, since these events are cyclic events dependent on the gravitational forces of the Sun and the Moon as the rotation of the Earth(NOAA, 2021). However, a part of the sea-level variability is due to low-frequency signals. This low-frequency sea-level variability is harder to predict since it seems to be non-stationary. The main research question that arises discussing this phenomena is in this case:

What is the contribution of the low-frequency signals to the sea-level variability along the Dutch coast?

To be able to answer this question, one should try to analyse data along the Dutch coast and see if there is a correlation between these low-frequency signals and non-tidal mechanisms(e.g. meteorological factors). Making use of mathematical tools to decompose the sea-level record into separate signals with different frequencies makes it possible to analyse this data. The fundamental mathematical tools that will be used are the so-called Fast Fourier Transformation and the Wavelet Transformation.

1.1. Background

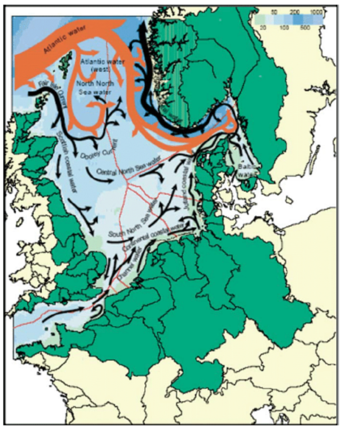

Figure 1.1: Circulation pattern of the water in the North Sea(reproduced from Hofmann et al. (2005))

The variability in the sea-level in the North Sea is due to multiple factors. As one can see in figure 2.2 the North Sea can be seen as a big basin. The water flows from the Atlantic Ocean into this big basin and flows along the coasts. Because of this phenomenon the variability in the sea-level in the North

Sea is partly driven by so-called seiches. Seiches are standing waves that occur in a (semi-)enclosed body of water because of the oscillation between the boundaries of this body(NOAA, 2019). Moreover, tides are important factors in the sea-level variability. Tidal events are due to gravitational forces of the Sun and the Moon as the rotation of the Earth(NOAA, 2021). The variability in the sea-level due to the different driving forces consist of different frequency waves. In this study, the main focus will be laid on the low-frequency waves one can find in the sea-level. It is of big interest to find the contribution of these low-frequency waves and examine the driving force behind it. It is believed that these low-frequency waves are mainly due to meteorological factors(Munk, 1950). Therefore, the relation between the meteorological factors and these low-frequency waves will be closely examined in this study.

A study to the correlation of meteorological factors and the variability in the sea-level has already been carried out in the Netherlands. In the article written by Gerkema and Duran-Matute(2017) the relation between the annual mean sea-level and the annual wind records has been studied. It is of big interest to understand the variation in relative mean sea-level, since the sea-level affects the coastal areas(Gerkema & Duran-Matute, 2017). According to FitzGerald et al.(2008) as cited by Gerkema and Duran-Matute(2017) the risk of flooding and the forming of barrier island systems for instance are all affected due to this variability in sea-level. As mentioned before, this variability in sea-level is due to multiple factors, such as seiches and tides. The frequencies of the tidal constituents and the driving force of these tides are known and therefore relatively easy to predict. Researches about seiches and the correlation between these seiches and non-tidal mechanisms have been carried out all over the world. de Jong et al.(2003) has carried out such a research in the Netherlands to understand the behaviour of these seiches in the Port of Rotterdam. However, because seiches consist of relatively high-frequency waves, the variability in the sea-level due to these seiches is relatively easy averaged out(Gerkema & Duran-Matute, 2017). However, the interannual sea-level variability because of the low-frequency waves is found to be irregular and large(Philip L. Woodworth | National Oceanography Centre et al., 2011 as cited by Gerkema and Duran-Matute, 2017). Therefore, to identify any climatic trend one has to examine a record that is long enough(Gerkema & Duran-Matute, 2017).

From the study carried out by Gerkema and Duran-Matute(2017) a few important things can be understood and taken into account when carrying out this current study. To find a trend in the sea-level variability due to low-frequency waves one has to examine a long record. The record used in the study of Gerkema and Duran-Matute(2017) is a 20-year record. The record used in this study dates from 1900 up to 2018. However, the examination of the low-frequency waves will be done from the 1990 up to 2016, thus a 26-year old record will be examined closely. Moreover, the correlation between the wind and the sea-level has been examined in the study of Gerkema and Duran-Matute(2017). However, Gerkema and Duran-Matute(2017) carried out the correlation study between the wind and the sea-level variability without excluding the tides and the frequency waves, such as seiches, from the record. In this study, the correlation between the meteorological factors and low-frequency waves only will be examined. Thus, the main goal of this study is to understand the contribution of these low-frequency waves on the sea-level variability in its entirety.

From the study carried out by Gerkema and Duran-Matute(2017) important conclusions can be made. There is a correlation found between the wind and the annual mean sea level, dependent on the wind direction(Gerkema & Duran-Matute, 2017). Wind direction plays an important role in the correlation coefficient(Gerkema & Duran-Matute, 2017). Furthermore, one finds a trend in the mean sea-level rise when distinguishing two main periods in a year(Gerkema & Duran-Matute, 2017). The mean sea-level rise during the winter half-year, that is the first and fourth quarter of a year combined, is steeper than the sea-level rise during the summer half-year, that is the second and third quarter of a year combined(Gerkema & Duran-Matute, 2017). This finding implies seasonal dependency in the sea-level variability indicating the importance of this feature in view of coastal protection(Gerkema & Duran-Matute, 2017). Moreover, it is concluded that changes in the wind speed due to climate change have a significant influence on the sea-level(Gerkema & Duran-Matute, 2017).

1.2. Goal and Objectives

The goal of this research is to understand the behaviour of low-frequency waves along the Dutch coast so they are easier to predict. This will be achieved by finding the driving force behind these low-frequency sea-level variability. In this study, mainly local factors will be studied, such as wind records along the Dutch coast. However, this study can be seen as a preliminary study for researches to non-local factors influencing the sea-level variability such as low-frequency standing waves due to the previously discussed phenomenon in a basin. Understanding and predicting the sea-level variability is of big relevance for hydraulic engineering. Using the gained knowledge and insights one can design prediction models of the sea-level variability. These models help improving the coastal protection for instance. Moreover, the usage time of harbours and canals can be extended. Generally, understanding the behaviour of the sea-level variability makes it possible to optimise civil engineering structures along the Dutch coast. Optimising these structures will prevent or limit the chance of failure in the future.

1.3. Method of Approach

First, an analysis will be carried out on the entire data that expresses the sea-level variability. By computing the seasonal sea-level means of each year from 1900 up to 2018 and their corresponding standard deviation deviation, one can find a trend on the long term. Thus, one can understand the variation in the sea-level as a function of the four seasons.

In order to analyse the frequencies in the sea-level variability a couple mathematical tools should be considered. The sea-level record that will be analysed cannot be examined for the described purposes before decomposing it in multiple separate waves with different frequencies. Some of the frequencies are understood, such as the tidal driven waves, or not relevant, such as seiches and high-frequency wind waves.

Using the Fast Fourier Transformation and the filtering criteria, the irrelevant frequencies can be filtered out the data. Fast Fourier Transformation is handy to observe tidal related frequencies since they can be considered as stationary waves. Because low-frequency waves are more difficult to predict and do not have specific predictable patterns, it is not possible to say anything essential about these waves using Fast Fourier Transformation. However, using Inverse Fast Fourier Transformation makes it possible to obtain the data only containing the low-frequency waves which are not related to tides, seiches or wind waves. Using wind records along the Dutch coast one can then search for the correlation between these low-frequency residuals and the wind. To observe the correlation as a function of time between these low-frequency waves and the wind, for instance, one could make use of the so-called Wavelet transformation.

In the final step of the research one should try to link the information between the low-frequency waves, so-called residuals, in the sea water and (extreme) weather events. Using the spectrogram, which is the visualisation of the Wavelet Transformation, it is possible to find specific low frequencies that may be triggered due to meteorological factors or due to seasonal dependency.

A similar method of approach has been applied in the research to seiches in the Port of Ferrol by López et al.(2012). Research in the Port of Ferrol has shown that low wave energy related to seiches are strongly correlated with the off-shore swell energy(López et al., 2012). In this study the focus will be mainly laid on waves that have periods greater than 24 hours.

In chapter 2 the used data and methods will be explained. In chapter 3 the results obtained will be presented with a brief explanation of the observations made. In chapter 4 these results will be discussed and compared to earlier studies. In chapter 5 the conclusion is given and the research question is answered. In the appendices one can find the used source code and further results.

2

Data and Methods

Introduction

In this chapter the used data and the used methods for this study will be discussed. In section 2.1 the type of waves one can differentiate in sea-level will be discussed. In section 2.2 the used datasets and their corresponding locations of the measurement gauges will be presented. Furthermore, the interpolation method will be explained. In section 2.3 the Fast Fourier Transformation and its background will be briefly explained. In section 2.4 the visualisation of the Fast Fourier Transformation using the power spectrum will be explained. In section 2.5 the Wavelet Transformation and a brief explanation about its background will be explained. Finally, the extraction method used to obtain the low-frequency residuals will be explained.

2.1. Type of Waves

First, a clear distinction should be made in the type of waves that occur in the sea-level variability. These frequencies will be considered in this research. The waves that occur at the coastal waters can be seen as the summation of separate waves with different frequencies and different amplitudes. An ocean wave can be broken down into dozens of classes, each class with its own characteristics. For the sake of simplicity, the distinction between three (main) classes will be considered: tides, seiches, and trans-tidal waves. These three classes of waves have their own bandwidth of frequencies and are generated by different forces.

Tides

So-called tides are waves with known frequencies that are due to cyclic events dependent on the gravitational attraction forces of the Sun and the Moon as the rotation of the Earth(NOAA, 2021). These tides are considered to be stationary, which mean they do not change in time.

Seiches

So-called seiches occur when water is considered to be flowing in a basin. In a (semi-)enclosed body of water a wave will oscillate between these boundaries, resulting in standing waves, known as seiches(NOAA, 2019). This phenomenon will be primarily strengthened due to the presence of meteorological conditions and earthquakes(Toffoli & Bitner-Gregersen, 2017).

Trans-tidal Waves

So-called trans-tidal waves are waves which have periods longer than 24 hours. There are tidal waves with similar periods that could be explained due to solar- and lunar-phenomena, but it is believed that these are over-classed by other meteorological factors(Munk, 1950). The definition of meteorological events can be considered to be broad, but for the convenience of this study the definition will be limited to specific influence factors. These trans-tidal waves are the low-frequency waves this study will primarily focus on with the goal to find a clear correlation between them and the meteorological events.

Type of waves	Classification	Period bandwidth	Generating force(s)
Seiche	Long-period waves	5 min to 12 h	Atmospheric pressure gradients and earthquake
Tides	Ordinary tidal waves	12-24 h	Gravitational attraction
Trans-tidal waves	Trans-tidal waves	>24 h	Trans-tidal waves

Table 2.1: Classification of the ocean waves (adapted from Toffoli and Bitner-Gregersen(2017))

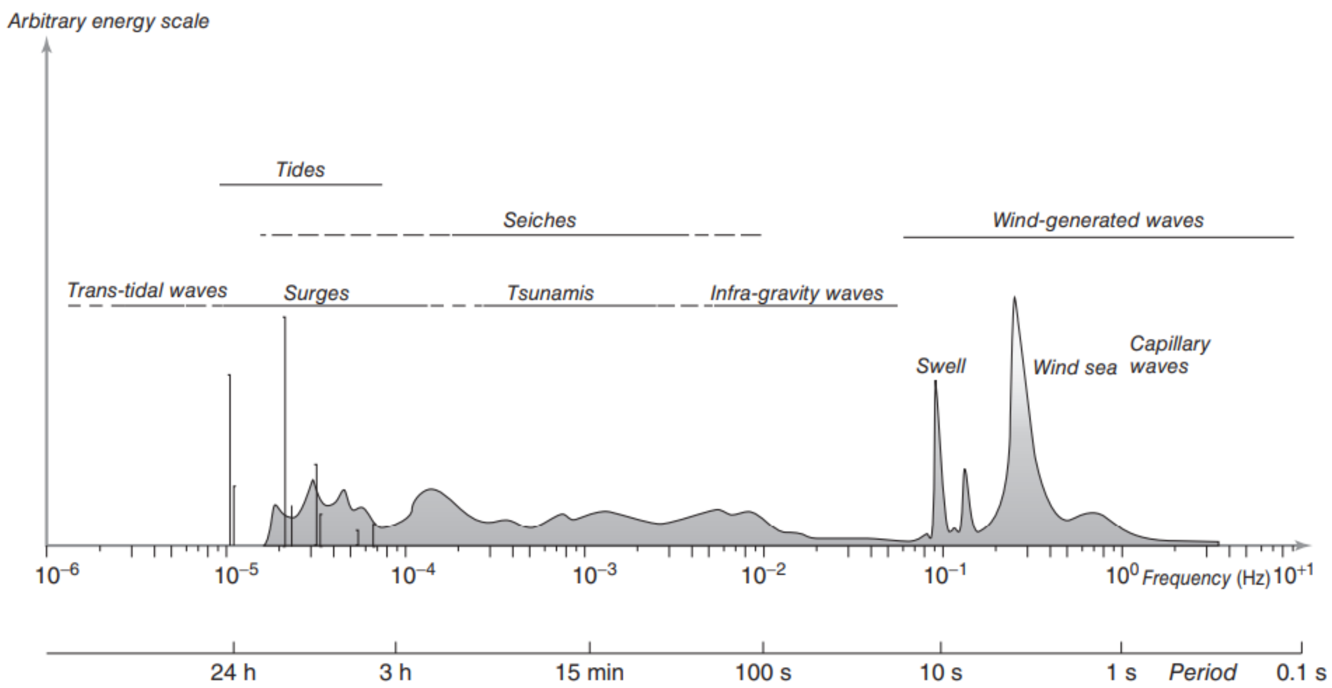

Figure 2.1: Classification of the ocean waves (reproduced from Toffoli and Bitner-Gregersen(2017))

As one can see in table 2.1, the variability in sea-level is due to three main type of waves, which are: long-period seiche waves, tides and so-called trans-tidal waves. Higher frequency waves should not be considered as relevant waves for the sea-level variability, since higher frequency waves mainly affect the variability in the sea surface.

As one can see in figure 2.1, there are higher frequency waves if one considers the entire variation in the sea-level as in the sea surface. However, this study is mainly focused on the analysis of the sea-level variability. Thus, because of the Nyquist theorem (explained later in the report) and the sampling frequency of the data, waves with periods less than 20 minutes should not be considered relevant.

Considering the interest in the correlation between the low-frequency trans-tidal waves and non-tidal mechanisms, one should extract the long-period trans-tidal signal, so-called residuals, from the data. These residuals should then be analysed in combination with the local meteorological events.

2.2. Datasets and Locations

Sea-level Record

The data collected to do this research is from a coastal area in the Netherlands known as Hoek van Holland. The dataset contain information about the sea-level expressed in meters. The data dates from a relatively long time ago. The data obtained in Hoek van Holland dates from 1900 up to 2018. Because the data is relatively long, the problem arises that for some time intervals the data is inconsistent. Some of the datapoints are missing and cannot be used for analysis. Furthermore, the time intervals between the measurements of the sea-level have changed during the course of the years. For the analysis part using the Fast Fourier Transformation and the Wavelet Transformation, it is chosen to consider only the part of the data where this time-step between the measurements is chosen to be 10 minutes.

Figure 2.2: Location corresponding to data measurements

Meteorological Measurements

In order to answer the research question, it is necessary to have data of meteorological events at the same location of the gauge used for the sea-level measurements. The data used is of the Koninklijk Nederlands Meteorologisch Instituut, known as the KNMI. The data can be downloaded from the official website of the KNMI. Multiple meteorological observations are represented as time series. The dataset with the measurements at Hoek van Holland dates from 1971.

Interpolation

A problem which arises when working with these kind of datasets is inconsistency within the data, such as missing datapoints. A consistent dataset is a necessity when performing any kind of Fast Fouier Transformation analysis, this will be clarified later in the report. Using Python, interpolation of the data can be carried out to obtain consistent time-steps between the datapoints. The following steps should be carried out:

1. First, an array should be constructed that begins with a chosen starting date and ends with a chosen end date. The build-up from the starting date up to the end date can be specified by passing the chosen discrete timestep. The discrete timestep used in the initial sea-level record for instance is 10 minutes. Now, an array containing consistent discrete timesteps(10 minutes) has been constructed.
2. Using the NumPy interpolation function in Python, three things need to be passed: the array containing the consistent timesteps, the initial time array containing inconsistency in timesteps and its related datapoints.
3. Based on the initial time array and its related datapoints, an array containing interpolated datapoints will be returned that matches the constructed time array with consistent discrete timesteps.

In order to check for the correctness of interpolation a plot should be carried out before continuing with any analysis using the interpolated data. This to prevent any critical manipulation of the initial data.

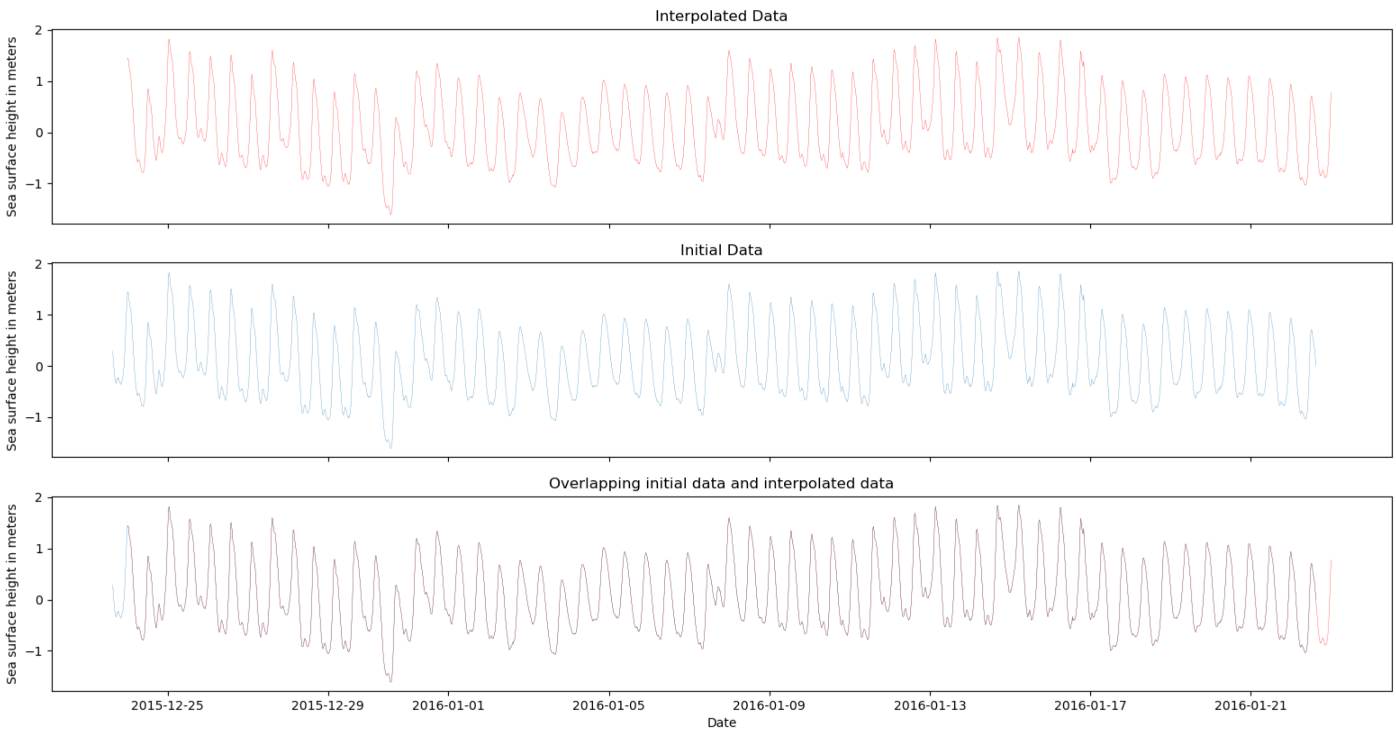

Figure 2.3: The upper graph shows the interpolated data. The middle graph shows the actual data. The lower graph shows the overlapping of the interpolated data and the actual data.

Figure 2.3 shows that the interpolation, over 1 month, worked correctly since the overlapping of the interpolated data on the initial data does not contradict the initial data.

