MASTER'S THESIS IN STATISTICS

Statistical Analysis of CGPS Time Series

Mohammed Ouassou

August, 2007

FACULTY OF SCIENCE
Department of Mathematics and Statisctics
University of Tromsø

Contents

List of Figures

List of Tables

Abstract

All points on the surface of the Earth are moving. To define the velocity of a given point, we can place a GPS receiver there and measure the coordinates every day. After collecting enough data, we can generate a time series of three coordinates, North, East and Height directions. The most used technique to determine such displacements, is the linear model.

The main objective of this thesis is to show how to estimate the velocity of a given point, using statistical methods to improve the results.

The improvement of the site velocity achieved by exluding all signals that are not tectonic origine (seasonal variations, spacially correlated noise reduction).

Time series for all directions contain gaps(missing data), outliers, offsets and various data length.

The data discontinuities are detected and corrected by a simple algorithm, based on binary search to detect the time of abruption. The outliers are eliminated by using robust estimation techniques. Simulation is used to fill the gaps.

The data obtained from permanent GPS-stations in Norway and some other European countries are unevenly sampled. We therefore use the Lomb-Scargle method to perform spectral analysis. This allows us to detect annual and interannual variations.

The methods of Principal Components (also known as Empirical Orthogonal Functions, or EOF) and Factor Analysis are used to correct for common fluctuations. We use data from 8 permanent GPS-stations (SATREF) in these investigations.

Acknowledgements

This thesis could not have been realized without the support of many people. First and foremost I would like to express my sincere gratitude to my supervisor Professor Fred Godtlibsen for his inspiring guidance and encouragement. I would also like to thank the Institute of Mathematics and Statistics (IMS), TromsøUniversity, special thanks to my teachers, Elinor Ytterstad, Sigrid Sorbye, Georg Elvebakk, and administrative leader Helge Johanssen.

I would like to thank my colleague at NMA, Doctor Oddgeir Kristianssen who formulated the objective of this thesis, for his help and providing me with the necessary data to be analysed.

I am grateful to the Geodetic Institute of Norwegian Mapping and Cadastral Authority of Norway where I work, the head of the Infrastructure and Reference Data Section Rune Ivar Hanssen, and the head of the Geodesy Division Per Erik Oppset for the financial support and authorisation to use SATREF data.

Finally, I would like to express my deepest gratitude to my wife Gunn and my children for being patient all those weekends and evenings when I have been "**married to my books**". Special thanks to my son Jabir Ali who helped me keeping my computer running all the time.

Table 1: *Frequently Used Acronyms*

Name	Description
AIC	Akaike Information Creterion
AICc	Corrected AIC
ACVF	Auto-Covariance Function
ACF	Auto-Correlation Function
AR	Auto-Regressive
ARMA	Auto-Regressive Moving Average
ARIMA	Auto-Regressive Intergared Moving Average
AS	Anti Spoofing
CGPS	Continuous GPS
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
DOP	Dilution Of Precision
DRSO	Detection Removable Simulation Outliers
ECEF	Earth Centered Earth Fixed
EF	Exponential Smoother
FA	Factor Analysis
IGS	International GPS Service
IPP	Ionospheric Piercing Point
IGP	Ionospher Grid Point
JPL	Jet Propulsion Laboratory, Pasadena, California, USA
MDRSO	Modified DRSO
MF	Median Smoother
MA	Moving Average
PC	Principal Component
PFA	Principal Factor Analysis
WMAF	Weighted Moving Average Filter
SMAF	Symetric Moving Average Filter
SA	Selective Availability
SATREF	SATelitebasert REFerensesystem
SIS	SATREF Ionospheric System
MLE	Maximum Likelihood Estimation
TEC	Total Electron Content
TS	Time Series
UTM	Universal Transverse Mercator

Continued on next Page. . .

Table 1 –Continued

Name	**Description**
WAAS	Wide Area Augmentation System
WGS84	World Geodetic System 1984
RMS	Root Mean Square
VLBI	Very Long Base Interfferometry

Frequently Used Definitions	
Name	**Definition**
Accuracy	A measure of how close an estimate of a GPS position is to the true location.
Clock Bias	The difference between the indicated clock time in the GPS receiver and true universal time (or GPS satellite time).
Clock offset	A constant difference in the time reading between two clocks, normally used to indicate a difference between two time zones.
DOP	Dilution of precision, a measure of the satellite/receiver geometry, describing factors which affects the geometric satellite distributions
GPS	A global system based on 24 satellite orbiting the earth at an altitude of 12000 statue miles and providing precise worldwide positioning and navigation information 24 hours a day.
Kalman Filter	Recursive data processing algorithm
Magnetic North	Represents the direction of the magnetic north pole from the observers position. The direction a compass points
Multipath	Interference caused by reflected GPS signal arriving at the receiver.
Navigation Message	The message transmitted by each GPS satellite containing system time, clock correction parameters, ionospheric delay model parameters, and the satellite's ephemeris data and health. The information is used to process GPS signals to give the user time, position, and velocity. Also known as the data message.
Position	An exact unique location based on geographic coordinate system
Residual	Difference between observation and Prediction
Smoothing	Backward processing algorithm
True North	The direction of the north pole from your current position. Magnetic compasses indicate north differently due to the variation between true north and magnetic north. A GPS receiver can display headings referenced to true north or magnetic north.
Troposphere	The lowest region of the atmosphere between the surface of the earth and the tropopause, characterized by decreasing temperature with increasing altitude. GPS signals travel through the troposphere (and other atmospheric layers).
WGS-84	World Geodetic System, 1984 the primary map datum used by GPS. Secondary datums are computed as differences from the WGS 84 standard.

Table 2: *List Of Definitions*

Chapter 1

Introduction and motivation

During the past decade, GPS has grown rapidly and become the most important observational technique used to the study the rotation of the Earth, tectonic plate motions, crustal deformations and seismic activities. The IGS (International GPS Service) GPS network now has more than 330 permanent tracking stations around the world. The network has played a key role in deploying and operating a common and comprehensive global tracking system.

Data for CGPS used during numerical investigation is obtained from NMA, SATREF (SATellittbasert REFeransesystem) and from some IGS stations in Europe. The data is processed by GIPSY software modules from JPL.

The site velocity is determined by fitting a straight line through a series of N points x_i taken at time t_i. The parameters that define the line are its x-intercept x_0 and the slope r so that:

$$x_{t_i} = x_0 + rt_i + \epsilon(t)$$

r determines the rate of change at the site.

But it is not that simple. Our time series X_t contains outliers (O_t), gaps (G_t), jumps(J_t), linear trends (L_t), seasonal components (S_t) and noise (N_t). The main objective is to improve the L_t.

$$X_t = L_t + S_t + N_t + O_t + J_t + G_t$$

This thesis is made up by three major parts:

1. The first part *(ch. 4 - ch. 8)* will handle outliers, jumps, gaps, seasonal components and estimation of the linear trend.

2. The second part *(ch. 9)* will examine the common fluctuations (variations) in our network.

3. The last part *(ch. 10)* will implement a CGPS filter for our time series. This is optional, and not an obligatory part of my thesis.

Let us conclude with an outline of the subsequent chapters:

- **Chapter 2: Exploring the dynamic Earth**
  This chapter introduces the internal structure of the Earth, and how the Earth is affected by the gravitational forces of the Sun and the Moon. The conclusion is that the Earth is constantly changing, both on the inside and on the outside.

- **Chapter 3: GPS Theories**
  This chapter gives an short introduction to the GPS and its error sources. It shows how GPS is used to determine the position of a user, and should introduce the GPS as an important tool for researchers.

- **Chapter 4: Smoothing CGPS time series**
  This chapter introduces the smoothing techniques (parametric and non-parametric) used when smoothing CGPS time series. The methods of Weighted Moving Average, Exponential Smoother, LOWESS and kernel smoothing will be explained.

- **Chapter 5: Outliers Detection in CGPS time series**
  This chapter introduces different algorithms that can be used to remove outliers (O_t) from our time series. As we know, outliers lead to biased site velocity estimation and model misspecification.

- **Chapter 6: Handling apparent discontinuities in CGPS time series**
  This chapter introduces different algorithms to detect and correct offsets in CGPS time series.

- **Chapter 7: Handling Gaps in CGPS time series**
  This chapter will introduce algorithms to fill the gaps (G_t) in our time series. The

gaps can be caused at a site for different reasons, e.g. hardware/software failure or downtime.

- **Chapter 8: Frequency Analysis of CGPS time series**
  This chapter will handle the component (S_t) in our model. It is used to detect seasonal variations by analysing the Fourier frequencies. The Lomb periodogram will be introduced and implemented to handle unevenly sampled data.

- **Chapter 9: Correction for common mode error in CGPS TS**
  This chapter investigates the common variation in our network. We know that the signals from satellites are received by all sites and are common, but non-tectonic signals are present (and common at some sites). Different algorithms will be introduced and implemented to identify this common signal.

- **Chapter 10: CGPS Filter Implementation**
  This chapter will introduce the building blocks needed to implement a CGPS filter and estimation of parameters. An introduction to Power-Law processes will be given.

- **Chapter 11: Summary and Conclusion**
  This chapter gives a summary of all the operations we have done to give a better estimate of L_t.

Chapter 2

Exploring the Dynamic Earth

2.1 Earth structure

Since intensive studies of seismic waves was carried out in early 20th century, we today know that the interior of the Earth has a radially layered structure, like that of an onion (see fig. 2.1). Each layer is characterized by a specific set of physical properties determined by the composition, pressure and temperature in the layer. The four main layers are the crust, mantle, and the outer and inner core.

According to the theory of plate tectonics, the crust of the Earth is made up by moving plates. The theory explains the reasons behind volcanoes, earthquakes and mountains. The most widely accepted version of the theory is the one based on convection in the mantle. Near the core, magma is heated by radioactivity, making it less dense and thus lighter. The magma will then rise towards the crust. Near the crust, the magma is cooled, making it denser and heavier again. This causes the magma to sink deep into the mantle again. This repeated heating and cooling of magma causes continuous currents in the mantle, like illustrated below:Since the tectonic plates basically float on top of the mantle, currents in the mantle move the overlying tectonic plates. These plates move in different directions with differing velocity, forming different types of plate boundaries.

2.1.1 Tectonic Plates

The Earth's surface is broken into several large and many small moving plates. A plate may be as wide as 10,000 km (the Pacific Plate)- or as small as a few thousand km (the

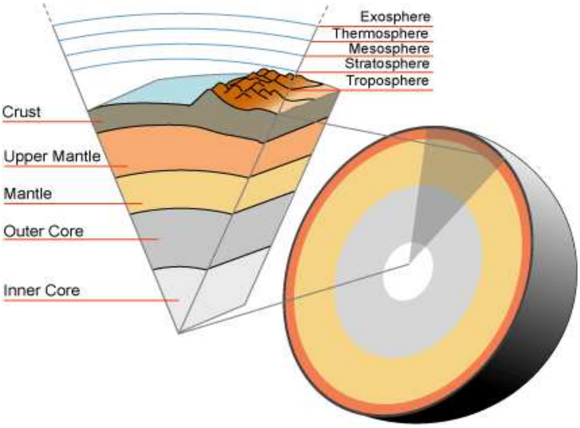


Figure 2.1: *Earth Crust*

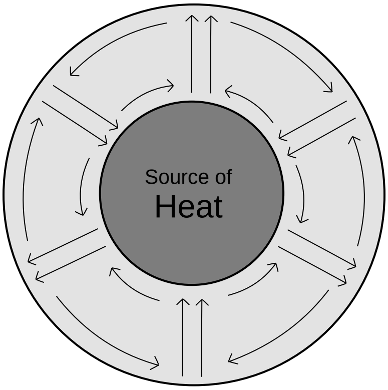


Figure 2.2: *Convection in the Mantle*

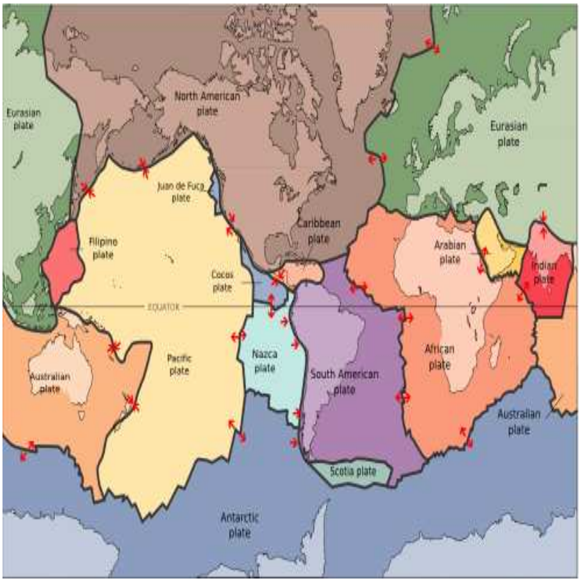


Figure 2.3: *Tectonic Plates*

Philippines Plate). The are twelve major plates(Antarctica, Africa, Eurasia, India, Australia, Arabia, Philippines, North America, South America, Pacific, Nazca and Cocos) and several minor plates (Scotia, Caribbean, Juan de Fuca). These plates move relative to each other on an average of a few centimeters a year. The *figure 2.3* shows the major and minor lithospheric plates.

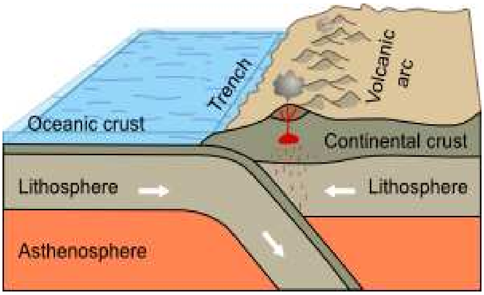


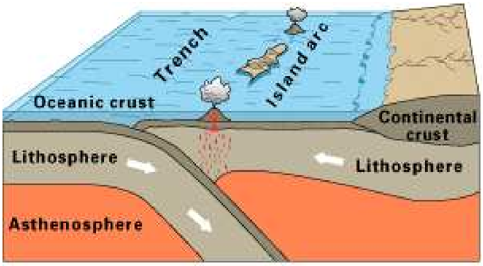
Figure 2.4: *Continental-Oceanic Convergence*

2.1.2 Plate Boundaries

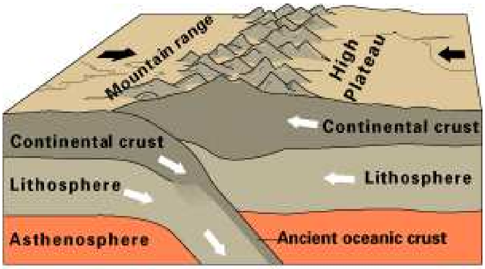
In some places, tectonic plates move apart, forming a "*divergent boundary*". If this process takes place on land, no crust is immediately created, the process is known as "*rifting*", and simply breaks a continent apart. If the process continues, it may eventually form a Mid-Oceanic Ridge. The ridge ("crack") between the moving plates is then filled with magma from the mantle, forming new crust. This process is referred to as "seafloor spreading".

Two plates may also move in opposite directions beside each other. These are referred to as "*transform-fault plate boundaries*", and often causes earthquakes because of the friction between the plates but crust is neither created nor consumed.
If two tectonic plates move toward each other, we find a "*convergent plate boundary*". These plate boundaries experience many earthquakes, but otherwise the scenario depends on what kind of tectonic plates we are dealing with:

1. When a continental plate and an oceanic plate collide, the oceanic plate is subducted under the continental one and melted. A trench is formed between the plates, while a volcanic mountain range is created on the continental side of the boundary.



Oceanic-oceanic convergence



Continental-continental convergence

2. When two oceanic plates collide, one of the plates is usually subducted under the other. The process is basically the same as when a continental plate and an oceanic plate collides, but since it takes place under water, the volcanic mountain range becomes an island arc the islands being tall mountains rising from the seafloor .

3. If two continental plates collide, the plates will crush and buckle each other, thus thickening the plates near the boundary and creating a mountain range. These kind of plate boundaries are also called "collisional plate boundaries". A good example of one would be the Himalayas, where the Indian Plate collides with the Eurasian plate.

The last type of boundary is simply called "*plate-boundary zones*". These are large areas where the involvement of so called microplates between the major plates complicate the situation, thus making the simple terms mentioned above inappropriate.

2.1.3 Hotspot

But volcanism is not always connected to the plate boundaries. Narrow streams of hot magma from as deep as the mantle-core boundary exist at fixed places in the mantle, making those spots especially hot. The tectonic plates are melted when they move over

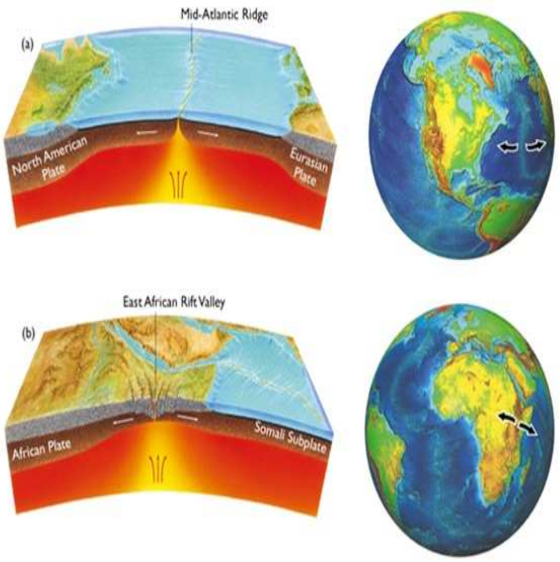


Figure 2.5: *Divergent Plate Boundaries*

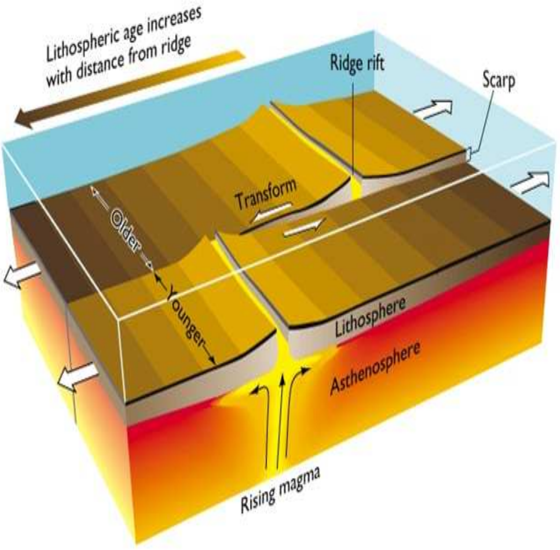


Figure 2.6: *Transform Plate Boundaries*

such “hot spots”, resulting in volcanic activity. A good example of this is the Hawaiian islands. A hot spot resulted in the creation of a volcanic island, but each time this island moved out of reach (because of the movements of the Pacific Plate), a new one would be created.

2.2 The tides

The gravitational forces of the Sun and the Moon deform the shape of the Earth, causing tides in the oceans, atmosphere and solid body of the Earth. The most visible tidal effects is the deplacement of the ocean surface.

The Earth and the Moon are coupled together by gravitational attraction. Their common motion is like that of a pair of ballroom dancers. Each partner moves around the center of the mass of the pair. Determination of the center of the mass is easily defined by the principle of “ *revolution without rotation*”.

The gravitation of the Sun keeps the Earth in it’s orbit, while the Moon stabilizes the rotation of the Earth. But these are not their only tasks. While their gravitation is too weak to for example pull rocks and houses out in space, they do affect matter on Earth. Most known is the phenomenon known as tides. The Moon and the Sun attract the water on Earth, making it move towards the places on Earth that faces them – but the gravitation of the Earth itself is strong enough to keep the water from leaving the planet. During a solar eclipse, the Sun and the Moon stand in a line, causing a double effect. This does not only mean bigger tidal waves, but the gravitational pull can in these cases be strong enough to cause destructive earthquakes and volcanic eruptions.

The last theory is regarded as controversial, but scientists supporting this theory have been able to successfully predict several disasters just by comparing the placement of the Sun and the Moon with the state of the tectonic plates.

2.3 Earth Rotation and Euler Poles

The rotation of the Earth is a vector. This means that it is a quantity characterized by both magnitude and direction. The Earth behaves as an elastic body and deforms in response to the forces generated by its rotation, becoming slightly flattened at the poles with a compensating bulge at the equator.

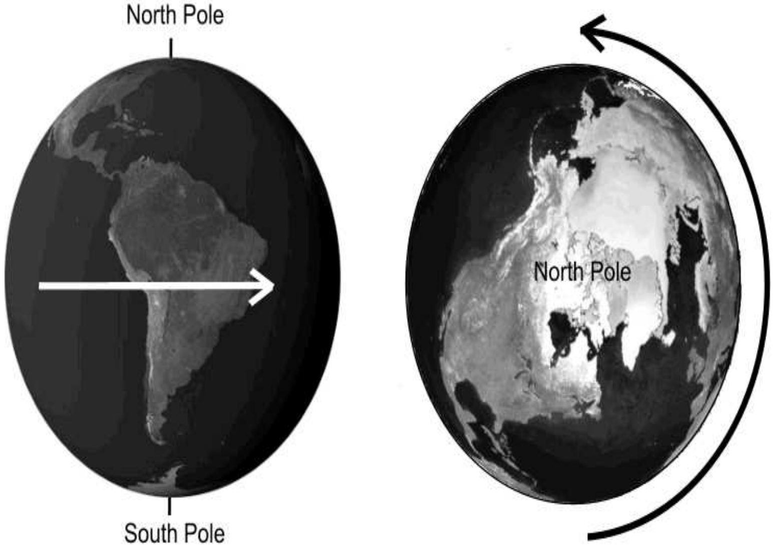


Figure 2.7: *Earth Rotation*

When something moves on a sphere, constantly moving in a fixed direction also means rotating around a fixed point. To illustrate this, let's take a look at the rotation of the Earth (fig. 2.7).

As you can see, everything on the Earth moves in the same direction eastwards along the Equator. But this also means that everything is rotating around the North Pole (or the South Pole). The Equator is way broader than the Arctic Circle, yet a point on the Arctic Circle uses the same time to rotate once as a point on the Equator (24 hours). Because of this, we must also conclude that the area around the Equator moves faster than the area near the poles.

This concept is also applied for tectonic plates. The tectonic plates are assigned "*Euler Poles*" that they rotate around, and these poles tend to remain fixed for long periods of time (indicating that the tectonic plates hold a relatively steady course). Euler poles have different "rotation rates" that can be used to calculate the velocity of a given point on a tectonic plate if we know the distance from the Euler pole. As with the rotation of the Earth, the tectonic plates move fastest about 90 degrees from the Euler pole.

Chapter 3

GPS Theories

3.1 GPS Overview

NAVigation Satellite Timing And Ranging(NAVSTAR) GPS is a satellite navigation system capable of providing accurate, continuous global positioning and navigation services.

The system consists of 24 operational satellites in space, approximately uniformly dispersed around six circular orbits with four or more satellites each. The orbits are inclined at an angle of 55^o relative to the equator and separated from each other by multiple of 60^o.Theoretically, three or more GPS satellites will always be visible anywhere on the surface of the Earth.

The heart of the GPS satellite is the precise atomic clocks (2 rubidium and 2 cesium) that are used to generate two coherent carriers L1 and L2 in the L-band. The GPS satellites transmit on two L-band frequencies with one at 1575.42 MHz(L1) and the other on 1227.60 MHz(L2).These two frequencies are integral multiples $f_1 = 1540 f_0$ and $f_2 = 1200 f_0$, where the fundamental frequency $f_0 = 10.23$MHz

The GPS system is composed of four segments:

- **The Space Segment**:
  Includes the satellite constellation of GPS satellites and is run by the US Air Force (responsible for operation and maintenance). The main Control Centre is at Falcon Air Force Base, Colorado Springs, USA.

- **The Control Segment**:
  Responsible for monitoring the space segment by using signals from channels L1 and L2 to estimate and predict the satellite orbits and clock errors. This information is uploaded to the satellite, which broadcasts the navigation message part by part in frames or subframes to the Earth (users).

- **The User Segment**:
  Different types of GPS receivers and users of various applications are considered the user segment of the GPS. The receiver uses the signals and navigation messages to compute the position, velocity and presice time.

- **The Ground Segment**:
  Includes civilian tracking networks that provide the user segment with reference control, precise ephemeris, and real-time services(DGPS), which mitigate the effect of s̈elective Availability¨.

There are three types of codes in the carrier signals:

- **Coarse/Acquisition(C/A) code**:
  Modulated on the carrier L1. Each satellite has a different C/A code, so that they can be uniquely identified.

- **Precision(P) code**:
  Modulated on the carrier L1 and L2 and is better for more precise positioning.

- **Navigation Message**:
  Can be found on L1 and L2 and includes information on the broadcast Ephemeris, used to determine the satellite orbital parameters, satellite clock corrections, almanac data, ionosphere information, and satellite health information status.

3.2 GPS Observable

The observation types provided by the GPS receivers depend heavily on the technique used. A single frequency receiver can only provide the user with observation on L1 frequency, while dual frequency receivers can provide observations on both frequencies. There are three basic types of observations, namely pseudorange, carrier phase and Doppler measurements.

Suppose that the C/A, P, or Y-code(Encrypted P code) are transmitted by satellite k at time t^k and registered by receiver i at time t_i. The fundamental observation equation is defined by:

$$P_i^k = c(t_i - t^k) = c\tau_i^k \tag{3.1}$$

where

$$\begin{aligned}
P_i^k &: \quad Pseudo-range\ ,expressed\ in\ unit\ of\ length. \\
c &: \quad Speed\ of\ light \\
t^k &: \quad Transmission\ time\ of\ the\ signal \\
t_i &: \quad Observation\ time\ of\ the\ signal. \\
\tau_i^k &: \quad The\ signal\ travelling\ time.
\end{aligned}$$

3.2.1 Pseudorange

The pseudo-range measurement P_i^k is the time difference between the receiver signal and the receiver-generated signal, and is related to the geometric range ρ, distance between the receiver i at time $t_i - \Delta t_i$ and the satellite k at time $t^k - \Delta t^k$, and to the delays due to the Earths atmosphere. The observation equation of pseudo-range can be expressed as:

$$P_i^k = \rho + c.(\Delta t_i - \Delta t^k) + \Delta\rho_{i,trop}^k + \Delta\rho_{i,ion}^k + \Delta\rho_{i,mult}^k + \epsilon_P$$

Where :

$$\begin{aligned}
\rho &: \quad The\ geometric\ range \\
\Delta t^k &: \quad The\ satellite\ clock\ error\ including\ SA\ with\ respect\ to\ GPS\ system\ time \\
\Delta t_i &: \quad The\ receiver\ clock\ error\ with\ respect\ to\ GPS\ system\ time \\
\Delta_{i,ion} &: \quad The\ signal\ delay\ due\ to\ ionosphere \\
\Delta_{i,trop} &: \quad The\ signal\ delay\ due\ to\ troposphere \\
\Delta_{i,mult} &: \quad The\ pseudorange\ multipath\ effect. \\
\epsilon_P &: \quad The\ pseudorange\ measurement\ noise
\end{aligned}$$

3.2.2 Carrier Phase

The carrier phase observation is the measured phase difference between the receiver signal and the receiver-generated signal. It can be expressed as:

$$\Phi_i^k = \rho + (\Delta t_i - \Delta t^k) + \Delta\rho_{i,trop}^k - \Delta\rho_{i,ion}^k + \lambda B_i^k + \epsilon_P$$

Where:

λ : *The wavelength of The GPS signal*

B_i^k : *Denotes constant bias, expressed in cycles, containing the initial carrier phase ambiguity* N_i^k.

3.3 GPS error sources

From the equations mentioned above, Eq , the GPS measurements are subject to many error sources, which degrades the obtainable accuracy of the GPS positions. These are:

3.3.1 Ionosphere error

The ionosphere delay in GPS pseudorange observations is the second largest error source after SA. The ionosphere (an atmospheric layer with the presence of free electrons) can retard radio waves from their velocity in free space by more than 300 ns, in a worst case scenario, corresponding to range error of 100 m.

The ionosphere layer extends from approximately 50 to 1000 km above the surface of the Earth and consists of gases that have been ionized by solar radiation. The ionization process produces clouds of free electrons that act as a dispersive medium for GPS signals (the velocity is a function of frequency).

The important parameter for ionosphere time delay is the total number of electrons (TEC) encountered by the radio waves on its path from satellite to the GPS receiver. All users have to correct this error, and the technique used differ among the GPS receivers.