2.3. Fast Fourier Transformation (FFT)

To extract the low-frequency waves from the data, a Fast Fourier Transformation should be carried out on the data. This Fast Fourier Transformation is an optimised computational algorithm based on the Discrete Fourier Transformation. Using the Fast Fourier Transformation it is possible to transform the data from amplitude-time domain into an amplitude-frequency domain. Hence, one can compute the power spectrum of the frequencies, which indicates the dominance of each frequency in the signal. Making use of Inverse Fast Fourier Transformation and filtering criterion one can convert the signal back to amplitude-time domain, but now containing the relevant frequencies of the initial signal. This mathematical tool is used in this study. Therefore, one should have a clear understanding of the mathematics behind this algorithm.

Mathematical Background

The Fast Fourier Transformation is based on the mathematical concept which is known as the Fourier series. Using Fourier series one can decompose a periodical function into a summation of cosine and sine series and a constant, this comes down to the following formula (Xu, 2015):

$$f(t) = \frac{1}{2} \cdot a_0 + \sum_{k=1}^{\infty}(a_k \cdot cos(2\pi kt) + b_k \cdot sin(2\pi kt)) \quad (2.1)$$

As one can see in equation 2.1, the summation consists of multiple cosine and sine functions with different frequencies because of the changing k in the summation. In Fast Fourier Transformation this same concept is applied, by transforming a wave from its frequency-time domain into a frequency-amplitude domain. In case one is dealing with a continuous function, say $x(t)$, this can be expressed in the following mathematical way (Xu, 2015):

$$X(F) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi Ft}dt \quad (2.2)$$

The formula 2.2 describes how the product between the function $x(t)$ and the analysing function $e^{-j2\pi Ft}$ is computed. The analysing function is is the complex notation for sinusoids, where F stands for the frequency. This analysing function can be linked to the cosine and sine functions in equation 2.1 having Euler's identity in mind. After taking the product between the function x(t) and this analysing sinusoidal function, an integration is carried out over a infinite time domain. This operation returns a complex coefficient related per frequency, which contains information about the amplitude and the phase of that related frequency(Xu, 2015).

Since in this study computational power is used to carry out this Fourier transformation, it is not convenient to continue considering this formula which is only valid for continuous functions. In this study, one is dealing with discrete datapoints. Therefore, a discrete notation of formula 2.2 is needed, which comes down to the following (Xu, 2015):

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}} \quad (2.3)$$

Formula 2.3 is directly derived from 2.2, but is valid for discrete points. The following relation can be found:

$$\frac{k}{N} \cong F \quad ; \quad n \cong t \quad (2.4)$$

The summation is evaluated at all N discrete sample points. Therefore, it is necessary to have consistent data without missing discrete sample points. Formula 2.3 returns a frequency coefficient at the k-th frequency bin of the considered frequency bins. The set of frequency bins is dependent on the number of samples and the sample frequency (Xu, 2015).

The formula 2.3 gives a complex coefficient for the related frequency bin, which has the following form (Xu, 2015):

$$X_k = A_k + B_k j \tag{2.5}$$

The information this complex number carries corresponding to its frequency bin is stored in the imaginary and real part of the complex number. Hence, one can know the amplitude of the corresponding frequency and its phase by plotting this complex number in a complex plane, where the following relations follow (Xu, 2015):

$$Amp = \sqrt{A_k^2 + B_k^2} \quad ; \quad \theta = tan^{-1} \frac{B_k}{A_k} \tag{2.6}$$

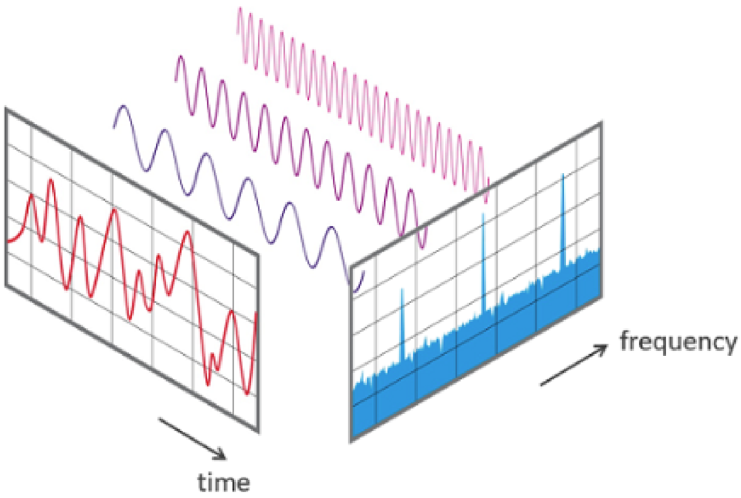

Figure 2.4: Illustration of a signal with amplitude on the y-axis(reproduced from Audio(n.d.))

In figure 2.4 the essence of the Fourier Transformation is illustrated, expressing a signal in the amplitude-frequency domain, instead of amplitude-time domain. Therefore, one can have a clear image of the dominating frequencies in a signal.

Nyquist Frequency

The Nyquist theorem states that the highest frequency that can be found when performing a Fast Fourier Transformation on a discrete signal is at maximum the half of the sampling frequency(Garg & Wang, 2005). This means that if the sampling frequency is f_s Hz, then the highest frequency component that should be included to fully obtain the initial periodic waveform as accurate as possible is $\frac{f_s}{2}$ Hz(Weisstein, 2022). Hence, only waves with periods greater than 20 minutes will be able to get extracted from the data. This theorem should be kept in mind, when considering the Inverse Fourier Transformation when filtering the data.

2.4. Power Spectrum (PS)

After performing the Fast Fourier Transformation on a signal, it is handy to plot the results visually so the filtering step can be carried out. However, the Fast Fourier Transformation will return an array containing the frequency coefficients expressed in complex numbers. The so-called power spectrum is the mathematical concept one could use to obtain real numbers that give information about the energy per frequency bin(DEMPSTER, 2001).

One should realize that the intensity, thus the energy, of a specific frequency in a real signal is directly related to the magnitude of the amplitude squared(Semmlow, 2012). Hence, the definition of the power spectrum for the discrete case is(Semmlow, 2012):

$$PS(k) = X_k \cdot X_k^* \tag{2.7}$$

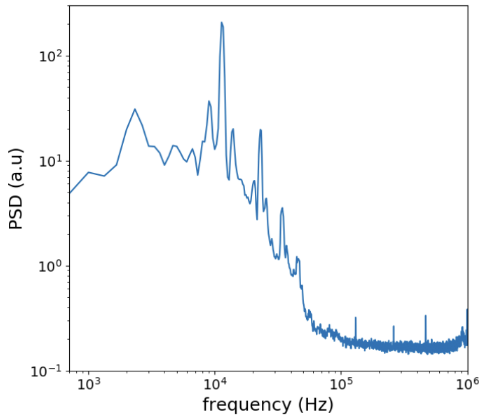

Figure 2.5: An example of a visualised power spectrum of a signal in a so-called periodogram(reproduced from Kawachi et al.(2018))

As can be seen in the example in figure 2.5, the power has been set out along a logarithmic scaled y-axis, whereas the considered frequencies have been set out along a logarithmic scaled x-axis. There is a clear peak at 10^4 Hz, which indicates that the signal analysed is strongly dominated by a wave with 10^4 Hz frequency. Information about the phase per frequency is not given in this periodogram.

Inverse Fast Fourier Transformation (IFFT)

The Inverse Fast Fourier Transformation is the mathematical concept used to transform the Fast Fourier Transformation complex coefficients back to a real signal. Using computational power and software, like Python, this can be achieved. As discussed before, the power spectrum gives the ability to filter certain frequencies out of the Fast Fourier Transformation. That can be simply achieved by only Inverse Fourier transforming the frequencies one is interested in. The frequencies which are not relevant can be filtered out by disabling specific entries in the Fast Fourier Transformation array in accordance with the filtering criterion.

As mentioned before, the Nyquist frequency is the largest frequency which should be included to fully recover the initial signal when doing an Inverse Fast Fourier Transformation. For the discrete case when using computational power, only the upper half of the Fast Fourier Transformation array should be considered. Therefore, one should only pass the upper half of the Fast Fourier Transformation array to the Inverse Fast Fourier Transformation function. Fortunately, the NumPy package in Python takes this into account.

2.5. Wavelet Transformation

The Wavelet Transformation is an alternative for those who are interested in the relation between amplitude-frequency in a signal as function of time. As one can see in formula 2.3, the Fourier Transformation for each frequency bin is carried out on the entire signal, from the first discrete signal point up to the last discrete signal point. Thus, the Fourier Transform does not show when in time the frequencies occur(Nicoll, 2020). Therefore, Fourier Transformation is more suitable for stationary signals(Nicoll, 2020).

This limitation of Fourier Transformation is solved by the invention of the so-called Wavelet Transformation. The Wavelet Transformation is a mathematical concept where the signal is split up in portions, so-called windows. With the assumption that the non-stationary frequencies in the entire signal can be considered stationary in these windows, one can compute the same concept of the Fourier Transformation on these windows(Nicoll, 2020). However, according to the uncertainty principle it is not possible to know the exact frequencies at what time instances, but instead it is possible to know the frequency bins at what time intervals(Nicoll, 2020). This theorem is expressed as the following(Nicoll, 2020):

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{2.8}$$

One should realize that the window width is directly related to Δt in formula 2.8. To correctly find the low-frequency components in a signal, one should have high frequency resolution since these low frequency components can last a long period of time(Nicoll, 2020). According to formula 2.8 this can be achieved by using a big Δt and therefore a bigger window. Vice versa goes for high frequency components, since these last for a short period of time and therefore need a higher time resolution(Nicoll, 2020). This can be achieved by using small Δt and therefore a smaller window according to formula 2.8.

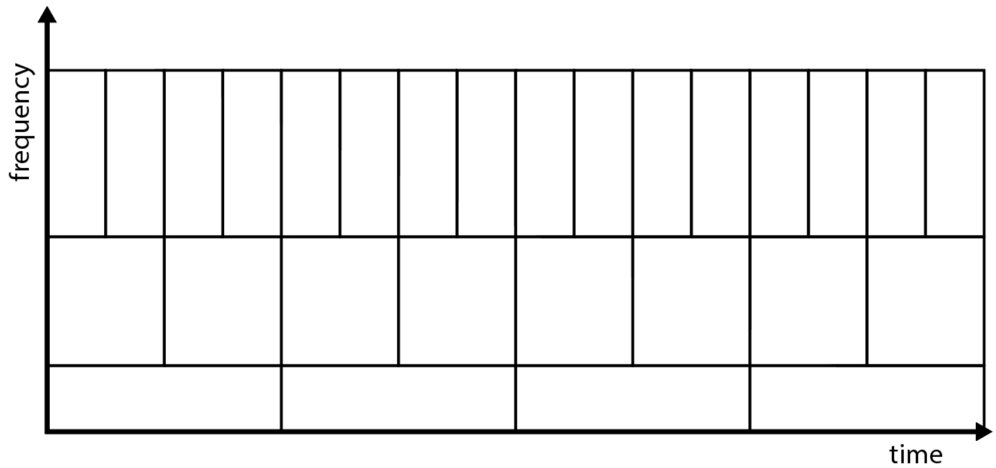

Figure 2.6: The concept of Wavelet Transformation and varying window width(adapted from Nicoll(2020))

As one can see in 2.6, for high frequencies smaller windows are used to obtain high time resolution, whereas for low frequencies larger windows are used to obtain high frequency resolution.

Instead of using sine and cosine as analysing functions, one uses so-called wavelets. These wavelets are used as window functions, meaning the same concept described in formula 2.2 can be applied, such as at the Fourier Transformation, but now at each separate window using this windows function(Nicoll, 2020). This is done so that the amplitude-frequency relation of that specific window can be computed(Nicoll, 2020). An example of such a wavelet is the so-called Morlet wavelet, which will be used in this study inspired by the study carried out by de Jong et al.(2003).

2.6. Extracting the Low-frequency Waves

Combining these previously discussed concepts one can start to extract the low-frequency waves from the initial sea-level data. To give a clear illustration of this process, it is convenient to look at the data over a course of 1 year at Hoek van Holland. The same process goes for other time-frames.

As discussed before, interpolating the data should be done for further analysis. Using a plot of the interpolated data is used to prevent manipulation of the initial data. This can be seen in figure 2.7.

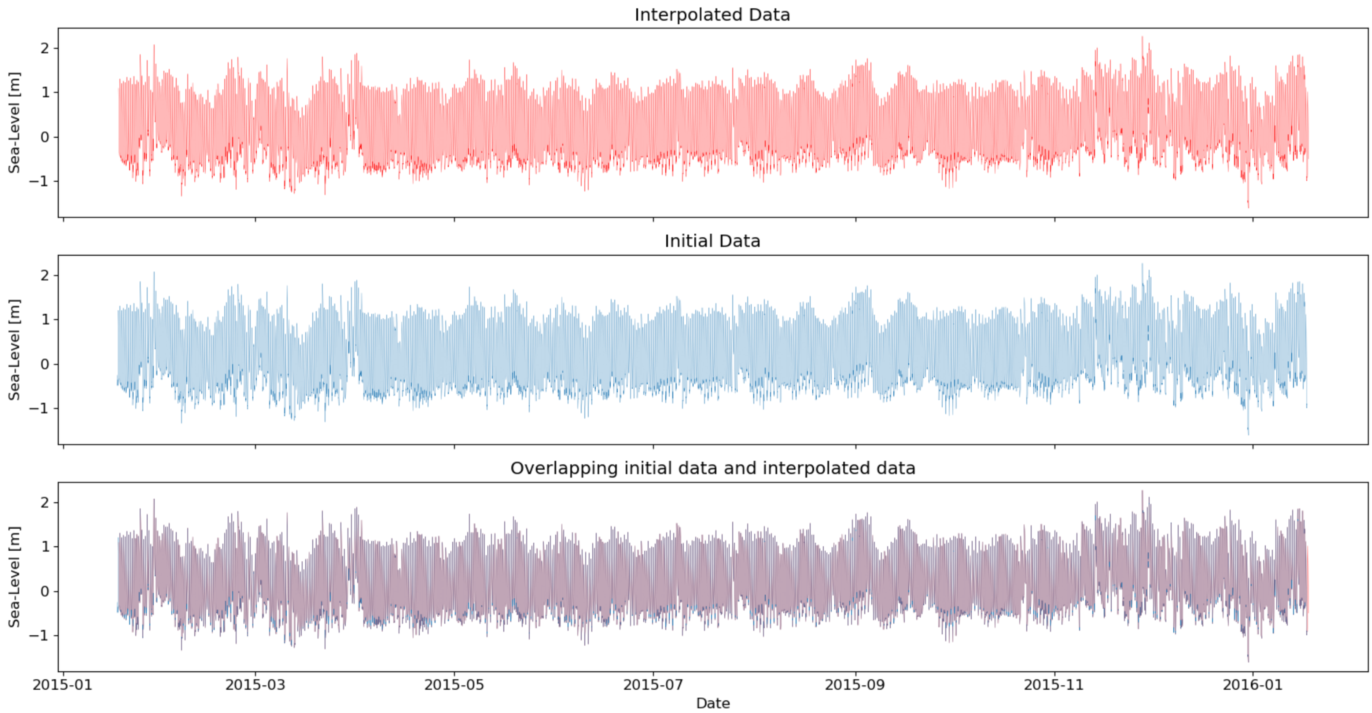

Figure 2.7: The initial sea surface height data and the interpolated sea surface height data

The interpolated data can then be passed to the Fast Fourier Transformation function of the NumPy package in Python. This returns an array with the complex coefficients corresponding to the considered frequencies. The power spectrum can then be computed according to the definition 2.7. Plotting the power spectrum in a so-called periodogram shows the dominant frequencies. An example can be seen in figure 2.8. The frequencies that are clearly visible in the periodogram are the frequencies due to tides. Some of the tides one can see clearly in figure 2.8 are the $M2$, $K1$ and M_f tides, which can be found in table 2.2 with their corresponding frequencies.

Tidal Constituents	Symbol	Frequency per day
Principal lunar	$M2$	1.93
Luni-solar diurnal	$K1$	1.00
Lunar fortnightly	M_f	0.07

Table 2.2: Tidal constituents(adapted from Chelton and Enfield(1986))

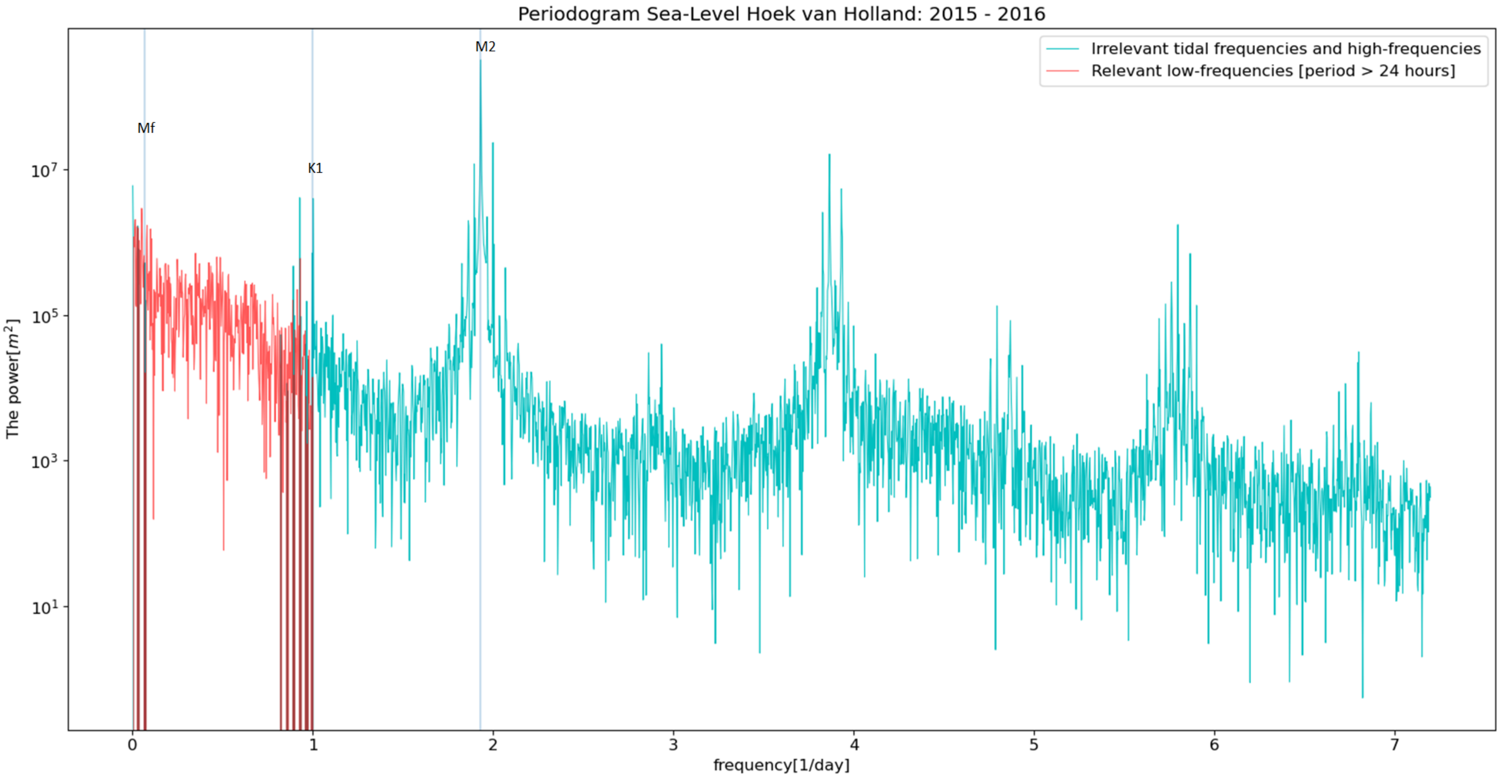

Figure 2.8: The red part in the periodogram shows the frequencies of the residuals, whereas the blue part shows the frequencies of the filtered tides and irrelevant high-frequency signals.

As mentioned before, the goal of this study is to extract the low-frequency waves found in the sea-level variability. Obviously, all signals with periods less than 24 hours should get filtered out. However, some of the tidal constituents have periods larger than 24 hours. Fortunately, by knowing the exact frequencies of these tidal constituents it is possible to filter those out of the signal. Thus, one should remain with the low-frequency signals not due to the tides. The low-frequency tidal-constituents that are filtered out can be found in table B.1.

From figure 2.8 the periodogram of one year can be seen. The red part in the periodogram shows the extracted frequencies to obtain the residuals. The long red stripes that dip to zero power are the low-frequency tidal constituents that had to be filtered out. That is the reason these frequencies dip to a power of zero, since they are silenced out in the low-frequency residuals.

Using the previously described filtering criterion and the Inverse Fast Fourier Transformation function in Python, the low-frequency signals can be extracted from the data. In figure 2.9, these so-called residuals can be seen.

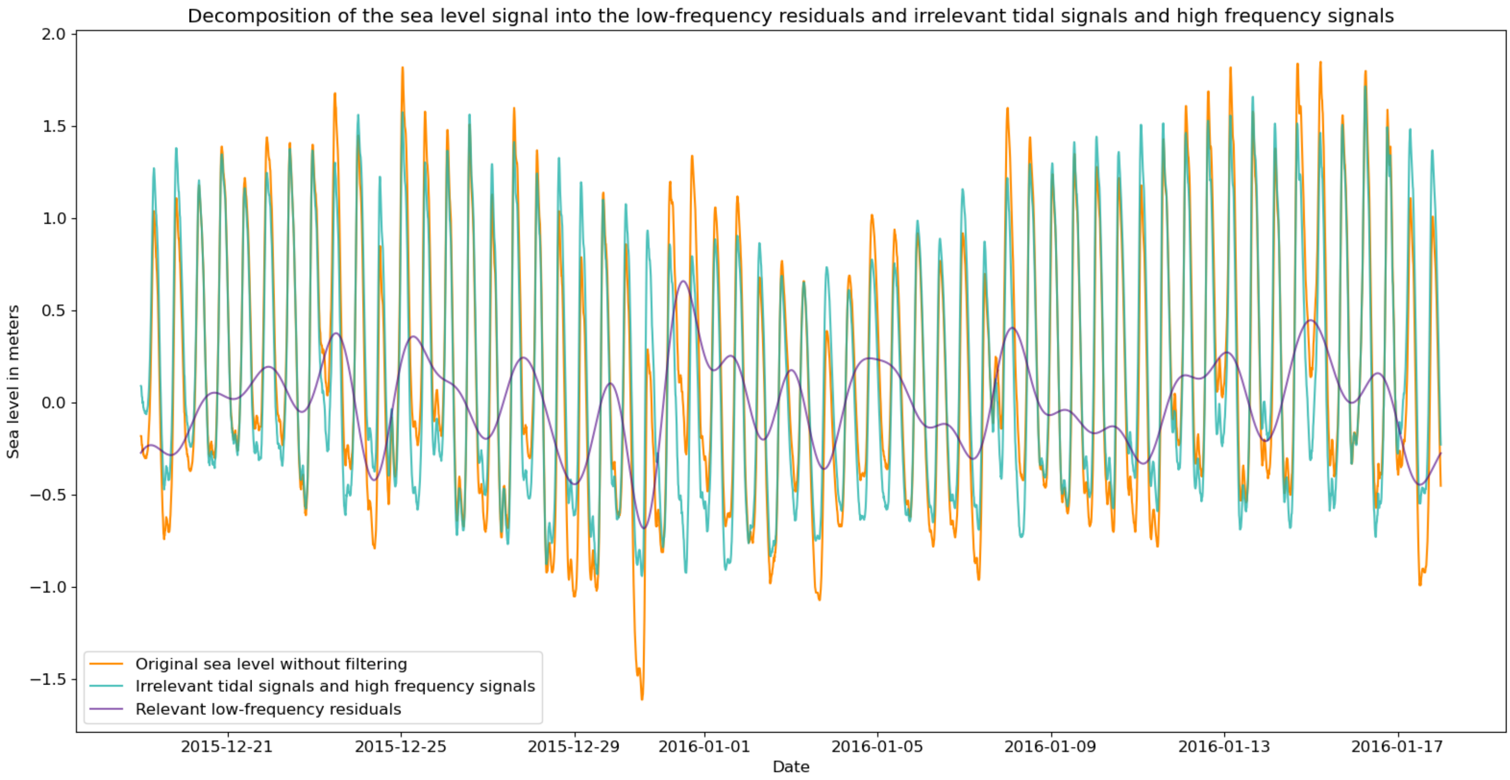

Figure 2.9: The yellow signal is the initial signal without filtering. The green signal shows the signal consisting of the tides and irrelevant high-frequency signals. The purple signal is the extracted low-frequency residuals one is interested in.

Now having the residuals, one can be interested to do the Wavelet Transformation. Since these residuals contain non-stationary frequencies, the Wavelet Transformation comes in handy.

As one can see in appendix B, the Wavelet Transformation gives a visualisation of the power of the frequencies as a function of time, the brighter the color, the bigger the power of the frequency bin. Furthermore, a white dashed line is plotted in the figure, which is the boundary of the so-called cone of influence. The cone of influence is the area within the two dashed lines that should be considered accurate and relevant for study. As discussed before and as can be seen from the figures in appendix B, the higher the frequency, the bigger the frequency bandwidth and thus the higher the time resolution. Vice versa goes for the low frequencies, the lower the frequency, the smaller the frequency bandwidth and thus the lower the time resolution. This goes in accordance with the uncertainty principle expressed in 2.8.

3
Results

In this section the obtained results will be presented. The results will reflect the long-term trend found in the sea-level and the correlation found between the low-frequency residuals and the wind records. Moreover, in order to satisfy the main goal of this study, that is understanding the contribution of the low-frequency residuals to the sea-level variability along the Dutch coast, the low-frequency residuals as function of time will be examined using the Wavelet Transformation. This in order to distinguish particular low-frequencies and their energies as a function of time.

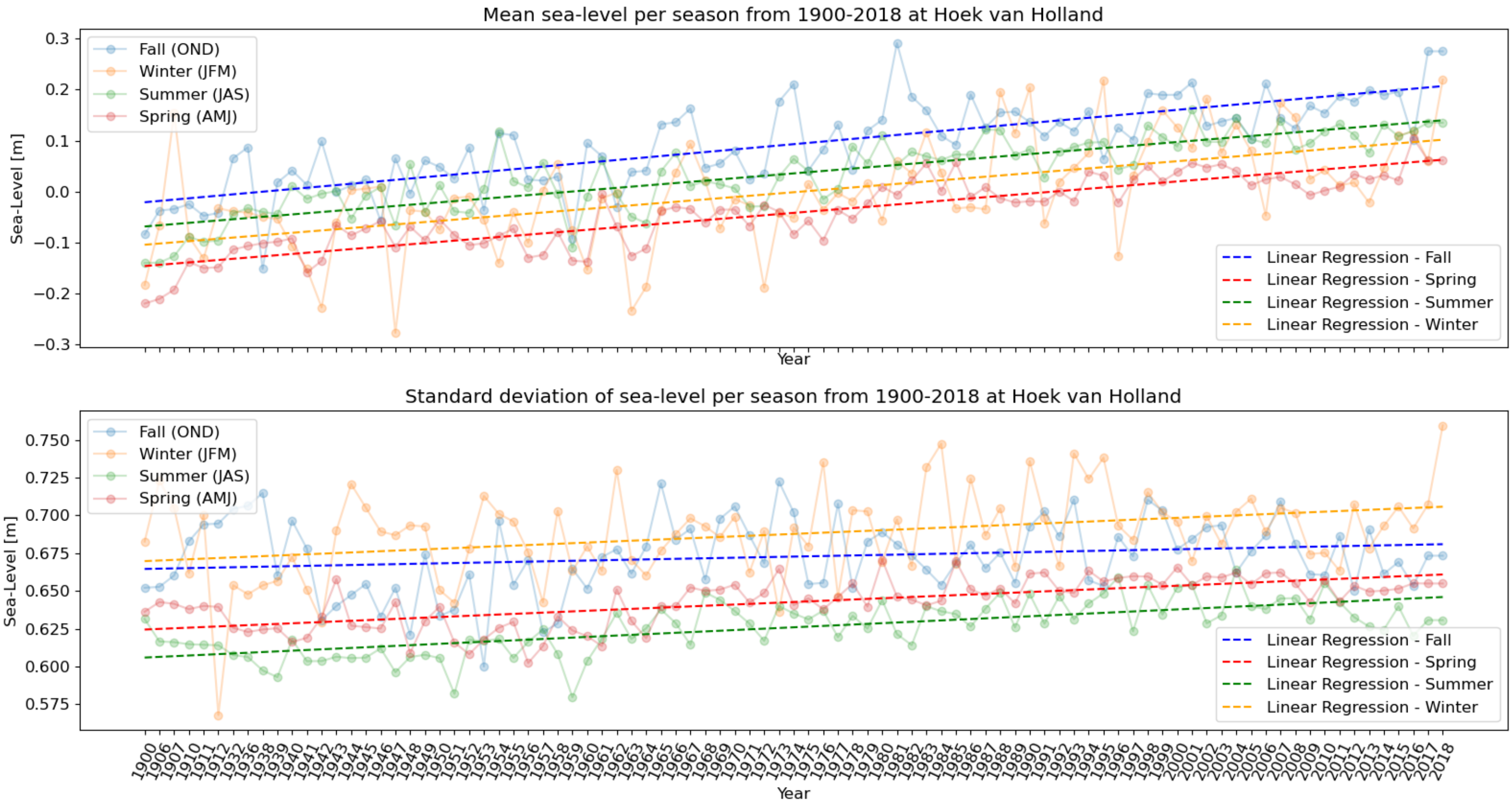

Figure 3.1: The upper graph shows the mean sea-level per season per year computed over the entire length of the sea-level data. The lower graph shows the corresponding standard deviation per season per year computed over the entire length of the sea-level data.

From figure 3.1 the seasonal sea-level means per year can be seen. The upper graph in figure 3.1 shows that the mean sea-level each year during fall is the highest among the other seasons. Moreover, the rise in seasonal sea-level mean for all seasons goes in approximately same speed. However, when observing the standard deviations corresponding to the seasonal means per year, one can find something interesting. The standard deviation, thus the variation in sea-level, is the highest during the winter. Moreover, there is an clear increase in the standard deviation during spring, summer and winter. The standard deviation corresponding the fall season does not increase as hard as the other seasons.

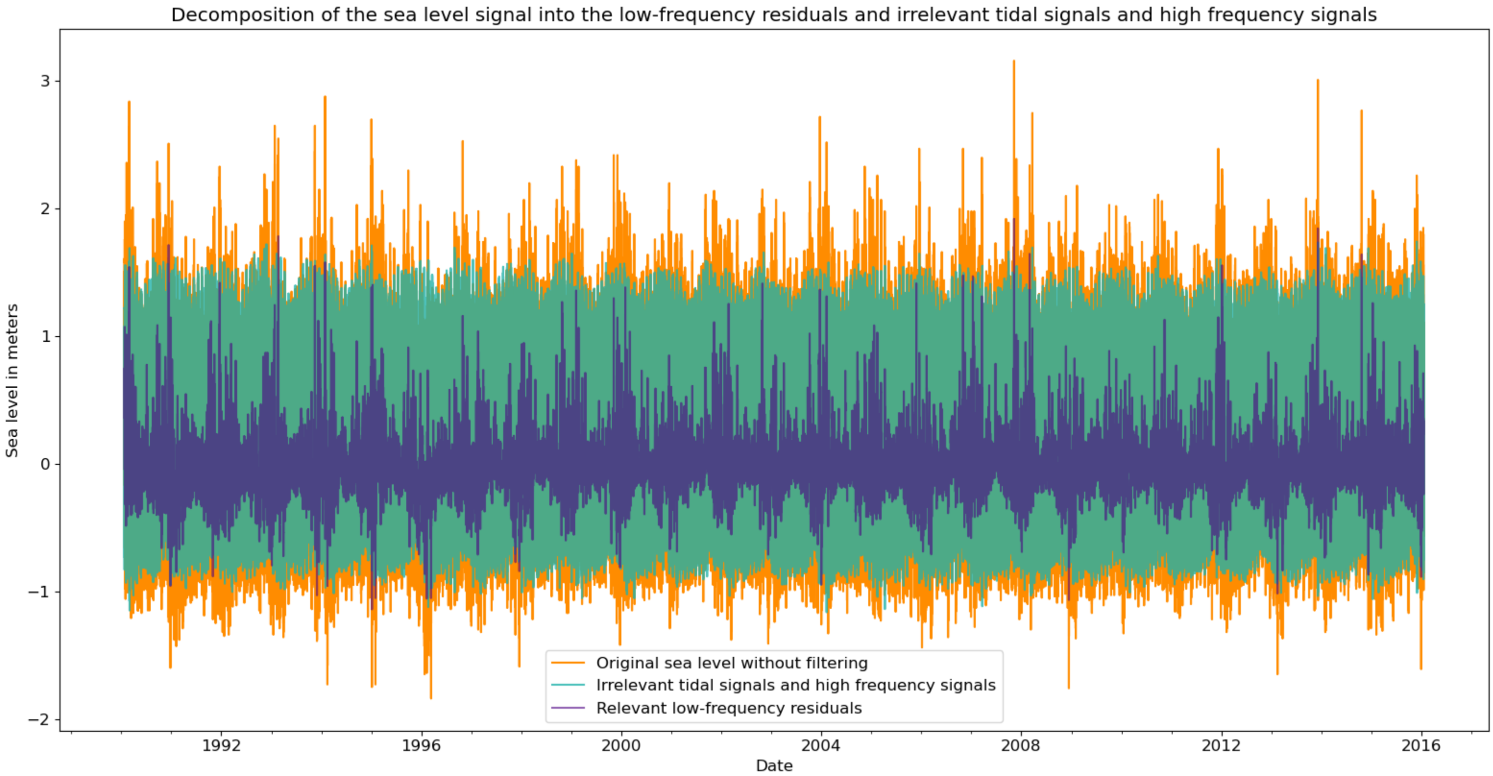

Figure 3.2: The yellow signal is the initial signal without filtering. The green signal shows the signal consisting of the tides and irrelevant high-frequency signals. The purple signal is the extracted low-frequency residuals one is interested in.

From figure 3.2(see B.1 for the bigger figure) one can see the extracted low-frequency residuals as the purple signal, the filtered out tides and high-frequency signal as the green signal and the original sea-level as the orange signal. The figure is obtained from the sea-level record from the year 1990 up to 2016. This figure is obtained by performing the Fast Fourier Transformation and using the previously described filtering criterion to decompose the sea-level record using the Inverse Fast Fourier Transformation.

From figure 3.3 one can see the computed correlation between the wind and the low-frequency residuals dependent on the wind direction. The correlation coefficient used is the Pearson correlation coefficient defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{3.1}$$

Using the Wavelet Transformation it is possible to obtain the spectrogram visualized in figure 3.4. From figure 3.4 the energy of the low-frequencies can be examined as a function of time. Hence, using this spectrogram it is possible to not only discuss the correlation between the low-frequency residuals and the wind speed, but to find specific frequencies triggered by the meteorological factors. In appendix B the spectrograms from year 1990 up to 2016 can be found.

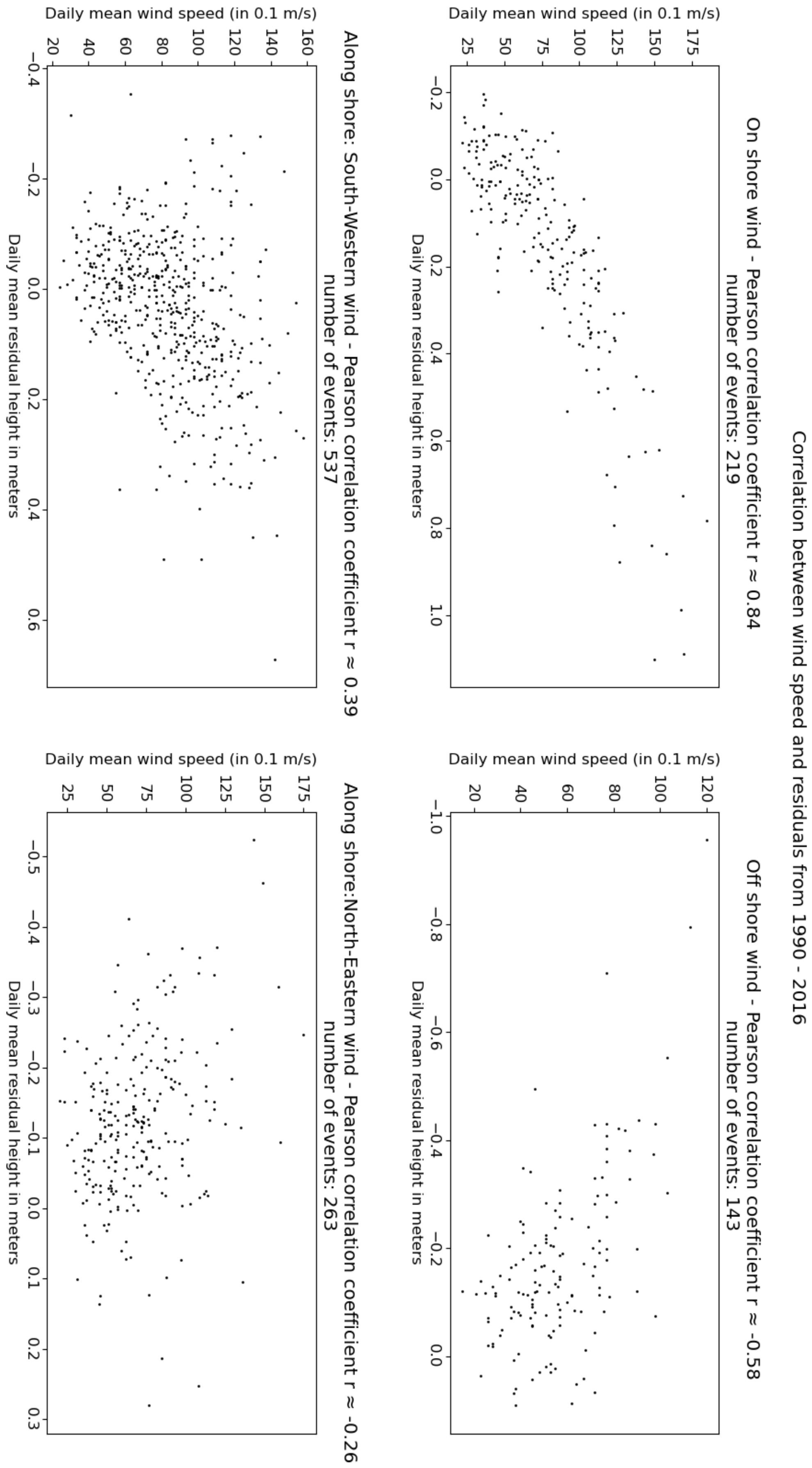

Figure 3.3: The four graphs show the correlation between the wind speed and the low-frequency residuals depending on the wind direction computed from the year 1990 to 2016.