- **Dual frequency receivers:** For dual frequency receivers(L1 and L2), this error can be eliminated by forming a linear combinations on P or C/A codes.
- **Single frequency receivers:** The ionosphere broadcast algorithms developed by J. A. Klobuchar can be used to reduce the effect of the ionosphere. It can correct up to 50%. The algorithm assumes that all electrons are consentrated in a single layer at 350 km above the Earth surface *fig:3.1*. The algorithm is simple and take into consideration the number of coefficients used and the computation time. All angles are in units of semi-circle, and the time is in seconds.

    1. Compute Earth-centered angle Ψ

    $$\Psi = \frac{0.0137}{E + 0.11} - 0.022 \quad (semicircles)$$

    2. Compute the subionosphere latitude

    $$\Phi_I = \phi_U + \Psi \cos(A) \quad (semicircles)$$

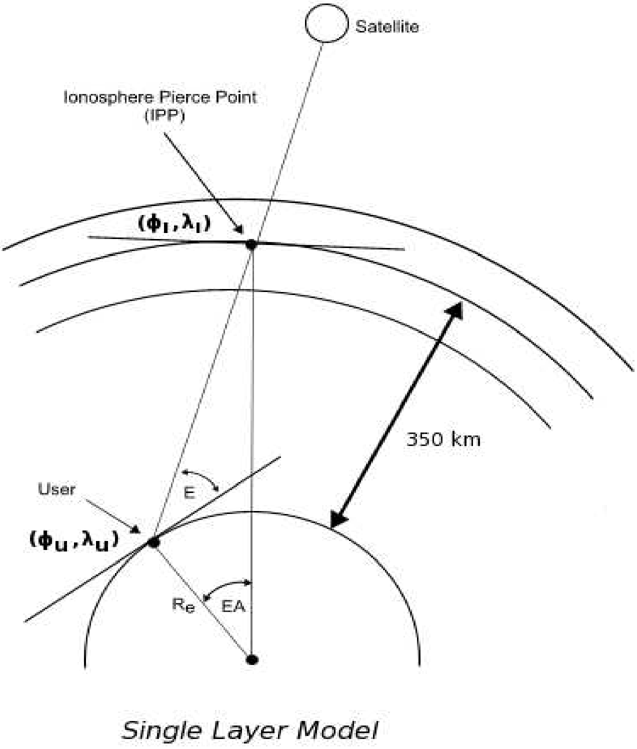


Figure 3.1: *Single Layer Model*

3. Compute the subionosphere longitude

$$\lambda_I = \lambda_U + \frac{\Psi\ \sin(A)}{\cos(\Phi_I)} \quad (semicircles)$$

4. Find the geomagnetic latitude

$$\Phi_m = \Phi_U + 0.0 + 64\ \cos(\lambda_I - 1.617) \quad (semicircles)$$

5. Find the local time:

$$\begin{array}{llll} & t = 4.32\,.\,10^4.\lambda_I + \text{GPS time (sec)} & & \\ \text{if} & t > 86400 & \Rightarrow & t = t - 86400 \\ \text{if} & t < 0 & \Rightarrow & t = t + 86400 \end{array}$$

6. Compute the slant factor:

$$F = 1.0 + 1.6.(0.53 - E)^3$$

7. Compute the ionospheric time delay :

$$\begin{aligned} T_{iono} &= F.\left[5.10^{-9} + \sum_{n=0}^{3} \alpha_n \left(1 - \frac{x^2}{2} + \frac{x^4}{24}\right)\right] \\ \text{where} x &= \frac{2\pi(t - 50400)}{\sum_{n=0}^{3} \beta_n \Phi_m^n} \end{aligned}$$

The alpha and beta coefficients are uploaded by the master station to the satellite and broadcasted from the satellite to the user in the navigation message. In addition, the algorithm uses the elevation angle E of satellite and azimuth angle A. These parameters must be calculated before calling the algorithm to estimate the ionospheric time-delay.

The Klobuchar broadcast algorithm is very popular and is used in more sophisticated ionospheric monitoring algorithms like Ionospheric Delay Inverse Distance Weighting (IDW) implemented in EGNOS and GAGAN systems. IDW uses Klobuchar as prior information about IPP and IGP points.

The ionosphere time delay algorithm is implemented and tested with alpha and beta values at 20 degree satellite elevation and 210 deg azimuth. The algorithm name is klobuchar.R and is placed under the directory libs.

3.3.2 Troposphere error

The lower part of the atmosphere, called the troposphere, is electrically neutral and nondispersive for frequencies as high as about 15 GHz.The raw tropospheric delay in pseudo range units depends on the vertical delay related to the dry and wet components.

About 90 % of the tropospheric refraction arise from the dry component and about 10 % from the wet component(due to temperature, pressure and humidity variations) of the atmosphere.

Several algorithms exist for correction. The most popular is the Empirical Saastamoinen Model and the Hopfield model. If the reader is interested, these algorithms are well described in most GPS textbooks.

3.3.3 Satellite clock error

The satellite clock offset can be eliminated by forming a single difference of two phase (or code) observations .

The GPS time of transmission of the GPS message is $t = t_{sv} - \Delta t_{sv}$, where the the t_{sv} is the SV PRN(Space Vehicle PseudoRandom Noise) code phase time at the time of transmission. The satellite clock correction term Δt_{sv} is approximated by a second order polynomial:

$$\Delta t_{sv} = a_{f0} + a_{f0}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_R$$

Here a_{f0}, a_{f1},a_{f2} are the polynomial correction coefficients corresponding to phase error, frequency error and rate of change of frequency error(in second, second per second and second per square second), and t_{oc}

3.4 User Position Determination

In this section, we will present an algorithm how to calculate the range with no errors, including clock bias. As we know that the receiver record data at regular, specified interval. It is the reading of the receiver clock time T, which used to say exactly when the measurement is sampled. Therefore the value of T is known. If we let the reading of satellite clock be T^s when the signal was transmitted, and c the speed of light in vacuum, then the actual observation to satellite s can be written:

$$\text{distance} = (\text{speed}) \text{ x } (\text{time})$$

$$\rho_r^s = (T - T^s)c$$

We are interesting of defining the pseudorange as a function of the true time t, when the signal was received. This is done by setting:

$$\begin{aligned} T &= t + \tau \\ T^s &= t^s + \tau^s \\ \Rightarrow \rho_r^s &= (t - t^s) + c\tau - c\tau^s \end{aligned}$$

$$\begin{aligned} \rho_r &= Pseudorange(Known) \\ x, y, z &= Satellite\ position(calculated\ from\ ephemeris) \\ X, Y, Z &= User\ position\ coordinates(Unknown) \end{aligned}$$

x,y,y and X,Y, Z are in the earth-centred, earth-fixed(ECEF) coordinates system. From Pythagoras theorem, the position calculation with no errors is:

$$\begin{aligned} \rho_r &= \left[(x - X)^2 + (y - Y)^2 + (z - Z)^2\right]^{\frac{1}{2}} \\ \Rightarrow \rho_r^2 &= (x - X)^2 + (y - Y)^2 + (z - Z)^2 \\ \Rightarrow \rho_r^2 &= (x^2 + y^2 + z^2) + (X^2 + Y^2 + Z^2) - 2xX - 2yY - 2zZ \\ \Rightarrow \rho_r^2 - (x^2 + y^2 + z^2) - R^2 &= \tau - 2xX - 2yY - 2zZ \end{aligned}$$

Where: τ is Clock Bias Correction and R is Radius of the earth.

The problem we still facing, is that the satellite position must be calculated at transmission time, t^s, because the satellite range can change as much as 60 meters from the signal was transmitted to the time the signal was received.An efficient iterative algorithm known as **Light Time Equation** allows us to do the job, by starting with the receive time t

$$\begin{aligned} t^s(0) &= t = (T - \tau) \\ t^s(1) &= t - \frac{\rho^s(t, t^s(0))}{c} \\ t^s(2) &= t - \frac{\rho^s(t, t^s(1))}{c} \end{aligned}$$

$$\vdots$$

$\rho^s(t, t^s)$ is calculated at each step from navigation message, and the algorithm stops when the convergence is reached.

The navigation message allows us to compute the satellite position(x, y, z) and the satellite clock bias τ^s. Therefore we are left with 4 unknowns, the receiver position (X, Y, Z) and the receiver clock bias τ.

For 4 satellites in view of the receiver, we can write the pseudorange to each satellite as:

$$\begin{aligned}
\rho_{r_1}^2 - (x_1^2 + y_1^2 + z_1^2) - R^2 &= \tau - 2Xx_1 - 2Yy_1 - 2Zz_1 \\
\rho_{r_2}^2 - (x_2^2 + y_2^2 + z_2^2) - R^2 &= \tau - 2Xx_2 - 2Yy_2 - 2Zz_2 \\
\rho_{r_3}^2 - (x_3^2 + y_3^2 + z_3^2) - R^2 &= \tau - 2Xx_3 - 2Yy_3 - 2Zz_3 \\
\rho_{r_4}^2 - (x_4^2 + y_4^2 + z_4^2) - R^2 &= \tau - 2Xx_4 - 2Yy_4 - 2Zz_4
\end{aligned}$$

Writing the four eequationsin matrix form, we get:

$$\begin{bmatrix} \rho_{r_1}^2 - (x_1^2 + y_1^2 + z_1^2) - R^2 \\ \rho_{r_2}^2 - (x_2^2 + y_2^2 + z_2^2) - R^2 \\ \rho_{r_3}^2 - (x_3^2 + y_3^2 + z_3^2) - R^2 \\ \rho_{r_4}^2 - (x_4^2 + y_4^2 + z_4^2) - R^2 \end{bmatrix} = \begin{bmatrix} -2x_1 - 2y_1 - 2z_1 & 1 \\ -2x_2 - 2y_2 - 2z_2 & 1 \\ -2x_3 - 2y_3 - 2z_3 & 1 \\ -2x_4 - 2y_4 - 2z_4 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \tau \end{bmatrix}$$

or

$$\begin{aligned}
R &= MV_p \\
\Rightarrow M^{-1}R &= M^{-1}MV_p \\
&= V_p \\
&= \begin{bmatrix} X \\ Y \\ Z \\ \tau \end{bmatrix}
\end{aligned}$$

Pseudorange observation equations are not linear, therefore we apply Taylor methods to linearize our observations. Before doing that , write our observations as the observed model, plus error (noise).

$$\begin{aligned}
P_{obs} &= P_{model} + Noise \\
&= P(X, Y, Z, \tau) + W
\end{aligned}$$

Applying Taylor series eexpansionmethod about (X_0, Y_0, Z_0, τ_0) and ignoring the higher order.

$$
\begin{aligned}
P(X,Y,Z,\tau) &= P(X_0,Y_0,Z_0,\tau_0) + (X - X_0)\frac{\partial P}{\partial X} + (Y - Y_0)\frac{\partial P}{\partial Y} \\
&\quad +(Z - Z_0)\frac{\partial P}{\partial Z} + (\tau - \tau_0)\frac{\partial P}{\partial \tau} \\
&= P_{computed} + \frac{\partial P}{\partial X}\Delta X + \frac{\partial P}{\partial Y}\Delta Y + \frac{\partial P}{\partial Z}\Delta Z + \frac{\partial P}{\partial \tau}\Delta \tau \\
\Rightarrow \Delta P &= P_{observed} - P_{computed} \\
&= \frac{\partial P}{\partial X}\Delta X + \frac{\partial P}{\partial Y}\Delta Y + \frac{\partial P}{\partial Z}\Delta Z + \frac{\partial P}{\partial \tau}\Delta \tau + W \\
&= \begin{bmatrix} \frac{\partial P}{\partial X} & \frac{\partial P}{\partial Y} & \frac{\partial P}{\partial Z} & \frac{\partial P}{\partial \tau} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \Delta \tau \end{bmatrix} + W
\end{aligned}
$$

For each satellite in view, say n, we have:

$$
\begin{bmatrix} \Delta P^1 \\ \Delta P^2 \\ \Delta P^3 \\ . \\ . \\ \Delta P^n \end{bmatrix} = \begin{bmatrix} \frac{\partial P^1}{\partial X} & \frac{\partial P^1}{\partial Y} & \frac{\partial P^1}{\partial Z} & \frac{\partial P^1}{\partial \tau} \\ \frac{\partial P^2}{\partial X} & \frac{\partial P^2}{\partial Y} & \frac{\partial P^2}{\partial Y} & \frac{\partial P^2}{\partial \tau} \\ . & . & . & \\ . & . & . & \\ \frac{\partial P^n}{\partial X} & \frac{\partial P^1}{\partial Y} & \frac{\partial P^n}{\partial Z} & \frac{\partial P^n}{\partial \tau} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \Delta \tau \end{bmatrix} + \begin{bmatrix} W^1 \\ W^2 \\ W^3 \\ . \\ . \\ W^n \end{bmatrix}
$$

We show that the user position can be determined by the Least Square method. The GPS is a very complicated area, but we will in this chapter only try to give a short introduction. This method is described in most GPS textbooks.

3.5 Satellite position calculations

In this section, we are going to present the algorithm to calculate the satellite position (X_s, Y_s, Z_s) from navigation message.

3.5.1 Components of ephemeris

The parameters needed to determine the satellite position are given in the table below.

Name	Description	Unit
M_0	Mean Anomaly at reference time	semicircle
Δn	Mean motion difference from computed value	semicircle/s
e	Eccentricity	dimensionless
$\sqrt{a}$	Square Root of semi-major axis	$m^{\frac{1}{2}}$
ω_0	Longitude of ascending node of orbital plane at weekly epoch	semicircle
i_0	Inclination angle at reference time	semicircle
ω	Argument of perigee	semicircle
$\dot{\Omega}$	Rate of right ascention	semicircle/s
$IDOT$	Rate of inclination angle	semicircle/s
C_{uc}	Amplitude of cosine harmonic correction term to the argument of latitude	rad
C_{us}	Amplitude of sine harmonic correction term to the argument of latitude	rad
C_{rc}	Amplitude of cosine harmonic correction term to the orbit radius	m
C_{rs}	Amplitude of sine harmonic correction term to the orbit radius	m
C_{ic}	Amplitude of cosine harmonic correction term to the angle of inclination	rad
C_{is}	Amplitude of sine harmonic correction term to the angle of inclination	rad
t_{oe}	Ephemeris reference time	s
$IODE$	Issue of data, ephemeris	s

Table 3.1: *Components Of Ephemeris*

3.5.2 Satellite position Algorithm

In the next table, we present all steps and formulas needed to calculate the satellite position (X_s, Y_s, Z_s) from navigation message.

t is in GPS system time at time of transmission, i.e, GPS time corrected for transit time(range/speed of light). Furthermore, t_k shall be the actual total time difference between the time t and the time epoch t_{oe} (time of ephemeris), and we must account for beginning or end of week crossover. That is, if t_k is greater than 302400 s, subtract 604.800 s from t_k. If t_k is less than -302400 s add 604.800 s to t_k.

Eccentric Anomaly E_k shall be estimated iteratively by Performing Newton-Raphson algorithm.

Algorithm can be checked against the result obtained from IGS in sp3 format, that can be downloaded from any IGS site by ftp.

Name	**Description**
$\mu = 3.986005.10^{14}[m^3/s^2]$	WGS84 value of the Earth's universal gravitational parameter
$\dot{\Omega}_e = 7.292115167.10^{-5}[rad/s]$	WGS84 value of the Earth's rotation rate.
$a = (\sqrt{(a)})^2$	Semimajor axis
$n_0 = \sqrt{(\frac{\mu}{a^3})}$	Computed mean motion [rad/s]
$t_k = t - t_{oe}$	Time from ephemeris reference epoch
$n = n_0 + \Delta n$	Corrected mean motion
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - e\sin(E_k)$	Kepler's equation for eccentric anomaly
$f_k = \cos^{-1}\left(\frac{\cos(E_k)-e}{1-e\cos()E_k}\right)$	True anomaly from cosine
$f_k = \sin^{-1}\left(\frac{\sqrt{(}1-e^2)\sin(E_k)-e}{1-e\cos()E_k}\right)$	True anomaly from cosine
$E_k = \cos^{-1}\left(\frac{e+\cos(f_k)}{1+e\cos()f_k}\right)$	Eccentric Anomaly from cosine
$\phi_k = f_k + \omega$	Argument of latitude
$\delta\mu_k = C_{uc} * \cos(2\phi_k) + C_{us}\sin(2\phi_k)$	Second harmonic correction to argument of latitude
$\delta r_k = C_{rc} * \cos(2\phi_k) + C_{rs}\sin(2\phi_k)$	Second harmonic correction to radius
$\delta i_k = C_{ic} * \cos(2\phi_k) + C_{is}\sin(2\phi_k)$	Second harmonic correction to inclination
$\mu_k = \phi_k + \delta\mu_k$	Corrected argument of latitude
$r_k = a\,(1 - e\cos(E_k)) + \delta r_k$	Corrected Radius
$i_k = i_0 + \delta i_k + (IDOT)t_k$	Corrected inclination
$\acute{x}_k = r_k + \cos(\mu)$	X coordinate in orbit plan
$\acute{y}_k = r_k + \sin(\mu)$	Y coordinate in orbit plan
$\Omega_k = \left(\dot{\Omega} - \dot{\Omega}_e\right) t_k - \dot{\Omega}_e t_{oe}$	Corrected longitude of ascending node.
$x_k = \acute{x}_k\cos(\Omega_k) - \acute{y}_k\cos(i_k)\sin(\Omega_k)$	ECEF X coordinate
$y_k = \acute{x}_k\sin(\Omega_k) + \acute{y}_k\cos(i_k)\cos(\Omega_k)$	ECEF Y coordinate
$z_k = \acute{y}_k\sin(i_k)$	ECEF Z coordinate

Table 3.2: *Satellite Position Algorithm*

Chapter 4

Smoothing Time Series

The main goal of this chapter is to introduce the principles of smoothing techniques, and to provide a background of later chapters. Our signal is composed of a long-term trend, seasonal components and a noise signal. It's very important for data analysts to master different techniques of smoothing, as we know application differs and techniques used shall match the nature of problem under examination. We will discuss the moving Average filter, exponential smoother, Locally Weighted Scatter Plot Smoothing(LOWESS) and kernel smoothing

4.1 Moving Average Filter

Smoothing is just a transformation we use to create a new data set from the observed one with reduced noise. If let our observations be presented by x_t by applying the Symmetrical Moving Average Filter, the new observations is given by

$$y_t = \sum_{i=-k}^{k} A_i x_{t-k}$$

Where $A_i = A_{-i} \geq 0$ and $\sum_{i=-k}^{k} A_i = 1$. This filter regards each data point in the window to be equally important when calculating the average (filtered) value.The smoothing parameter h, the size of the span window used under smoothing plays an important role in smoothing, the small value of h allows us to discover the periodic components in our signal and largest value allows us to discover the trend if is present. Choosing window size is an important step in smoothing.

We will focus now on improvement in computational efficiency. If we perform the calculation of the mean in a recursive fashion, we can reduce the computation tasks. Suppose that at any instant j, the average of the latest n samples of the data sequence, X_i is given by

$$\bar{x}_j = \frac{1}{n} \sum_{i=j-n+1}^{j} x_i$$

Similarly, at the previous time instant, $j-1$, the average of the latest n samples is

$$\begin{aligned}
\bar{x}_{j-1} &= \frac{1}{n} \sum_{i=j-n}^{j-1} x_i \\
\Longrightarrow \bar{x}_j - \bar{x}_{j-1} &= \frac{1}{n} \left[\sum_{i=j-n+1}^{j} x_i - \sum_{i=j-n}^{j} x_i \right] \\
&= \frac{1}{n} \left[x_j - x_{j-n} \right] \\
\Longrightarrow \bar{x}_j &= \bar{x}_{j-1} + \frac{1}{n} \left[x_j - x_{j-n} \right]
\end{aligned}$$

It can be seen that we need only to perform 1 division, 1 addition and 1 subtraction operation, regardless of the number of data point n.

4.2 Exponentially Weighted Moving Average Filter

This type of filter places more emphasis on the most recent data points and it's useful in prediction (forecasting). Such filter can be designed by the following procedures. As before, the starting points is the mean value expressed as:

$$\bar{x}_j = \frac{1}{n} \sum_{i=j-n+1}^{j} x_i$$

Similarly, consider the mean with additional point

$$\begin{aligned}
\bar{x}_{j+1} &= \frac{1}{n+1} \sum_{i=j-n+1}^{j+1} x_i \\
&= \frac{1}{n+1} \left[x_{j+1} + \sum_{i=j-n+1}^{j} x_i \right]
\end{aligned}$$

$$\begin{aligned} &= \frac{1}{n+1}\left[x_{j+1} + n * \bar{x_j}\right] \\ &= \frac{1}{n+1}x_{j+1} + \frac{n}{n+1}\bar{x}_j \end{aligned}$$

Shifting the time index back one time-step, we obtain the corresponding expression for x_j as:

$$\bar{x}_j = \frac{1}{n+1}x_j + \frac{n}{n+1}\bar{x}_{j-1}$$

To simplify the notation, let $\alpha = n/(n+1)$, which implies that $(1-\alpha) = 1/(n+1)$. We can write the equation of the filter as:

$$\bar{x}_j = \alpha\bar{x}_{j-1} + (1-\alpha)x_j$$

Calculation of $\bar{x}_j$ requires only 1 addition, 1 subtraction and 2 multiplication operations. The value of α, dictates the degree of filtering, i.e how strong the filtering action will be. Since $n \geq 0$, this means that $0 \leq \alpha \leq 1$. When n is larger, $\alpha \rightarrow 1$, and $\bar{x}_j \rightarrow \bar{x}_{j-1}$. This means that the degree of filtering is so great, the measurement does not play a part in the calculation of the average. If $n \rightarrow 0$, then $\bar{x}_j \rightarrow x_j$, this means that no filtering is being performed.

The Exponentially Weighted Moving Average filter places more importance to more recent data by discounting older data in an exponential manner. This characteristic can be illustrated simply by describing the current average value in terms of past data.

$$\begin{aligned} \bar{x}_j &= \alpha\bar{x}_{j-1} + (1-\alpha)x_j \\ &= \alpha\left[\alpha\bar{x}_{j-2} + (1-\alpha)x_{j-1}\right] + (1-\alpha)x_j \\ &= \alpha^2\bar{x}_{j-2} + \alpha(1-\alpha)x_{j-1} + (1-\alpha)x_j \\ &. \\ &. \\ &= \alpha^k\bar{x}_{j-k} + \alpha^{k-1}(1-\alpha)x_{j-k-1} + .. + (1-\alpha)x_j \end{aligned}$$

If we continue expanding $\bar{x}_j$ backwards, we se that the older values of x_i are weighted by increasing power of α. Since α is less than 1, the contribution of old values get smaller.

4.3 Kernel Density Estimation

Before getting into more details of kernel density estimation, it's worth to distinguish between models:

- **Parametric Models**:
  These models assume that our observations came from known distribution function, with unknown parameters. The main statistical challenge is to estimate the unknown parameters and test if the model chosen estimates better our observations. They lead to an easy interpretation of the resulting fit.

- **Nonparametric Models**:
  These models let data speak for themselves. Therefore they are said to be non-parametric. They open the way for a new models by their flexibility.

- **Semiparametric Models**:
  These models combine parametric and nonparametric parts.

Suppose now that our observations $X = (X_1, X_2, X_3, ..., X_n)$ are from an unknown density function $f(x)$ and that our aim is to estimate $f(x)$ and display it graphically. The density estimation function is defined as follow :

$$\widehat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)$$

Where h is smoothing parameter and $K(.)$ is the kernel function. Required properties of the kernels are:

- $K(.)$: is density .

$$\int K(u)du = 1 \qquad \text{and} \quad K(u) \geq 0$$

- $K(.)$: Symmetric

$$\int uK(u)du = 0$$

The different kernel functions used are given in next table. Choosing $K(u)$ is less important than choosing the smoothing parameter h.

4.3.1 MSE and MISE

To investigate the performance of the kernel density estimation at a single point or over the whole real line, and find out how close our estimator $\hat{\theta}$ is to its target θ. The Mean Square Error and Mean Integrated Square Error can be used to measure such performance

Kernel	$K(u)$
Uniform Kernel	$\frac{1}{2} I\left(\lvert u\rvert \leq 1\right)$
Triangle Kernel	$(1-\lvert u\rvert)\, I\left(\lvert u\rvert \leq 1\right)$
Epanechnikov Kernel	$\frac{3}{4}\left(1-u^2\right) I\left(\lvert u\rvert \leq 1\right)$
Quartic Kernel	$\frac{15}{16}\left(1-u^2\right)^2 I\left(\lvert u\rvert \leq 1\right)$
Triweight Kernel	$\frac{35}{32}\left(1-u^2\right)^3 I\left(\lvert u\rvert \leq 1\right)$
Gaussian Kernel	$(2\pi)^{-1/2} \exp\left(-\frac{1}{2}u^2\right)$
Cosinus Kernel	$\frac{\pi}{4}\left(\cos(\frac{\pi}{2}u)\right) I\left(\lvert u\rvert \leq 1\right)$

Table 4.1: *Kernel Functions*

or efficiency of non-parametric.

The Mean Square Error in estimating $f(x)$ by $\hat{f}(x)$ at the point x is given by:

$$\begin{aligned}
\mathrm{MSE}\left\{\hat{f}(x)\right\} &= E\left[f(x) - f\hat{(x)}\right]^2 \\
&= E\left[(f(x) - E[f(x)]) + \Big(E[f(x)] - \hat{f}(x)\Big)\right]^2 \\
&= E\left[(f(x) - E[f(x)])\right]^2 + E\left[\Big(E[f(x)] - \hat{f}(x)\Big)\right]^2 + \\
&\quad 2\ E\left[(f(x) - E[f(x)])\right]\ E\left[\Big(E[f(x)] - \hat{f}(x)\Big)\right] \\
&= \mathrm{var}\{\hat{f}(x)\} + \left\{\mathrm{bias}\{\hat{f}(x)\}\right\}^2
\end{aligned}$$

We se that MSE is composed of two parts, namely the variance and the bias part. On the contrary of the parametric model, the bias is ignored.