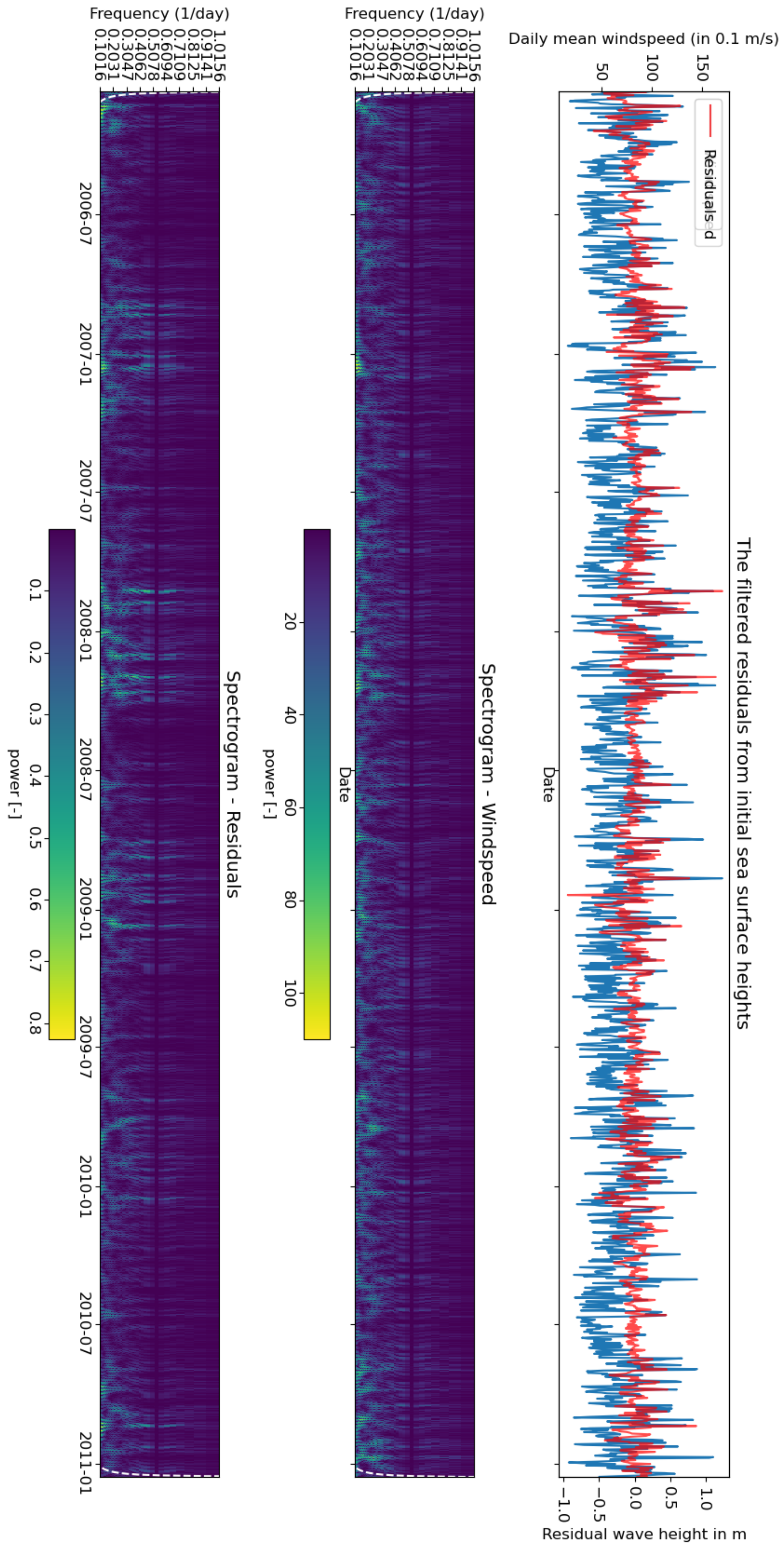

Figure 3.4: The upper graph shows the overlapping of the low-frequency residuals on the wind speed record. The middle figure is the spectrogram computed on the wind speed. The lower graph is spectrogram computed on the low-frequency residuals. This figure applies to the wind speed records and low-frequency residuals from 2006 up to 2011.

4
Discussion

In this section the results and the observations of this study will be discussed. First, the entire sea-level data will be discussed by the computed means and standard deviations. Thereafter, the low-frequency residuals obtained from the sea-level record over a period of 26 years will be discussed. Using data containing the wind speeds, one can then search for correlation between the wind speed and these low-frequency residuals. Finally, using the spectrograms we can observe the low frequencies closely.

4.1. Key findings

The main goal of this study is to understand the contribution of the low-frequency residuals to the sea-level variability in its entirety. Figure 3.1 indicates that the sea-level on the long term rises. This same finding was found in the study carried out by Gerkema and Duran-Matute(2017). However, Gerkema and Duran-Matute(2017) distinguished two seasons per year, whereas in this study four seasons per year have been distinguished. Gerkema and Duran-Matute(2017) found that the annual summer half-year mean sea-level, that is the summer and spring combined, is higher than the annual winter half-year mean sea-level, that is the fall and winter combined. This corresponds to the same finding illustrated in figure 3.1. However, according to Gerkema and Duran-Matute(2017), the rise of the annual mean sea-level in the summer half-year is steeper compared to the rise of the annual mean sea-level in the winter half-year. From figure 3.1 the annual mean sea-level seems to rise in one pace for all the considered seasons. Something to notice from figure 3.1 is the standard deviation of the mean sea-level found per season. The standard deviation corresponding to the sea-level seems barely to rise during the fall season. Whereas, during the summer, winter and spring season one can see that the standard deviation increases on the long term.
From figure 3.2 the variation in the low-frequency residuals compared to the variation in sea-level due to tides and high-frequency waves can be observed. When observing this figure closely, one can notice the large variation in the low-frequency residuals during the first and last quarter of the years compared to the small variation in the low-frequency residuals during the second and third quarter of the years. This observation seems not to be the case for the sea-level variation due to tidal events and high-frequency waves. Figure 3.3 indicates that there is a correlation between the found low-frequency residuals and the wind. Depending on the wind direction the correlation coefficient differs in magnitude and sign. The study carried out by Gerkema and Duran-Matute(2017) found a correlation between the wind and the mean sea-level. However, in this study the correlation is computed between the low-frequency residuals and the wind. Figure 3.4 illustrates the energy contained in the frequencies of the residuals and the frequencies of the wind speed. From the lower spectrogram one can see that the frequencies contain more energy during the first and fourth quarter of the years.

4.2. Interpretation of the results

From the first finding we not only find that the sea-level rises, but more interesting we find that the variation in the sea-level seems to get intenser on the long term. This finding is relevant considering the risk of flooding. The definition of the risk of flooding is been defined based on the annual mean sea-level(Gerkema & Duran-Matute, 2017). However, the forming of barrier island systems is strongly dependent on the intensity of variation in sea-level. It is commonly assumed that the rise in sea-level is mainly due to the climate change. Therefore, we can assume that the rise in sea-level found in figure 3.1 is due to the climate change. Moreover, we find that the standard deviation in sea-level does not rise during fall. This means that the variation in sea-level during fall season is approximately the same during the entire length of the sea-level records. Moreover, we find that the standard deviation of the sea-level during spring and summer are the lowest. Whereas, the variation in sea-level during winter and fall are the highest. Since the tidal events can be considered to be stationary during all seasons in a year we can assume that the difference in standard deviation dependent on the season is because of non-stationary factors. During fall and winter extreme weather events occur which are considered to be non-stationary events. Therefore, we decompose the sea-level record in two signals, the stationary signal due to tidal events and the non-stationary signal containing the low-frequency residuals.

From figure 3.2 this difference in standard deviation dependent on the season can be explained. The variation in amplitude of the low-frequency residuals seem to be much bigger during the first and last quarter of the years, whereas the variation in amplitude of the low-frequency residuals seem to be much smaller during the second and third quarter of the year. Hence, we can assume that the variation in sea-level is strongly influenced by the variation found in the low-frequency residuals.

As similarly done in the study of Gerkema and Duran-Matute(2017) a correlation can be found between these low-frequency residuals and the wind speed. In case the wind blows on-shore, we find the highest positive correlation. Thus, the amplitude of the low-frequency residuals increase when the wind speed increases. In case the wind blows off-shore we find a negative correlation coefficient, thus the low-frequency residuals decrease when the wind speed increases. In case the wind blows along-shore we find the correlation coefficient to have relatively small magnitudes. However, something that can be noticed is the difference in sign. The South-Western wind has a positive correlation coefficient, whereas the North-Eastern wind has a negative correlation coefficient. This can be explained due to the so-called Ekman transport. The Ekman transport is due to the Coriolis effect, where because of the rotation of the Earth the water will flow in the direction of the wind and in the direction perpendicular to the direction of the wind. These findings support the assumption that the intense variation in sea-level due to the low-frequency residuals is correlated with the wind speed. This finding is important in case one would like to design a model that describes the low-frequency residuals. Moreover, in line with the previously made interpretation of figure 3.1 we can say that the rise in standard deviation of the sea-level indicates the rise of the standard deviation of the wind speed. Thus, more extreme storm surges occur on the long term. This corresponds to the conclusion made in the study of Gerkema and Duran-Matute(2017), where it is shown that the annual mean sea-level is influenced significantly due to climatic change in wind speed from any wind direction(Gerkema & Duran-Matute, 2017).

From figure 3.4 we observe that the frequencies in the low-frequency residuals contain more energy during the first and last quarter of the year. This means that the amplitude of the low-frequency residuals have a bigger magnitude during fall and winter, which was already expected from the previous findings. This indicates that the low-frequency residuals are seasonal dependent. Observing the spectrogram closely we find that the frequencies that are the most triggered during fall and winter are all below 0.60 $\frac{1}{day}$, especially the frequencies around 0.10 $\frac{1}{day}$ contain the most energy during fall and winter. This finding has not been found in the research done by Gerkema and Duran-Matute(2017). Finding these frequencies is relevant in case one would like to design a simple model that describes these low-frequency residuals. In appendix B we can observe the spectrogram of specific years. We see that triggered signals in the low-frequency residuals can be found in the spectrogram of the residuals. Thus, when a high-energy signal in the low-frequency residual is observed we find a high-energy signal in the wind speed at the same instance. However, the opposite finding is not found, which may indicate the dependency of the wind direction.

4.3. Limitations

During this study we used one record of the sea-level. This record presented the sea-level at Hoek van Holland from 1900 up to 2018. However, unfortunately the entire record was not homogeneous. Therefore, the Fast Fourier Transformation and the Wavelet Transformation could only by performed on a small part of the data. The Fast Fourier Transformation and the Wavelet Transformation are carried out on a 26-year long part of the record. Moreover, the Wavelet Transformation is a relatively intense computational operation. This made it more complicated to observe a larger frequency spectrum.

4.4. Recommendation

This study was primarily focused on the correlation between the low-frequency residuals and local meteorological events. However, using the Fast Fourier Transformation and the Wavelet Transformation it is possible to research non-local factors. Since the southern North Sea can be seen as a basin, one can wonder if certain frequencies in the sea-level can be observed because of the standing waves occurring in the southern North Sea.

5 Conclusion

The goal of this study is to answer the following research question:

What is the contribution of low-frequency signals to sea-level variability along the Dutch coast?

The low-frequency signals are the low-frequency residuals in the sea-level. From the results obtained one can conclude that these low-frequency residuals are highly correlated to the wind. The correlation coefficients found are dependent on the wind direction. For on-shore wind, one finds the highest positive correlation. This means that in case the wind is blowing on-shore, these low-frequency residuals increase in amplitude. For the case where wind blows off-shore, one finds the highest negative frequency. This means that in case the wind is blowing off-shore, these low-frequency residuals decrease in amplitude. Furthermore, one can conclude that the low-frequency residuals are dependent on the seasons based on the results. During fall, these low-frequency residuals have a larger amplitude, whereas during summer these low-frequency residuals have a relatively small amplitude. This observation can be confirmed by performing the Wavelet Transformation, where one can see greater power in the low-frequency residuals during winter and fall. These findings can be summarised into a brief answer to the research question:

The contribution of the low-frequency signals, thus the low-frequency residuals, is correlated to meteorological events, such as the wind. The amplitude of these low-frequency residuals are seasonal dependent. Thus, the sea-level variability is directly influenced by these low-frequency residuals since the tidal driven waves are stationary, whereas the low-frequency residuals are non-stationary. Because of this non-stationary low-frequency residuals, the sea-level variability can be assumed to be non-stationary. One can therefore conclude that the intensity of variation in sea-level is in a large extent influenced by the intensity of variation in the low-frequency residuals.

References

Audio, N. (n.d.). *Fast fourier transformation fft - basics.* https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft.

Chelton, & Enfield, D. (1986, 08). Ocean signals in tide gauge records. *Journal of Geophysical Research*, *91*, 9081-9098. doi: 10.1029/JB091iB09p09081

de Jong, M. P. C., Holthuijsen, L. H., & Battjes, J. A. (2003). Generation of seiches by cold fronts over the southern north sea. *Journal of Geophysical Research: Oceans*, *108*(C4). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002JC001422 doi: https://doi.org/10.1029/2002JC001422

DEMPSTER, J. (2001). Chapter six - signal analysis and measurement. In J. DEMPSTER (Ed.), *The laboratory computer* (p. 136-171). London: Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780122095511500398 doi: https://doi.org/10.1016/B978-012209551-1/50039-8

FitzGerald, D. M., Fenster, M. S., Argow, B. A., & Buynevich, I. V. (2008). Coastal impacts due to sea-level rise. *Annual Review of Earth and Planetary Sciences*, *36*(1), 601-647. Retrieved from https://doi.org/10.1146/annurev.earth.35.031306.140139 doi: 10.1146/annurev.earth.35.031306.140139

Garg, V. K., & Wang, Y.-C. (2005). 5 - data communication concepts. In W.-K. CHEN (Ed.), *The electrical engineering handbook* (p. 983-988). Burlington: Academic Press. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780121709600500736 doi: https://doi.org/10.1016/B978-012170960-0/50073-6

Gerkema, T., & Duran-Matute, M. (2017). Interannual variability of mean sea level and its sensitivity to wind climate in an inter-tidal basin. *Earth System Dynamics*, *8*(4), 1223–1235. Retrieved from https://esd.copernicus.org/articles/8/1223/2017/ doi: 10.5194/esd-8-1223-2017

Hofmann, J., Behrendt, H., Gilbert, A., Janssen, R., Kannen, A., Kappenberg, J., … Windhorst, W. (2005, 06). Catchment-coastal zone interaction based upon scenario and model analysis: Elbe and the german bight case study. *Regional Environmental Change*, *5*, 54-81. doi: 10.1007/s10113-004-0082-y

Kawachi, Y., Inagaki, S., TOMITA, K., Yamasaki, K., KIN, F., KOSUGA, Y., … Fujisawa, A. (2018, 08). Determination of spatiotemporal structure of fluctuations by statistical averaging method. *Plasma and Fusion Research*, *13*, 3401105-3401105. doi: 10.1585/pfr.13.3401105

KNMI. (n.d.). *Dag gegevens van het weer in nederland.* https://www.knmi.nl/nederland-nu/klimatologie/daggegevens.

López, M., Iglesias, G., & Kobayashi, N. (2012). Long period oscillations and tidal level in the port of ferrol. *Applied Ocean Research*, *38*, 126-134. Retrieved from https://www.sciencedirect.com/science/article/pii/S0141118712000673 doi: https://doi.org/10.1016/j.apor.2012.07.006

Munk, W. H. (1950, Jan.). Origin and generation of waves. *Coastal Engineering Proceedings*, *1*(1), 1. Retrieved from https://icce-ojs-tamu.tdl.org/icce/index.php/icce/article/view/904 doi: 10.9753/icce.v1.1

Nicoll, A. (2020, 07). *The Wavelet Transform for Beginners.* Retrieved from https://www.youtube.com/watch?v=kuuUaqAjeoA

NOAA. (2019, Mar). *What is a seiche?* https://oceanservice.noaa.gov/facts/seiche.html/.

NOAA. (2021, Nov). *What are tides?* https://oceanservice.noaa.gov/facts/tides.html.

Philip L. Woodworth | National Oceanography Centre, U., Liverpool, W. Roland Gehrels | School of Geography, E., Environmental Sciences at the University of Plymouth, U., for Astrodynamics Research, R. S. N. . C. C., & Cooperative Institute for Research in Environmental Sciences, B. C. U., University of Colorado. (2011, June). Nineteenth and twentieth century changes in sea level. *Oceanography*.

Semmlow, J. (2012). Chapter 4 - the fourier transform and power spectrum: Implications and applications. In J. Semmlow (Ed.), *Signals and systems for bioengineers (second edition)* (Second Edition ed., p. 131-165). Boston: Academic Press. Retrieved from https://www.sciencedirect

.com/science/article/pii/B9780123849823000043 doi: https://doi.org/10.1016/B978-0-12-384982-3.00004-3

Tide-forecast. (n.d.). *Hoek van Holland, Netherlands Tide Station Location Guide.* Retrieved from https://www.tide-forecast.com/locations/Hoek-van-Holland-Netherlands

Toffoli, A., & Bitner-Gregersen, E. M. (2017). Types of ocean surface waves, wave classification. In *Encyclopedia of maritime and offshore engineering* (p. 1-8). John Wiley Sons, Ltd. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118476406.emoe077 doi: https://doi.org/10.1002/9781118476406.emoe077

Weisstein, E. W. (2022, May). *Nyquist frequency.* https://mathworld.wolfram.com/NyquistFrequency.html.

Xu, S. (2015, 08). *Discrete Fourier Transform - Simple Step by Step.* Retrieved from https://www.youtube.com/watch?v=mkGsMWi_j4Q

A

Source Code - Python

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 19 19:41:32 2022

@author: Kareem El Sayed
"""

import numpy as np
import matplotlib.pyplot as plt
import netCDF4 as nc
import pandas as pd
import scipy as sc
import pywt
import pycwt
from scipy.signal import chirp
from scipy import signal
import datetime as datetimee
import calendar
from scipy.interpolate import interp1d
import matplotlib.dates as mdates
from matplotlib.dates import date2num
import xarray as xr

plt.rcParams['figure.figsize'] = [8, 6]
plt.rcParams.update({'font.size' : 12})
#%% Loading in data
fn = 'C:/Users/Kelsa/OneDrive/Bureaublad/BEP/WaterLevel_records/HOEKVHLD.nc' #path to netcdf
    file
ds = nc.Dataset(fn) # read as netcdf dataset

tides_const = pd.read_csv("C:/Users/Kelsa/OneDrive/Bureaublad/BEP/WaterLevel_records/
    tidal_const.txt") #Tidal Constituents
tides_const['freq'] = tides_const['freq'] * 24 #Frequency from 1/h to 1/day

ds_pandas = xr.open_dataset('C:/Users/Kelsa/OneDrive/Bureaublad/BEP/WaterLevel_records/
    HOEKVHLD.nc')
df_pandas = ds_pandas['sea_surface_height'].to_dataframe()
df_pandas['sea_surface_height'] = ds_pandas['sea_surface_height'].to_dataframe()
df_pandas['time'] = ds_pandas['time'].to_dataframe()

with nc.Dataset(fn) as dataset:
    time = nc.num2date(dataset['time'][:],
                       dataset['time'].units,
                       only_use_cftime_datetimes=False)
    sea_height = dataset['sea_surface_height'][:] #get data for sea height in m - stable part
        - beginning from 1987-01-01

mul = 1                                    #mul is used to interpolate between the data and
    get a higher frequency spectrum, however this is not used

```

```
day_ = (144)                                    # 1 day = 144 data points if mul = 1
month_ = (144 * 30)                             # 1 month = 144 * 30 data points
year_ = (144 * 365)                             # 1 year = 144 * 365 data points

day = (144) * mul                                    # 1 day = 144 data * mul points if mul >= 1
month = (144 * 30) * mul                             # 1 month = 144 * 30 * mul data points if
    mul >= 1
year = (144 * 365) * mul                             # 1 year = 144 * 365 * mul data points if
    mul >= 1

SCH_lon = ds['lon'][:]
SCH_lat = ds['lat'][:]

ds_SCH = datetimee.date(1987,1,1)
de_SCH = datetimee.date(2018,1,12)

def ndays(date1, date2):
    return (date2-date1).days

numofdays = ndays(ds_SCH, de_SCH)
#%%
df_pandas["time"] = pd.to_datetime(df_pandas["time"])


def MaandnaarSeizoen(x):
    global seizoen
    if x == 1 or x == 2 or x == 3:
        seizoen = "Winter"
    elif x == 4 or x == 5 or x == 6:
        seizoen = "Spring"
    elif x == 7 or x == 8 or x == 9:
        seizoen = "Summer"
    elif x == 10 or x == 11 or x == 12:
        seizoen = "Fall"
    else:
        seizoen = np.nan
    return seizoen

def SeizoenElkJaar(x):
    global Seizoen_Jaar
    if x.month == 1 or x.month == 2 or x.month == 3:
        Seizoen_Jaar = "Winter" + str(x.year)
    elif x.month == 4 or x.month == 5 or x.month == 6:
        Seizoen_Jaar = "Lente" + str(x.year)
    elif x.month == 7 or x.month == 8 or x.month == 9:
        Seizoen_Jaar = "Zomer" + str(x.year)
    elif x.month == 10 or x.month == 11 or x.month == 12:
        Seizoen_Jaar = "Herfst" + str(x.year)
    else:
        Seizoen_Jaar = np.nan
    return Seizoen_Jaar


def MaandnaarMaand(x):
    global maand
    if x == 1:
        maand = "January"
    elif x == 2:
        maand = "February"
    elif x == 3:
        maand = "March"
    elif x == 4:
        maand = "April"
    elif x == 5:
        maand = "May"
    elif x == 6:
        maand = "June"
    elif x == 7:
        maand = "July"
    elif x == 8:
        maand = "August"
```