Mean Integrated Square Error(global measurement) is defined as:

$$\mathrm{MISE}\left\{\hat{f}(x)\right\} = \int \mathrm{MSE}\left\{\hat{f}(x)\right\} dx$$

4.3.2 Asymptotic Statistical Properties of KDE

We define

$$\begin{aligned} \mu_2(K) &= \int_{-\infty}^{\infty} u^2 K(u)du \\ \text{and} \quad \|K\|_2^2 &= \int_{-\infty}^{\infty} \{K(u)\}^2 du \end{aligned}$$

Derivation of Bias approximation

We start by calculating the expression of expectation of the kernel density $\hat{f}_h(x)$

$$\begin{aligned} E\left[\hat{f}_h(x)\right] &= \frac{1}{nh}\sum_{i=1}^{n} E\left[K(\frac{x-X_i}{h})\right] \\ &= \frac{1}{h}E\left[K(\frac{x-X}{h})\right] \\ &= \int_{-\infty}^{\infty} \frac{1}{h}K(\frac{x-z}{h})f(z)dz \quad \# \, Definition \; of \; expectation \\ &= \int_{-\infty}^{\infty} K(s)f(x+sh)ds \quad \# \, Variable \; changes(s = \frac{x-z}{h}) \\ &= \int_{-\infty}^{\infty} K(s)\left\{f(x) + shf^{(1)}(x) + \frac{s^2h^2}{2}f^{(2)}(x) + o(h^2)\right\} ds \quad \# \, Taylor \\ &= \int_{-\infty}^{\infty} K(s)f(x)ds + \int_{-\infty}^{\infty} shK(s)f^{(1)}(x)ds \\ &\quad + \int_{-\infty}^{\infty} \frac{s^2h^2}{2}f^{(2)}(x)K(s)ds + o(h^2) \\ &= f(x) + \frac{h^2}{2}f^{(2)}(x)\int_{-\infty}^{\infty} s^2K(s)ds + o(h^2) \\ &\approx f(x) + \frac{h^2}{2}f^{(2)}(x)\mu_2(K) \quad h \longrightarrow 0 \\ \Longrightarrow Bias\{\hat{f}_h(x)\} &= E\left[\hat{f}_h(x)\right] - f(x) \\ &= \frac{h^2}{2}f^{(2)}(x)\mu_2(K) \end{aligned}$$

Noting that the kernel $K(u)$ is a density and symmetric. Here we presents some remarks

1. Bias $\sim \quad o(h^2)$. Bigger value of $h \Longrightarrow$ **Oversmoothing**

2. Sign and direction are decided by $f^{(2)}(x)$

3. We can reduce the error term to order of (h^4) by including more terms in Taylor series expansion.

Derivation of variance approximation

The variance of stochastic variable X is defined as $\text{var}(X) = E(X^2) - \{E(X)\}^2$ and using the Taylor expansion of $f(x+sh$ and $\int_{-\infty}^{\infty} K(s)^2 sds = 0$(from the symmetry of the kernel.

$$\begin{aligned}
\text{var}\left[\hat{f}_h(x)\right] &= \frac{1}{n^2}\text{var}\left[\frac{1}{h}\sum_{i=1}^{n} K(\frac{x-X_i}{h})\right] \\
&= \frac{1}{n}\text{var}\left[\frac{1}{h}K(\frac{x-X_i}{h})\right] \\
&= \frac{1}{n}E\left[\frac{1}{h^2}K(\frac{x-X_i}{h})^2\right] - \frac{1}{n}E\left[\frac{1}{h}K(\frac{x-X_i}{h})\right]^2 \\
&= \frac{1}{nh}\int_{-\infty}^{\infty} K(s)^2 f(x+sh)ds - \frac{1}{n}E\left[\hat{f}_h(x)\right]^2 \\
&= \frac{1}{nh}\|K\|_2^2\{f(x)+o(h)\} - \frac{1}{n}\{f(x)+o(h)\}^2 \\
&= \frac{1}{nh}\|K\|_2^2 f(x) + o(\frac{1}{nh}) \\
&\approx \frac{1}{nh}\|K\|_2^2 f(x) \quad for \quad (nh) \longrightarrow \infty
\end{aligned}$$

Some remarks about the variance of $\hat{f}^(x)$

1. var $\sim o(h^{-1})$. Small value of $h \Longrightarrow$ **Undersmoothing**

2. (nh) gives the number of the observations inside the window

3. Variance is bigger when $f(x)$ is bigger.

The mean square error (MSE) of the kernel density estimator $\hat{f}_h(x)$ is given by:

$$\begin{aligned}
\text{MSE}\left[\hat{f}_h(x)\right] &= E\left[\{\hat{f}_h(x) - f(x)\}^2\right] \\
&\approx \frac{h^4}{4}f^{(2)}(x)^2\mu_2(K)^2 + \frac{1}{nh}\|K\|_2^2 f(x) \quad for\ h \longrightarrow 0\ and\ (nh) \longrightarrow \infty
\end{aligned}$$

The mean integrated square error(MISE) of the kernel density estimator $\hat{f}_h(x)$ is given by:

$$\begin{aligned} \text{MISE}\left[\hat{f}_h(x)\right] &= E\left[\int_{-\infty}^{\infty}\{\hat{f}_h(x)-f(x)\}^2dx\right] \\ &= \int_{-\infty}^{\infty}\text{MSE}\{\hat{f}_h(x)\}dx \\ &= \frac{h^4}{4}f^{(2)}(x)^2\mu_2(K)^2+\frac{1}{nh}\|K\|_2^2+o(\frac{1}{nh})+o(h^4) \end{aligned}$$

For $h \longrightarrow 0$ and $(nh) \longrightarrow \infty$, the asymptotic mean integrated square error (AMISE) is given by

$$\text{AMISE}\left[\hat{f}_h(x)\right]=\frac{h^4}{4}f^{(2)}(x)^2\mu_2(K)^2+\frac{1}{nh}\|K\|_2^2$$

4.3.3 Bandwidth selection for KDE

To choose the smoothing parameter h, we must find the bandwidth h_{opt} that minimizes the MISE, that is

$$\begin{aligned} \partial\left\{\text{MISE}\left[\hat{f}_h(x)\right]\right\}/\partial h &= \frac{d}{dh}\left\{\frac{h^4}{4}f^{(2)}(x)^2\mu_2(K)^2+\frac{1}{nh}\|K\|_2^2\right\} \\ &= -\frac{1}{nh^2}\|K\|_2^2+h^3f^{(2)}(x)^2\mu_2(K)^2=0 \\ \Leftrightarrow \frac{1}{nh^2}\|K\|_2^2 &= h^3f^{(2)}(x)^2\mu_2(K)^2 \\ \Leftrightarrow h_{opt} &= \left(\frac{\|K\|_2^2}{f^{(2)}(x)^2\mu_2(K)^2n}\right)^{\frac{-1}{5}} \\ &\sim n^{\frac{1}{5}} \end{aligned}$$

4.4 LOWESS: Locally Weighted Scatter Plot Smoothing

Locally Weighted Scatter Plot Smoothing(LOWESS, see Cleveland 1977) is an iterative procedure combining the ideas of Law Pass smoothing with robust to outliers. The first iteration of LOWESS fits a polynomial to the data in a time window centred at t.

The LOWESS algorithm requires several parameters:

- d : Polynomial degree. The same degree used in each window.
- h : " *half window*", the number of records in the window to the side of the center data point, so that the total number of data points is $2h+1$.
- r : The number of iterations for the robust smoothing.

For a given choice of these parameters and a given data set, $X_{t=1}^{n}$:

1. For each $1 \leq t \leq n$ define a neighborhood of size h as follow:

$$N_h(t) \;= \{s : |s-t| \leq h\}$$

2. For each $1 \leq t \leq n$ define weights $w_t(s)$ for all $1 \leq s \leq n$:

$$N_h(t) \;= \begin{cases} \left(1 - |(t-s)/h|^3\right)^3, & s \in N_h(t) \\ 0, & s \notin N_h(t) \end{cases}$$

3. For each t, fit a dth degree polynomial using weighted least squares with the weights defined in step (2). The value at each t is denoted by $\hat{X}_t$.

4. Define the residuals to be $\mu_s = X_s - \hat{X}_s$ and let m be the median of $|\mu_s|$ for $s \in N_h(t)$. Assign another set of weights, δ_s as follow:

$$\delta_s \;= \begin{cases} \left(1 - |\mu_s/6m|^2\right)^2, & |\mu_s| \leq 6m, s \in N_h(t) \\ 0, & |\mu_s| > 6m \end{cases}$$

5. Iterate steps 3 and 4 r times but with $w_t(s) = \delta_s w_t(s)$ or until convergence.

4.5 Median Smoother

The steps of the median filter is defined as follow

- Choosing a span window of size k, k but be an odd number, say 3, 5, ...
- Sorting the data in increasing order.
- The central value of the sort is selected as the median.
- The smoothed value is replaced by the median of the selected window.

We can ameliorate the algorithm by adding a threshold value T_h, . The replacement is done if the absolute value of the difference between the median and the observation under examination is greater that T_h, otherwise the observation left unchanged.

4.6 Conclusion

We have presented different algorithms of smoothing and there exists other methods that are not presented her like

- **Kernel Regression:** Consider a set of observations of two variables $\{y_i, x_i\}_{i=1}^{N}$ which are connected via unknown regression function m as follow:

  $$y_i = m(x_i) + \epsilon_i$$

  Where the ϵ_i are independent random variables with 0 mean and constant variance. Kernel estimate can be used to estimate the unknown function m.

  The most popular nonparametric regression smoothers are the Nadaraya-Waston (NW) estimator $\hat{m}_{NW}$ and Gasser-Muller estimator $\hat{m}_{GM}$. The first one consists in choosing weights by direct kernel evaluation while the second by convolution of the kernel with a histogram representing the data. The kernels are presented bellow:

$$\begin{aligned}\hat{m}_{NW} &= \sum_{i=1}^{n} y_i K\left((x - x_i)/h\right) / \sum_{i=1}^{n} K\left((x - x_i)/h\right) \\ &\text{and} \\ \hat{m}_{GM} &= \sum_{i=1}^{n} y_i \int_{(x_{(i)}+x_{(i-1)})/2}^{(x_{(i+1)}+x_{(i)})/2} K\left((x - x_i)/h\right)\end{aligned}$$

- **K-nearest neighbors:** This method is based on linear regression. To predict the value of x_t, uses the $K/2$ values bigger than x_t and $K/2$ values smaller than x_t
- **Local Polynomial Regression:** This method uses weighted least squares(WLS) regression to fit a $\mathbf{d^{th}}$ degree polynomial($d \neq 0$) to data locally. For more information, se Wand and Jones.
- **Periodic Regression:** We try to fit a trigonometric polynom to our data, $A_0 + \sum_{j=1}^{K} (A_j \cos(\omega_j\ t) + B_j \sin(\omega_j\ t))$
- **Eliminating high frequencies:** Applying Fourier transform to data, many coefficients are usually very small and can be ignored to some threshold value T_h, then performing the inverse transform to get our data back.

These methods presented here will be used in next chapters.

Chapter 5

Outliers Detection in Time series

The outlier detection is an important step in time series analysis, because the outliers can easily bias parameter estimations and introduce unwanted signals in time series that can lead to model misinterpretation and poor forecasts. Several detections methods have been proposed for univariate time series, but here we will introduce some very simple algorithms to detect and simulate outliers for CGPS time series. The outliers can be caused at the station for many different reasons, e.g. by high ionosphere activity, earthquake or other unknown natural phenomenon, but for many outliers it is impossible to say why they lay far from neighbouring data points.

5.1 DRSO Algorithm

After examination of the data from all the sites, we concluded that the time series of the East, North and the Vertical component contain outliers, and shall be removed. We prefer to analyse each component independently even if the underlying physic suggests dependency between all components(if an observation is outlier shall be observed in all 3 components !).

The DRSO stands for **D**etection,**R**emovable, **S**imulation, **O**utliers. This algorithm is capable of detecting the outliers, remove them, simulate a new candidate without changing the structure on data.

The algorithm is controlled by two parameters:

- α : Length of sub-intervals JUsuallyly between 40 and 140 data points.

- β : the slope of the sub-interval J.

For detection of outliers, I used the median value rather than the mean value(the median is more robust estimator than the mean) and the slope(coefficient director) for the whole data.

If the data point under investigation is greater than $(k + \beta)$ times the median of the sub-interval J. We simulate a new candidate from normal distribution with the mean is the median of sub-interval J and variance of ±0.5.

The algorithm is tested with different values of α and β and gives a very good promising results for East and North components.

The DRSO procedure is defined as follow:

1. Divide the whole interval of East, North and Vertical components into J sub-intervals of length L. Data points are between [40, 140] elements.
2. Calculate the slope β for whole data points, by fitting a straight line to the data.
3. The robust estimator chosen is the median. Calculate the median of each sub-interval J
4. The scaling factor k is computed by the formula $k = \tan(\gamma) \times L$, where γ is the angle between the slope of sub-interval J and the horizontal line of length L
5. Go through the raw data and examine each observation in turn, if the value of observation is greater than $(k+\beta)$ times the median of J, then an outlier is detected.
6. Remove the detected outlier and replace it with a new candidate. The new candidate is simulated from normal distribution with the mean equals to the median of sub-interval J and variance equals to ±0.5.

To solve the problem of the vertical component , I used a low pass filter to smooth the data and then apply the algorithm to remove the outliers. This filter is an Symmetric Moving Average Filter(SMAF).

The drawbacks of this algorithm are:

- Operating directly on the data is not a good idea, it's better to analyse residuals.
- Fitting a straight line is a very poor representation of data.

- The algorithm handle vertical direction poorly, due to the presence of strong periodic components in the signal.

- A good algorithm is the algorithm that can handle all components(East, North and Vertical directions).

Some corrections are made to make the DRSO algorithm more robust, and the procedure of modified DRSO are:

1. Get a god approximation of the data by applying **K**ernel smoothing, LOESS or simply fit a higher polynomial to the data.

2. Get the residuals X_i .

3. Compute a new vector say $Y_i = \text{abs}(X_i)$

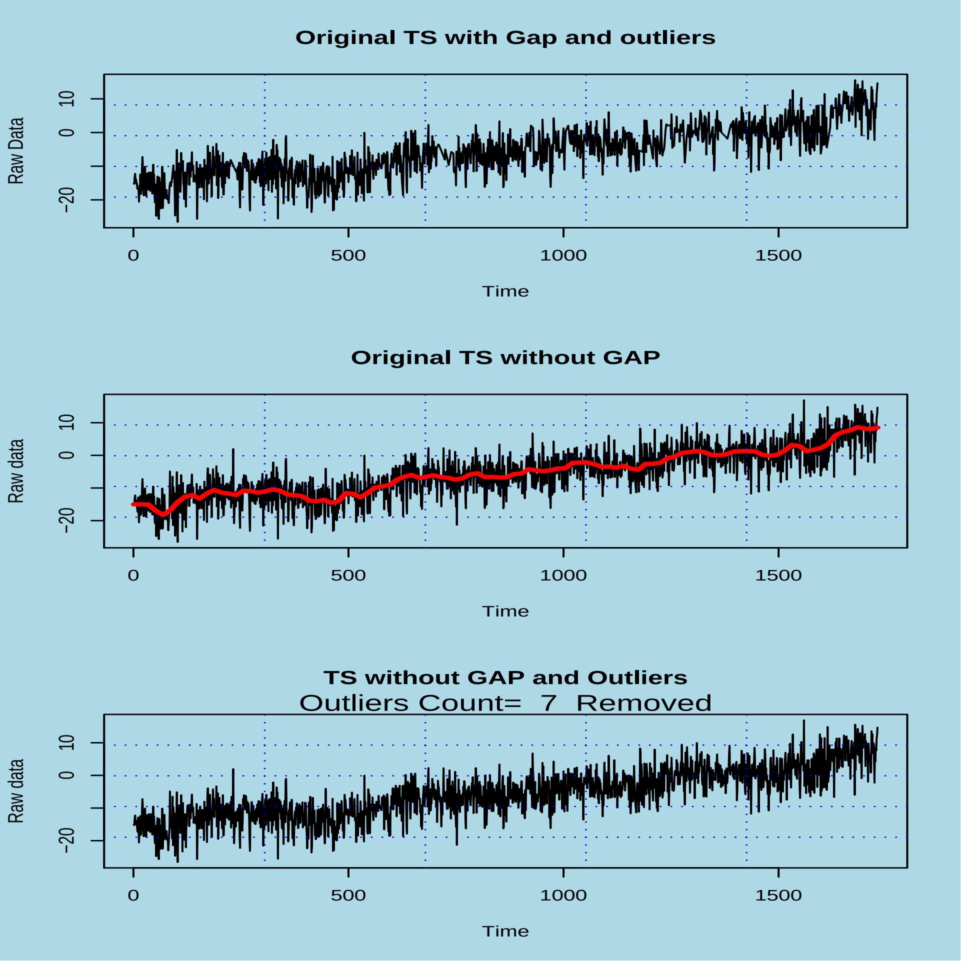
4. Divide the whole interval of each components into J sub-intervals of length L. Data points are between [40, 140] elements as before, and calculate the variance for each sub-interval J.

5. If the value of $Y_i \geq K \times \text{var}(Y_i)$ of sub-interval J, then the observation under examination is an outlier. The scaling factor $k \in [2, 3]$, control the outlier detection level.

6. To generate a new candidate, we simulate from a normal distribution where the mean is the median of sub-interval J and the variance is the variance of sub-interval J.

7. We are done, and we need to get our data back as $Z_i = Y_i \times (\text{sign of } \mathrm{X_i}) + (\text{fitted data})$

The next figure shows an output generated by the algorithm. The advantage of this algorithm is that we conserve the location of the data and that we move only suspected observations.



5.2 Outlier Detection Algorithm 2

The algorithm is similar to the previous one, the difference is that it requires two additional parameters:

- G : The gain factor, controls how far an observation is from its neighbours not considered to be an outlier. The value of 1 to 20 can be used).
- S : The scaling factor, just a value from 1 to 5.
- J : The data length under examination from 30 to 200 elements.

For detection of outliers, the median value of sub-interval J will be used and the criteria for detection is defined as follow:

$$|x(t_i) - \bar{x}| > GS\delta_J$$

The procedure is defined as follow:

1. Run the *offsetDetection.R* algorithm in next chapter to detect and correct for jumps(offsets).
2. Fit a straight line or higher order polynomial and get the residuals.
3. Divide the whole interval of the East, North and the Vertical components into J sub-intervals of length k. The data points are between [30, 200] elements each
4. The robust estimator chosen is the median. Calculate the median of each sub-interval J.
5. Go through the residuals and examine each observation in turn. If the criteria is fullfilled, the observation is detected as outlier. The simulation of a new candidate is generated from Gaussian where the mean is the median of sub-interval J and the variance is the variance of sub-interval J not including the outlier.
6. When we are done, add the residuals to the fitted data.

The drawback of this algorithm is that for each time series, a convenable scaling factor S, sub-interval J and gain factor G must be identified. This means that, for each site a different value of parameters must be tested.

The advantage is that once these parameters are known and stored in the database for each site, the processing is carried out easily.

5.3 Grubbs Test for Outliers

Most outlier tests look at some measure of the relative distance of a suspect point from the mean value. This measure is then assessed to discover if the extreme value could reasonably be expected to have arisen by chance.

Grubbs test for outliers is based on the assumption of normality, that is, Kolmogorov-Smirnov test for normality can be carried out. For simplicity, we assume that our residuals are Gaussian distributed with mean μ and variance δ.

If more that 20% of our observations are outliers, then our algorithm is wrong. Grubbs test values can detect one outlier in data set (left or right side), 2 outliers(one in the left and one to the right side) and 4 outliers(2 in the left and 2 to the right side) and are defined by G_1 , G_2 and G_3:

$$G_1 = |\bar{x} - x_i|/S \qquad G_2 = (x_n - x_1)/S \qquad G_4 = 1 - \left\{(n-3)S^2_{n-2}/(n-1)S^2\right\}$$

S is the sample variance.

5.4 Dixon's Test for Outliers

Dixon investigated the performance of several statistical tests in terms of their ability to reject bad values in data sets taken from a Gaussian distribution. As a conclusion, we assume that our residuals are ordered in increasing order, such that $(\epsilon_1 < \epsilon_2 < .... < \epsilon_n)$

1. For a single outlier x_1

$$r_{10} = \frac{x_2 - x_1}{x_n - x_1} \qquad \left(\text{OR} \quad \frac{x_n - x_{n-1}}{x_n - x_1}\right)$$

2. For outlier x_1 avoiding x_n

$$r_{11} = \frac{x_2 - x_1}{x_{n-1} - x_1} \qquad \left(\text{OR} \quad \frac{x_n - x_{n-1}}{x_n - x_2}\right\}$$

3. For outlier x_1 avoiding x_n and x_{n-1}

$$r_{12} = \frac{x_2 - x_1}{x_{n-2} - x_1} \left(\text{OR} \quad \frac{x_n - x_{n-1}}{x_n - x_2}\right\}$$

4. For outlier x_1 avoiding x_2

$$r_{20} = \frac{x_3 - x_1}{x_n - x_1} \left(\text{OR} \quad \frac{x_n - x_{n-2}}{x_n - x_1}\right\}$$

5. For outlier x_1 avoiding x_2 and x_n

$$r_{10} = \frac{x_3 - x_1}{x_{n-1} - x_1} \left(\text{OR} \quad \frac{x_n - x_{n-1}}{x_n - x_1}\right\}$$

6. For outlier x_1 avoiding x_2 and x_n, x_{n-1}

$$r_{10} = \frac{x_3 - x_1}{x_{n-2} - x_1} \left(\text{OR} \quad \frac{x_n - x_{n-2}}{x_n - x_3} \right\}$$

5.5 Robust Outlier Detection Simulation Algorithm

For CGPS time series, the most common technique used to reduce the effect of high frequency components is to apply a Low Pass Filter (Moving Average Filter), this filter is too weak for outliers. I implemented and tested with different Filters(MS, WMAF, EWF,..), and we concluded with the kernel smoothing and Robust LOWESS algorithms instead.

We will introduce both algorithms, and finally introduce the Robut Outlier Detection Simulation algorithm that handle outliers in CGPS almost perfectly.

The main reason of doing thing this way, is that we would like to detect outliers, remove them and replace them without smoothing the data and destruction of modes. Therefore we choose to process the data this way.

1. Use Kernel Smoothing with optimal bandwidth or robust LOWESS to get a good estimation of data.

2. Get the residuals by extracting the fitted values from the original data.

3. Test residuals for normality by applying Kolmogorov-Smirnov test or qqplot. For simplicity we assume that the residuals are normal distributed.

4. Apply Grubbs and Dixon's tests for outliers, this test detects one outlier at a time. For each outlier detected , it will be replaced by simulating a new candidate from normal distribution, with a the mean is the median and the variance of the sub-interval under processing. The length chosen is $N = 25$ points

5. Now we like to get our data without outliers, this is done by adding the new residuals with fitted values found in the first step.

This algorithm is similar to the Modified Detection Removable Simulation algorithm, the only difference is that the detection criteria of MDRSO uses $Y_i \geq K \times \text{var}(Y_i)$ and RODS uses Grubbs and Dixon's test for detection.

Se the **appendix B** for output generated from this algorithm.

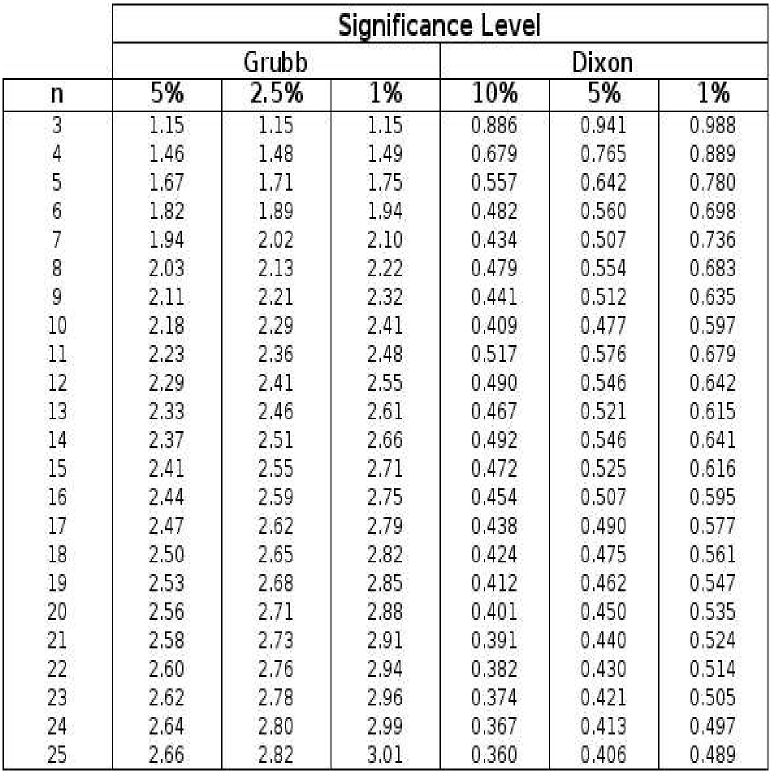
5.6 Conclusion

My philosophy is that the data analyst is like a mayor of a city. He wants to keep the city nice. Bad buildings are like outliers or extreme values. The mayor can apply the Shanghai model and demolish all bad areas and build everything from scratch. This model has several drawbacks; you have to move people from their homes and historical buildings will vanish.

The data analyst can apply a smoother. If the smoothing parameter is not right, the destruction of modes and data moving will take place.

If the mayor try to fix only bad houses and keep the rest in the same location, the city will conserve the historical thing and avoid moving people around. That is what we have done in this chapter. We have tried to develop algorithms that keep data in their locations and replace by simulation only the suspected observations.

For the Grubb and Dixon $p - value$ used are given in table.



	Significance Level					
	Grubb			Dixon		
n	5%	2.5%	1%	10%	5%	1%
3	1.15	1.15	1.15	0.886	0.941	0.988
4	1.46	1.48	1.49	0.679	0.765	0.889
5	1.67	1.71	1.75	0.557	0.642	0.780
6	1.82	1.89	1.94	0.482	0.560	0.698
7	1.94	2.02	2.10	0.434	0.507	0.736
8	2.03	2.13	2.22	0.479	0.554	0.683
9	2.11	2.21	2.32	0.441	0.512	0.635
10	2.18	2.29	2.41	0.409	0.477	0.597
11	2.23	2.36	2.48	0.517	0.576	0.679
12	2.29	2.41	2.55	0.490	0.546	0.642
13	2.33	2.46	2.61	0.467	0.521	0.615
14	2.37	2.51	2.66	0.492	0.546	0.641
15	2.41	2.55	2.71	0.472	0.525	0.616
16	2.44	2.59	2.75	0.454	0.507	0.595
17	2.47	2.62	2.79	0.438	0.490	0.577
18	2.50	2.65	2.82	0.424	0.475	0.561
19	2.53	2.68	2.85	0.412	0.462	0.547
20	2.56	2.71	2.88	0.401	0.450	0.535
21	2.58	2.73	2.91	0.391	0.440	0.524
22	2.60	2.76	2.94	0.382	0.430	0.514
23	2.62	2.78	2.96	0.374	0.421	0.505
24	2.64	2.80	2.99	0.367	0.413	0.497
25	2.66	2.82	3.01	0.360	0.406	0.489

Figure 5.1: *Grubb and Dixon p-values*

Chapter 6

Handling apparent discontinuities in TS

Discontinuities in time series due to Earthquake or to the antenna movements can easily bias parameters estimations and introduce unwanted signals in time series that can lead to model misinterpretation as outliers. Therefore, it might be important to analyze the time series by breaking them into homogeneous segments, or by removing discontinuities. Therefore, it is important to find the time instants when the abrupt changes occur and to estimate the different models for different segments during which the system does not change. The second approach is to connect the two segments to obtain a single time series.

We can divide offsets into three categories or groups, the first one is due to changes made at the CGPS site (e.g antenna changes, firmware, ..). This type of offset is easily detected and corrected if we establish good management procedures and careful logging all events. The second type, larger and unexpected offsets, will be visible in the time series and can be resolved. The last one, is the difficult one to handle, because is smaller and unknown. Any algorithm used to detect and correct this type, must define some limit or threshold.

In this chapter we will present two different algorithms to correct CGPS time series offsets.

6.1 Connecting two time series algorithm

The main reason to connect segments of time series and to obtain one homogenous is to improve the precision. Te main objective of this thesis is to estimate the CGPS site velocities from their time series. I am to present two simple algorithms capable to detect and correct for offsets. The first algorithm is based on a standard binary search algorithm for detection and a simple formula for connection. The second one is based on off-line estimation of the change time.

6.2 The Offset Detection Correction Algorithm 1

This algorithm works in two steps, the first step is to detect the time of offset occurrence and the second step is to calculate the value of jump in time series.

1. Perform the standard binary search using two comparisons per level. This means, divide the data interval in two equal intervals, and calculate the slope of each interval
2. Fit a straight line to each segment and choose the slope with highest absolute value.
3. If data length of the remaining segment is greater that 3 elements, repeat step 1 and 2. Otherwise, perform the next step.
4. Find the largest absolute value of the remaining elements. The index of the largest element represent the time of offset occurrence..
5. To connect the two segments, I used this procedure:
    (a) Fit a straight line to data segment with largest data length and get the slope b
    (b) Get two small segments with 40 to 100 elements from the detected offset,one from left and other from right
    (c) For each segment, calculate the median or the mean
    (d) calculate the offset value that separate the two segment by this Formula.

    $$\Delta Y = \bar{Y_2} - \bar{Y_1} - b * (\bar{t_2} - \bar{t_1})$$

    Where b is the slope, calculated by linear regression or by weighted linear regression over the largest possible time span. The indexes 1 and 2 indicate the time windows before and after the offset.$\bar{Y}$ is the coordinate average and $\bar{t}$ the mean time of the indicated window.

Below you will find the output from the program summarized in the table 6.1. The results shows very good approximation.

- **First column:** This column contains the test numbers, we have performed 20 tests.