```
    elif x == 9:
        maand = "September"
    elif x == 10:
        maand = "October"
    elif x == 11:
        maand = "November"
    elif x == 12:
        maand = "December"
    else:
        maand = np.nan
    return maand


def MaandElkJaar(x):
    global maand
    if x.month == 1:
        maand_jaar = "January" + str(x.year)
    elif x.month == 2:
        maand_jaar = "February" + str(x.year)
    elif x.month == 3:
        maand_jaar = "March" + str(x.year)
    elif x.month == 4:
        maand_jaar = "April" + str(x.year)
    elif x.month == 5:
        maand_jaar = "May" + str(x.year)
    elif x.month == 6:
        maand_jaar = "June" + str(x.year)
    elif x.month == 7:
        maand_jaar = "July" + str(x.year)
    elif x.month == 8:
        maand_jaar = "August" + str(x.year)
    elif x.month == 9:
        maand_jaar = "September" + str(x.year)
    elif x.month == 10:
        maand_jaar = "October" + str(x.year)
    elif x.month == 11:
        maand_jaar = "November" + str(x.year)
    elif x.month == 12:
        maand_jaar = "December" + str(x.year)
    else:
        maand_jaar = np.nan
    return maand_jaar


def YearlyMean(x):
    global jaar
    jaar = str(x)
    return jaar



df_pandas['Season'] = df_pandas['time'].dt.month.apply(lambda x : MaandnaarSeizoen(x))
df_pandas['Maand'] = df_pandas['time'].dt.month.apply(lambda x : MaandnaarMaand(x))
df_pandas['Jaar'] = df_pandas['time'].dt.year.apply(lambda x : YearlyMean(x))
df_pandas['Season Per Year'] = df_pandas['time'].dt.date.apply(lambda x : SeizoenElkJaar(x))
df_pandas['Month Per Year'] = df_pandas['time'].dt.date.apply(lambda x : MaandElkJaar(x))

SeasonalVariations = df_pandas.groupby(df_pandas["Season"]).agg(['mean','std'])
SeasonalVariations = SeasonalVariations.reindex(['Winter', 'Spring', 'Summer', 'Fall'])

MonthlyVariations = df_pandas.groupby(df_pandas["Maand"]).agg(['mean','std'])
MonthlyVariations = MonthlyVariations.reindex(['January', 'February', 'March', 'April', 'May'
    , 'June', 'July', 'August', 'September', 'October', 'November', 'December'])

YearlyVariations = df_pandas.groupby(df_pandas["Jaar"]).agg(['mean','std'])

SeasonPerYearVariation = df_pandas.groupby(df_pandas["Season Per Year"]).agg(['mean','std'])
#%%
Jaartallen = YearlyVariations.index

SeasonalVariations = np.array(SeasonalVariations)
```

```
MonthlyVariations = np.array(MonthlyVariations)
YearlyVariations_ = np.array(YearlyVariations)

SV = np.reshape(SeasonalVariations, -1)
MV = np.reshape(MonthlyVariations, -1)
YV = np.reshape(YearlyVariations_, -1)

m, b = np.polyfit(np.arange(len(YearlyVariations_)), YV[::2], 1) #Fitting a linear line for
    MSL variability


Seizoenen = ['Winter (JFM)', 'Spring (AMJ)', 'Summer (JAS)', 'Fall (OND)']
Maanden = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', '
    September', 'October', 'November', 'December']

fig, axs = plt.subplots(4, 1, sharex=False)
fig.subplots_adjust(hspace=0.8)

axs[0].bar(np.arange(4), SV[::2], yerr = SV[1::2], align='center', alpha=0.75, ecolor='black'
    , capsize=2, color = 'black')
axs[0].xaxis.set_ticks(np.arange(4), Seizoenen)
axs[0].set_xticks(np.arange(4))
axs[0].set_xticklabels(Seizoenen)
axs[0].set_ylabel("Sea Level Variability [m]")
axs[0].set_title("Seasonal means from 1900-2018 with standard deviations")
axs[0].grid(True)

axs[1].bar(np.arange(12), MV[::2], yerr = MV[1::2], align='center', alpha=0.75, ecolor='black
    ', capsize=2, color = 'black')
axs[1].set_xticks(np.arange(12))
axs[1].set_xticklabels(Maanden)
axs[1].set_ylabel("Sea Level Variability [m]")
axs[1].set_title("Monthly means over 1900-2018 with standard deviation")
axs[1].grid(True)


axs[2].plot(np.arange(len(YearlyVariations_)), YV[::2], 'ko'
            ,np.arange(len(YearlyVariations_)), m*np.arange(len(YearlyVariations_))+b,
            '--k')
axs[2].set_xticks(np.arange(len(YearlyVariations_)))
axs[2].set_xticklabels(Jaartallen, rotation = 90)
axs[2].set_ylabel("Sea Level Variability [m]")
axs[2].set_title("Trend - Yearly means from 1900-2018")
axs[2].set_xlabel('Date')
axs[2].legend(['Datapoints', 'Linear Regression'])
axs[2].grid(True)

n, v = np.polyfit(np.arange(len(YearlyVariations_)), YV[1::2], 1) #Fitting a linear line for
    MSL variability

axs[3].plot(np.arange(len(YearlyVariations_)), YV[1::2], 'ko',np.arange(len(YearlyVariations_
    )), n*np.arange(len(YearlyVariations_))+v,
            '--k')
axs[3].set_xticks(np.arange(len(YearlyVariations_)))
axs[3].set_xticklabels(Jaartallen, rotation = 90)
axs[3].set_ylabel("Sea Level Variability [m]")
axs[3].set_title("Trend - Yearly standard deviation from 1900-2018")
axs[3].set_xlabel('Date')
axs[3].legend(['Datapoints', 'Linear Regression'], loc = 4)
axs[3].grid(True)
#%%
herfstjaren = []
winterjaren = []
lentejaren = []
zomerjaren = []

herfstmeanyear = []
wintermeanyear = []
lentemeanyear = []
zomermeanyear = []
```

```
herfststdyear = []
winterstdyear = []
lentestdyear = []
zomerstdyear = []

for i in range(len(SeasonPerYearVariation)):
    if SeasonPerYearVariation.index[i][0] == 'H':
        herfstjaren.append(SeasonPerYearVariation.index[i][-4:])
        herfstmeanyear.append(SeasonPerYearVariation['sea_surface_height']['mean'][i])
        herfststdyear.append(SeasonPerYearVariation['sea_surface_height']['std'][i])
    if SeasonPerYearVariation.index[i][0] == 'Z':
        zomerjaren.append(SeasonPerYearVariation.index[i][-4:])
        zomermeanyear.append(SeasonPerYearVariation['sea_surface_height']['mean'][i])
        zomerstdyear.append(SeasonPerYearVariation['sea_surface_height']['std'][i])
    if SeasonPerYearVariation.index[i][0] == 'L':
        lentejaren.append(SeasonPerYearVariation.index[i][-4:])
        lentemeanyear.append(SeasonPerYearVariation['sea_surface_height']['mean'][i])
        lentestdyear.append(SeasonPerYearVariation['sea_surface_height']['std'][i])
    if SeasonPerYearVariation.index[i][0] == 'W':
        winterjaren.append(SeasonPerYearVariation.index[i][-4:])
        wintermeanyear.append(SeasonPerYearVariation['sea_surface_height']['mean'][i])
        winterstdyear.append(SeasonPerYearVariation['sea_surface_height']['std'][i])


herfstmeanyear_ip = np.interp(np.array(Jaartallen).astype(float),
               herfstjaren,
              herfstmeanyear)

wintermeanyear_ip = np.interp(np.array(Jaartallen).astype(float),
               winterjaren,
              wintermeanyear)

lentemeanyear_ip = np.interp(np.array(Jaartallen).astype(float),
               lentejaren,
              lentemeanyear)

zomermeanyear_ip = np.interp(np.array(Jaartallen).astype(float),
               zomerjaren,
              zomermeanyear)

herfststdyear_ip = np.interp(np.array(Jaartallen).astype(float),
               herfstjaren,
              herfststdyear)

winterstdyear_ip = np.interp(np.array(Jaartallen).astype(float),
               winterjaren,
              winterstdyear)

lentestdyear_ip = np.interp(np.array(Jaartallen).astype(float),
               lentejaren,
              lentestdyear)

zomerstdyear_ip = np.interp(np.array(Jaartallen).astype(float),
               zomerjaren,
              zomerstdyear)
#%%

fig, axs = plt.subplots(1, 1, sharex=False)
#fig.subplots_adjust(hspace=0.8)

nh, vh = np.polyfit(np.arange(len(Jaartallen)), herfstmeanyear_ip, 1) #Fitting a linear line
    for MSL variability
nl, vl = np.polyfit(np.arange(len(Jaartallen)), lentemeanyear_ip, 1)
nz, vz = np.polyfit(np.arange(len(Jaartallen)), zomermeanyear_ip, 1)
nw, vw = np.polyfit(np.arange(len(Jaartallen)), wintermeanyear_ip, 1)

nhs, vhs = np.polyfit(np.arange(len(Jaartallen)), herfststdyear_ip, 1) #Fitting a linear line
     for MSL variability
nls, vls = np.polyfit(np.arange(len(Jaartallen)), lentestdyear_ip, 1)
nzs, vzs = np.polyfit(np.arange(len(Jaartallen)), zomerstdyear_ip, 1)
nws, vws = np.polyfit(np.arange(len(Jaartallen)), winterstdyear_ip, 1)
```

```
p1, = plt.plot(np.array(Jaartallen), herfstmeanyear_ip, label = 'Fall (OND)', marker = 'o',
    alpha = 0.25)
p2, = plt.plot(np.array(Jaartallen), wintermeanyear_ip, label = 'Winter (JFM)', marker = 'o',
     alpha = 0.25)
p3, = plt.plot(np.array(Jaartallen), zomermeanyear_ip, label = 'Summer (JAS)', marker = 'o',
    alpha = 0.25)
p4, = plt.plot(np.array(Jaartallen), lentemeanyear_ip, label = 'Spring (AMJ)', marker = 'o',
    alpha = 0.25)

p5, = plt.plot(np.arange(len(Jaartallen)), nh*np.arange(len(Jaartallen)) + vh, label = '
    Linear Regression - Fall', linestyle = '--', color = 'blue')
p6, = plt.plot(np.arange(len(Jaartallen)), nl*np.arange(len(Jaartallen)) + vl, label = '
    Linear Regression - Spring', linestyle = '--', color = 'red')
p7, = plt.plot(np.arange(len(Jaartallen)), nz*np.arange(len(Jaartallen)) + vz, label = '
    Linear Regression - Summer', linestyle = '--', color = 'green')
p8, = plt.plot(np.arange(len(Jaartallen)), nw*np.arange(len(Jaartallen)) + vw, label = '
    Linear Regression - Winter', linestyle = '--', color = 'orange')

plt.ylabel('Sea-Level [m]')
plt.xlabel('Year')

plt.xticks(np.arange(len(Jaartallen)), Jaartallen, rotation = 65)

l1 = plt.legend(handles=[p1,p2,p3,p4], loc = 'upper left')
plt.gca().add_artist(l1)
plt.legend(handles=[p5,p6,p7,p8], loc = 'lower right')

plt.title('Mean sea-level per season from 1900-2018 at Hoek van Holland')
#%%
p1, = plt.plot(np.array(Jaartallen), herfststdyear_ip, label = 'Fall (OND)', marker = 'o',
    alpha = 0.25)
p2, = plt.plot(np.array(Jaartallen), winterstdyear_ip, label = 'Winter (JFM)', marker = 'o',
    alpha = 0.25)
p3, = plt.plot(np.array(Jaartallen), zomerstdyear_ip, label = 'Summer (JAS)', marker = 'o',
    alpha = 0.25)
p4, = plt.plot(np.array(Jaartallen), lentestdyear_ip, label = 'Spring (AMJ)', marker = 'o',
    alpha = 0.25)

p5, = plt.plot(np.arange(len(Jaartallen)), nhs*np.arange(len(Jaartallen)) + vhs, label = '
    Linear Regression - Fall', linestyle = '--', color = 'blue')
p6, = plt.plot(np.arange(len(Jaartallen)), nls*np.arange(len(Jaartallen)) + vls, label = '
    Linear Regression - Spring', linestyle = '--', color = 'red')
p7, = plt.plot(np.arange(len(Jaartallen)), nzs*np.arange(len(Jaartallen)) + vzs, label = '
    Linear Regression - Summer', linestyle = '--', color = 'green')
p8, = plt.plot(np.arange(len(Jaartallen)), nws*np.arange(len(Jaartallen)) + vws, label = '
    Linear Regression - Winter', linestyle = '--', color = 'orange')

plt.ylabel('Sea-Level [m]')
plt.xlabel('Year')

plt.xticks(np.arange(len(Jaartallen)), Jaartallen, rotation = 65)

l1 = plt.legend(handles=[p1,p2,p3,p4], loc = 'upper left')
plt.gca().add_artist(l1)
plt.legend(handles=[p5,p6,p7,p8], loc = 'lower right')

plt.title('Standard deviation of sea-level per season from 1900-2018 at Hoek van Holland')
#%%

fig, axs = plt.subplots(2, 1, sharex=True)
#fig.subplots_adjust(hspace=0.8)

nh, vh = np.polyfit(np.arange(len(Jaartallen)), herfstmeanyear_ip, 1) #Fitting a linear line
    for MSL variability
nl, vl = np.polyfit(np.arange(len(Jaartallen)), lentemeanyear_ip, 1)
nz, vz = np.polyfit(np.arange(len(Jaartallen)), zomermeanyear_ip, 1)
nw, vw = np.polyfit(np.arange(len(Jaartallen)), wintermeanyear_ip, 1)

nhs, vhs = np.polyfit(np.arange(len(Jaartallen)), herfststdyear_ip, 1) #Fitting a linear line
     for MSL variability
```

```
nls, vls = np.polyfit(np.arange(len(Jaartallen)), lentestdyear_ip, 1)
nzs, vzs = np.polyfit(np.arange(len(Jaartallen)), zomerstdyear_ip, 1)
nws, vws = np.polyfit(np.arange(len(Jaartallen)), winterstdyear_ip, 1)

p1, = axs[0].plot(np.array(Jaartallen), herfstmeanyear_ip, label = 'Fall (OND)', marker = 'o', alpha = 0.25)
p2, = axs[0].plot(np.array(Jaartallen), wintermeanyear_ip, label = 'Winter (JFM)', marker = 'o', alpha = 0.25)
p3, = axs[0].plot(np.array(Jaartallen), zomermeanyear_ip, label = 'Summer (JAS)', marker = 'o', alpha = 0.25)
p4, = axs[0].plot(np.array(Jaartallen), lentemeanyear_ip, label = 'Spring (AMJ)', marker = 'o', alpha = 0.25)

p5, = axs[0].plot(np.arange(len(Jaartallen)), nh*np.arange(len(Jaartallen)) + vh, label = 'Linear Regression - Fall', linestyle = '--', color = 'blue')
p6, = axs[0].plot(np.arange(len(Jaartallen)), nl*np.arange(len(Jaartallen)) + vl, label = 'Linear Regression - Spring', linestyle = '--', color = 'red')
p7, = axs[0].plot(np.arange(len(Jaartallen)), nz*np.arange(len(Jaartallen)) + vz, label = 'Linear Regression - Summer', linestyle = '--', color = 'green')
p8, = axs[0].plot(np.arange(len(Jaartallen)), nw*np.arange(len(Jaartallen)) + vw, label = 'Linear Regression - Winter', linestyle = '--', color = 'orange')

axs[0].set_ylabel('Sea-Level [m]')
axs[0].set_xlabel('Year')

plt.xticks(np.arange(len(Jaartallen)), Jaartallen, rotation = 65)

l1 = axs[0].legend(handles=[p1,p2,p3,p4], loc = 'upper left')
axs[0].add_artist(l1)
axs[0].legend(handles=[p5,p6,p7,p8], loc = 'lower right')

axs[0].set_title('Mean sea-level per season from 1900-2018 at Hoek van Holland')

p1, = axs[1].plot(np.array(Jaartallen), herfststdyear_ip, label = 'Fall (OND)', marker = 'o', alpha = 0.25)
p2, = axs[1].plot(np.array(Jaartallen), winterstdyear_ip, label = 'Winter (JFM)', marker = 'o', alpha = 0.25)
p3, = axs[1].plot(np.array(Jaartallen), zomerstdyear_ip, label = 'Summer (JAS)', marker = 'o', alpha = 0.25)
p4, = axs[1].plot(np.array(Jaartallen), lentestdyear_ip, label = 'Spring (AMJ)', marker = 'o', alpha = 0.25)

p5, = axs[1].plot(np.arange(len(Jaartallen)), nhs*np.arange(len(Jaartallen)) + vhs, label = 'Linear Regression - Fall', linestyle = '--', color = 'blue')
p6, = axs[1].plot(np.arange(len(Jaartallen)), nls*np.arange(len(Jaartallen)) + vls, label = 'Linear Regression - Spring', linestyle = '--', color = 'red')
p7, = axs[1].plot(np.arange(len(Jaartallen)), nzs*np.arange(len(Jaartallen)) + vzs, label = 'Linear Regression - Summer', linestyle = '--', color = 'green')
p8, = axs[1].plot(np.arange(len(Jaartallen)), nws*np.arange(len(Jaartallen)) + vws, label = 'Linear Regression - Winter', linestyle = '--', color = 'orange')

axs[1].set_ylabel('Sea-Level [m]')
axs[1].set_xlabel('Year')

#plt.xticks(np.arange(len(Jaartallen)), Jaartallen, rotation = 65)

l1 = axs[1].legend(handles=[p1,p2,p3,p4], loc = 'upper left')
axs[1].add_artist(l1)
plt.legend(handles=[p5,p6,p7,p8], loc = 'lower right')

plt.title('Standard deviation of sea-level per season from 1900-2018 at Hoek van Holland')
#%%
duration = 5*year
duration_ = 5*year_
                                                #From 31-12-1986 23:00 the sea height was measured every 10 minutes
end = 6 * year + 51840                          # x years from end(2018-01-13) of dataset
end_ = 6 * year_ + (893340)

#From ~04-04-2016 the data is each minute (HOEKVHLD)
```

```
f = sea_height.flatten()                                # in meters
t = np.array(time).astype(datetimee.datetime).flatten() # in days

def to_float(d, epoch=t[0]):
    return (d - epoch) / datetimee.timedelta(minutes=10)

to = np.arange(ds_SCH, de_SCH,
               datetimee.timedelta(days=1/(day))).astype(datetimee.datetime)

fo = np.interp(to_float(to)[-end-duration:-end+1].astype(float),
               to_float(t).astype(float),
               f)

inc = 1

tod = to[::inc]
fod = fo[::inc]

fig, axs = plt.subplots(3, 1, sharex=True)
fig.subplots_adjust(hspace=0.2)

#axs[0].xaxis.set_minor_locator(mdates.DayLocator())
axs[0].plot(tod[-end-duration:-end+1], fod, LineWidth = 0.2, color = 'red')
axs[0].set_ylabel("Sea-Level [m]")
axs[0].set_title("Interpolated Data")

axs[1].plot(t[-end_-duration_:-end_+1], f[-end_-duration_:-end_+1], LineWidth = 0.2)
axs[1].set_ylabel("Sea-Level [m]")
axs[1].set_title("Initial Data")

axs[2].plot(tod[-end-duration:-end+1], fod, LineWidth = 0.2, color = 'red')
axs[2].plot(t[-end_-duration_:-end_+1], f[-end_-duration_:-end_+1], LineWidth = 0.2)
axs[2].set_title("Overlapping initial data and interpolated data")
axs[2].set_ylabel("Sea-Level [m]")
axs[2].set_xlabel("Date")

plt.show()

print(tod[-end-duration:-end+1])
#%%
n = len(fod)                            # Total number of discrete points
T = duration_/144                       # Total sampling time in days
dt = T/n                                # sampling period
fs = 1/dt                               # sampling frequency

fhat = np.fft.rfft(fod)                 # Compute the FFT for a real signal
freq = np.fft.rfftfreq(n,dt)            # Natural frequencies (1/dt)

PSD = np.abs((fhat * np.conj(fhat)))                #Power spectrum(power per frequency)
L = np.arange(1, np.floor(n/(20*mul)), dtype = 'int')    #Spectrum of frequencies


#plt.axvline(1.00, alpha = 0.25) #M2 Tide
#plt.axvline(0.07, alpha = 0.25) #K1 Tide
#plt.axvline(1.93, alpha = 0.25) #Mf Tide

plt.xscale("log")
plt.yscale("log")
plt.plot(freq[L], PSD[L], color = 'c', LineWidth = 0.75)
plt.xlabel('frequency[1/day]')
plt.ylabel(r'The power[$m^{2}$]')
plt.title('Periodogram Sea Level: 1990 - 2018')
#%% Getting the low frequency waves
L = np.arange(1, np.floor(n/(100*mul)), dtype = 'int')          #Spectrum of frequencies
plt.xscale("log")
plt.yscale("log")
plt.xticks(np.linspace(0,7,100), rotation = 65)
plt.plot(freq[L], PSD[L], color = 'c', LineWidth = 0.75)
plt.xlabel('frequency[1/day]')
plt.ylabel(r'The power[$m^{2}$]')
```