Test Number	Position	Offset Added	Position Detected	Offset Corrected
1	100.00	5.00	99.00	5.23
2	104.00	9.00	104.00	9.59
3	108.00	13.00	108.00	13.03
4	112.00	17.00	112.00	17.04
5	116.00	21.00	116.00	21.16
6	120.00	25.00	120.00	24.95
7	124.00	29.00	126.00	27.86
8	128.00	33.00	128.00	32.96
9	132.00	37.00	132.00	36.96
10	136.00	41.00	136.00	40.94
11	140.00	45.00	140.00	45.09
12	144.00	49.00	145.00	48.01
13	148.00	53.00	148.00	53.05
14	152.00	57.00	154.00	54.43
15	156.00	61.00	156.00	60.83
16	160.00	65.00	160.00	64.88
17	164.00	69.00	164.00	68.73
18	168.00	73.00	168.00	72.98
19	172.00	77.00	174.00	73.15
20	176.00	81.00	176.00	81.52

Table 6.1: *Offset Detection Output*

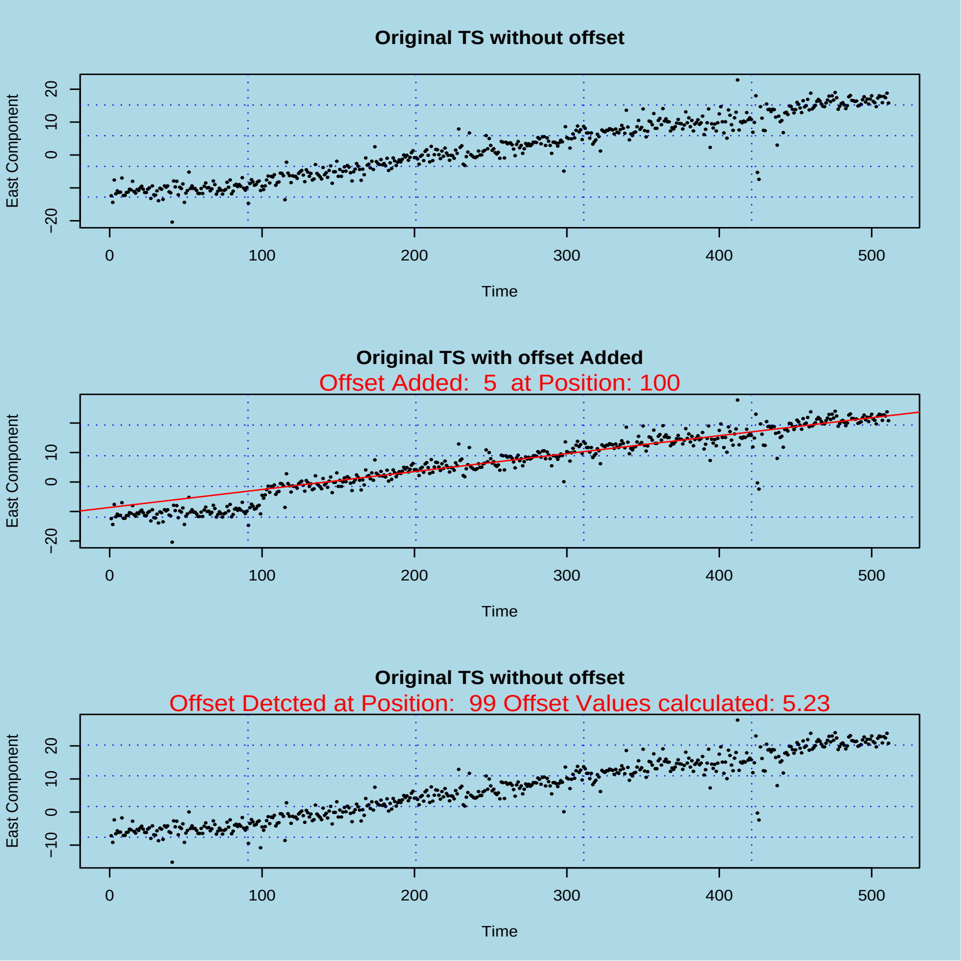
- **Second column:** Gives the position in our time series to introduce an offset(a jump).

- **Third column:** Gives the size the offset

- **Fourth column:**The position in our time series detected by the algorithm

- **Last column:** The offset corrected by algorithm

One snapshot from above test is presented by the graph below. The first graph shows the original time seres, the second graph two segments of the original time series and the last one the connection of the time series.


Original TS without offset
East Component
20
10
0
−20
0
100
200
300
400
500
Time
Original TS with offset Added
Offset Added: 5 at Position: 100
East Component
10
0
−20
0
100
200
300
400
500
Time
Original TS without offset
Offset Detcted at Position: 99 Offset Values calculated: 5.23
East Component
20
10
0
−10
0
100
200
300
400
500
Time

6.3 A change Detection Correction Algorithm 2

Before describing in more details the second algorithm for detecting offset in time series, let us introduce some notation and definitions. Let $(\Omega, \aleph, \Im)$ be a probability space, where Ω is an abstract space (the sample space), $\aleph$ is a sigma algebra of subsets of Ω and $\Im$ is a probability measure defined over all members of $\aleph$.

Let our observations be presented by $x_{i=1}^{N}$, parameters be presented by $\theta = (\theta_1, ..., \theta_k)$ and let $S_j^k = \sum_{i=j}^{k} s_i$. Where $s_i = \log\{P_{\theta_0}(x_i)/P_{\theta_1}(x_i)\}$ is log likelihood ratio and is a sufficient statistic.

Definition 6.3.1. *(Unbiased Estimator): An estimator $H = h(x_1, x_2, ..., x_N)$ is said to be unbiased estimator for θ if $E(H) = \theta$.*

Definition 6.3.2. *(Efficient Estimator): Let H_1 and H_2 be two unbiased estimator of θ with variances* $\mathrm{var}(H_1)$ *and* $\mathrm{var}(H_2)$. *We will call H_1 more efficient than H_2 if:*

$$\mathrm{var}(H_1) \leq \mathrm{var}(H_2) \tag{6.1}$$

Also, the relative efficiency of H_1 with respect to H_2 will be defined as ratio

$$\mathrm{var}(H_1)/\mathrm{var}(H_2) \tag{6.2}$$

Definition 6.3.3. *(Consistent Estimator): We say that $H = h(x_1, ..., x_N)$ is a consistent estimator for θ, if it converges in probability to θ- That is, if for all $\epsilon \geq 0$ and $\delta > 0$, there exists an $N(\epsilon, \delta)$ such that:*

$$P\{|H_N - \theta| < \epsilon\} > 1 - \delta \qquad for \quad N > N(\epsilon, \delta) \tag{6.3}$$

To prove the asymptotic property (consistency) of an estimator, usually we use the Chebyshev's inequality.

Theorem 6.3.1. *(Chebyshev's inequality): Let $x = (x_1, x_2, ..., x_n)$ be any random variable with mean μ and variance σ^2. For any $\epsilon > 0$*

$$P\{\|x - \mu\| \geq \epsilon\} \leq \sigma^2/\epsilon^2 \tag{6.4}$$

Definition 6.3.4. *(Sufficient Estimator): We say that S is a sufficient statistic for parameter θ, if there exists a determination of the conditional distribution $P(x_1^N \in B)|S)$ that is independent of θ*

In other words, the information about θ contained in the sample x_1^N is consentrated in the statistic S.

Let us introduce the ideas behind the algorithm, it's based on continuously inspection on data samples and a decision rule is computed to test the parameter θ between the two following hypotheses

$$H_0 : \theta = \theta_0 \qquad H_1 : \theta \neq \theta_0$$

The sampling and test continue as long it is in favor of H_0, it stop when the first observation for which the decision is taken in favor of H_1.

Let N be the sample size, the optimal decision rules D is defined by

$$D = \begin{cases} 0 & S_1^N < h \quad H_0 \text{ is chosen} \\ 1 & S_1^N \geq h \quad H_1 \text{ is chosen} \end{cases}$$

Where the S_1^N is the decision function and h is the threshold chosen for detection.

The stopping rule that help us to make a decision is defined by:

$$t_a = N.\min\{K : d_k = 1\}$$

Where the d_k is the the sample number k of size N and t_a is the alarm time.

Let us focus on implementation,the algorithm procedure is defined as follow :

- **STEP 1:** Run the kernel smoothing or LOWESS algorithm and get residuals.
- **STEP 2:** We assume that the residuals are white noise presented by $\{x_i\}_{i=1}^N$, with mean μ_x and standard deviation σ.

  Now two situations are possible, either all residuals have the same mean or there exists an unknown change time $t_1 < t_{off} < t_N$ such that, before t_{off}, the mean is equal to μ_0 and after t_{off} the mean is equal to $\mu_1 \neq \mu_0$.

  In this case the changing parameter θ is μ.The probability density is given by:

$$p_\theta(x) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left\{-1/(2\sigma^2)\,(x_i - \mu)^2\right\}$$

Define the sufficient statistic test S_i

$$\begin{aligned} S_i &= \ln \{p_{\theta_1}(x)/p_{\theta_0}(x)\} \\ &= \frac{\mu_1 - \mu_0}{\sigma^2}\left\{x_i - \frac{\mu_0 - \mu_1}{2}\right\} \\ &= \frac{A}{\sigma}\left(x_i - \mu_0 - B/2\right) \end{aligned}$$

Where $B = \mu_1 - \mu_0$ is the change in magnitude(offset size) and $A = (\mu_1 - \mu_0)/\sigma$ is the signal to noise ratio.

The decision function is defined by:

$$S_1^N = A/\sigma \sum_{i=1}^{N} \{x_i - \mu_0 - B/2\} \tag{6.5}$$

The stopping rule for the change in mean is defined as before:

$$D = \begin{cases} 0 & S_1^N(K) < h \quad H_0 \text{ is chosen} \\ 1 & S_1^N(K) \geq h \quad H_1 \text{ is chosen} \end{cases}$$

The alarm is set for the first time when $\mu_1 > \mu_0$ at which

$$\bar{x}(K) \geq \mu_0 + \kappa\sigma/\sqrt(N) \quad \text{where} \quad \bar{x}(K) = 1/N \sum_{i=N(N-k)+1}^{NK} x_i$$

The problem is that we like to detect deviation in both direction (increasing and decreasing). Therefore the alarm is set if $\bar{x}(K) \geq |\mu_0 + \kappa\sigma/\sqrt(N)|$

- **STEP 3:** Detect the time of abrupt if exists by maximizing the function

$$f(k) = k(N-k)(\mu_0 - \mu_1)^2$$

The estimated time change $\hat{t}_{off} = \max\ _{1 \leq k \leq N} \left[k(N-k)(\mu_0 - \mu_1)^2\right]$. The mean $\mu_0 = \frac{1}{k}\sum_{i=1}^{k} x_i$ and $\mu_1 = \frac{1}{N-k}\sum_{i=k+1}^{N} x_i$.

- **STEP 4:** The time of abrupt $\hat{t}_{off}$ and the offset size are estimated. For correction, we add the offset from the beginning to the time of abrupt,finally we have one homogene time series.

Chapter 7

Handling gaps in CGPS time series

Evenly spaced data in CGPS time series is hard to obtain, due to hardware failure (GPS receiver, antenna, router, etc) or software failure or other reasons.
The site can be down for a while and we are facing missing data problems. In this chapter we are going to present two algorithms *to handle the gap in our CGPS time series, by transforming Unevenly sampled data to uniform sampled data.* Filling the gaps with well known methods like Simulation or Prediction. Both methods will be presented and implemented. *the interpolation method can be used to fill gaps but is too weak if a lot of consecutive points are missing, and we have* to interpolate missing data points.

7.1 Filling gaps by Simulation:

We are going to fill the gaps with values that are acceptable without destruct the data structure. As we know that our signal is composed from several components,namely linear trend's component, periodic component and white noise components. If we like to simulate to fill gaps, we must take into account all components. The algorithm is very simple and is defined as follows:

1. **Step 1:** Run the ***drsoAlgorithm.R*** to remove and simulate outliers.
2. **Step 2:** Run the ***offsetDetection.R*** to remove offsets in our time series.
3. **Step 3:** Fit a polynomial of degree 1 to data, and extract the linear trend from original data.
4. **Step 4:** Now we must estimate or make a good guess to define the periodic components, we like to fit a periodic component of the form $C(\cos) = \mu + \beta \cos\left(\frac{2\pi(t-\alpha)}{\tau}\right)$.

The question is now, how we define the parameters μ, α and β ?.

We start by choosing the data for a period(1 year) without gaps and plotting the monthly mean against the month index and define visually if the cos or sin will be well fit.

(a) **Overall mean μ:**
A good initial guess for the overall mean μ can be obtained by taking the average of the lowest and highest monthly mean.

(b) **Signal amplitude β:**
A good initial guess for the amplitude β can be obtained by dividing the difference between the lowest and highest monthly mean variations by two.

(c) **The shift α:**
A good initial guess for the shift α is equal to the month of the highest mean variations.

(d) **Period τ**
We expect that $\tau = 12$ months, this value is fixed.

5. **Step 4:** For a missing value, simulate from normal distribution where the mean is the value from the fitted line and the variance of 0.5. or equivalently using the fitted value, plus white noise plus estimated periodic component.

6. **Step 5:** Control visually that the simulated values follows the same direction as other observations. Make sure that we are not generating the outliers.

The implemented algorithm is located under the *libs* directory and the name is xtbf*gapWithSim.*

7.2 Filling gaps by Prediction:

This method lets the data speech for it self. The drawback is that if the consecutive observations are missing from the beginning of the file, it can give unwanted results.

With forecasting, we mean predicting the future values of time series x_t from previous observations .We restrict attention to predictors that are linear functions of the data, that is, prediction of the form

$$x^n_{n+m} = \alpha_0 + \sum_{i=1}^{n} \alpha_t x_t$$

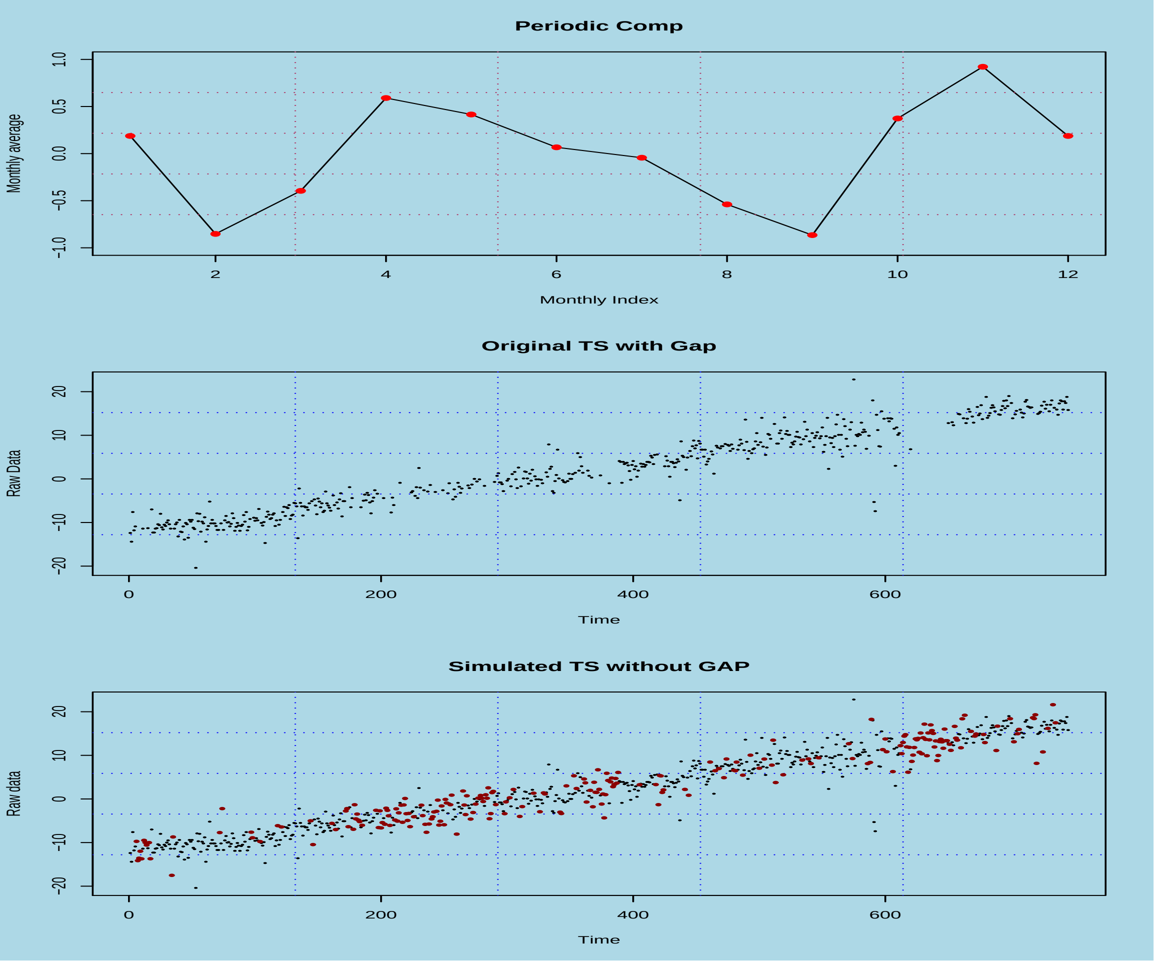


Figure 7.1: *Gap By Simulation*

Definition 7.2.1. *(Best Linear Predictor): The best linear predictor for x_{n+m} is the linear predictor that minimizes the mean square error*

$$\hat{x}_{n+m} = E(x_{n+m}|x_1, x_2, ..., x_n)$$

The best linear predictor of $x_{n+m}, for \quad m = 1, 2, ..., k$ if it minimizes the predictor error

$$S(\alpha) = E(x_{n+m} - x^n_{n+m})^2$$

Where the α is the vector of the coefficients α_i. Since $S(\alpha)$ is quadratic function of α there is at least one value of α that minimizes (α). Its satisfy the equations

$$\frac{\partial S(\alpha)}{\partial \alpha_i} = 0, \quad i = 0, 1, 2, 3, ..., n$$

The evaluation of the derivatives gives us so called Prediction Equations

$$\frac{\partial S(\alpha)}{\partial \alpha_0} = E\left[x_{n+m} - \alpha_0 - \sum_{i=1}^{n} \alpha_i x_{n+1-i}\right] = 0$$

$$\frac{\partial S(\alpha)}{\partial \alpha_j} = E\left[(x_{n+m} - \alpha_0 - \sum_{i=1}^{n} \alpha_i x_{n+1-i})x_{n+1-j}\right] = 0$$

Assuming that $E(x_t) = \mu$ the first equation can be written as:

$$\mu - \alpha_0 - \sum_{i=1}^{n} \alpha_i.\mu = 0 \quad \Rightarrow \alpha_0 = \mu(1 - \sum_{i=1}^{n} \alpha_i)$$

Solving the second equation, we get :

$$\begin{aligned} 0 &= E(x_{n+m}.x_{n+1-j}) - \alpha_0.\mu - \sum_{i=1}^{n} \alpha_i.E(x_{n+1-i}.x_{n+1-j}) \\ 0 &= E(x_{n+m}.x_{n+1-j}) - \mu^2(1 - \sum_{i=1}^{n} \alpha_i) - \sum_{i=1}^{n} \alpha_i.E(x_{n+1-i}.x_{n+1-j}) \\ 0 &= \gamma(m - (1 - j)) - \sum_{i=1}^{n} \alpha_i \gamma(i - j) \end{aligned}$$

That is, we obtain the following form of the prediction equations:

$$\gamma(m - 1 + j) = \sum_{i=1}^{n} \alpha_i \gamma(i - j), \quad j = 1, 2, ..., n$$

We obtain the same set of equation if $E(x_t) = 0$. Let us focus on One -Step ahead Prediction.

7.2.1 One -Step ahead Prediction

Given $x_1, x_2, x_3, ..., x_n$, our aim is to forecast the value of x_{n+1} the best linear predictor is $x^n_{n+1} = \sum_{i=1}^n \alpha_i x_{n+1-i}$

The coefficients α_i are satisfying the prediction equations:

$$\sum_{i=1}^n \alpha_i \gamma(i-j) = \gamma(j), \quad j = 1, 2, ..., n$$

We write the following equations in Matrix form:

$$\Gamma_n \alpha_n = \gamma_n$$

Where

$$\begin{aligned} \Gamma &= \{\gamma(i-j)\}_{j,i=1,2,...,n} \\ \alpha_n &= (\alpha_1, ..., \alpha_n)^t \\ \gamma_n &= (\gamma(1), ..., \gamma(_n))^t \end{aligned}$$

if Γ_n is nonsigular, a unique solution exist and is equal to:

$$\alpha_n = \Gamma_n^{-1} \gamma_n$$

The the forecast of x_{n+1} based on $X = (x_1, x_1, ..., x_n)^t$ can be written as:

$$x^{(n)}_{n+1} = \alpha^t_n X$$

The mean square one-step-ahead prediction error denoted by P^n_{n+1} is

$$\begin{aligned} P^n_{n+1} &= E\{X_{n+1} - X^{(n)}_{n+1}\}^2 \\ &= E\{X_{n+1} - \alpha^t_n X\}^2 \\ &= E\{X_{n+1} - \gamma^t_n \Gamma^{-1}_n X\}^2 \\ &= E\{X^2_{(n+1)} - 2\gamma^t_n \Gamma^{-1}_n X X_{n+1} + \gamma^t_n \Gamma^{-1}_n X X^t \Gamma^t_n \gamma_n\} \\ &= \gamma(0) - 2\gamma^t_n \Gamma^{-1}_n \gamma_n + \gamma^t_n \Gamma^{-1}_n \Gamma_n \Gamma^{-1}_n \gamma_n\} \\ &= \gamma(0) - \gamma^t_n \Gamma^{-1}_n \gamma_n \end{aligned}$$

For large values of n, it's difficult to calculate the inverse of matrix Γ_n of large dimension. A recursive algorithm known as Durbin-Levinson Algorithm can be used to calculate the predictor and mean square error. The Algorithm is defined as follow:

- **STEP 1:** Put ϕ_{00} and $P_1^0 = \gamma(0)$
- **STEP 2:** For $n \geq 1$ calculate

  $$\phi_{nn} = \frac{\rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k}\rho(n-k)}{1 - \sum_{k=1}^{n-1} \phi_{n-1,k}\rho(k)}$$

  Where, for $n \geq 2$

  $$\phi_{nk} = \phi_{n-1,k} - \phi_{nn}\phi_{n-1,n-k}$$

- **STEP 3:**For $n \geq 1$ calculate:

  $$P_{n+1}^{(n)} = P_n^{(n-1)}(1 - \phi_{nn}^2)$$

Chapter 8

Frequency Analysis of Time Series

Time series often contains unknown periodicities, some of which may be of interest. I will start by presenting some historical progress done by identifying the periodogram as important instrument to detect the periodicity in time series.

In 1898 Schuster defined the periodogram as a method to discover the frequencies of the hidden harmonics in the signal. Yule and Kendal (1937) motivate the idea of the periodogram in the following way. If we go to a city that has one single radio station and we don't know the frequency used by the station. We move the dial through all frequencies and when the frequency in the receptor coincide, the intensity of the sound would go up.

The periodogram considers all frequencies $\frac{1}{N}, \frac{1}{N}, \frac{2}{N}, \frac{3}{N}, \ldots .\frac{1}{2}$ *and correlate each frequency with the data of the series in order to measure the* ***Intensity****.*

A common technique used to study such periodicities is the Fast Fourier Transform(FFT). Periodicities are found by searching the standard periodogram from Fourier analysis and looking for sharp peaks. Usually these peaks correspond to periodicity in the time series.

The spectral density estimation is a very useful technique for examining certain cyclic patterns of a stationary time series.

If we are dealing with uniform sampled data, we can use the standard procedures to extract the periodic components from time series. Our model can be expressed by the following formula:

$$X_{t_i} = a + bt_i + A\cos(2\pi f t_i) + B\sin(2\pi f t_i) + w_{t_i}$$

The first step is estimate the constants a and b and extract the linear trend from our original time series X_{t_i} we have:

$$\begin{aligned} Y_{t_i} &= X_{t_i} - (a + bt_i) \\ &= A\cos(2\pi f t_i) + B\sin(2\pi f t_i) + w_{t_i} \end{aligned}$$

8.1 Spectral Density and estimation of Spectrum

Definition 8.1.1. *(Spectral Density): If the $\{x_t\}_{t=1}^{N}$ has autocovariance γ satisfying*

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$$

Then we define the spectral density as

$$f(v) = \sum_{h=-\infty}^{\infty} |\gamma(h)e^{-2\pi i v h}$$

We give some facts about Spectral Density Function

- **Fact 1:** The series $\sum_{h=-\infty}^{\infty} |\gamma(h)e^{-2\pi i v h}$ is absolutely summable. This is because $|e^{i\theta}| = |\cos(\theta) + i\sin(\theta)| = \left[\cos(\theta)^2 + \sin(\theta)^2\right]^{1/2} = 1.$. Because γ is absolute summable.

- **Fact 2:** f is periodic, with period 1.

- **Fact 3:** f is even, this means that $f(v) = f(-v)$ To demonstrate that, we write

$$\begin{aligned} f(v) &= \sum_{h=-\infty}^{\infty} \gamma(h)e^{-2\pi i v h} \\ &= \sum_{h=-\infty}^{-1} \gamma(h)e^{-2\pi i v h} + \gamma(0) + \sum_{h=1}^{\infty} \gamma(h)e^{-2\pi i v h} \\ f(-v) &= \sum_{h=-\infty}^{-1} \gamma(-h)e^{-2\pi i v(-h)} + \gamma(0) + \sum_{h=1}^{\infty} \gamma(-h)e^{-2\pi i v(-h)} \\ &= \sum_{h=1}^{\infty} \gamma(-h)e^{-2\pi i v(h)} + \gamma(0) + \sum_{h=-\infty}^{-1} \gamma(-h)e^{-2\pi i v(h)} \\ &= f(v) \end{aligned}$$

We know that $\gamma(h) = \gamma(-h)$.

- **Fact 4:** $f(v) \geq 0$
- **Fact 5:** $\gamma(h) = \int_{-1/2}^{1/2} e^{2\pi iv(h)} f(v)dv$. To show that, write

$$\begin{aligned}
\gamma(h) &= \int_{-1/2}^{1/2} e^{2\pi iv(h)} f(v)dv \\
&= \int_{-1/2}^{1/2} e^{2\pi iv(h)} \sum_{h=-\infty}^{\infty} \gamma(h)e^{-2\pi ivh}dv \\
&= \sum_{h=-\infty}^{\infty} \gamma(j) + \int_{-1/2}^{1/2} e^{2\pi iv(j-h)}dv \\
&= \gamma(h) + \sum_{j\neq h} \frac{\gamma(j)}{2\pi i(j-j)}(e^{\pi i(j-h)} - e^{-\pi i(j-h)}) \\
&= \gamma(h) + \sum_{j\neq h} \frac{\gamma(j)\sin(\pi(j-h))}{\pi(j-h)} \\
&= \gamma(h)
\end{aligned}$$

We see that the spectral density is an alternative view of stationary time series. The question now how can we estimate the spectral density ?

Two possibilities, we can replace $\gamma(h)$ in the definition of spectral density $f(v)$ with the sample autocovariance

$$\widehat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-|h|}(x_{t+|h|} - \bar{x})(x_t - \bar{x}) \qquad for - n < h < n$$

The second alternative used to construct a spectral density estimate is the periodogram:

$$I^{(n)}(v) = \frac{1}{2\pi n} \mid \sum_{t=1}^{n} e^{-ivt} X_t \mid^2, \quad v \in [-\pi, \pi]$$

The periodogram as an asymptotically unbiased estimate of the spectral density

$$f(v) = \frac{1}{2\pi}\sum_{u=\infty}^{\infty} \gamma(u)\, e^{-i\,uv} \quad v \in [-\pi, \pi]$$

where $\gamma(u)$ is the autocovariance function.

8.2 Evenly Sampled data

If data is sampled uniformly, we will use the standard procedure to extract the periodic components, but we are dealing with situations where the evenly spaced data cannot be obtained. A common case where the electronic components fails and we have gaps on observations(missing data problems). We can transform Unevenly sampled data to uniform sampled data by filling the gaps with well known methods like Simulation or Prediction. Both methods will be presented and implemented. the interpolation method can be used to fill gaps but is to weak if a lot of consecutive points are missing, and we have to interpolate on missing data points.

8.3 Unevenly Sampled data

Missing values causes gap, and standard method used for Evenly sampled data is not effective, Lomb and Scargle proposed a modified periodogram and is preferable to the classical periodogram , its weights the data on a “per point “ basis instead of on a “per time interval basis”.