```
plt.legend()
#%% Getting the tides and high-frequency signals
tides_24h = np.array(tides_const['freq'][tides_const['freq'] < 1]) #Tides with periods larger
     than 24 hour
indices = np.zeros(len(PSD))

for i in tides_24h:
    nearest_freq_tide = np.abs(freq - i)
    indices[np.where(nearest_freq_tide == np.amin(nearest_freq_tide))] = 1

indices[freq > 1] = 1
indices = np.array(indices, dtype = bool)

PSDClean = PSD * indices
fhatclean = fhat * indices
ffilt = np.real_if_close(np.fft.irfft(fhatclean), tol = 1000)
plt.figure()
plt.xscale("linear")
plt.yscale("log")
plt.title('Periodogram Sea-Level Hoek van Holland: 2015 - 2016')
#plt.plot(tod[-end-duration:-end], ffilt, LineWidth = 0.5)
plt.plot(freq[L], PSDClean[L], color = 'c', LineWidth = 0.75, label = 'Irrelevant tidal
    frequencies and high-frequencies')

plt.show()
#%% Residuals
tides_24h = np.array(tides_const['freq'][tides_const['freq'] < 1]) #Tides with periods larger
     than 24 hour
indices = np.ones(len(PSD))

for i in tides_24h:
    nearest_freq_tide = np.abs(freq - i)
    indices[np.where(nearest_freq_tide == np.amin(nearest_freq_tide))] = 0

indices[freq > 1] = 0
indices = np.array(indices, dtype = bool)

PSDClean = PSD * indices
fhatclean = fhat * indices
resid = np.real_if_close(np.fft.irfft(fhatclean), tol = 1000)
#plt.figure()
plt.xscale("linear")
plt.yscale("log")
plt.axvline(1.00, alpha = 0.25) #M2 Tide
plt.axvline(0.07, alpha = 0.25) #K1 Tide
plt.axvline(1.93, alpha = 0.25) #Mf Tide
#plt.plot(tod[-end-duration:-end], resid, LineWidth = 0.5)
plt.plot(freq[L], PSDClean[L], color = 'red', LineWidth = 0.75, alpha = 0.65, label = '
    Relevant low-frequencies [period > 24 hours]')
plt.xlabel('frequency[1/day]')
plt.ylabel(r'The power[$m^{2}$]')
plt.legend()
plt.show()
#%%
fig, axs = plt.subplots(1, 1, sharex=True)

axs.plot(tod[-end-duration:-end], fod[:-1], color = 'darkorange', label = 'Original sea level
     without filtering')
axs.plot(tod[-end-duration:-end], ffilt, color = 'lightseagreen', alpha = 0.8, label = '
    Irrelevant tidal signals and high frequency signals')
axs.plot(tod[-end-duration:-end], resid, color = 'indigo', alpha = 0.6, label = 'Relevant low
    -frequency residuals')

axs.set_xlabel("Date")
axs.set_ylabel("Sea level in meters")
axs.set_title("Decomposition of the sea level signal into the low-frequency residuals and
    irrelevant tidal signals and high frequency signals")

axs.xaxis.set_minor_locator(mdates.YearLocator())

axs.legend()
```

```
#%% Wavelet Transformation and Plotting Scalogram
scales = np.arange(100,1101,25)
dt = T/n
coef, freqs = pywt.cwt(resid, scales, 'morl',dt)                                    #Finding
    CWT using a morlet wavelet
_,_,_,coi,_,_ = pycwt.wavelet.cwt(resid, dt, wavelet = 'morlet', freqs = freqs)
coif = 1/coi

fig, axs = plt.subplots(2, 1, sharex=True)
# Remove horizontal space between axes
fig.subplots_adjust(hspace=0.5)

axs[0].plot(tod[-end-duration:-end], resid)
axs[0].set_ylabel("Residual-level in meters")
axs[0].set_xlabel("Date")
axs[0].set_title("The extracted residuals from the original sea-level")

axs[1].pcolor(tod[-end-duration:-end], freqs, abs(coef))
axs[1].plot(tod[-end-duration:-end], coif, linestyle = "--", color = 'white')
axs[1].set_ylim(freqs[-1], freqs[0])
axs[1].set_title("Spectrogram Wavelet Transformation using a Morlet wavelet")
axs[1].set_ylabel("Frequency (1/day)")
axs[1].set_xlabel("Date")

plt.show()
#%%
# BRON: KONINKLIJK NEDERLANDS  METEOROLOGISCH INSTITUUT (KNMI) (KNMI, n.d.)
weer = pd.read_csv("C:/Users/Kelsa/OneDrive/Bureaublad/BEP/WaterLevel_records/WeatherHVHLD.
    txt",
                   skiprows = 50, skipinitialspace=True, parse_dates = [1], index_col = 1,
                       dayfirst=True)
#%%
ds_weerHVHLD = datetimee.date(1971,1,1)
de_weerHVHLD = datetimee.date(2022,5,2)

def ndays(date1, date2):
    return (date2-date1).days

numofdays = ndays(de_weerHVHLD, de_weerHVHLD)
#%%
FG = weer['FG']
PG = weer['PG']
DDVEC = weer['DDVEC']
FHVEC = weer['FHVEC']
#%%
lowerbound = tod[-end-duration]
upperbound = tod[-end]

print(tod[-end-duration], tod[-end])

FGi = FG[lowerbound : upperbound].interpolate()
PGi = PG[lowerbound : upperbound].interpolate()
DDVECi = DDVEC[lowerbound : upperbound].interpolate()
FHVECi = FHVEC[lowerbound : upperbound].interpolate()

residmean = np.mean(resid[:].reshape(-1,144), axis = 1) #computing the daily mean of the low-
    frequency residuals

time_nw = FGi[((DDVECi[:] - 43) <= 275) & ((DDVECi[:] - 43) >= 265)].index
time_se = FGi[((DDVECi[:] - 43) <= 95) & ((DDVECi[:] - 43) >= 85)].index
time_sw = FGi[((DDVECi[:] - 43) <= 185) & ((DDVECi[:] - 43) >= 175)].index
time_ne = FGi[((DDVECi[:] - 43) <= 5) & ((DDVECi[:] - 43) >= -5)].index

fig, axs = plt.subplots(2, 1)
fig.subplots_adjust(hspace=0.2)

#axs[0].xaxis.set_minor_locator(mdates.DayLocator())
axs[0].plot(FGi)

# for i in range(len(time_ne)):
#     axs[0].axvline((time_ne[i]), color = 'green')
```

```
axs[0].set_ylabel("Daily mean wind speed expressed in 0.1 m/s") (KNMI, n.d.)

axs[1].plot(PGi)
axs[1].set_ylabel("Daily mean sea-level pressure expressed in 0.1 hPa") (KNMI, n.d.)

ax2 = axs[0].twinx()
ax2.plot(tod[-end-duration:-end:144], residmean, color = 'red', alpha = 0.7)
ax2.set_ylabel("Residual wave height in m")

plt.show()
#%%Wavelet windspeed
scales_windspeed = np.arange(0.8,8.1,0.1)
dt_ws = 1
FGi_ws = np.array(FGi)
coef_ws, freqs_ws = pywt.cwt(FGi_ws, scales_windspeed, 'morl',dt_ws)
                                      #Finding CWT using a morlet wavelet
_,_,_,coi_ws,_,_ = pycwt.wavelet.cwt(FGi_ws, dt_ws, wavelet = 'morlet', freqs = freqs_ws)
coif_ws = 1/coi_ws #Cone of Influence

#Wavelet residuals - For small timeframe(high computational power needed, higher resolution)
# scales_resid = np.arange(110,1211,50)
# dt_rs = T/n
# coef_rs, freqs_rs = pywt.cwt(resid, scales_resid, 'morl',dt_rs)
                                      #Finding CWT using a morlet wavelet
# _,_,_,coi_rs,_,_ = pycwt.wavelet.cwt(resid, dt_rs, wavelet = 'morlet', freqs = freqs_rs)
# coif_rs = 1/coi_rs


#Wavelet Residuals - For large timeframe(lower computational power needed, lower resolution)
scales_resid = np.arange(0.8,8.1,0.1)
dt_rs = 1
coef_rs, freqs_rs = pywt.cwt(residmean, scales_resid, 'morl',dt_rs)
                                      #Finding CWT using a morlet wavelet
_,_,_,coi_rs,_,_ = pycwt.wavelet.cwt(residmean, dt_rs, wavelet = 'morlet', freqs = freqs_rs)
coif_rs = 1/coi_rs

fig, axs = plt.subplots(3, 1, sharex=True)
fig.subplots_adjust(hspace=0.5)

axs[0].plot(FGi)
# for i in range(len(time_ne)):
#     axs[0].axvline((time_ne[i]), color = 'green')

axs[0].set_ylabel("Daily mean windspeed (in 0.1 m/s)")
axs[0].set_xlabel("Date")
axs[0].set_title("The filtered residuals from initial sea surface heights")
axs[0].legend(['Wind speed'], loc = 2)#, 'NE-Wind'], loc = 2)
ax2 = axs[0].twinx()
ax2.plot(tod[-end-duration:-end:144], residmean, color = 'red', alpha = 0.7, label = '
    Residuals')
ax2.set_ylabel("Residual wave height in m")
ax2.legend()

#---------------------------------------------------------------------------------
axs[1].pcolor(tod[-end-duration:-end+1:144], freqs_ws, abs(coef_ws))
# for i in range(len(time_ne)):
#     axs[1].axvline((time_ne[i]), color = 'white', alpha = 0.5)

axs[1].plot(tod[-end-duration:-end+1:144], coif_ws, linestyle = "--", color = 'white')
axs[1].set_ylim(freqs_ws[-1], freqs_ws[0])
axs[1].set_title("Spectrogram - Windspeed")
axs[1].set_ylabel("Frequency (1/day)")
axs[1].set_xlabel("Date")
axs[1].set_yticks(np.linspace(freqs_ws[0], freqs_ws[-1], 10))
axs[1].grid(True)

im_ws = axs[1].pcolor(tod[-end-duration:-end+1:144], freqs_ws, abs(coef_ws))
fig.colorbar(im_ws, ax=axs[1], orientation = 'horizontal', label = 'power [-]')
#---------------------------------------------------------------------------------
#axs[2].pcolor(tod[-end-duration:-end], freqs_rs, abs(coef_rs)) #Enable when using Wavelet
```

```
    for small timeframe
axs[2].pcolor(tod[-end-duration:-end:144], freqs_rs, abs(coef_rs)) #Enable when using Wavelet
     for large timeframe
# for i in range(len(time_ne)):
#     axs[2].axvline((time_ne[i]), color = 'white', alpha = 0.5)

#axs[2].plot(tod[-end-duration:-end], coif_rs, linestyle = "--", color = 'white') #Enable
    when using Wavelet for small timeframe
axs[2].plot(tod[-end-duration:-end:144], coif_rs, linestyle = "--", color = 'white')  #Enable
     when using Wavelet for large timeframe
axs[2].set_ylim(freqs_rs[-1], freqs_rs[0])
axs[2].set_title("Spectrogram - Residuals")
axs[2].set_ylabel("Frequency (1/day)")
axs[2].set_xlabel("Date")
axs[2].set_yticks(np.linspace(freqs_rs[0], freqs_rs[-1], 10))
axs[2].grid(True)

#im_rs = axs[2].pcolor(tod[-end-duration:-end:144], freqs_rs, abs(coef_rs)) #Enable when
    using Wavelet for small timeframe
im_rs = axs[2].pcolor(tod[-end-duration:-end:144], freqs_rs, abs(coef_rs)) #Enable when using
     Wavelet for large timeframe
fig.colorbar(im_rs, ax=axs[2], orientation = 'horizontal', label = 'power [-]')


plt.show()
#%%
FGii = FGi[:-1]

nw_resid = residmean[((DDVECi[:-1] - 43) <= 275) & ((DDVECi[:-1] - 43) >= 265)] #On shore
nw_wind = FGii[((DDVECi[:-1] - 43) <= 275) & ((DDVECi[:-1] - 43) >= 265)]       #On shore

se_resid = residmean[((DDVECi[:-1] - 43) <= 95) & ((DDVECi[:-1] - 43) >= 85)]   #Off shore
se_wind = FGii[((DDVECi[:-1] - 43) <= 95) & ((DDVECi[:-1] - 43) >= 85)]         #Off shore

sw_resid = residmean[((DDVECi[:-1] - 43) <= 185) & ((DDVECi[:-1] - 43) >= 175)] #Along shore
    south-west
sw_wind = FGii[((DDVECi[:-1] - 43) <= 185) & ((DDVECi[:-1] - 43) >= 175)]       #Along shore
    south-west

ne_resid = residmean[((DDVECi[:-1] - 43) <= 5) & ((DDVECi[:-1] - 43) >= -5)]    #Along shore
    north-east
ne_wind = FGii[((DDVECi[:-1] - 43) <= 5) & ((DDVECi[:-1] - 43) >= -5)]          #Along shore
    north-east


fig, axs = plt.subplots(2, 2)
fig.subplots_adjust(hspace=0.5)

fig.suptitle('Correlation between wind speed and residuals from 1990 - 2016')

covariance_nw = np.corrcoef(nw_resid,nw_wind) #Correlation between north-western wind and
    residuals
covariance_se = np.corrcoef(se_resid, se_wind) #Correlation between south-eastern wind and
    residuals
covariance_sw = np.corrcoef(sw_resid, sw_wind) #Correlation between south-western wind and
    residuals
covariance_ne = np.corrcoef(ne_resid,ne_wind) #Correlation between north-eastern wind and
    residuals

axs[0,0].plot(nw_resid, nw_wind, 'o', color = 'black', markersize = 1)
axs[0,0].set_ylabel("Daily mean wind speed (in 0.1 m/s)")
axs[0,0].set_xlabel("Daily mean residual height in meters")
axs[0,0].set_title(f"On shore wind - Pearson correlation coefficient r  {covariance_nw
    [0,1]:.2f} \n number of events: {len(nw_resid)}")


axs[0,1].plot(se_resid, se_wind, 'o', color = 'black', markersize = 1)
axs[0,1].set_ylabel("Daily mean wind speed (in 0.1 m/s)")
axs[0,1].set_xlabel("Daily mean residual height in meters")
axs[0,1].set_title(f"Off shore wind - Pearson correlation coefficient r  {covariance_se
    [0,1]:.2f} \n number of events: {len(se_resid)}")
```

```
axs[1,0].plot(sw_resid, sw_wind,'o', color = 'black', markersize = 1)
axs[1,0].set_ylabel("Daily mean wind speed (in 0.1 m/s)")
axs[1,0].set_xlabel("Daily mean residual height in meters")
axs[1,0].set_title(f"Along shore: South-Western wind - Pearson correlation coefficient r  {
    covariance_sw[0,1]:.2f} \n number of events: {len(sw_resid)}")

axs[1,1].plot(ne_resid, ne_wind,'o', color = 'black', markersize = 1)
axs[1,1].set_ylabel("Daily mean wind speed (in 0.1 m/s)")
axs[1,1].set_xlabel("Daily mean residual height in meters")
axs[1,1].set_title(f"Along shore:North-Eastern wind - Pearson correlation coefficient r  {
    covariance_ne[0,1]:.2f}\n number of events: {len(ne_resid)}")

plt.show()

print(covariance_nw[0][1])
print(covariance_se[0][1])
print(covariance_sw[0][1])
print(covariance_ne[0][1])

print(len(nw_resid))
print(len(se_resid))
print(len(sw_resid))
print(len(ne_resid))
```

B

Tables and Figures

Filtered Tidal Constituents	Frequency per day	Filtered Tidal Constituents	Frequency per day
Z0	0.000000	SIG1	0.861809
SA	0.002738	Q1	0.893244
SSA	0.005476	RHO1	0.898101
MSM	0.031435	O1	0.929536
MM	0.036292	TAU1	0.935012
MSF	0.067726	BET1	0.960970
MF	0.073202	NO1	0.966446
ALP1	0.825518	CHI1	0.971303
2Q1	0.856952	PI1	0.994524
		P1	0.997262

Table B.1: Filtered out low-frequency signals due to tide(periods larger than 24 hours)

Extracted low-frequency residuals: 1990 - 2016

Figure B.1: The yellow signal is the initial signal without filtering. The green signal shows the signal consisting of the tides and irrelevant high-frequency signals. The purple signal is the extracted low-frequency residuals one is interested in.

Spectrograms(per 1 year)

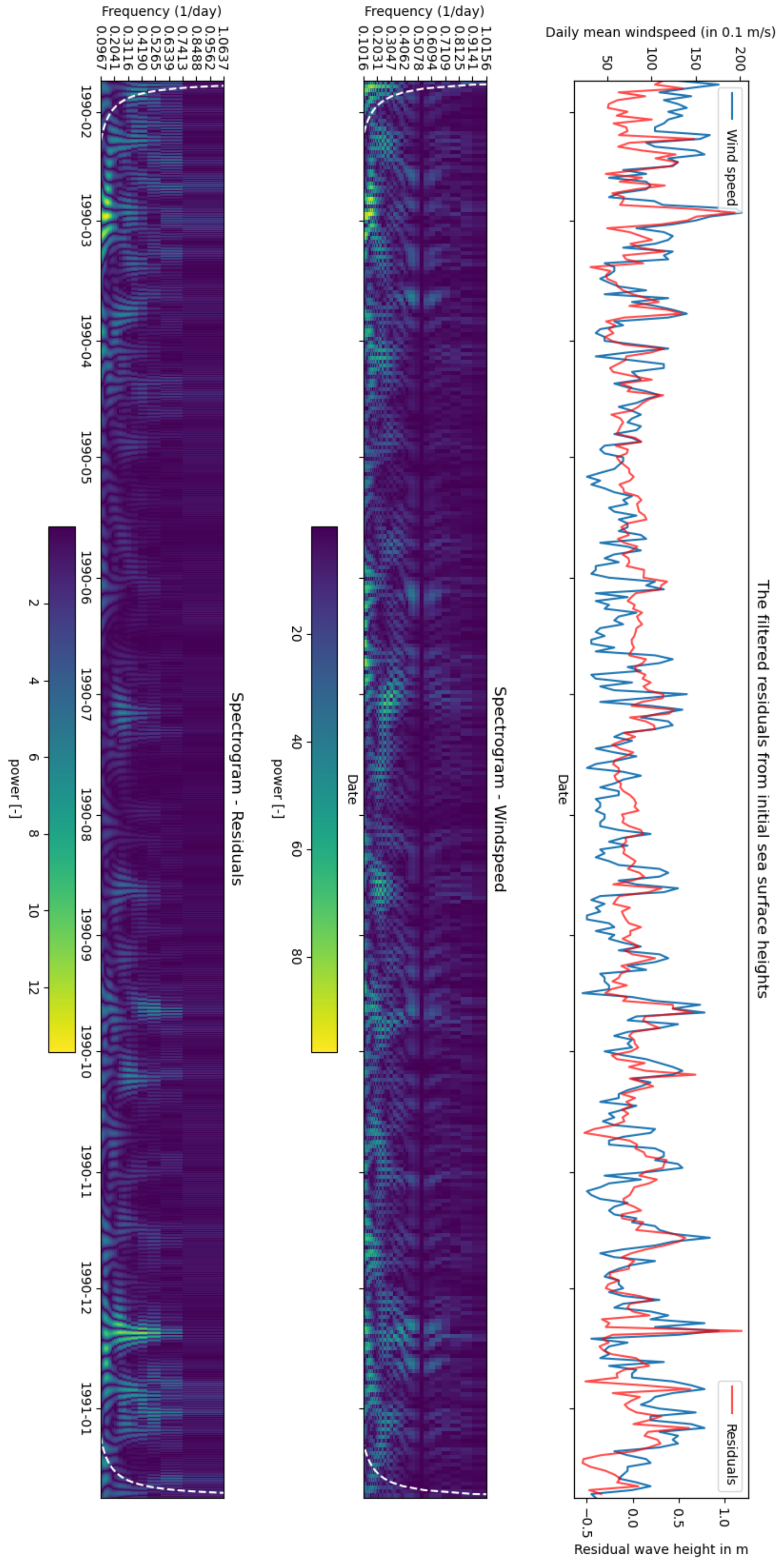

Figure B.2: Wavelet Transformation for sea level at Hoek van Holland: 1990-1991

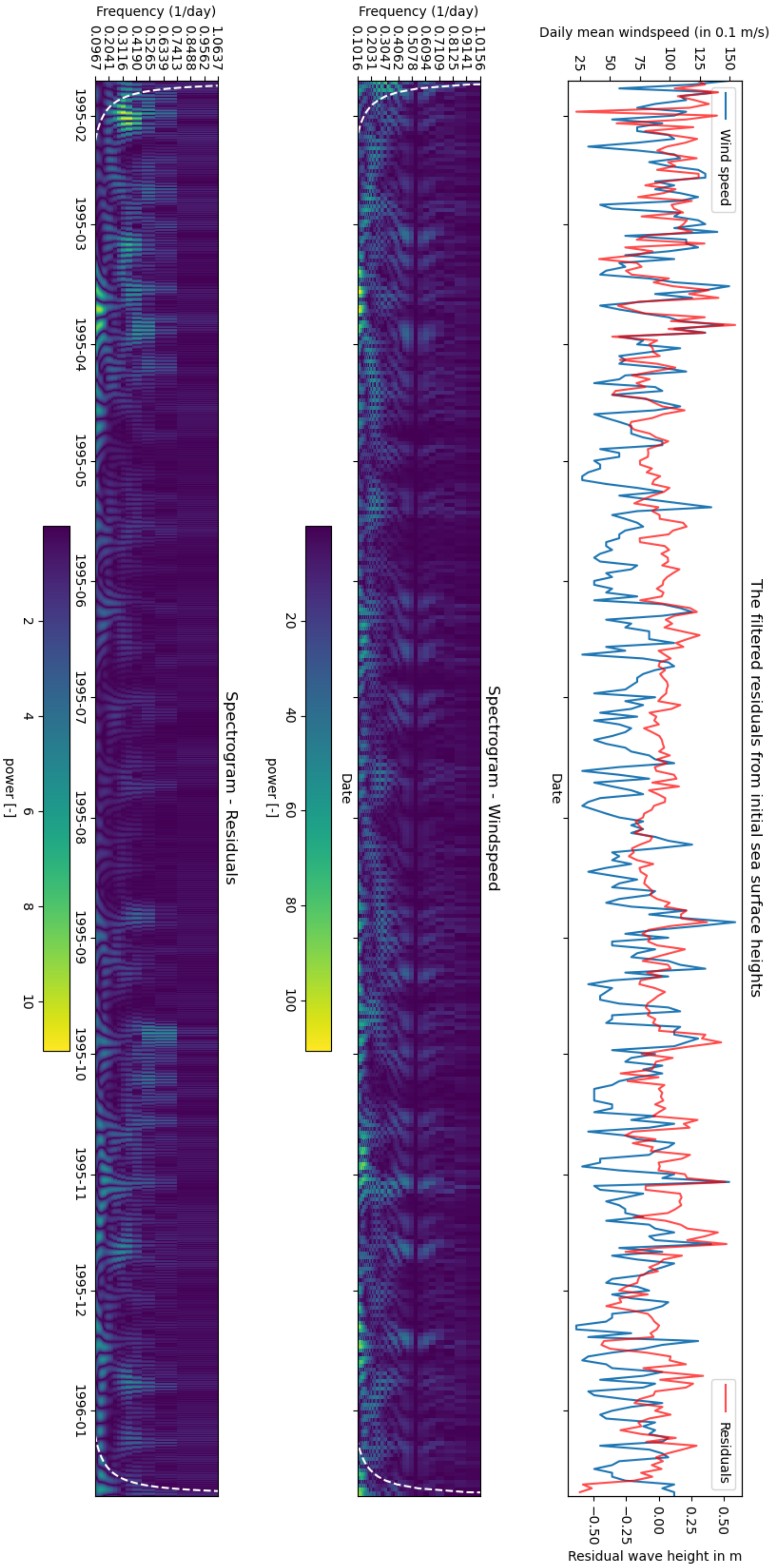

Figure B.3: Wavelet Transformation for sea level at Hoek van Holland: 1995-1996

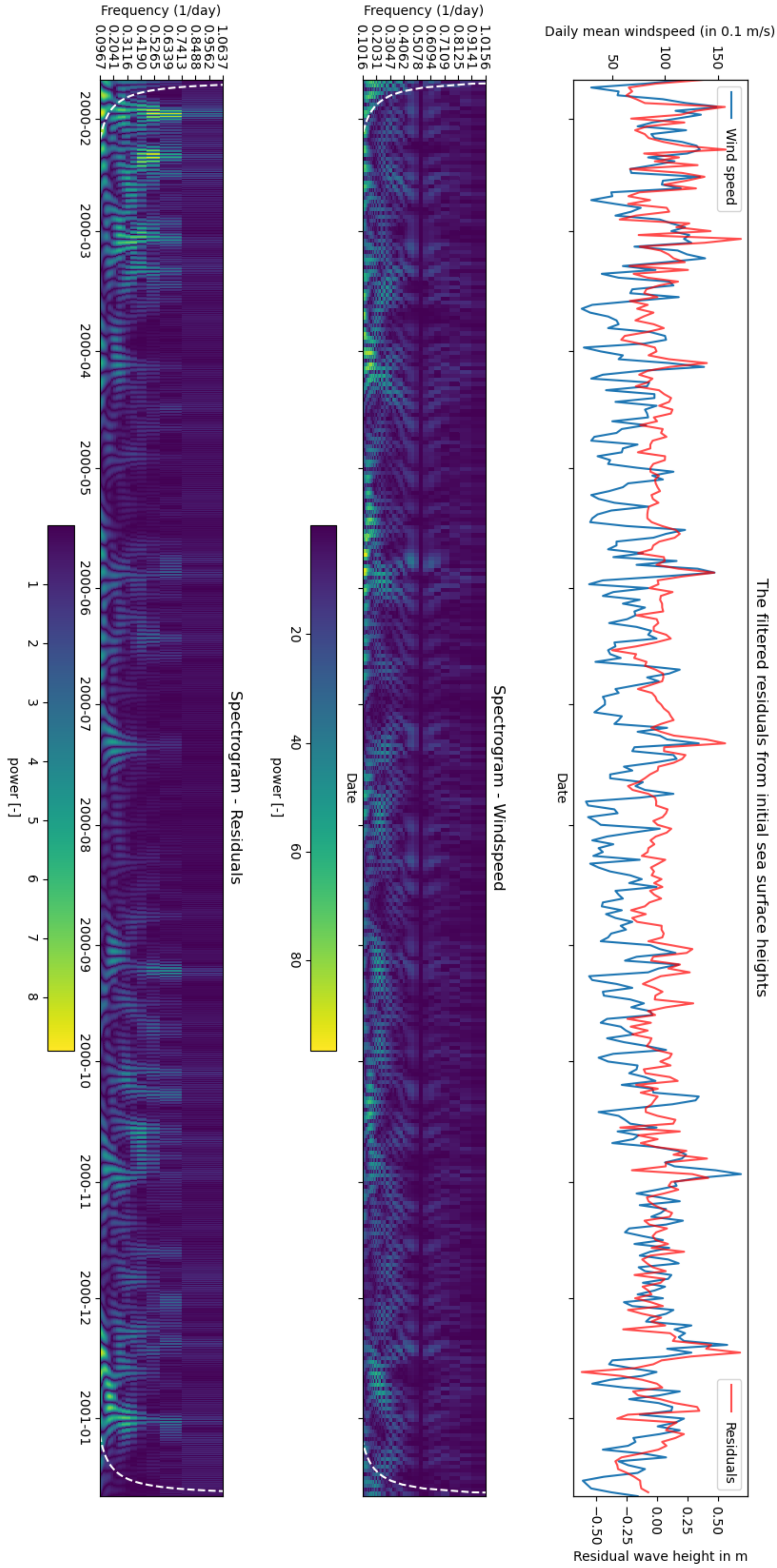

Figure B.4: Wavelet Transformation for sea level at Hoek van Holland: 2000-2001