Now, we present the work done by Lomb, given a set of n observations $x_t, t = 1, 2, 3, ..., n$, we can set up the model

$$x_{t_i} = a\cos(2\pi f t_i) + b\sin(2\pi t_i) + WN(t_i), \quad WN(t_i) \sim N(0, \sigma^2)$$

a and b are unknown and the frequency f is given. Define the least Square for determining a and b by $LS(a, b)$ and we have:

$$\begin{aligned}
LS(a,b) &= \sum_{i=1}^{n} [x_i - (a\cos(2\pi f t_i) + b\sin(2\pi f t_i))]^2 \\
&= \sum_{i=1}^{n} \left[x_i^2 - 2x_i\left(a\cos(2\pi f t_i) + b\sin(2\pi f t_i)\right)\right] + \\
&\quad \sum_{i=1}^{n} [a\cos(2\pi f t_i) + b\sin(2\pi f t_i)]^2 \\
\Rightarrow \frac{\partial LS(a,b)}{\partial a} &= \sum_{i=1}^{n} \left[-2x_i\cos(2\pi f t_i) - 2a\cos^2(2\pi f t_i) - 2b\cos(2\pi f t_i)\sin(2\pi f t_i)\right] = 0 \\
\Rightarrow \sum_{i=1}^{n} x_i\cos(2\pi f t_i) &= a\sum_{i=1}^{n}\cos^2(2\pi f t_i) + b\sum_{i=1}^{n}[\cos(2\pi f t_i)\sin(2\pi f t_i)]
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \frac{\partial LS(a,b)}{\partial b} &= \sum_{i=1}^{n}\left[-2x_i\sin(2\pi f t_i) - 2b\sin^2(2\pi f t_i) - 2b\cos(2\pi f t_i)\sin(2\pi f t_i)\right] = 0 \\
\Rightarrow \sum_{i=1}^{n} x_i\sin(2\pi f t_i) &= b\sum_{i=1}^{n}\sin^2(2\pi f t_i) + a\sum_{i=1}^{n}\left[\cos(2\pi f t_i)\sin(2\pi f t_i)\right]
\end{aligned}$$

Adopting the notation

$$\begin{aligned}
CC &= \sum_{i=1}^{n}\cos^2(2\pi f t_i), \quad SS = \sum_{i=1}^{n}\sin^2(2\pi f t_i), \\
CS &= \sum_{i=1}^{n}\cos(2\pi f t_i)\sin(2\pi f t_i) \\
XC &= \sum_{i=1}^{n}x_i\cos(2\pi f t_i), \quad XS = \sum_{i=1}^{n}x_i\sin(2\pi f t_i),
\end{aligned}$$

We are led to normal equations:

$$\begin{bmatrix} CC & CS \\ CS & SS \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} XC \\ XS \end{bmatrix}$$

Focusing now on reduction in the sum of square of

$$\begin{aligned}
\Delta R(f) &= \begin{bmatrix} XC & XS \end{bmatrix}\begin{bmatrix} CC & CS \\ CS & SS \end{bmatrix}^{-1}\begin{bmatrix} XC \\ XS \end{bmatrix} \\
&= \begin{bmatrix} XC & XS \end{bmatrix}\begin{bmatrix} \frac{SS}{D} & -\frac{CS}{D} \\ -\frac{CS}{D} & \frac{CC}{D} \end{bmatrix}\begin{bmatrix} XC \\ XS \end{bmatrix}
\end{aligned}$$

Where $D = CC.SS - (CS)^2$. If we can express $\Delta R(f)$ in the form of $A^2 + B^2$, this will ease the statistical description of Least square (LS) and is done by introducing the time delay τ and fitting data to the new model:

$$x_i = \cos\left[\cos(2\pi f(t_i - \tau)\right] + \sin\left[\cos(2\pi f(t_i - \tau)\right]$$

The τ is chosen such that $CS = 0$ and the equation (1) becomes

$$\begin{aligned}
\Delta R(f) &= \begin{bmatrix} XC & XS \end{bmatrix}\begin{bmatrix} \frac{SS}{D} & 0 \\ 0 & -\frac{CC}{D} \end{bmatrix}\begin{bmatrix} XC \\ XS \end{bmatrix} \\
&= \left(\frac{1}{\sqrt{CC}}XC\right)^2 + \left(\frac{1}{\sqrt{SS}}XS\right)^2 \\
&= A^2(f) + B^2(f)
\end{aligned}$$

Now we can make some important remarks:

1. **Remark 1:** Expressing $\Delta R(f)$ in this compact form, gives us the similarity with usual periodogram Formula. In fact the periodogram is just approximation to this exact formula $\Delta R(f)$

2. **Remark 2:** If let the $CS = 0$ for all τ and $SS = CC = \frac{n}{2}$, the equation (2)can be written as

$$AR(f) = \left(\sqrt{\frac{2}{n}}XC\right)^2 + \left(\sqrt{\frac{2}{n}}XS\right)^2$$

3. **Remark 3:** If let $R = \sum_{i=1}^{n} x_i^2$ a normalized spectral function can be defined as follow and the range is between 0 and 1.

$$P(f) = \frac{A^2(f) + B^2(f)}{R}$$

We still have some questions we didn't answer.

- **Q 1: How many test frequencies we need to scan ?**

  Unlike Fourier Analysis, in which the Fourier frequencies are used. We assume that there are M test frequencies, $f_1, f_2, ..., f_M$ and their corresponding angular frequencies $\omega_j = 2\pi f_j, \quad for \quad j = 1, 2, ..., M$. The choice of M depends on the number of independent frequencies, N_0, the number of data points and data spacing. Horne and Baliunas(1986) performed extensive Monte-Carlo simulations to investigate the relationship between M and N_0. They gave a simple least squares formula to estimate the number of independent frequencies N_0 from the number of observations, N, in time series:

$$N_0 \approx -6.362 + 1.193N + 0.00098N^2$$

- **Q 2: Hypothesis testing for Periodicity ?**
  The null hypothesis states that the noise realizations at distinct times are uncorrelated Gaussian distributed, which in Lomb method leads to $P_N(\omega)$ with an exponential probability distribution with unit mean.

$$H0: \quad \epsilon_i \sim exp(1)$$

- **Q 3: False alarms ?**
  If we scan M frequencies, the probability that none has a spectral peak larger that z is $(1-e^z)^M$. The false alarm probability of the null hypothesis is then
  $$P(peak > z) = 1 - (1-e^z)^M$$

- **Q 4: Significance of the peak ?**
  A small value of the false alarm level indicates a highly significant periodic signal.

We have presented material backgrown of Lomb- Scargle Periodogram for Unevenly sampled data, now is time to focus on implementation. The Algorithm is defined as follows:

1. **STEP 1:** Calculate the data's mean and variance of x_t by :
   $$E(x_t) = \frac{1}{n}\sum_{i=1}^{n} x_i \quad ,\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

2. **STEP 2:** Foe each angular frequency $\omega = 2\pi f > 0$ of interest, compute a time offset τ by:
   $$\tan(2\omega\tau) = \frac{\sum \sin(2\omega t_i)}{\sum \cos(2\omega t_i)}$$

3. **STEP 3:** Calculate Lomb-Scargle normalized periodogram(spectral power as a function of ω is defined by:
   $$P_n(\omega) = (2\sigma^2)^{-1}\left(\frac{\left[\sum_i (x_i - \bar{x})\cos(\omega(t_i - \tau))\right]^2}{\sum(\cos^2(\omega(t_i - \tau))} + \frac{\left[\sum_i (x_i - \bar{x})\sin(\omega(t_i - \tau))\right]^2}{\sum(\sin^2(\omega(t_i - \tau))}\right)$$

The implemented algorithm is located under *libs* directory and the name is ***lombPeriodogram.R***

A test program ***testLombPeriodogram.R*** was developed to test the algorithm of Lomb periodogram and the test was performed on one site. The test algorithm steps are defined as follows:

- **STEP 1:** Run the *robustGrubbOutlier.R* to remove and simulate outliers.
- **STEP 2:** Fit a straight line to data and extract the linear trend from data.
- **STEP 3:** Run the the Lomb periodogram algorithm *lombPeriodogram.R*
- **STEP 4:** Two graphs are shown in the next figure, the first graph shows the spectrum. and the second shows the significance of the the peak.

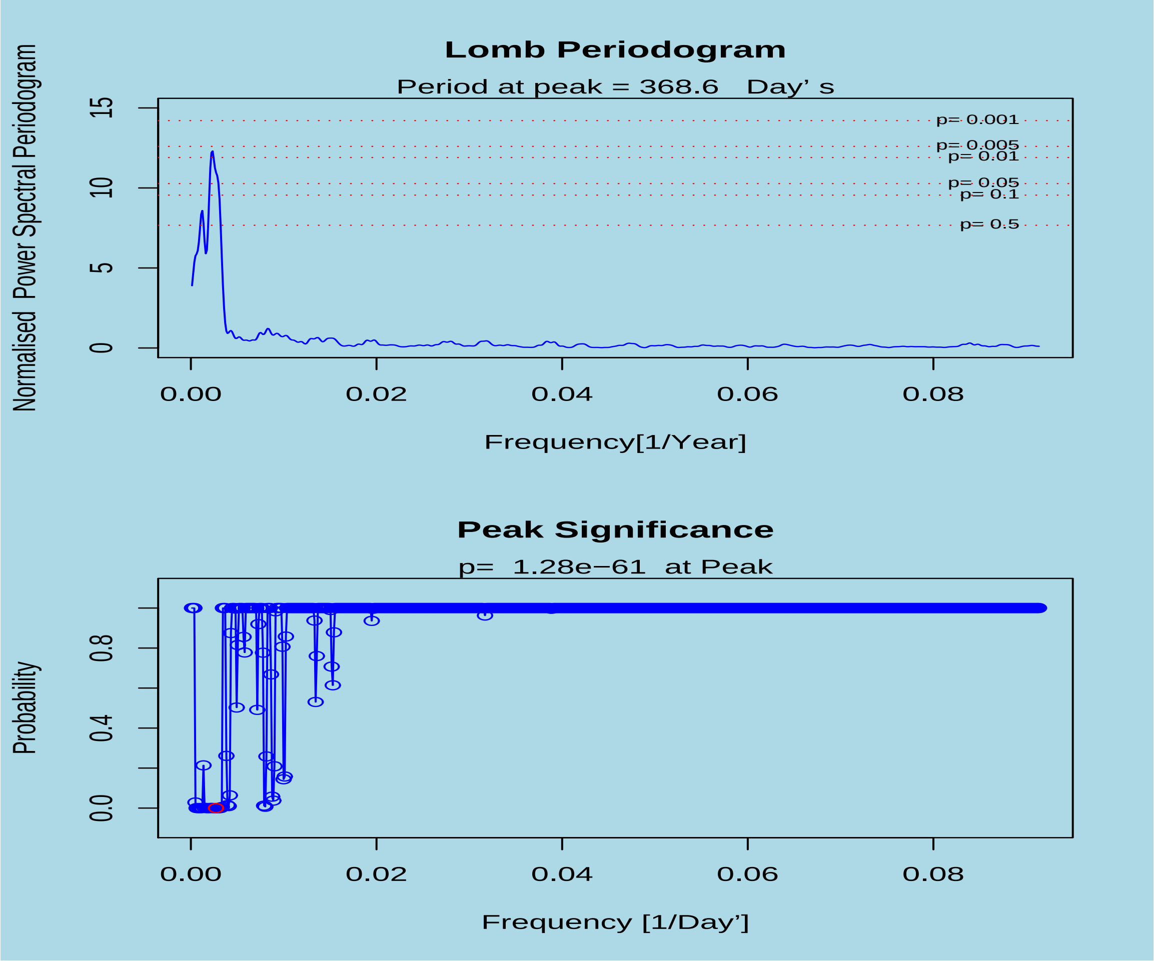


Figure 8.1: *Lomb Periodogram*

Chapter 9

Part II: Correction for Common Mode Error in TS

The first part of this thesis was focused on removing outliers, offsets and irregularities, which are not of tectonic origin. We have used different algorithms to accomplish these goals:

1. **Outliers :** Several algorithms were presented and all of them have the same goal; Detection, Removal and Simulation of outliers without destruction of the structure of the data.

2. **Data Gaps:** Filling gaps by simulation without destruction of the structure of the data, performed by the algorithm **gapWithSim.R**.

3. **Offset Detection :** Estimation and correction of antenna offset, performed by the algorithm **offsetDetection.R**

4. **Spectral Analysis :** Spectral analysis to extract periodic signals, by keeping data as it is **lombPeriodogram.R** or filling gaps by simulation and performing the usual FFT.

The main objective of part II of the thesis will be to focus on more improvements by excluding base stations with poor data and correction for common mode error. This is done by analyzing the residuals of time series of permanent GPS stations, generated by Part I of this thesis, after removing linear trend and periodic components.

The problem is that the time series are of different lengths and epochs with gaps. It is very important to choose data that is common for all sites before starting heavy computation.

The following algorithm will be introduced and implemented to achieve such improvements.

1. **Principal Components :** Principal components, also known as Empirical Orthogonal Functions (EOF), will be introduced and implemented.

2. **Factor Analysis :** If principal components can be used, why can't Factor Analysis be used to achieve the same goal as PCA?

3. **Wdowinski Algorithm :** This algorithm will be introduced, but not implemented.

9.1 Data preparation

The main reason behind most of the major irregularities in the time series is the contribution by stations with data of poor quality. This can significantly influence the whole network or a part of it. We assume that data we have is good and approved by quality check processes by the GIPSY-OASIS-II software.

The only job needed is to choose data with the same start epoch, the same data lengths and possibly without gaps. A preparation job will therefore be carried out before we analyze the data. A program has been written and produces the table presented below. From that table, the first station will be excluded from our analysis. The start period is 2001-05-20, and the end epoch is 2006-01-31.

Site Name	Gaps	Start Epoch	End Epoch	Total Length
aber.res	234	2004-02-02	2006-02-15	511
ales.res	314	2001-05-08	2006-01-31	1416
berg.res	310	2001-05-09	2006-01-31	1419
bodo.res	302	2001-05-08	2006-01-31	1428
hers.res	224	2001-09-04	2006-02-28	1415
hofn.res	236	2001-09-20	2006-02-28	1387
kris.res	513	2001-05-08	2006-01-31	1217
nya1.res	299	2001-05-08	2006-02-28	1459
oslo.res	309	2001-05-08	2006-01-31	1421
pots.res	248	2001-05-08	2006-02-28	1510
stav.res	290	2001-05-08	2006-01-31	1440
trol.res	236	2001-05-08	2006-02-28	1522
tron.res	297	2001-05-08	2006-01-31	1433
vard.res	342	2001-05-08	2006-01-31	1388
wsrt.res	282	2001-05-08	2006-02-28	1476

Table 9.1: *Data Preparation Table*

9.2 Weighted common-mode error filtering

The final step when improving the data is a common-mode filtering (Wdowinski, 1997). It is assumed that signals common to all sites, which are located on different lithosphere blocks, are most likely not of tectonic origin. Such signals can be filtered out of the time series without loosing tectonic signals. The algorithm presented by Wdowinski is as follows:

1. **Detrending:**
   Obtaining the daily residuals for each site by removing the main trend by applying linear regression or weighted linear regression.

2. **Stacking :** Calculation of the daily common-mode residuals using a weighted average.

$$\bar{R}(t) = \frac{\sum_{s=1}^{S} [p_s(t).r_s(t)]}{\sum_{s=1}^{S} p_s(t).}$$

   Where

   $\bar{R}(t)$: *The weighted average residual(common mode) for day t.*

   S : *The number of sites included in analysis stacking procedure.*

   $r_s(t)$: *The residual for the site s on t.*

   $p_s(t)$: *The weight of $r_s(t)$ computed as $r_s(t) = \dfrac{1}{(a_s.\sigma_s(t))^2}$ with $\sigma_s(t)$ equal to the coordinate RMS of day t and a_s as additional quality factor.*

3. **Filtering :**
   Removal of the corresponding common-mode residuals from daily coordinates of each site.

9.3 Principal Component Analysis

A principal component analysis is used to explain the variance-covariance structure of a set of variables through a few linear combinations of these variables. The objectives of the Principal Component Analysis are:

- **Data Reduction**: Data reduction or dimension reduction without significant loss of information We try to describe p dimensional data in as small a number of dimensions (less that p) as possible, while preserving as much as possible of the structure involved
- **Interpretation**: Working in small dimension, say $q < p$ is more attractive that working in p dimensional data.

Algebraically, principal components are particular linear combinations of the p random variables $X_1, X_2, ..., X_p$. Geometrically, these linear combination represent the selection of a new coordinate system obtained by rotating the original system with $X_1, X_2, ..., X_p$ as the coordinate axes. The new axes represent the direction with maximum variability and provide a simple description of the covariance structure.

9.3.1 Theories of principal Components

Let us focus on how to calculate the mean and variance of a linear function. Let X be a any $p-$ dimension random variable and β any constant $p-$dimension vector and let $Y = \beta^T\ X$

The mean value of Y

$$\begin{aligned} E(Y) &= E(\beta^T\ X) \\ &= E\left\{\sum_j \beta_j X_j\right\} \\ &= \sum_j \beta_j E\left\{X\right\} \end{aligned}$$

The variance of Y

If we write

$$\begin{aligned} \sigma_{ij} &= \operatorname{cov}(x_i, x_j) = E\left\{(x_i - E(x_i))(x_j - E(x_j))\right\} \\ \sigma_{ii} &= \operatorname{var}(x_i) = E\left\{(x_i - E(x_i))^2\right\} \end{aligned}$$

$$\begin{aligned}
\Longrightarrow \mathrm{var}(Y) &= E\left\{(\beta^T X - E(\beta^T X)\right\}^2 \\
&= E\left\{\sum_i \beta_i (X_i - E(X_i)\right\}^2 \\
&= E\left\{\sum_i \sum_j \beta_i \beta_j (X_i - E(X_i)(X_j - E(X_j)\right\}^2 \\
&= \sum_i \sum_j \beta_i \beta_j E\left\{(X_i - E(X_i)(X_j - E(X_j)\right\}^2 \\
&= \sum_i \sum_j \beta_i \beta_j \sigma_{ij}
\end{aligned}$$

If we let $\Sigma = (\sigma_{ij})$, then the variance-covariance of Y can be written as: $\mathrm{var}(Y) = \beta^T \; X = \beta^T \Sigma \beta$.

Matrix Theory of Principal Component

We note that if Y_i are uncorrelated and have the variance λ_i, then the variance-covariance matrix of the vector Y take the form:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix}$$

Sometimes it is useful to write:

$$A = (\alpha_1 \quad \alpha_2 \cdots \alpha_p) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_2 & \cdots & a_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix}$$

With this in mind, we can write $Y = A^T X$ and the equations:

$$\begin{aligned}
\alpha_i^T \; \alpha_i &= 1 \\
\alpha_i^T \; \alpha_j &= 1 \quad (i \neq j)
\end{aligned}$$

can be written $A^T A = I$, that is A is an orthogonal matrix. Now let examine the equations:

$$\Sigma \; \alpha_i = \lambda_i \alpha_i$$

can be written

$$\begin{aligned}\Sigma\left(\alpha_1 \quad \alpha_2 \cdots \alpha_p\right) &= \left(\alpha_1\lambda_1 \quad \alpha_2\lambda_2 \cdots \alpha_p\lambda_p\right)\\ \Longrightarrow \Sigma A &= A\Lambda\end{aligned}$$

The principal components in terms of matrix theory, is to find an orthogonal matrix satisfying the above theory, that is,

$$\begin{gathered}\textit{Given } \Sigma, \textit{find } A \textit{ such that} A^T A = I \textit{ and } \Sigma A = A\Lambda\\ \textit{where } \Lambda \textit{ is diagonal with decreasing elements down the main diagonal}\end{gathered}$$

Because of the orthogonality, the pre-multiplication and post-multiplication of $\Sigma A = A\Lambda$ by A^T gives respectively $A^T\Sigma A = \Lambda$ and $\Sigma = A\Lambda A^T$. We have

$$\begin{aligned}I &= AA^T\\ &= \left(\alpha_1 \quad \alpha_2 \cdots \alpha_p\right)\begin{pmatrix}\alpha_1^T\\ \alpha_2^T\\ \vdots\\ \alpha_p^T\end{pmatrix}\\ &= \left(\alpha_1\alpha_1^T \quad \alpha_2\alpha_2^T \cdots \alpha_p\alpha_p^T\right)\end{aligned}$$

and

$$\begin{aligned}A\Lambda A^T &= \left(\alpha_1\lambda_1 \quad \alpha_2\lambda_2 \cdots \alpha_p\lambda_p\right)\begin{pmatrix}\alpha_1^T\\ \alpha_2^T\\ \vdots\\ \alpha_p^T\end{pmatrix}\\ \Longrightarrow \Sigma &= A\Lambda A^T = \left(\lambda\alpha_1\alpha_1^T \quad \lambda\alpha_2\alpha_2^T \cdots \lambda\alpha_p\alpha_p^T\right)\end{aligned}$$

which is called the **Spectral Decomposition of Σ**

9.3.2 Computation of Principal Component

Suppose we have a set of $p-$dimensional random variable $X = [X_1, X_2, ..., X_p]$ with covariance Σ with associated eigenvalues $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_p \geq 0$. Note that we often subtract the mean value from our data.

- **(a)** The first step is to look for a linear function, say $Z_1 = \alpha_1^T X$ of the element of X which has the maximum variance, where α_1 is a vector of p components

$\alpha_{11}, \alpha_{21}, \alpha_{31}, ..., \alpha_{p1}$. Then

$$\begin{aligned} \alpha_1^T\ X &= \alpha_{11}X_1 + \alpha_{21}\ X_2...., \alpha_{p1}\ X_p \\ &= \sum_{j=1}^{p} \alpha_{j1}\ X_j \end{aligned}$$

- **(b)** We look for a linear function $Z_2 = \alpha_2^T\ X$ uncorrelated with with $Z_1 = \alpha_1^T\ X$ which has maximum variance subject to this condition, and so on, so the k'th stage to being uncorrelated with $\alpha_1^T\ X, \alpha_2^T\ X, ..., \alpha_{k-1}^T\ X$. We refer to k'th derived variable $\alpha_k^T X$ as the k'th principal component and we stop the m'th stage when the majority of the variance has been accounted for.

- **(c)** α_1 cannot be taken just to maximize the variance of Z_1. We have to set some requirements, and the easiest one is to choose α_1 to be an eigenvector of unit length.

$$\alpha_1^T \alpha_1 = 1$$

The process needed to determine the first principal component is reduced to:

$$\text{Maximize}\ \ \alpha_1^T \Sigma \alpha_1 \quad \text{subject to} \quad \alpha_1^T \alpha_1 = 1$$

This is a standard constrained maximization problem, often solved by using Lagrange Multiplier, so we can consider

$$F(\alpha_1, \lambda) = \alpha_1^T \Sigma \alpha_1 - \lambda(\alpha_1^T \alpha_1 - 1)$$

Differentiate with respect to scalar α_1, we get

$$\begin{aligned} \partial F / \partial \alpha_1 &= 2\Sigma\alpha_1 - 2\lambda\alpha_1 \\ &= 2(\Sigma - \lambda I)\alpha_1 = 0 \\ \Longrightarrow (\Sigma - \lambda I)\alpha_1 &= 0 \end{aligned}$$

Now this is an equation which frequently turns up in matrix theory, it says that α_1 *is an eigenvector of* Σ *corresponding to the eigenvalue* λ.

Now let us find the variance of $\alpha_1^T\ X$.

$$\begin{aligned} \text{var}(\alpha_1^T\ X) &= \alpha_1^T\ \Sigma\alpha_1 \quad \alpha_1\ is\ an\ eigenvector \\ &= \lambda\alpha_1^T\alpha_1 \quad \alpha_1\ is\ of\ full\ length \\ &= \lambda \end{aligned}$$

As conclusion, we can find α_1 as an eigenvector of unit length corresponding to the largest eigenvalue λ_1 of the variance-covariance matrix Σ.

To find the next principal component, we look for the vector unit α_2 so that

$$\begin{aligned} Z_2 &= \alpha_2^T\ X \ \ and\ Z_1 = \alpha_1\ X\ are\ uncorrelated\ an(subject\ to\ this) \\ Z_2 &= \alpha_2^T\ has\ maximum\ variance. \end{aligned}$$

We see that the variance of $Z_2 = \lambda_2$. This process continues in the same way, if the eigenvalues of Σ are $\lambda_1 > \lambda_2 > ..... > \lambda_p$ and they have corresponding eigenvectors (normalized to be of unit length) $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_p$, the the linear combination $Z_1 = \alpha_1^T\ X, \ \ , Z_2 = \alpha_2^T\ X, ..., Z_p = \alpha_p^T\ X$ from the principal components and have variances $\lambda_1, \lambda_2, ..., \lambda_p$ respectively.

The value of Z_i is frequently referred to as scores or Principal Components Scores. The i'th principal component Z_i account for a proportion of total variance

$$\frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}$$

And the m'th components account for a proportion of total variance

$$\frac{\sum_{j=1}^{m} \lambda_j}{\sum_{j=1}^{p} \lambda_j}$$

- **(d)** In practice, we don't know Σ. It must be estimated by S, the sample variance-covariance matrix, in which case we get the estimates $\hat{\lambda}_i\ of \lambda$ and $\hat{\alpha}_i\ of \alpha_i$. The sample mean, sample variance and the sample covariance estimates for μ_i, σ_{ii} and σ_{ij} are given by:

$$\begin{aligned} \bar{x}_i &= \frac{1}{n}\sum_{r=1}^{n} x_{ri} \quad the\ sample\ mean \\ s_{ii} &= \frac{1}{n-1}\sum_{r=1}^{n}(x_{ri} - \bar{x}_i)^2 \quad the\ sample\ variance \\ s_{ij} &= \frac{1}{n-1}\sum_{r=1}^{n}(x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j) \quad the\ sample\ covariance \end{aligned}$$

- **(e)** So far we have been talking of computation of principal components from variance-covariance matrix Σ. It is quite common to find the principal components from the correlation matrix P, which effectively amounts to normalizing all the variates to have a unit variance before finding principal components. Choosing to do this, we make all variables equally important.

  The principal components obtain from correlation matrix are not the same as those obtained from variance-covariance matrix.

9.3.3 Number of Components to retain

Another question which arises is how many components that are necessary to give an adequate representation of the data. This question is difficult to answer.

- When analysing a correlation matrix, where the sum of the eigenvalues is p, we can disregard all eigenvalues less than 1. This arbitrary rule is a useful rule of thumb but has no theoretical justification (drawback).
- The most used informal *ad hoc* rules of thumb are these:

  1. **Percent of total variance:**
     Include enough components to explain some relatively large percentage of the total variance, between 70 and 90 per cent are usually suggested. This number decreases as n and p increases.
  2. **Plot examination:**
     Cattel(1965) suggests examination of a plot of λ_i against i (the magnitude of an eigenvalue versus its number). To determine the appropriate number of components, we look for an **elbow** in the curve, this point being considered to be where *large* eigenvalues cease and *small* eigenvalues begin. Such a plot is generally known as a *scree* diagram.
  3. **Correlation matrix examination:**
     Exclude those principal components whose eigenvalues are less than average (less than 1). This rule was originally put forward by Kaiser (1958), but Jolliffe (1972) has suggested to exclude components whose associated eigenvalues are less than 0.7.

9.3.4 Numerical Investigations

	East	**Direc.**	**North**	**Direc**	**Height**	**Direc**
site Name	Score	Accum. var	Score	Accum var	score	accum. var
ALES	13434	0.3398	13588	0.3169	47533	0.21435
BODO	8649	0.5586	7796	0.4987	37797	0.3848
KRIS	5186	0.6898	5272	0.6216	30396	0.5219
NYA1	3535	0.7792	5034	0.739	26956	0.6434
OSLO	2891	0.8523	3631	0.8237	24532	0.7541
STAV	2419	0.9135	3344	0.9016	21160	0.8495
TRON	1936	0.9625	2278	0.9548	17937	0.9304
VARD	1483	1	1940	1	15442	1

Table 9.2: *Total Variance Explanation*

This table shows the estimated eigenvalues for each direction (east, north and height), and the proportion of the total population variance explained by the principal components.

Almost 77.92% of the variance is explained by the four principal components for east direction. With five principal components 85.23% is explaned by east, 82.37% for north and 75.41% for height direction.

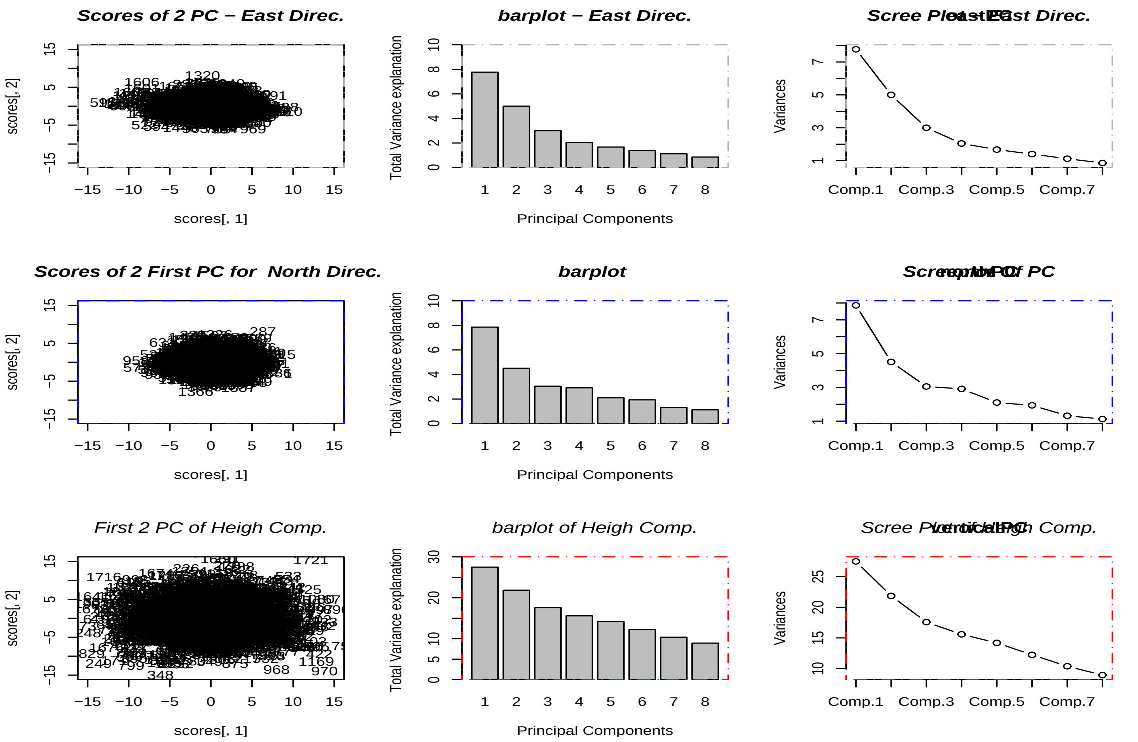


Figure 9.1: *Output from PCA*

This figure shows some graphs. The first row is for the East direction, the second one for North direction and the last one for Height. The first plot shows the scores of PC1 against PC2, the second one the contribution of each PC as barplot and the last one shows the contribution of each PC as Scree plot.

These plots show the information in the previous table graphically.

East Direction

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
ALES	−0.30	−0.50	0.07	0.40	−0.59	0.34	0.04	−0.20
BODO	−0.33	−0.11	0.03	0.50	0.60	−0.10	0.50	−0.07
KRIS	−0.15	−0.29	0.09	−0.37	−0.22	−0.73	0.26	−0.31
NYA1	−0.63	0.29	−0.69	−0.17	−0.09	0.05	−0.03	−0.01
OSLO	−0.16	−0.33	0.16	−0.63	0.34	0.53	0.12	−0.19
STAV	−0.18	−0.41	−0.00	−0.11	−0.00	−0.12	0.01	0.88
TRON	−0.27	−0.24	0.10	0.12	0.32	−0.21	−0.82	−0.17
VARD	−0.50	0.49	0.69	−0.06	−0.16	0.02	0.00	0.12

Table 9.3: *Loading East Direction*

This table shows the eigenvectors calculated from the variance-covariance matrix. After analysing this table, we see that we end up with 2 groups. Group 1, contains NYA1 and VARD with highest score in the first PC. The second group contains the rest of the sites from our network.

North Direction

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
ALES	−0.28	0.27	0.56	0.67	−0.27	0.12	0.00	0.05
BODO	−0.14	0.33	0.09	−0.12	0.19	−0.50	0.69	0.31
KRIS	−0.32	0.28	−0.22	0.01	0.52	0.61	−0.03	0.36
NYA1	0.70	0.62	−0.23	0.23	−0.00	0.04	−0.02	−0.08
OSLO	−0.38	0.33	−0.48	−0.19	−0.68	−0.00	−0.07	0.10
STAV	−0.34	0.21	−0.14	0.06	0.23	0.00	0.18	−0.86
TRON	−0.20	0.31	0.12	−0.09	0.29	−0.52	−0.70	0.05
VARD	0.09	0.32	0.57	−0.66	−0.13	0.31	−0.00	−0.13

Table 9.4: *Eigenvalues- North Direction*

This table shows the eigenvectors calculated from the variance-covariance matrix for the North direction. The same conclusion as for the East Component.

Height Direction

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
ALES	−0.42	0.24	0.68	0.24	0.22	0.43	0.03	−0.12
BODO	−0.16	0.03	−0.02	0.11	0.08	−0.27	0.92	0.15
KRIS	−0.14	0.24	0.16	−0.10	0.08	−0.68	−0.12	−0.64
NYA1	0.72	0.64	0.11	−0.03	0.18	0.09	0.12	0.02
OSLO	−0.29	0.37	−0.05	−0.72	−0.45	0.21	0.13	−0.00
STAV	−0.16	0.24	0.26	−0.01	−0.01	−0.47	−0.28	0.74
TRON	−0.34	0.29	−0.55	−0.09	0.68	0.10	−0.11	0.06
VARD	0.19	−0.44	0.36	−0.63	0.49	−0.02	0.07	0.06

Table 9.5: *Eigenvalues Vertical Direction*

This table shows the eigenvectors calculated from the variance-covariance matrix. We end up with the same conclusion as for the North and East directions.

9.3.5 Factor Loadings Of PC

Until now, the principal components have been estimated either from the variance-covariance matrix or from the correlation matrix. If we define the scaled vectors $\beta_i = \lambda^{\frac{1}{2}}\alpha_i$ where α_i is the eigenvector, clearly:

$$\begin{aligned}
\beta_i^T\beta_i &= (\lambda^{\frac{1}{2}}\alpha_i)^T\lambda^{\frac{1}{2}}\alpha_i \\
&= \alpha_i^T\lambda^{\frac{1}{2}}\lambda^{\frac{1}{2}}\alpha_i \\
&= \alpha_i^T\lambda\alpha_i \\
&= \lambda \\
&= \\
\beta_i^T\beta_i &= 0 \quad (i \neq j)
\end{aligned}$$

We define a new matrix L by:

$$\begin{aligned}
L &= (\beta_1 \quad \beta_2 \qquad\quad \beta_p) \\
&= A\Lambda^{\frac{1}{2}} \quad (\Lambda^{\frac{1}{2}} \quad is\ sym.) \\
\Longrightarrow \Sigma &= A\Lambda A^T \\
&= A\Lambda^{\frac{1}{2}}(A\Lambda^{\frac{1}{2}})^T \\
&= L\ L^T
\end{aligned}$$

The elements of L are computed so that the coefficients of the more important components are scaled in a manner that makes them larger than those of the less important components. We call the elements of L *components loading.* Some values in the components loading table are blank. This indicates that they are very small. The table is displayed at the end of the chapter, so it can be compared to Factor Analysis.

9.4 Factor Analysis

In factor analysis, we try to explain the covariances or correlations between a set of observed variables $X = (X_1, X_2, ..., X_p)^T$ in terms of a smaller set of unobservable latent variables $F = (f_1, f_2, ..., f_k)^T$, and we try to make k small as possible. We assume the mean is extracted from our observations, and can write:

$$\begin{aligned}
X_1 &= \lambda_{11}f_1 + \lambda_{12}f_2 + .. + \lambda_{1k}f_k + \epsilon_1 = Y_1 + \epsilon_1 \\
X_2 &= \lambda_{21}f_1 + \lambda_{22}f_2 + .. + \lambda_{2k}f_k + \epsilon_2 = Y_2 + \epsilon_2 \\
X_3 &= \lambda_{31}f_1 + \lambda_{32}f_2 + .. + \lambda_{3k}f_k + \epsilon_3 = Y_3 + \epsilon_3 \\
\vdots &= \vdots \\
X_p &= \lambda_{p1}f_1 + \lambda_{p2}f_2 + .. + \lambda_{pk}f_k + \epsilon_p = Y_p + \epsilon_p
\end{aligned}$$

Or in vector notation: $X = \Lambda F + \Psi = Y + \Psi$

$$\begin{aligned}
f_1, f_2, ..., f_k &: \quad Common\ Factors \\
\epsilon_1, \epsilon_2, ..., \epsilon_p &: \quad Specific\ Factors \\
\lambda_{ij} &: \quad Factor\ Loading
\end{aligned}$$

Since we are interested in covariances or correlations, some assumptions about unobserved random variables f_i and ϵ_i are:

- **A1:** The error terms (specific factors) ϵ_i are independent, and have a constant variance. This means that $E(\epsilon_i) = 0$ and $\mathrm{var}(\epsilon_i) = \sigma_i^2$
- **A2:** The unobserved factors f_i are independent, and have unity variance. This means that $E(f_i) = 0$ and $\mathrm{var}(f_i) = 1$
- **A3:** The error terms(specific factors) ϵ_i and f_i are independent.

Let us compute the variance of X_i. Remember that each observed variable X_i is linear combination of independent factors f_i and error terms ϵ_i. We have:

$$\begin{aligned}
\mathrm{var}(X_i) &= \mathrm{var}\left\{\lambda_{i1}f_1 + \lambda_{i2}f_2 + .. + \lambda_{ik}f_k + \epsilon_1\right\} \\
&= \lambda_{i1}^2\mathrm{var}(f_1) + ... + \lambda_{ik}^2\mathrm{var}(f_k) + \mathrm{var}(\epsilon) \\
&= \lambda_{i1}^2 + \lambda_{21}^2 + ... + \lambda_{k1}^2 + \sigma_i^2 \\
&= \sum_{j=1}^{k} \lambda_{ij} + \sigma_i^2
\end{aligned}$$

We see that the variance of the observed variable X_i is composed of two parts:

- **Communality:** This part is explained by the common factors f_i
- **Specific variance or unique variance:** This part is the variance of X_i that is not accounted by common factors

In our analysis of CGPS time series, we are interested in correcting for common variation at all sites under investigation. In this case, the first part is more interesting. If we are interested in identifying the dominant noise type at the site, the second part would be more interesting.

9.4.1 Covariance Structure of Orthogonal Factor model

In this section, we try to determine the covariance of our model, the Orthogonal Factor Model with k common factors. Our starting point is $X = \Lambda F + \Psi$

$$\begin{aligned}
XX^T &= (\Lambda F + \epsilon)(\Lambda F + \epsilon)^T \\
&= (\Lambda F + \epsilon)((\Lambda F)^T + \epsilon^T) \\
&= \Lambda F(\Lambda F)^T + \epsilon(\Lambda F)^T + \Lambda F^T + \epsilon\epsilon^T
\end{aligned}$$

$$\begin{aligned}
\Leftrightarrow \Sigma &= \mathrm{cov}(X) \\
&= \mathrm{E}(XX^T) \\
&= \mathrm{E}\left\{\Lambda F(\Lambda F)^T + \epsilon(\Lambda F)^T + \Lambda F\epsilon^T + \epsilon\epsilon^T\right\} \\
&= \Lambda \mathrm{E}(FF^T)\Lambda^T + \mathrm{E}(\epsilon F^T)\Lambda^T + \Lambda\mathrm{E}(F\epsilon^T) + \epsilon\epsilon^{\mathrm{T}} \\
&= \Lambda\Lambda^T + \Psi
\end{aligned}$$

where the Ψ is a diagonal matrix with $\psi_1 = \sigma_i^2, ..., \psi_p = \sigma_p 2$ down the main diagonal.

If let $h_i^2 = \sum_{j=1}^{k} \{\lambda_{ij}\}^2$, then the variance $\sigma_i^2 = \psi_i =$

9.4.2 Parameter estimations

This subsection focuses on how to estimate parameters in our model.

$$\Sigma = \Lambda\Lambda^T + \Psi$$

The sample covariance S can replace Σ. S is a $p \times p$ matrix and can be determined by $\frac{1}{2}p(p-1)$ quantities. We can estimate Λ by $\widetilde{\Lambda}$; this requires pk quantities. We can estimate $\widetilde{\Psi}$ for Ψ; this requires p quantities.

Now, the total of $p + pk$ quantities on the right side are constrained by requiring $\frac{1}{2}k(k-1)$. Quantities therefore need to be estimated from S is $p(1+k) - \frac{1}{2}k(k-1)$.

If we let d be the difference between the number of available quantities (to construct the estimates from) and the number of quantities to be estimated, then we end up with:

$$\begin{aligned} d &= \frac{1}{2}p(p-1) - \left[p(1+k) - \frac{1}{2}k(k-1)\right] \\ &= \frac{1}{2}\left[(p-k)^2 - (p+k)\right] \end{aligned}$$

Depending on the value of d, there are three alternatives:

- **Alternative 1:** If $d < 0$, fewer equations than parameters to be estimated. Infinite number of solutions.
- **Alternative 2:** If $d = 0$, there is one unique solution.
- **Alternative 3:** If $d > 0$, this is the most important alternative.

Our sample model becomes:

$$S = \widetilde{\Lambda}\widetilde{\Lambda}^T + \widetilde{\Psi}$$

9.4.3 Numerical Investigation

Let us first define the value of d, which must be positive. We have $p = 8$ for $k \in \{1,2,3,4,5\}$, and found the value of $d \in \{18, 13, 7, 2, -2\}$. For our analysis, we chose $k = 3$.

The most widely used methods used to determine a first set of loading are the Principal Factor Analysis (PFA) and Maximum Likelihood (MLE). A computer program was implemented for estimating parameters, using both methods.

The PFA seeks values of loadings that bring the estimate of the total communality closer to the total of the observed variances, the covariances are ignored. This coincide loadings obtained in previous section (PC).

$$\begin{aligned} \Sigma &= \Lambda\Lambda^T \\ &= \lambda_1 e_1 e_1^T + ... + \lambda_p e_p e_p^T \end{aligned}$$

The MLE make assumptions about normality of f_i and ϵ. The observation $X_i = \Lambda f_i + \epsilon_i$ is normal distributed.

If the factor solution does not reveal the structure of the loading, we can apply rotation to find another set of loadings that fit our observations equally and interpretable. The two most widely used methods are:

1. **varimax:** Try to find the rotated loadings that maximize the variance of squared loadings for each factor. The objective is to make some of these loadings as large as possible, and the rest as small as possible.

2. **quartimax:** Try to maximize the variance of squared loadings for each variable. This tends to produce factors with high loadings for each variable.

We applied the **varimax** rotation to MLE when handling the **height direction**.

	Principale	**Factor**	**Analysis**		**Maximum**	**Likel.**	**Estimator**	
Site Name	$\widehat{f_1}$	$\widehat{f_2}$	$\widehat{\psi_i} = 1 - \widehat{h_i^2}$	$\widehat{h_i^2}$	$\widehat{f_1}$	$\widehat{f_2}$	$\widehat{\psi_i} = 1 - \widehat{h_i^2}$	$\widehat{h_i^2}$
ALES	-0.5371	0.1018	0.29887	0.8687	0.5311	-0.0467	0.284	0.7156
BODO	-0.4097	-0.2965	0.25585	0.7444	0.3653	0.29587	0.2210	0.77897
KRIS	-0.4688	0.1663	0.24746	0.7599	0.4402	-0.0920	0.2022	0.79771
NYA1	-0.2317	-0.4700	0.27465	0.7899	0.1860	0.41938	0.2105	0.7894
OSLO	-0.4478	0.1606	0.22638	0.7688	0.411	-0.0798	0.17550	0.824
STAV	-0.5778	0.2154	0.3802	0.6671	0.6491	-0.2071	0.46427	0.53572
TRON	-0.5443	-0.0748	0.3018	0.7431	0.5195	0.13471	0.2880	0.7119
VARD	-0.1114	-0.5201	0.2829	0.7999	0.0604	0.499	0.25362	0.7463

Table 9.6: *Factor Analysis- East Direction*

	Principale	**Factor**	**Analysis**		**Maximum**	**Likel.**	**Estimator**	
Site Name	$\widehat{f_1}$	$\widehat{f_2}$	$\widehat{\psi_i} = 1 - \widehat{h_i^2}$	$\widehat{h_i^2}$	$\widehat{f_1}$	$\widehat{f_2}$	$\widehat{\psi_i} = 1 - \widehat{h_i^2}$	$\widehat{h_i^2}$
ALES	-0.3585	-0.05213	0.1312	0.90504	0.30479	0.04526	0.09495	0.9050
BODO	-0.3739	-0.34024	0.25559	0.72594	0.36879	0.37154	0.274053	0.7259
KRIS	-0.4804	0.09626	0.2401	0.7729	0.46315	-0.11186	0.2270	0.77296
NYA1	0.35537	-0.2894	0.2101	0.85064	-0.31863	0.21869	0.149357	0.850
OSLO	-0.4745	0.07669	0.2311	0.79323	0.449	-0.0676	0.2067	0.7932
STAV	-0.5649	0.11713	0.33289	0.5872	0.6232	-0.15578	0.4127	0.587
TRON	-0.4297	-0.26859	0.25682	0.7501	0.410837	0.28467	0.2498	0.7501
VARD	0.02160	-0.44676	0.20006	0.9226	-0.02422	0.27704	0.07734	0.9226

Table 9.7: *Factor Analysis- North Direction*

	Principale	**Factor**	**Analysis**		**Maximum**	**Likel.**	**Estimator**	
Site Name	$\widehat{f_1}$	$\widehat{f_2}$	$\widehat{\psi_i} = 1 - \widehat{h_i^2}$	$\widehat{h_i^2}$	$\widehat{f_1}$	$\widehat{f_2}$	$\widehat{\psi_i} = 1 - \widehat{h_i^2}$	$\widehat{h_i^2}$
ALES	-0.3249	0.00353	0.10559	0.89440	0.24434	0.1585	0.08483	0.9151
BODO	-0.1942	0.20506	0.079773	0.9202	0.06088	0.1750	0.0343	0.9656
KRIS	-0.3160	-0.1905	0.13618	0.86381	0.29335	0.102	0.09643	0.90356
NYA1	0.1731	-0.2982	0.11892	0.881077	-0.0213	-0.2025	0.0414	0.95852
OSLO	-0.2835	-0.0358	0.0816	0.91832	0.1597	0.1951	0.063	0.936
STAV	-0.3301	-0.2132	0.15451	0.8454	0.53626	-0.0061	0.2876	0.7123
TRON	-0.2612	0.17891	0.1002	0.8997	0.0383	0.3838	0.14877	0.8512
VARD	0.2130	-0.0068	0.0454	0.9545	-0.093	-0.1753	0.03938	0.96061

Table 9.8: *Factor Analysis- Vertical Direction*

		East	**Direction**		**North**	**Direction**		**Height**	**Direction**
	L1	L2	L3	L1	L2	L3	L1	L2	L3
ALES	-0.300	-0.493	-0.092	-0.283	-0.202	0.875	-0.344	-0.295	0.649
BODO	-0.343	-0.090	-0.071	0.124	-0.344		0.178		
KRIS	-0.160	-0.296	-0.076	-0.328	-0.259	-0.124	-0.344	-0.295	0.649
NYA1	-0.635	0.267	0.699	0.699	-0.601	0.135	0.794	-0.561	0.040
OSLO	-0.151	-0.344	-0.160	-0.392	-0.365	-0.372	-0.245	-0.420	-0.049
STAV	-0.188	-0.405		-0.326	-0.215	-0.143	-0.280	0.264	
TRON	-0.269	-0.224	-0.102	-0.191	-0.330	-0.309	-0.294	-0.555	
VARD	-0.482	0.508	-0.676	0.115	-0.354	-0.248	0.181	0.430	0.383

Table 9.9: *Factor Loadings from Principale Comp.*

Chapter 10

PART 3: CGPS Filter Implementation and parameter estimations

10.1 Introduction

The main objective of this chapter is to implement a filter to our CGPS time series and parameters estimation. Modeling proceeds by a series of well-defined steps, variance stabilization, fitting an ARMA model by following Box-Jenkins approach (model identification, estimation, diagnostic checking and iteration by modifying the selected model) and extraction of periodic signal.

The Last step is to evaluate the randomness of the model, and to predict the future value of chosen model.

Prediction will be used as quality control, if the observed observations reach upper or lower bound of prediction, an alarm shall be generated.

We start by giving some definitions related to time series and then starting our framework to identify our model

Definition 10.1.1. *(Strict Stationarity):*
We say that the time series $x(t) : t \in Z$ is strictly stationary if for any $k > 0$ and any $\{t_1, t_2, ..., t_k \in Z\}$, the distribution of $\{x_{t_1}, ..., x_{t_k}\}$ is the same as that for $x_{t_{1+h}}, ..., x_{t_{k+h}}$ for every value of h

Definition 10.1.2. *(Weak Stationarity):*

We say that the time series $x(t) : t \in Z$ is strictly stationary if for any $k > 0$ and any $\{t_1, t_2, ..., t_k \in Z\}$, the distribution of $\{x_{t_1}, ..., x_{t_k}\}$ is the same as that for $x_{t_{1+h}}, ..., x_{t_{k+h}}$ for every value of h

Definition 10.1.3. *(Autocorrelation):*
We say that the process $w(t)$ is weakly stationary, if the following requirements are satisfied:

1. *The expectation $E(x_t) = k$ is constant and independent of time t*
2. *The variance is constant and independent of time t*
3. *The autocovarince function $\text{cov}(x_t, x_{t+h}) = E(x_t, x_{t+h})$ independent of time t and depend only on separation lag h*

Definition 10.1.4. *(Cross correlation):*
We say that the process $x(t)$ is weakly stationary ,if the following requirements are satisfied:

1. *The expectation $E(x_t) = k$ is constant and independent of time t*
2. *The variance is constant and independent of time t*
3. *The autocovarince function $\text{cov}(x_t, x_{t+h}) = E(x_t, x_{t+h})$ independent of time t and depend only on separation lag h*

Definition 10.1.5. *(White Noise Process):*
We say that $w(t)$ is a white noise, if the following requirements are satisfied:

1. *The expectation $E(w_t) = 0$*
2. *The variance is constant and independent of time t*
3. *The autocovarince function is independent of time t*

Definition 10.1.6. *(Random Walk Process):*
The Random Walk is defined by $x_t = x_{t-1} + w_t$. The process is not stationary:

1. *The expectation $E(w_t) = 0$*
2. *The variance is dependent of time t,* $\mathrm{var(x_t)0\gamma t}$

Definition 10.1.7. *Autoregressive process - AR(p):*
A process $\{x_t\}_{t=1}^{N}$ is said to be an AR(p) if X_t satisfy

$$\begin{aligned} X_t &= \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_q X_{t_q} + W_t \\ &= \sum_{i=1}^{p} \theta_i X_{t_i} + W_t \end{aligned}$$

Where $W_t \sim N(0, \sigma_w^2)$, white noise and ϕ_i are constants.

Definition 10.1.8. *Moving Average Process- MA(q):*
A process $\{X_t\}_{t=1}^N$ satisfying

$$\begin{aligned} X_t &= \theta_1 W_{t-1} + \theta_2 W_{t-2} + \cdots + \theta_q W_{t-q} + W_t \\ &= \sum_{i=1}^{q} \theta_i W_{t_i} \end{aligned}$$

Where $\theta_0 = 1$ and $W_t \sim N(0, \sigma_w^2)$.

Definition 10.1.9. *Autoregressive moving average process ARMA(p,q):*
A process $\{x_t\}_{t=1}^N$ is said to be an $ARMA(p,q)$ if

$$X_t = \sum_{i=1}^{q} \theta_i W_{t_i} + \sum_{i=1}^{p} \phi_i X_{t_i}$$

Where W_t is a white noise and the $\{\phi_i\}$ and $\{\theta_i\}$ are real constants.

Definition 10.1.10. *Invertibility of an ARMA(p,q) process*
An $ARMA(p,q)$ process is invertible only when the root of $\theta(z)$ lies outside the unit circle. The coefficients π_i of $\pi(B) = \sum_{i=0}^{\infty} \pi_i B^i$ can be determined by solving

$$\begin{aligned} \pi(z) &= \sum_{i=1}^{\infty} \pi_i z^i \\ &= \phi(z)/\theta(z) \qquad |z| \leq 1 \end{aligned}$$

Definition 10.1.11. *Causality of an ARMA(p,Q) Process:*
An $ARMA(p,q)$ process is causal only when the root of $\phi(z)$ lies outside the unit circle. That is, $\phi(z) = 0$ only when $|z| > 0$. The coefficients of the linear process $X_t = sum_{i=1}^{\infty} \psi_i W_{t_i} >= \psi(B) W_t$ can be determined by solving

$$\begin{aligned} \psi(z) &= \sum_{i=1}^{\infty} \psi_i z^i \\ &= \theta(z)/\phi(z) \qquad |z| \leq 1 \end{aligned}$$

10.2 Power Low Processes

Definition 10.2.1. *(Power Law Process):*
We say that the process x_t is a power law process, if it's power-law spectrum can be written as $P(f) = P_0 \left(f/f_0\right)^v$, where P_0 and f_0 are normalisation constants and v is the spectral index, that's often falls in the range $-3 \leq v \leq 0$ and f is spatial or temporal frequency.

1. *$v = 0$: White Noise*

2. *$v = -1$: Flicker Noise*

3. *$v = -2$: Random Walk Noise (Brownian Motion)*

10.3 Model Identification

10.3.1 Variance stabilization

The first step in any data analyses is to view data, this job is done in the previous chapters and we know that our time series is not stationary(containing trends and periodic components), we can perform some transformation to achieve stationarity. If we let our time series be presented by $\{x_t\}_{t=1}^{N}$, then $y_t = x_t - x_{t-1}$. Note that the differenced data y_t contains one less data points than the original x_t.

We can repeat the difference operator more than once until the stationarity achieved. Usually one difference is sufficient in our case.

Other methods exists, like taking logarithm or square root to our time series may stabilise the the variance. For negative data, we can add an offset to make all data positive before applying the transformation.

The Box-Cox transformation can be applied to achieve stationarity.

10.3.2 Model Selection

The next step is to identify the model order p and q, and are estimated by applying the Akaike Information Creterion.

$$\text{AIC} = \ln(\hat{\sigma}^2) + \frac{k}{n}$$

The coefficients of an $AR(p)$ are estimated by solving the Yule-Walker equations (see chapter 7), and $MA(q)$ by innovation algorithm. In general case of an $ARMA(p, q)$ process, the coefficients $\phi_1, .., \phi_p, \theta_1, ..., \theta_q$ can be estimated by maximum likelihood.

The model chosen must have the smallest AIC. From graphs of row 2 and 3, we see that the PACF decays and the ACF cutoff after one lag, this is the signature of ACF.As conclusion we must fit MA(1) model.

10.3.3 Extraction Of Periodic components

After fitting an $MA(1)$ to our differenced data, the next step is identify the periodic components. This step is carried out by computing the periodogram of all directions(East, North, and Height). We apply the least square method to extract periodic component(as we did in *part1*). The signal representing the annual and inter annual variation can be modeled as

$$\begin{aligned} S_j(t_i) &= \sum_{j=1}^{2} A_j \cos(\omega_j t_i - \phi_j) \\ &= \sum_{j=1}^{2} \{A_j \sin(\phi_j)\sin(\omega_j t_i) + A_j \cos(\phi_j)\cos(\omega_j t_i)\} \end{aligned}$$

where $\omega_j = 2\pi f_J$ is the angular frequency, A_j is the amplitude, ϕ_j is the phase. J is the number of frequencies(in our case is 2).

If let $SS_j = A_j \sin(\phi_j)$ and $CC_j = A_j \cos(\phi_j)$, the amplitude is given by $A_j = \{SS_J^2 + cc_J^2\}^{1/2}$. This means we must extract both frequencies from our signal.

Smoothing is very important operation in data analysis, especially when analysing the spectrum (periodogram). Periodogram fluctuates estimate of the spectrum with high variance, for a stable estimate, the periodogram must be smoothed.

In many text books, they recommend the Daniell window as smoothing filter for generating an stable estimate from the periodogram. The modified Daniell window of size K, is defined as:

$$\begin{aligned} W_i &= \frac{1}{2(k-1)} & i = 1 \text{ or } i = k \\ &= \frac{1}{(k-1)} & \text{otherwise} \end{aligned}$$

The fourth row in our graph shows that exists annual and interannual variations, and must be extracted from our time series.

10.4 Noise Identification

10.4.1 Diagnostic Checking

Typically, we judge the goodness of fit of a statistical model to a set of data by comparing the observed values with corresponding predicted values obtained from the fitted model.

The sample ACF of the residuals

The sample autocorrelations of an iid sequence with finite variance are approximately iid with distribution $N(1, 1/n$. In our case all noises are white noise.

Randomness of the residuals

The portmanteau test statistic can be used to check for compability of the residuals with white noise.

Other tests with accompanying good explanations exists (see Brockweel and Davis).