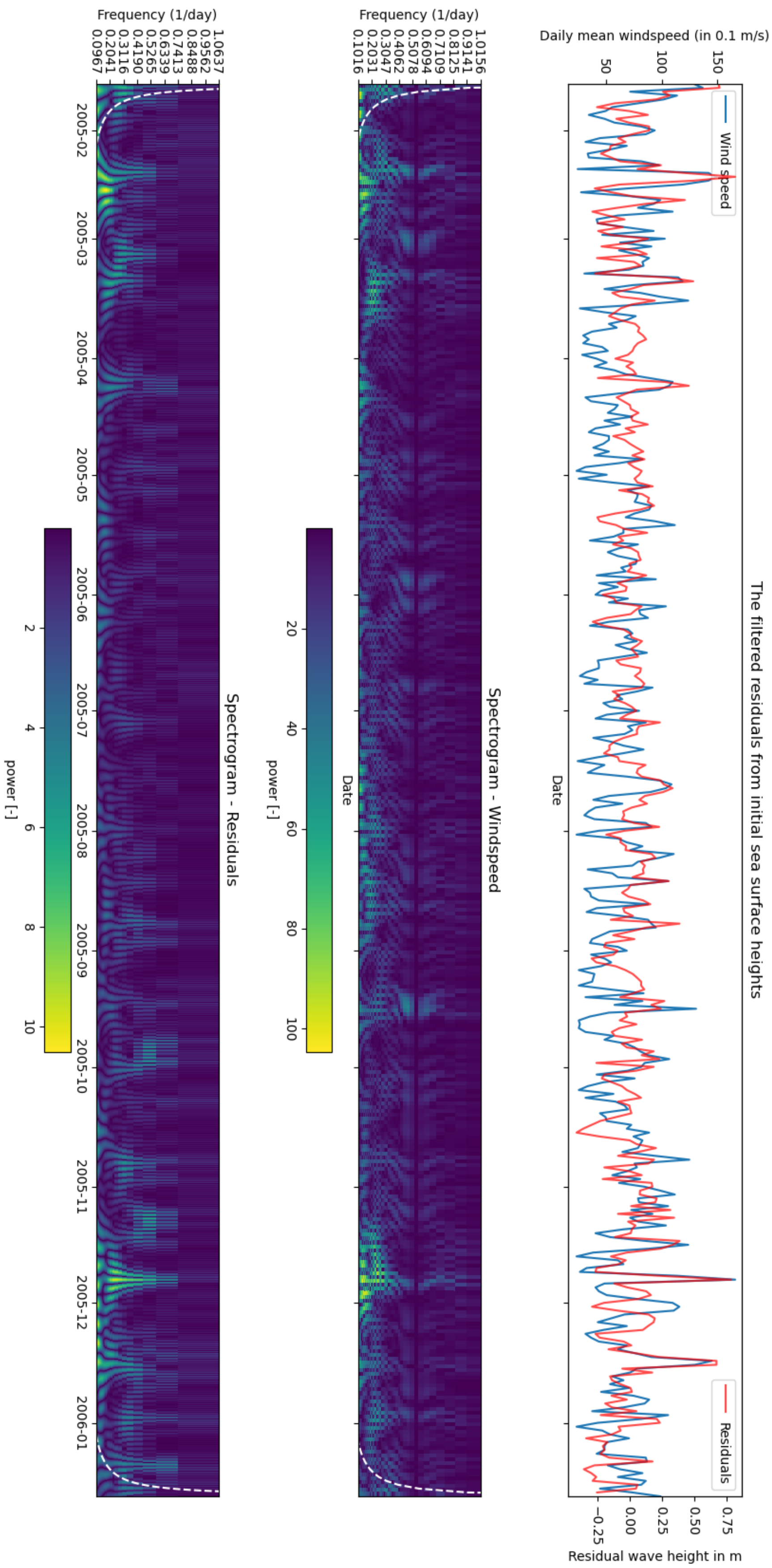

Figure B.5: Wavelet Transformation for sea level at Hoek van Holland: 2005-2006

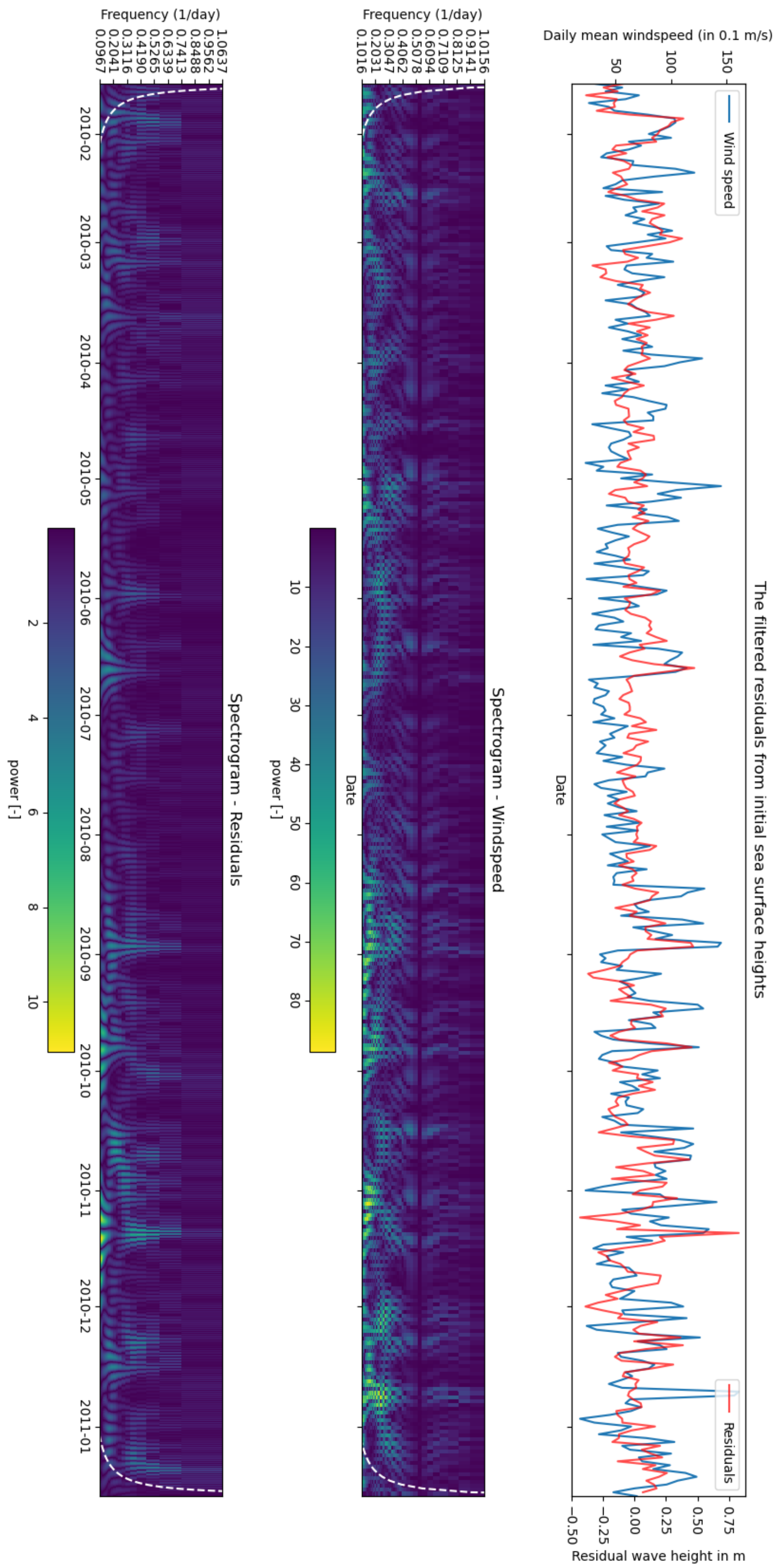

Figure B.6: Wavelet Transformation for sea level at Hoek van Holland: 2010-2011

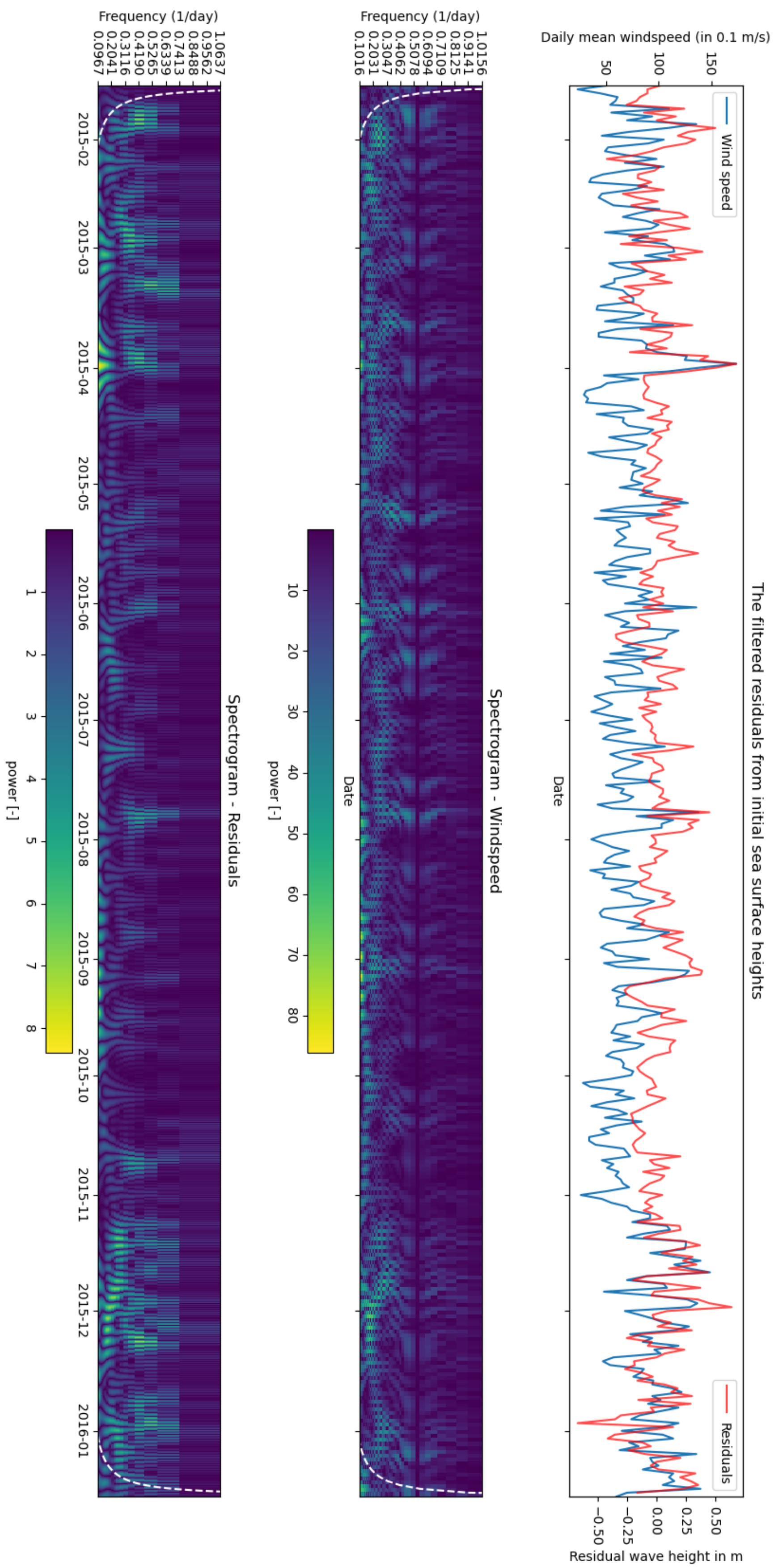

Figure B.7: Wavelet Transformation for sea level at Hoek van Holland: 2015-2016

Spectrograms 1990-2016

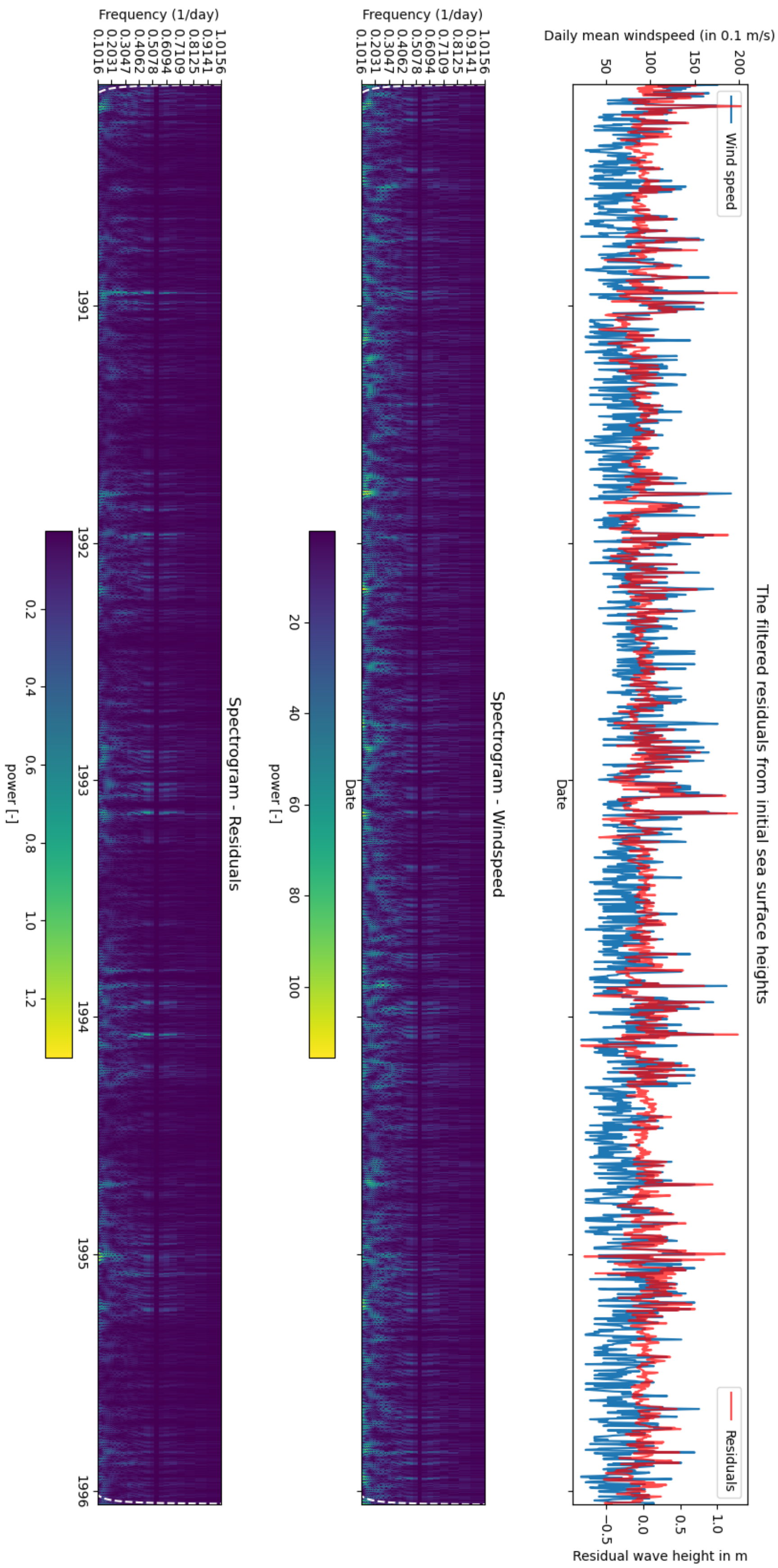

Figure B.8: Wavelet Transformation for sea level at Hoek van Holland: 1990-1996

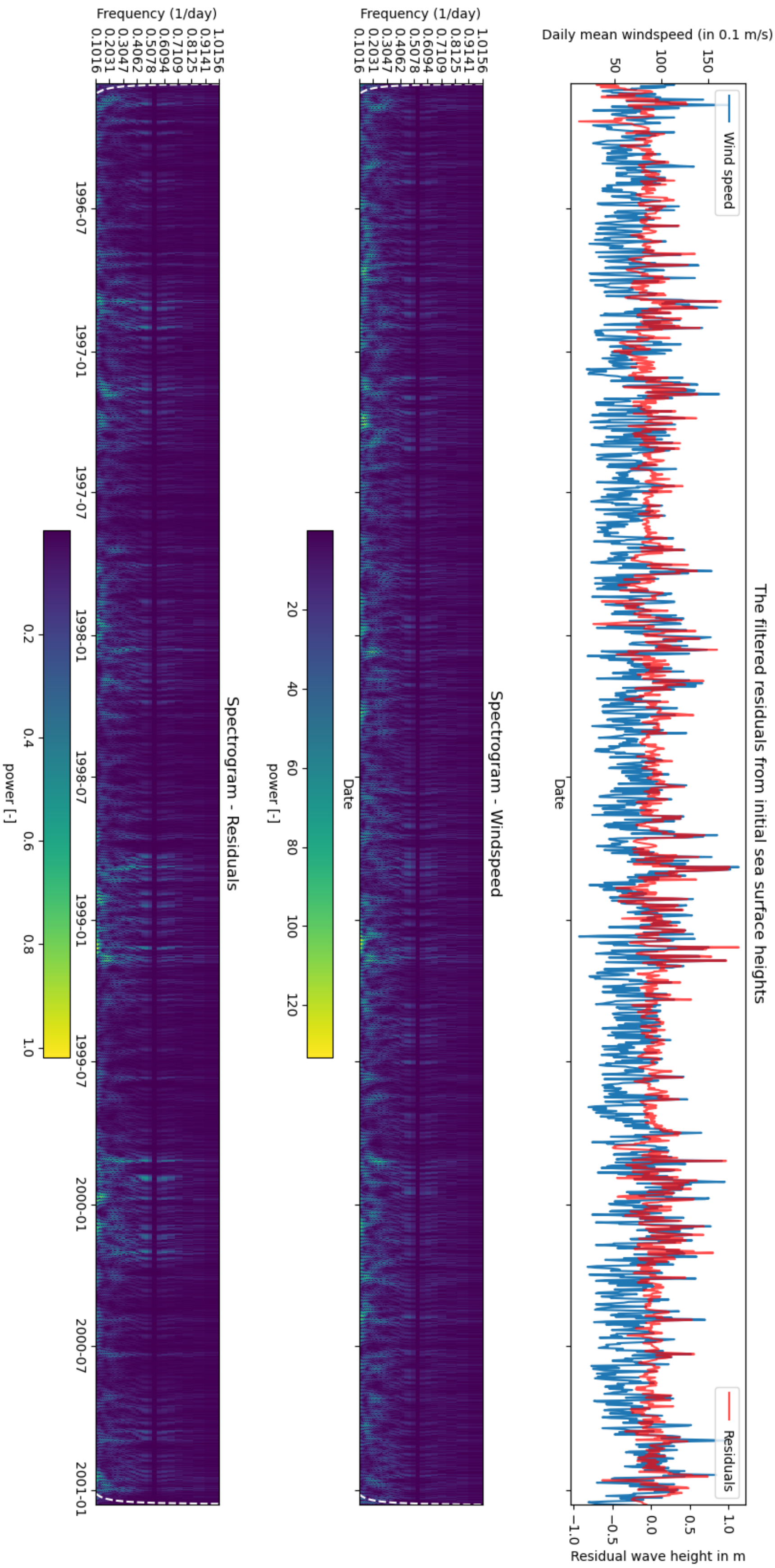

Figure B.9: Wavelet Transformation for sea level at Hoek van Holland: 1996-2001

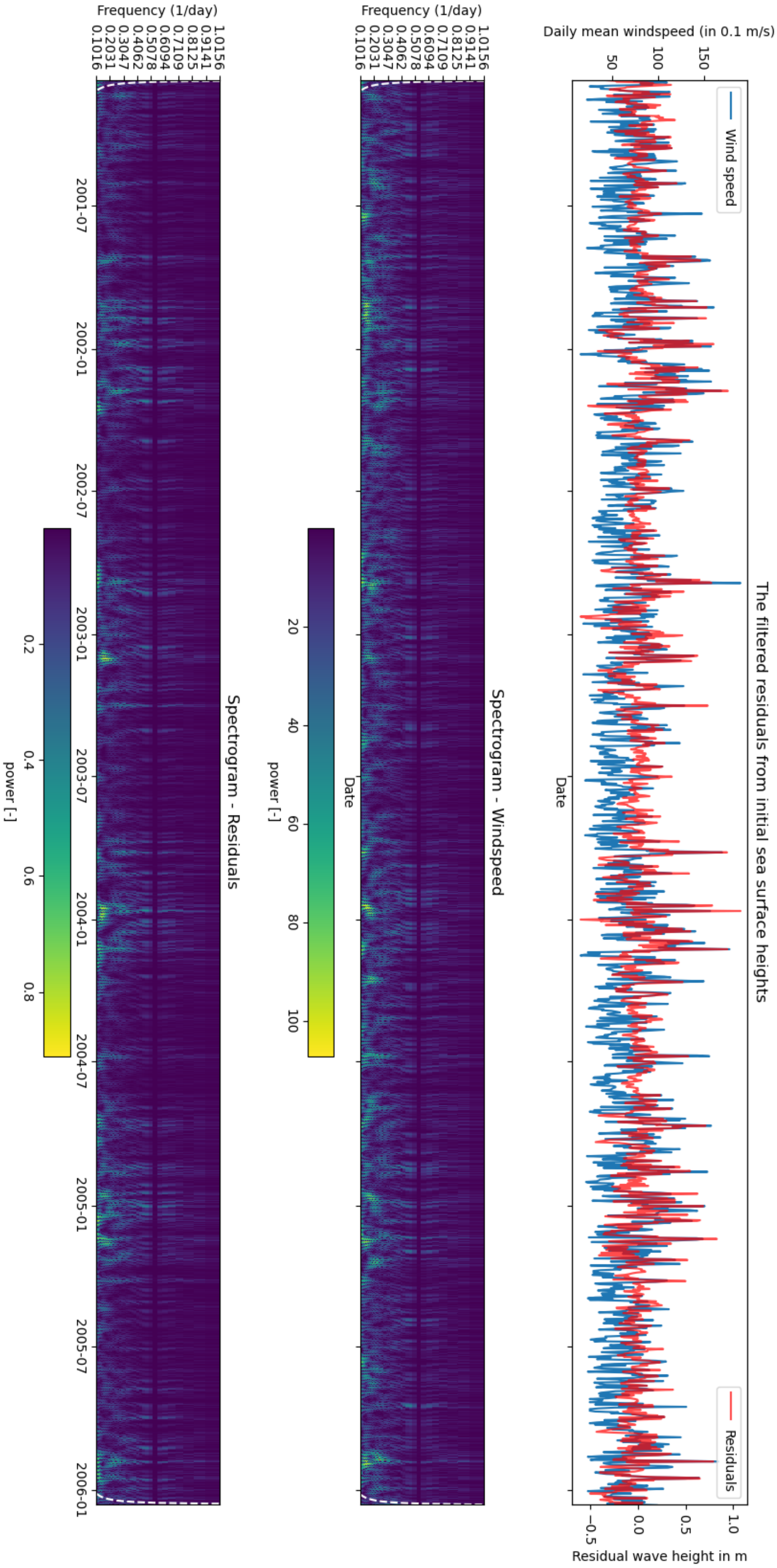

Figure B.10: Wavelet Transformation for sea level at Hoek van Holland: 2001-2006

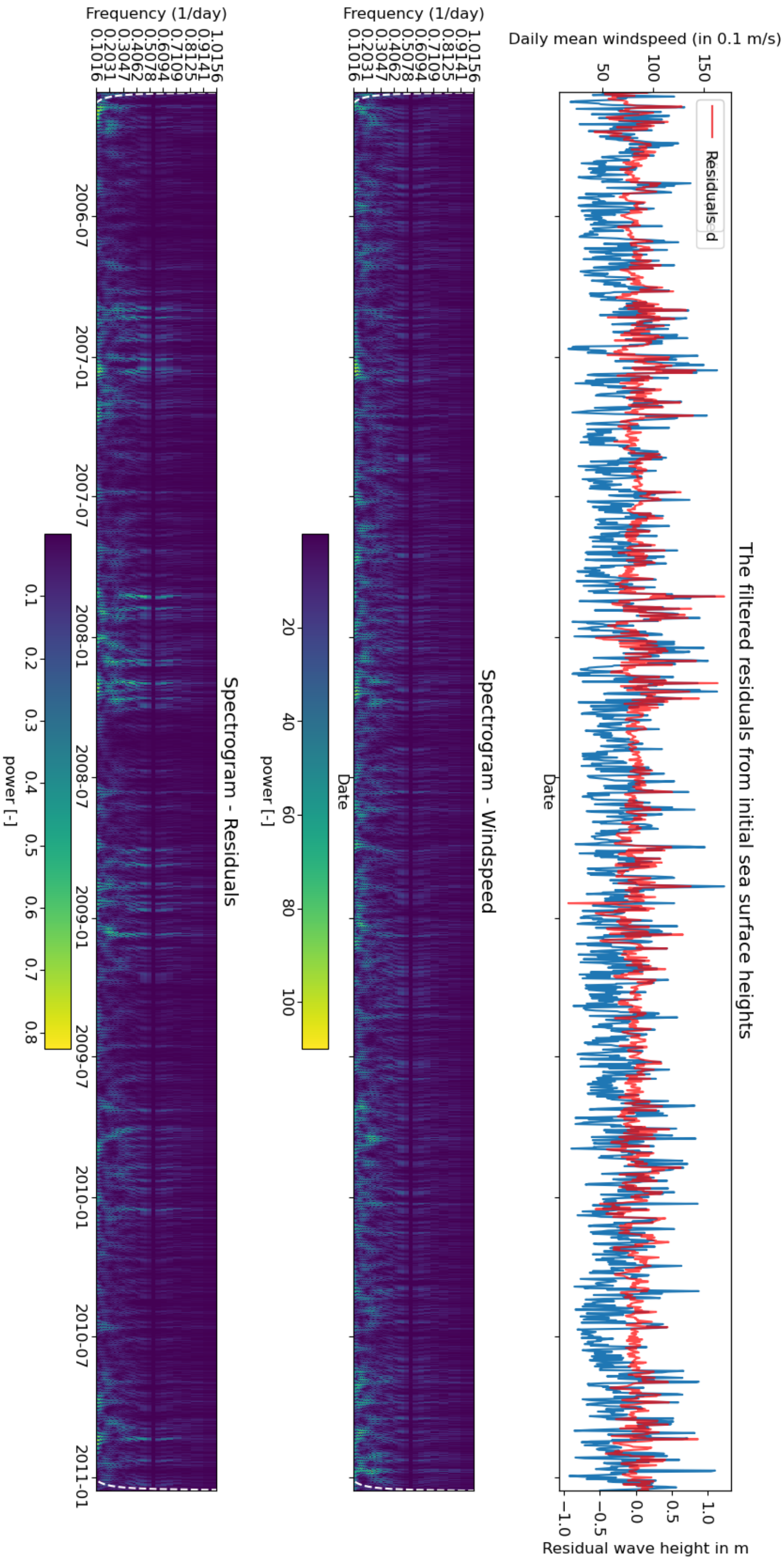

Figure B.11: Wavelet Transformation for sea level at Hoek van Holland: 2006-2011

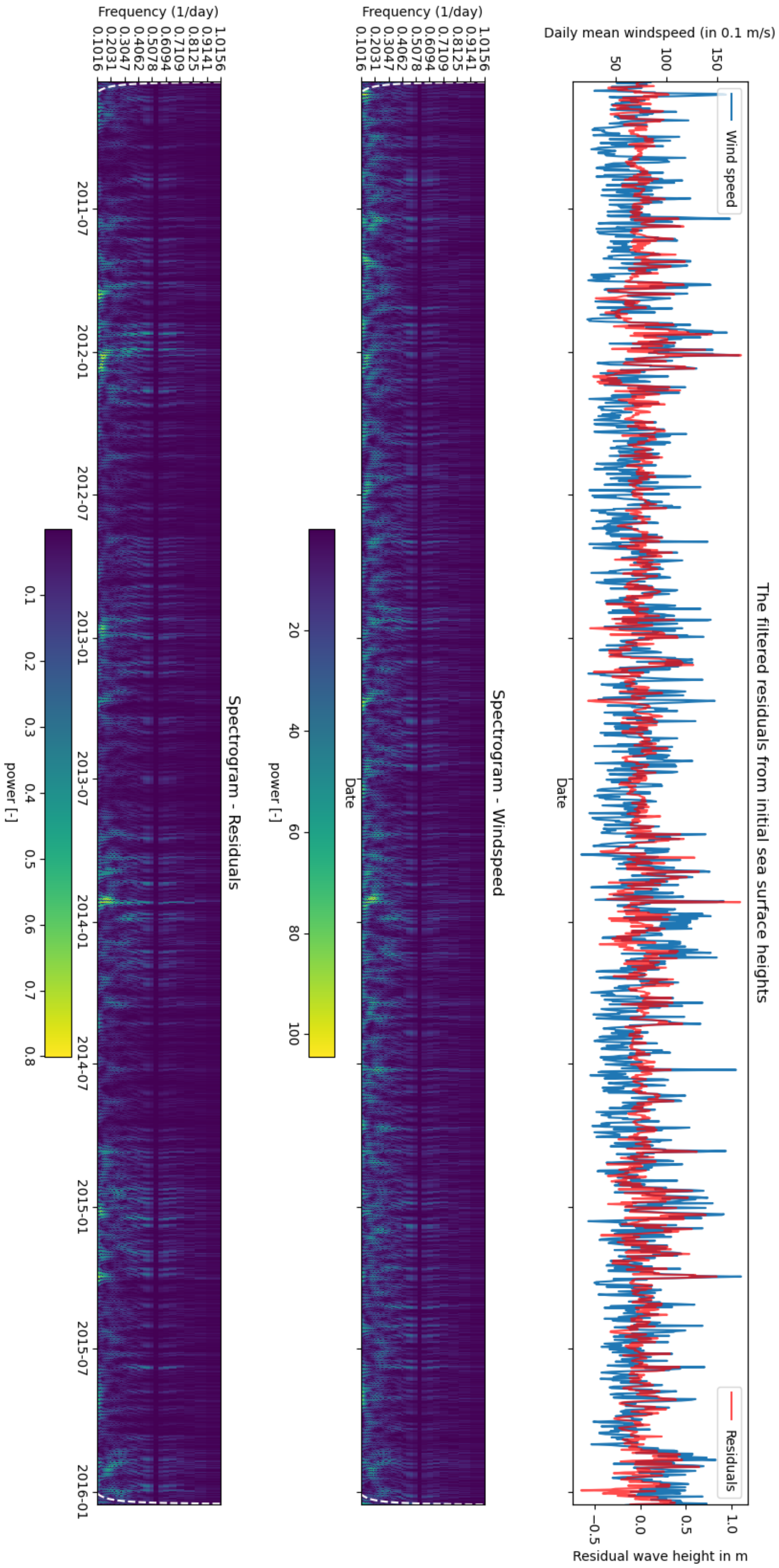

Figure B.12: Wavelet Transformation for sea level at Hoek van Holland: 2011-2016

Location Sea Level Gauge - Hoek van Holland

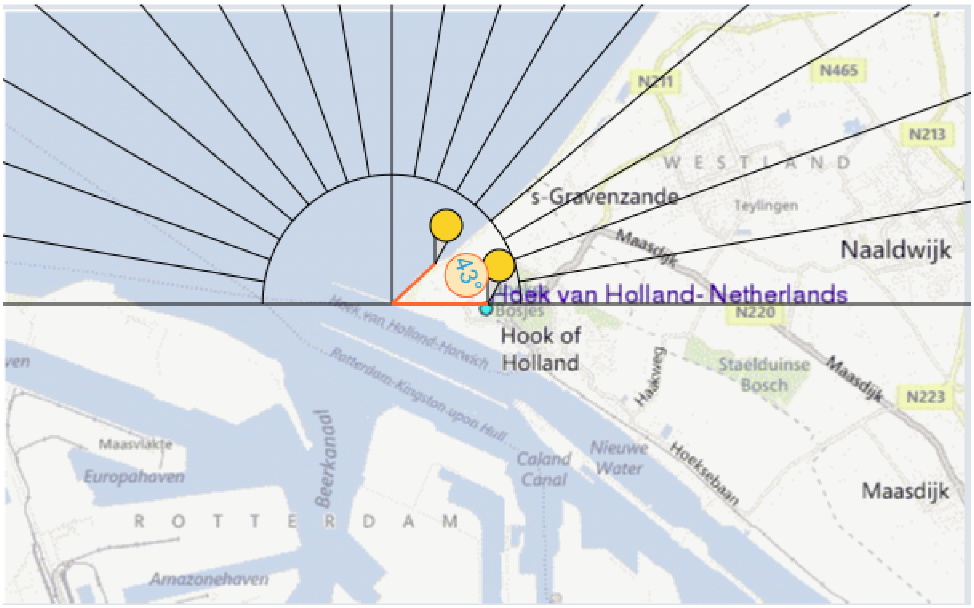

Figure B.13: Location sea level gauge: Hoek van Holland (Adapted from Tide-forecast(n.d.))