10.4.2 Integer Spectral Indices

The site coordinate component can be modelled by an initial value of the abscissa intercept x_0 and the velocity r such that

$$x(t_i) = x_0 + rt_i + \epsilon_x(t)$$

or in matrix form

$$\begin{aligned} X &= A\mathbf{x}(t) + \epsilon_x(t) \\ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} &= \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{bmatrix} \begin{bmatrix} x_0 \\ r \end{bmatrix} + \epsilon_x(t) \end{aligned}$$

Now, we assume that the error term is a linear combination of a sequence of independent unit variance uncorrelated random variables α_j and a sequence of temporally correlated random variables β_j

$$\epsilon_x(t) = a\alpha_{\mathrm{j}}(t) + b_\kappa \beta_{\mathrm{j}}(t)$$

The scaling coefficients a and b are the magnitude of white noise and colored noise.

If we assume that the colored noise is Random walk process, we can express the covariance of the measurements x_i as

$$C_x(t) = a^2 I + b^2 J_2(t)$$

Where I is the identity matrix of dimension N and J_2 is the random walk covariance matrix

$$J_2(t) \quad = \quad f_s^{-1} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & \cdots & N \end{bmatrix}$$

Where the sampling period in yr^{-1}, $f_s = (N-1)/T$, the T is the total observation span, and N is the number of observations .

Fitting a straight line to the time series, i.e.

$$\hat{x}(t_i) = \hat{x}_0 + \hat{r}\ t_i$$

by weighted linear regression(least squares) yields estimates for x_0 and r

$$\hat{y} = \left[A^T C_x^{-1} A\right]^{-1} A^T C_x^{-1} x$$

Where

$$x = (x_1, ..., x_N)^T$$

$$A = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_N \end{bmatrix}$$

$$y = (x_0, r)^T$$

Now is time to give all covariance matrices of underlying integer spectral models, white noise(WN), random walk noise(RWN), and flicker noise(FN)

- **White Noise ($k = 0$:** In this case, $b_k = 0$, and

$$C_x = a^2 \ I$$

That is, the covariance is diagonal and independent of time.

- **Random Walk Noise ($k = 2$:** In this case, $a = 0$, and

$$C_x(t) = J_2(t) = f_s^{-1} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & \cdots & N \end{bmatrix} \qquad f_s = (N-1)/T$$

- **Flicker Noise ($k = 1$:** In this case, $a = 0$, and

$$C_x = J_1 = \left[\left(\frac{3}{4}\right)^2 \frac{24 \ I - J_0}{12} \right]_{NxN}$$

Where the elements (i, k) of the symmetric matrix J_0 is given by

$$J_0 = \begin{cases} 0 & i = k \\ \frac{\log(k-i)}{\log 2} + 2 & i < k \end{cases}$$

10.5 Prediction from CGPS Filter

We are satisfy with our model, we like to predict the future values for our time series(east, north and height direction). The last figures shows the prediction values from our models with upper and lower bounds.

The design blocks for our filter are presented in the next figure. It's composed of 4 mains blocks, namely variance stabilizer, extraction of periodic components, the AR block, and the MA block.

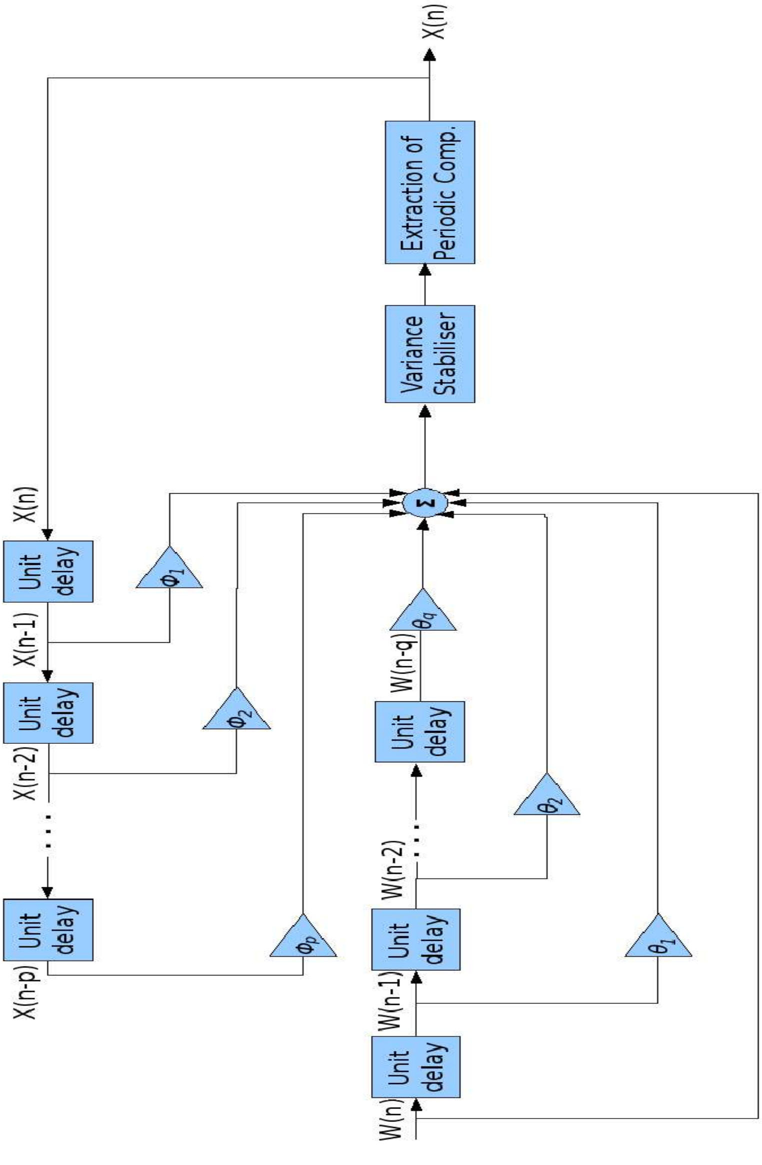


Figure 10.1: *CGPS Filter*

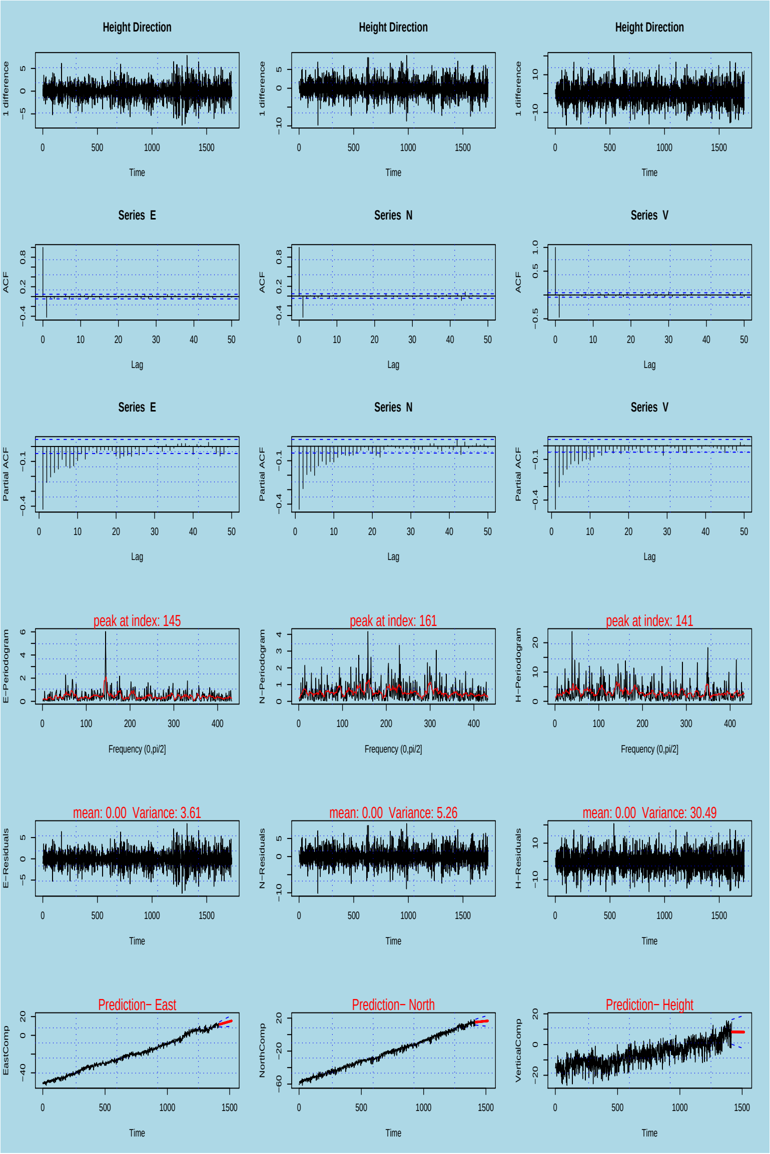


Figure 10.2: *Output from CGPS Filter*

Chapter 11

Summary and Conclusion

11.1 Part 1

Part 1 *of this thesis explained how we prepare the data by removing outliers and offsets and filling gaps. The seasonal signals are extracted, so that we finally can get a better estimate of the linear trend. We are satisfied with the results generated by the different algorithms.*

Our time series are modelled by an additive model $X_i = T_i + S_i + N_i$, where the T_i represents the trend component, S_i represents the seasonal component and N_i represents the noise component. We are primarily interested in the trend component, which estimates the velocity of the site under investigation. We give a summary of our actions here.

- **STEP1:**
  We implemented different algorithms to fix gaps, detect and correct offsets and handle outliers. We then extracted the linear trend from our data $Y_i = X_i - T_i = S_i + N_i$.

- **STEP2:**
  We analyzed the spectral of Y_i by Lomb periodogram. We were interested in the frequency of the peak in the Lomb periodogram. The next step is to estimate the coefficients A_f and B_f in S_i by the least square method.

  We use the normal equation from the Lomb method (Least Square, see ch. 8):

$$\begin{bmatrix} CC & 0 \\ 0 & SS \end{bmatrix} \begin{bmatrix} A_f \\ B_f \end{bmatrix} = \begin{bmatrix} XC \\ XS \end{bmatrix}$$

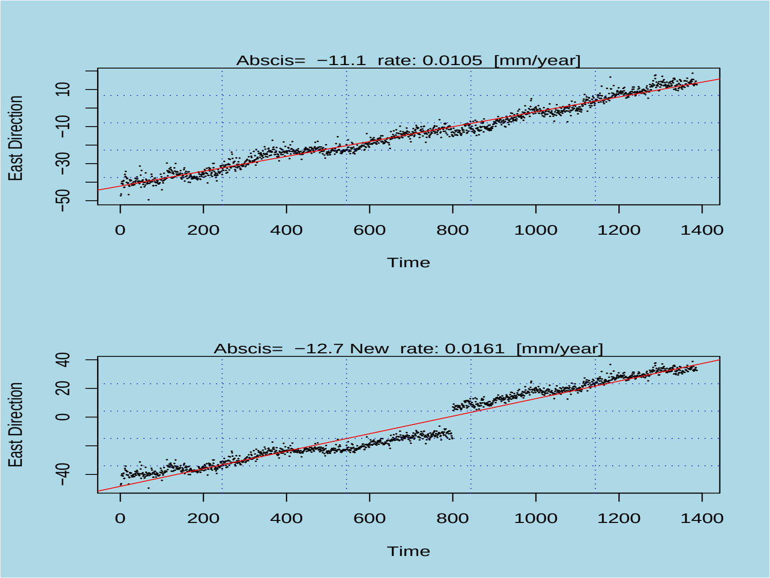


Figure 11.1: *Impact of jumps in TS*

$$\Longrightarrow \begin{bmatrix} A_f \\ B_f \end{bmatrix} = \begin{bmatrix} CC & 0 \\ 0 & SS \end{bmatrix}^{-1} \begin{bmatrix} XC \\ XS \end{bmatrix}$$

- **STEP3:**
  To get a better estimate of the site velocity, we need to extract the component S_i from the original time series $Z_i = X_i - S_i = T'_i + N'_i$. We then estimate the linear trend T'_i again, but this time we have to be more accurate than when we calculated from T_i. The main reason is that we have removed the seasonal component (improvement).

- **STEP4:**
  After the extraction of the linear trend T'_i, the last step is to remove the noise component. This is done by first identifying the type of noise at the site; this will be introduced in chapter **10**.

The next table shows the slope for each direction without extracting the seasonal component, and the second one with extraction of the seasonal components.

We see that the abscissa and the slope are changed in the second table. This shows the improvement by extracting S_i component.

To show the impact of outliers, gaps and offsets on site velocity estimations, I created an artificial offset of size $20mm$ at the middle of a time serie . The site is randomly chosen. I calculated the abscissa and the slope in both cases. When looking at fig.11.1, we see that the presence of offsets in our time series influences the site velocity heavily.

11.2 Part 2

***Part 2** of this thesis explained how we determine common fluctuations in our network. The removal of this common variance is not included in this thesis. Two methods are implemented: Factor Analysis and Principal Component. We conclude that Factor Analysis is the preferred method.*

- **East Direction:**
  In the table for the **East Direction** on the next pages, the values of the first loading component obtained from PFA are uniform. This means that the common global variation over the whole network is almost constant. The same information is obtained from MLE, with opposite sign.

  The second loading component bring us interesting results: we have two groups; the first group contains the sites ALES,KRIS,OSLO,STAV, and the second group BODO,NYA1,TRON,VAR. This component tries to explain the regional common variation, and is different for the southern and the northern Norway. The observation shows that Trondheim separates the two regions.

- **North Direction:**
  In the table for the **North Direction** on the next pages, the values of the first loading component obtained from PFA, shows that we end up with two groups. The first group is the site of NYA1 and VARD, while the second group contains the rest of network. Each group has a common global variation. The same information is obtained from MLE, but with opposite sign of loading.

  The second loading component bring the same results as for **East Direction**

- **Height Direction:**
  In the table for the **Height Direction** on the next pages, the values of the first loading component obtained from PFA is explained as in **East Direction**. The

same information is obtained from MLE, with opposite sign.

The second loading component bring the same results as for **East Direction** and **North Direction:**.

- **Choose between PFA or MLE:**
  We examined the residuals of $\widehat{R} = S - \widehat{\Lambda}\widehat{\Lambda}^T - \widehat{\Psi}$, and found that the MLE estimate does a better job of producing S than principal components (PFA). The entries of $\widehat{R}$ are almost zero compared to PFA.

- **Principal Component or Factor Analysis** :
  Since our purpose is to identify the latent variables that are contributing to the common fluctuations in our network, the Factor Analysis meets our needs better than Principal Component, which does not distinguish between the common variance and specific variance.

- **Including more Loading Components:**
  Including more sites into our network and including more factor loadings, allows us to discover more regarding the variations due to the location of the site.

- **Loading generated by PC:**
  The next table represents the factor loading generated by principal components from the previous section, and provide us with almost the same information.

- **Peaks from Lomb Periodogram - Part1 :**
  The number of peaks generated by frequency analysis (Lomb periodogram) from **Part 1** (see Appendix A), must comply with the number of factor loading explaining the fluctuations.

I believe it is amazing that we can determine how many factor loading components we need by counting the number of peeks from the spectral analysis.

11.3 Part 3

Part 3 *focused on the implementation of a CGPS Filter. The reason why I chose to proceed with this part, was that all the building blocks needed to construct the filter were already implemented in Part 1.*

This chapter is NOT an obligatory part of my thesis. I included it because this information is useful when working with CGPS time series and because I find this theory very interesting. I did not, however, have the time to implement the algorithms needed to identify the dominant type of noise in our network.

appendix A, the graphs generated from each steps are presented.

Site Name		East	Direction			North	Direction			Height	Direction	
	Intercept	**Corrected**	**Slope**	**Corrected**	**Intercept**	**Corrected**	**Slope**	**Corrected**	**Intercept**	**Corrected**	**Slope**	**Corrected**
ales.res	-50.08	-51.53	0.0092	0.0112	-56.64	-58.77	0.0108	0.0133	-16.237	-16.805	0.0032	0.00381
berg.res	-77.64	-80.96	0.0147	0.0181	-52.75	-43.08	0.01001	0.0078	-22.426	-21.205	0.00456	0.0046
bodo.res	-53.56	-55.18	0.0094	0.0114	-53.21	-55.13	0.01014	0.0124	-23.953	-24.976	0.0045	0.0056
hers.res	-52.34	-57.08	0.0116	0.0134	-49.7	-51.97	0.01055	0.0125	-8.973	-8.617	0.0021	0.0022
hofn.res	-40.38	-42.6	0.0088	0.0107	-51.8	-53.85	0.01039	0.0123	-51.5	-53.19	0.01083	0.01263
kris.res	-56.35	-54.19	0.012	0.0182	-51.72	-15.91	0.0117	0.0013	-13.584	-5.154	0.00320	0.00096
nya1.res	-42.78	-43.66	0.007	0.0085	-53.26	-54.81	0.00961	0.0116	-49.19	-51.23	0.0087	0.010827
oslo.res	-56.87	-57.92	0.01063	0.0129	-51.39	-51.18	0.00999	0.0115	-25.984	-26.705	0.00512	0.00614
pots.res	-68.69	-72.78	0.01221	0.0144	-50.03	-59.97	0.00974	0.013096	-4.566	-6.026	0.0012252	0.0017
stav.res	-53.84	-54.96	0.01005	0.0120	-54.81	-46.03	0.01036	0.00857	-11.451	-11.705	0.0022	0.0026132
tro1.res	-14.07	-22.794	0.0082	0.0118	-48.95	-51.31	0.00850	0.01015	-19.778	-20.629	0.00402	0.0046
tron.res	-52.2	-53.6	0.009411	0.0113	-55.08	-57.14	0.01045	0.0128	-23.071	-23.733	0.00444	0.00535
vard.res	-69.69	-71.7	0.012425	0.0155	-38.23	-39.85	0.00809	0.0101	-18.918	-19.699	0.00368	0.00466
wsrt.res	-60.6	-62.14	0.01168	0.01276	-55.07	-57.67	0.01039	0.012112	-0.6819	0.14106	0.0004642	0.00014

Table 11.1: *Site Velocity Improvement*

Appendix A

Graphs from Part I

This appendix presents graphs generated from time series for all sites examinated under Part I.

1. The first row presents the raw data of each component (namely east, north and height directions) of the site without gaps. The gaps are eliminated by runing the algorithm gapWithSim.R.

2. The second row presents the raw data of each component (namely east, north and height directions)) of the site without outliers. Outlierss are eliminated bu runing the algorithm robustGrubbOutlier.R

3. The third row presents our time series of each component (namely east, north and height) of the site by eliminating the trend. The trend is elliminated by fitting a polynome of degre 1.

4. The forth row presents the Smoothed Lomb periodogram for each components (namely east, north and height directions). To reduce the variance, we choose to overage over adjacent frequencies(smoothing the periodogram).

5. The last row presents the significance of the peak in Lomb periodogram.

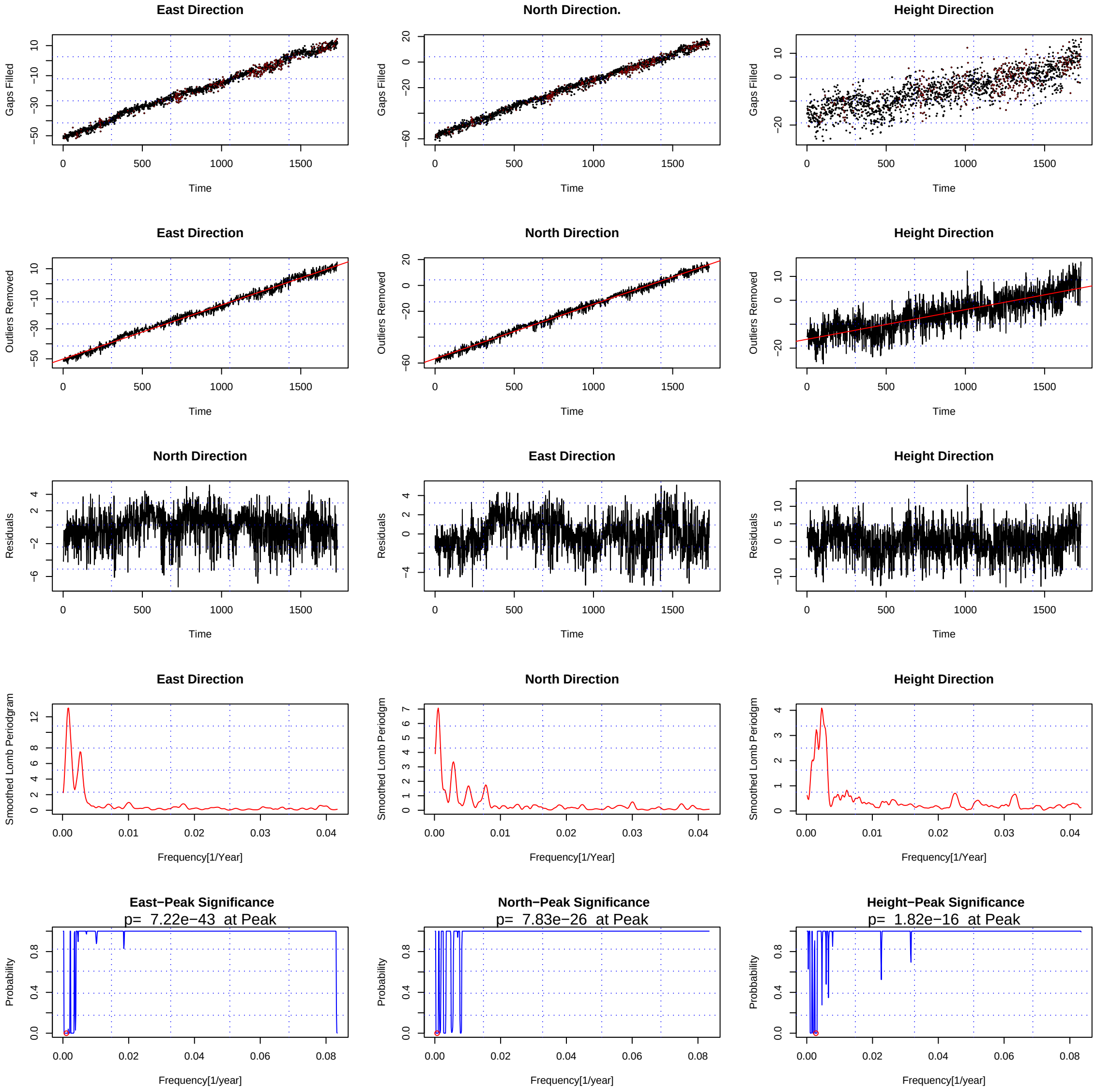


Figure A.1: *Site ALES*

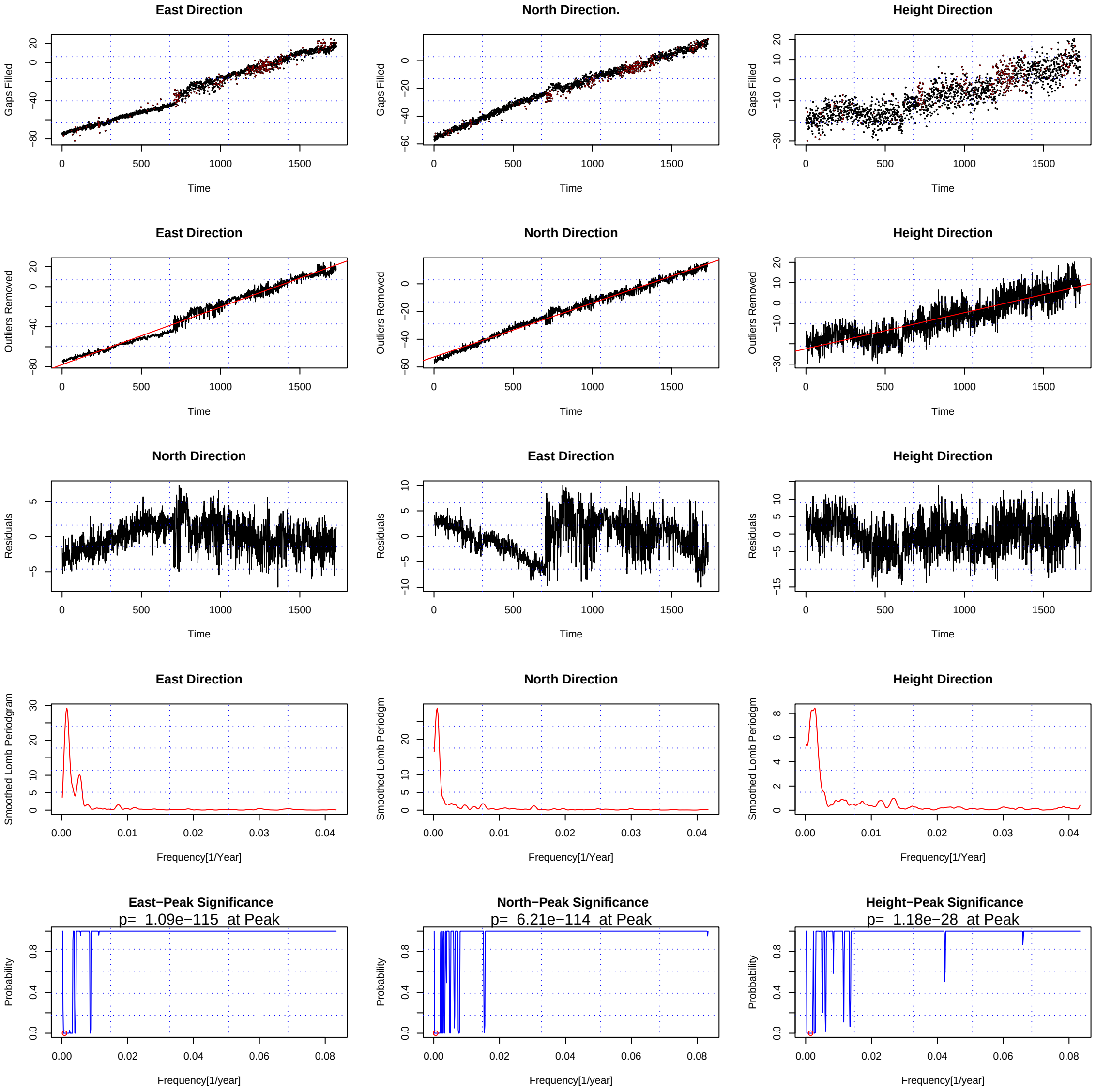


Figure A.2: *Site BERG*

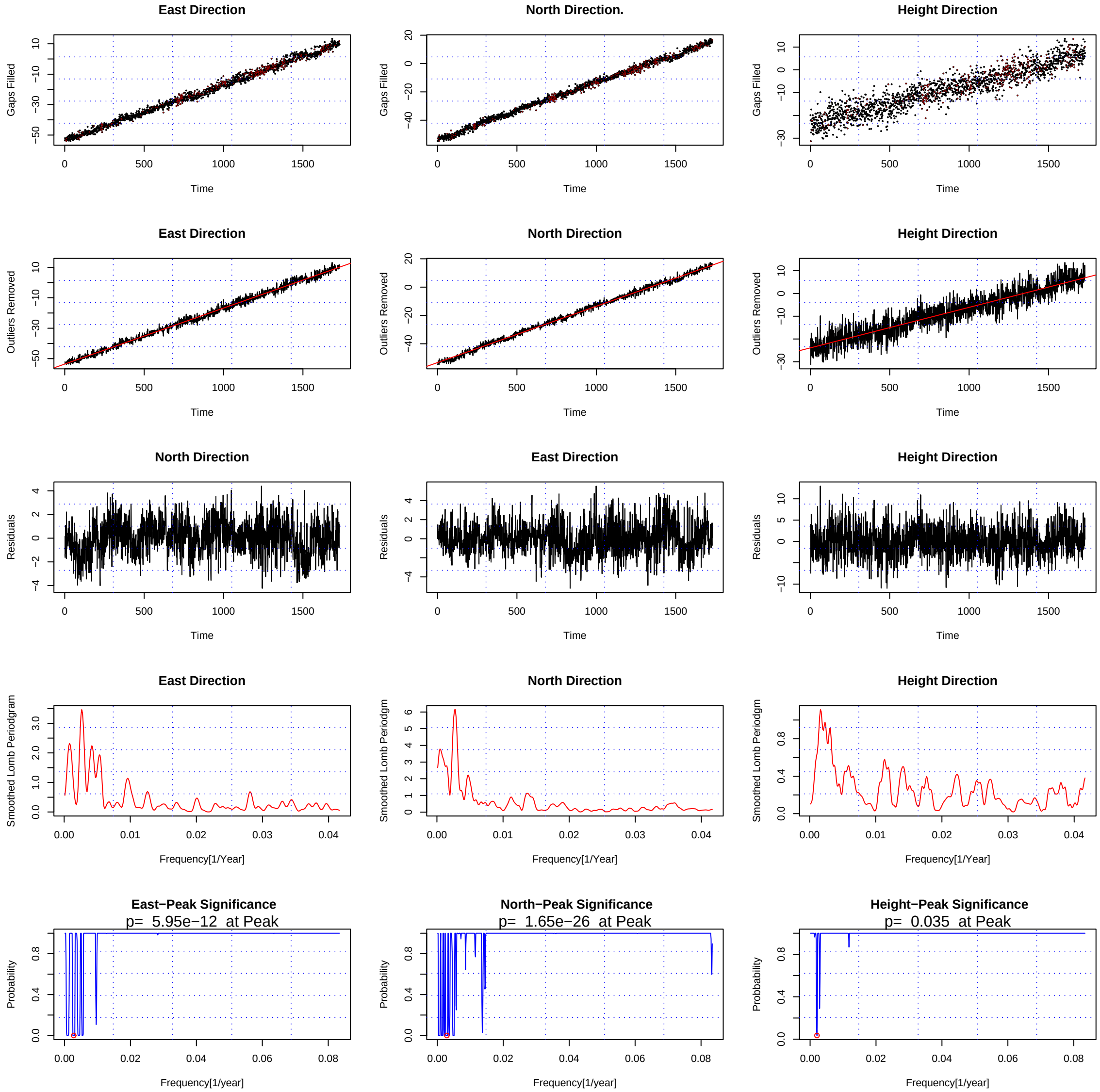


Figure A.3: *Site BODO*

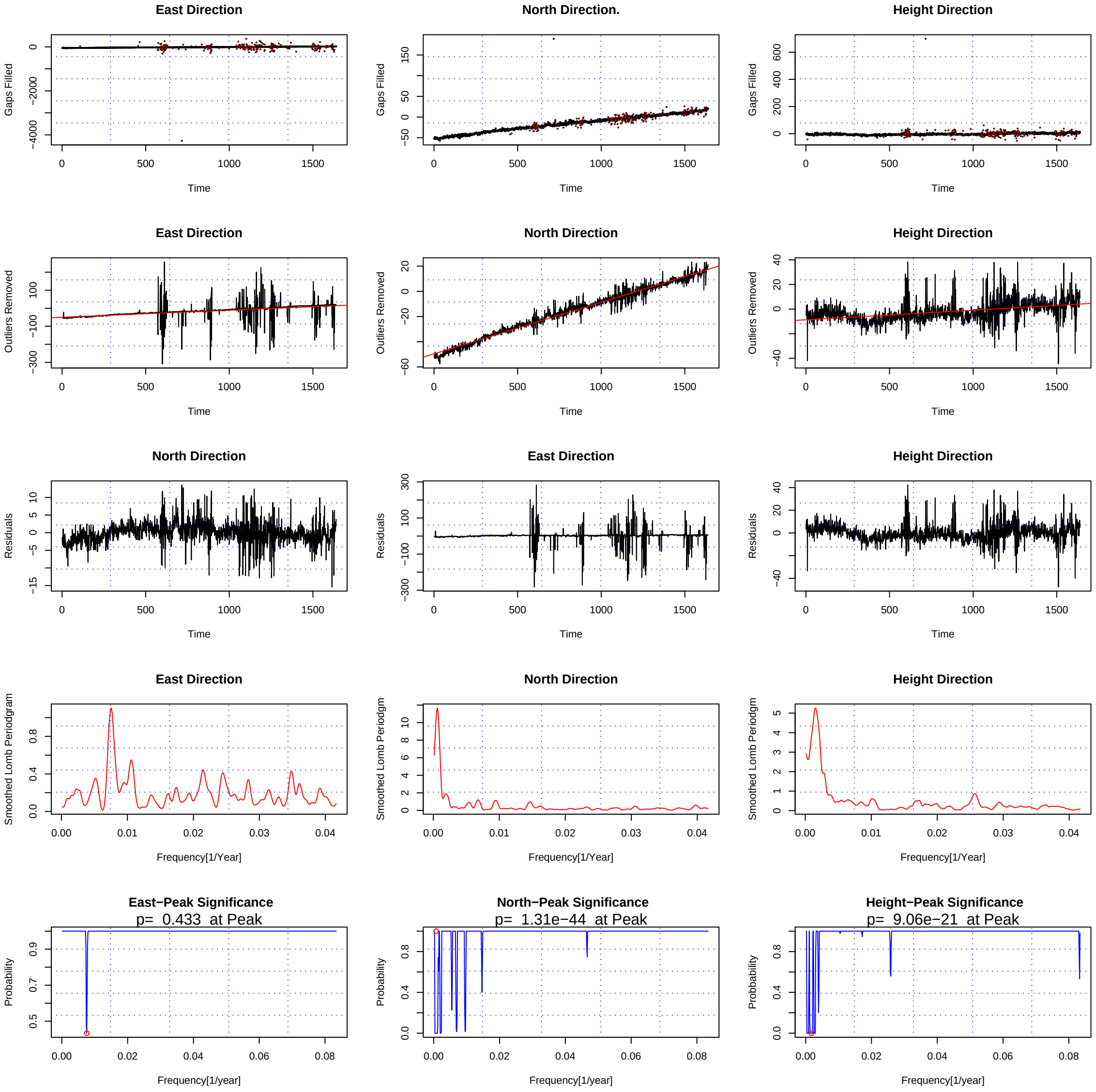


Figure A.4: *Site HERS*

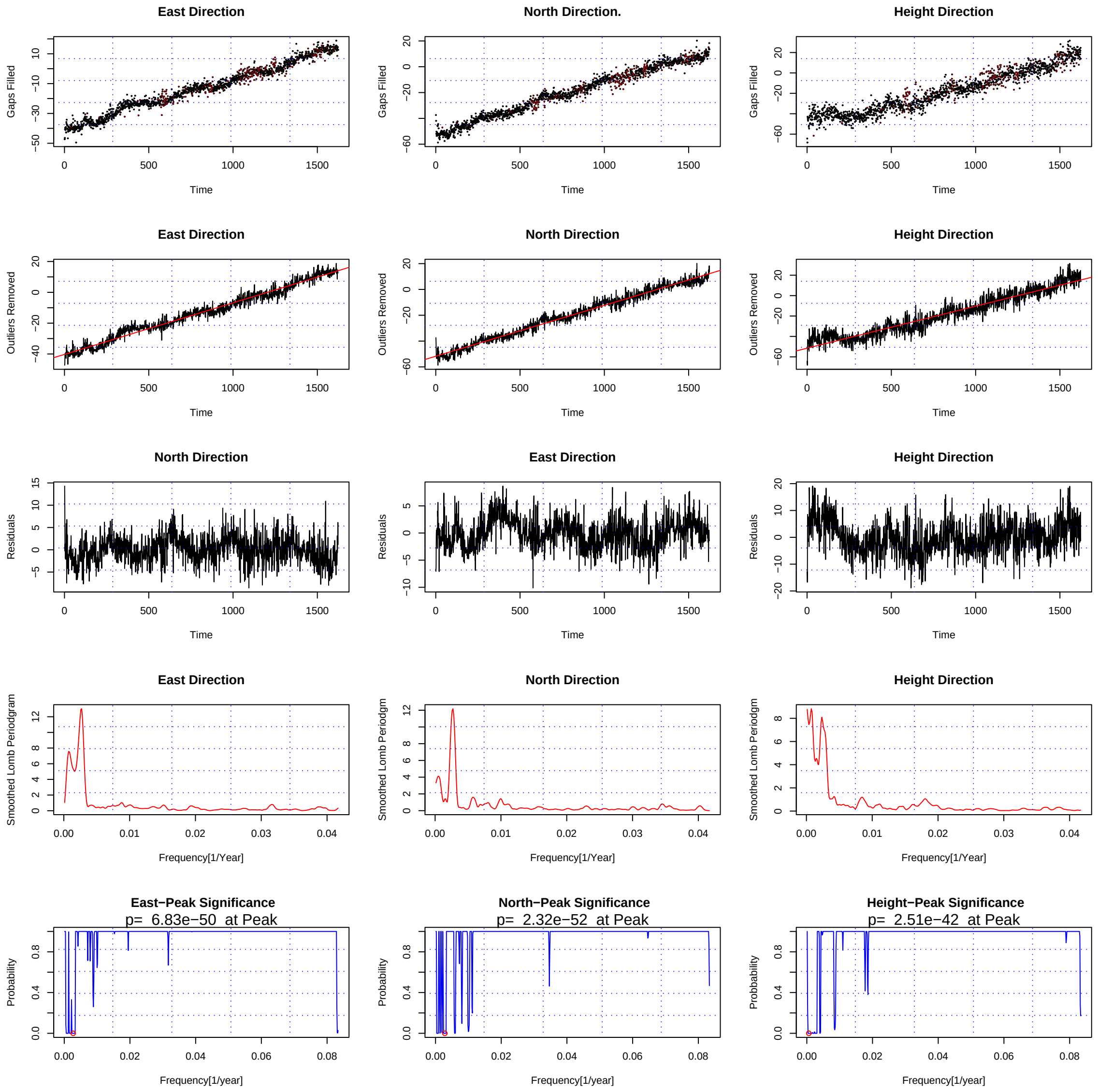


Figure A.5: *Site HOFN*

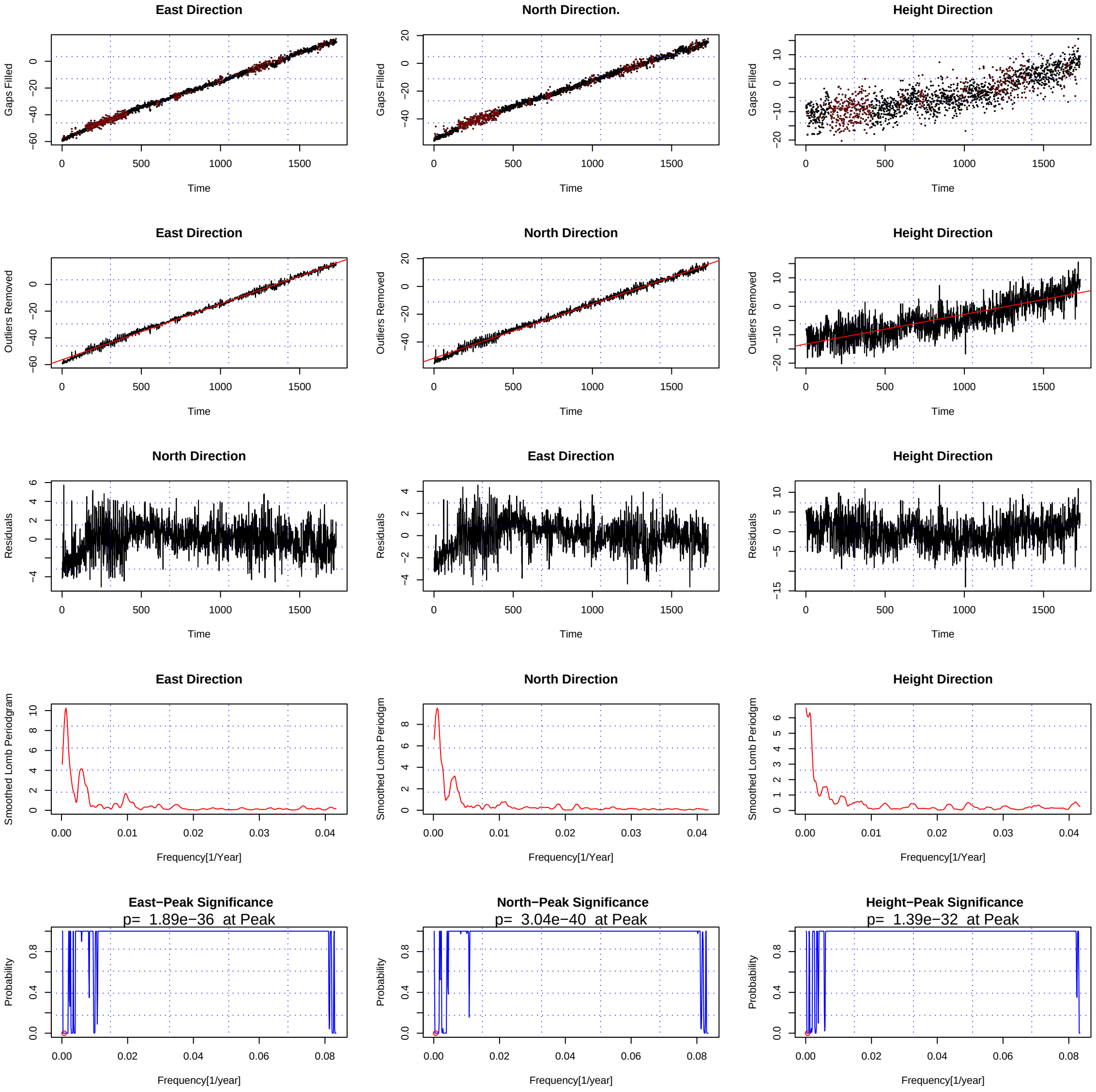


Figure A.6: *Site KRIS*

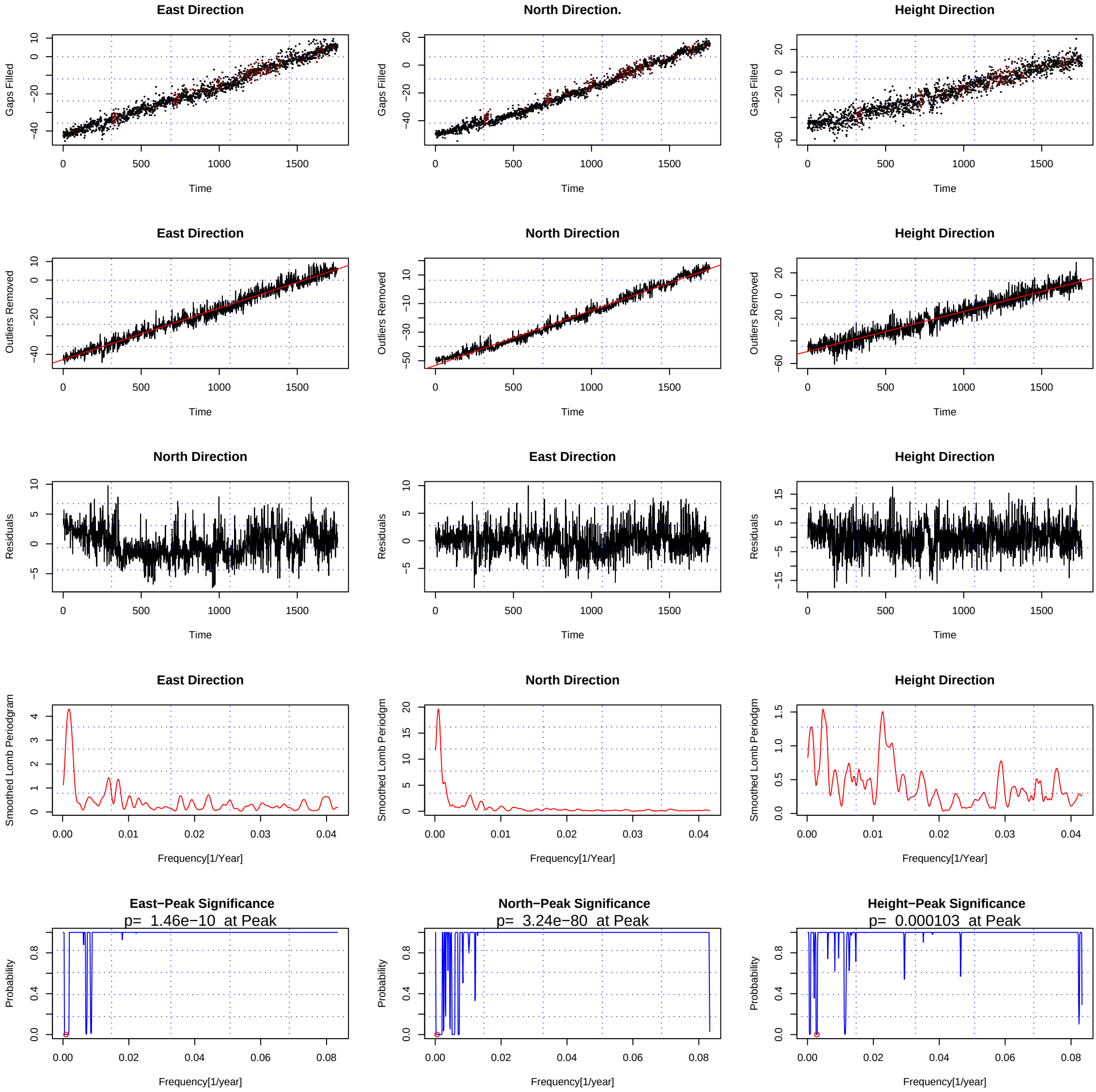


Figure A.7: *Site NYA1*

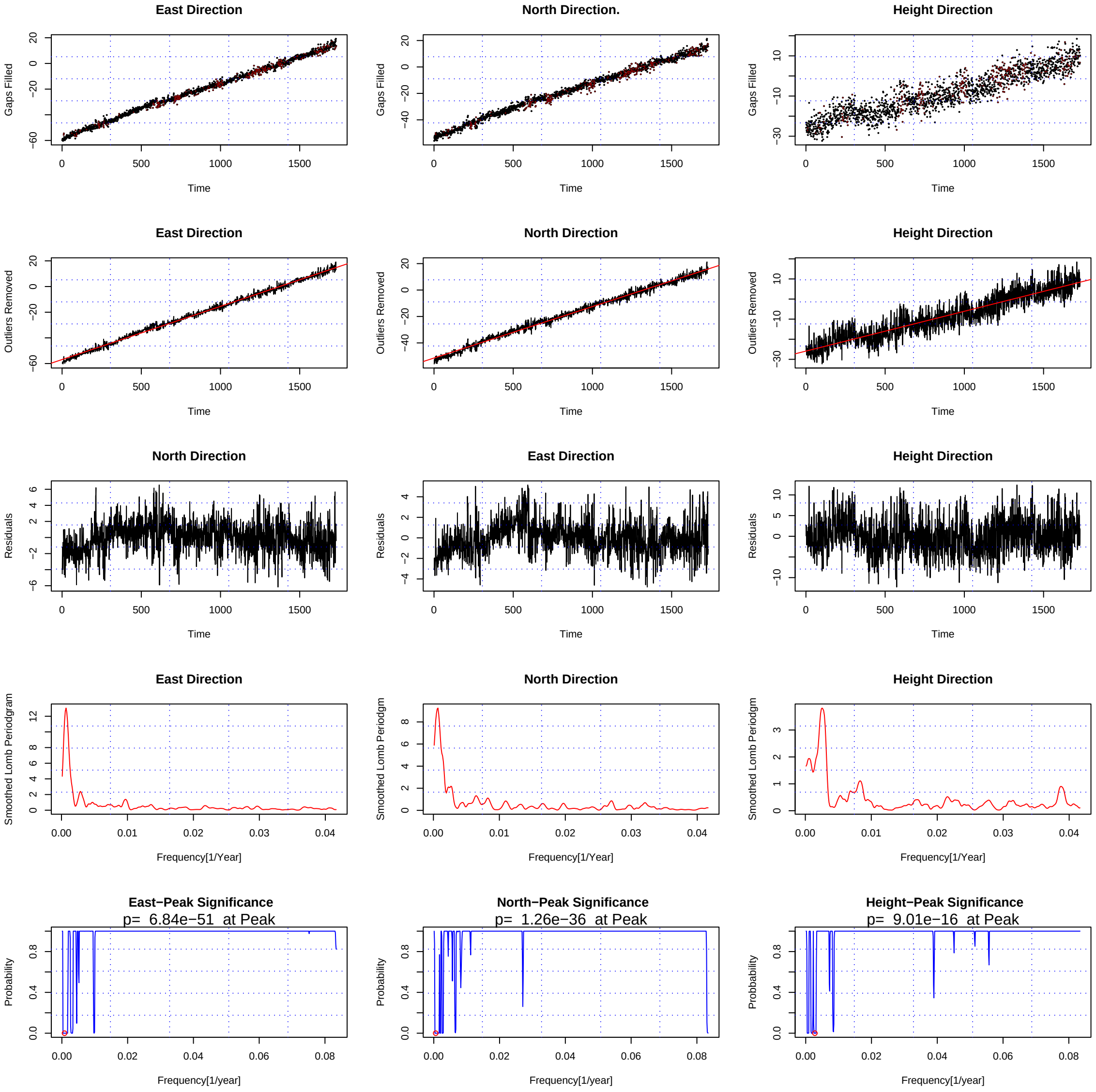


Figure A.8: *Site OSLO*

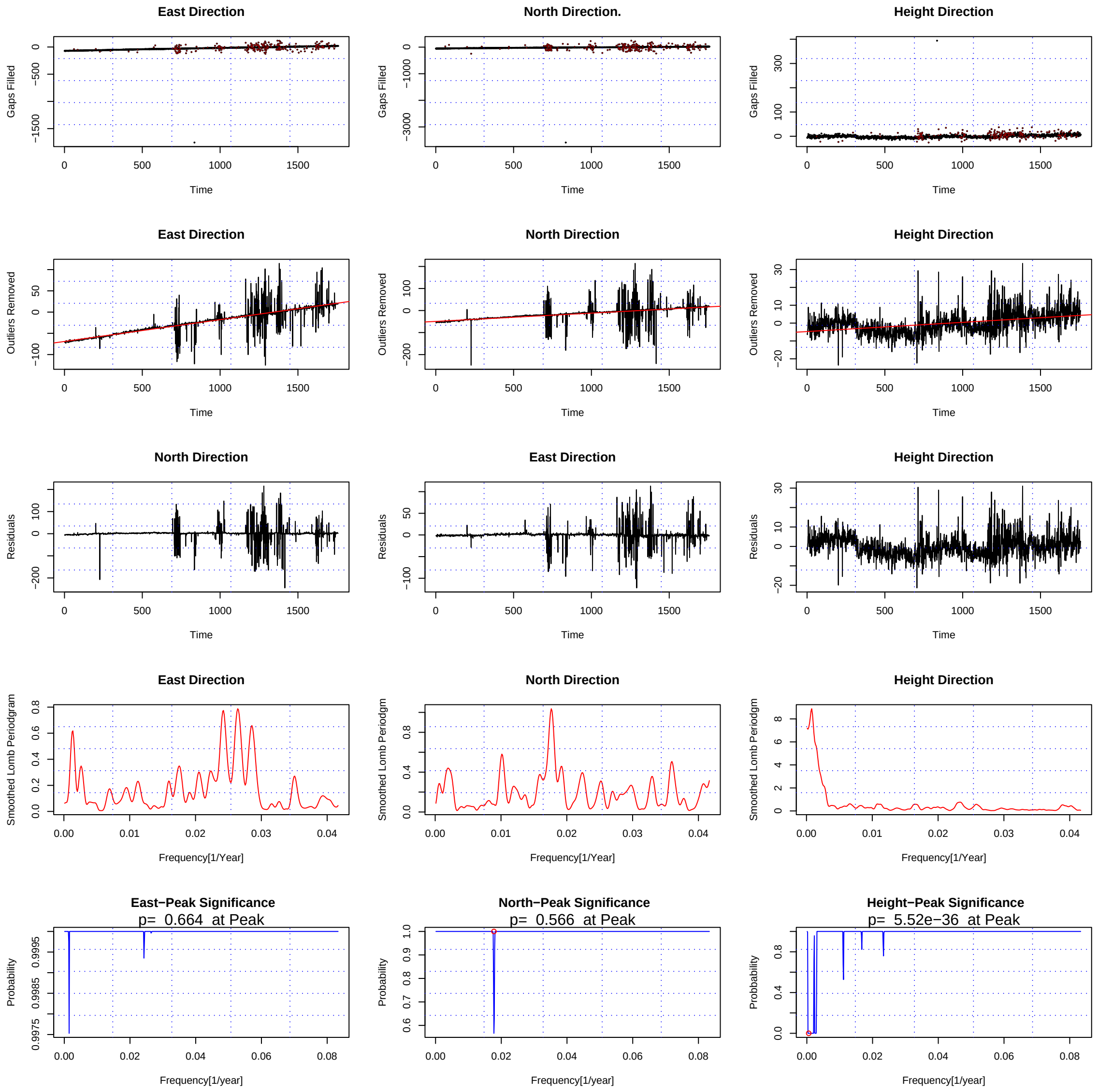


Figure A.9: *Site POTS*

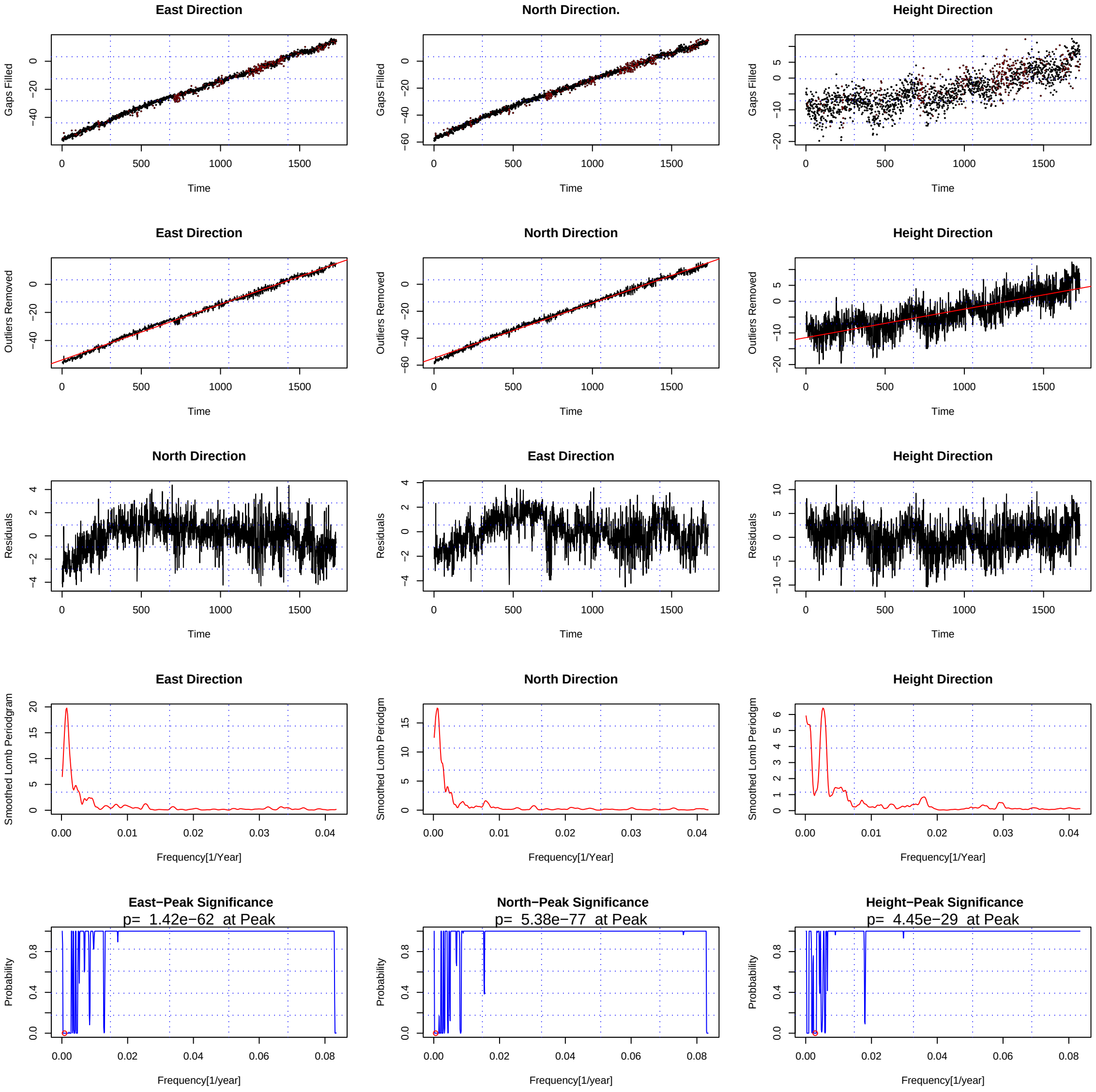


Figure A.10: *Site STAV*

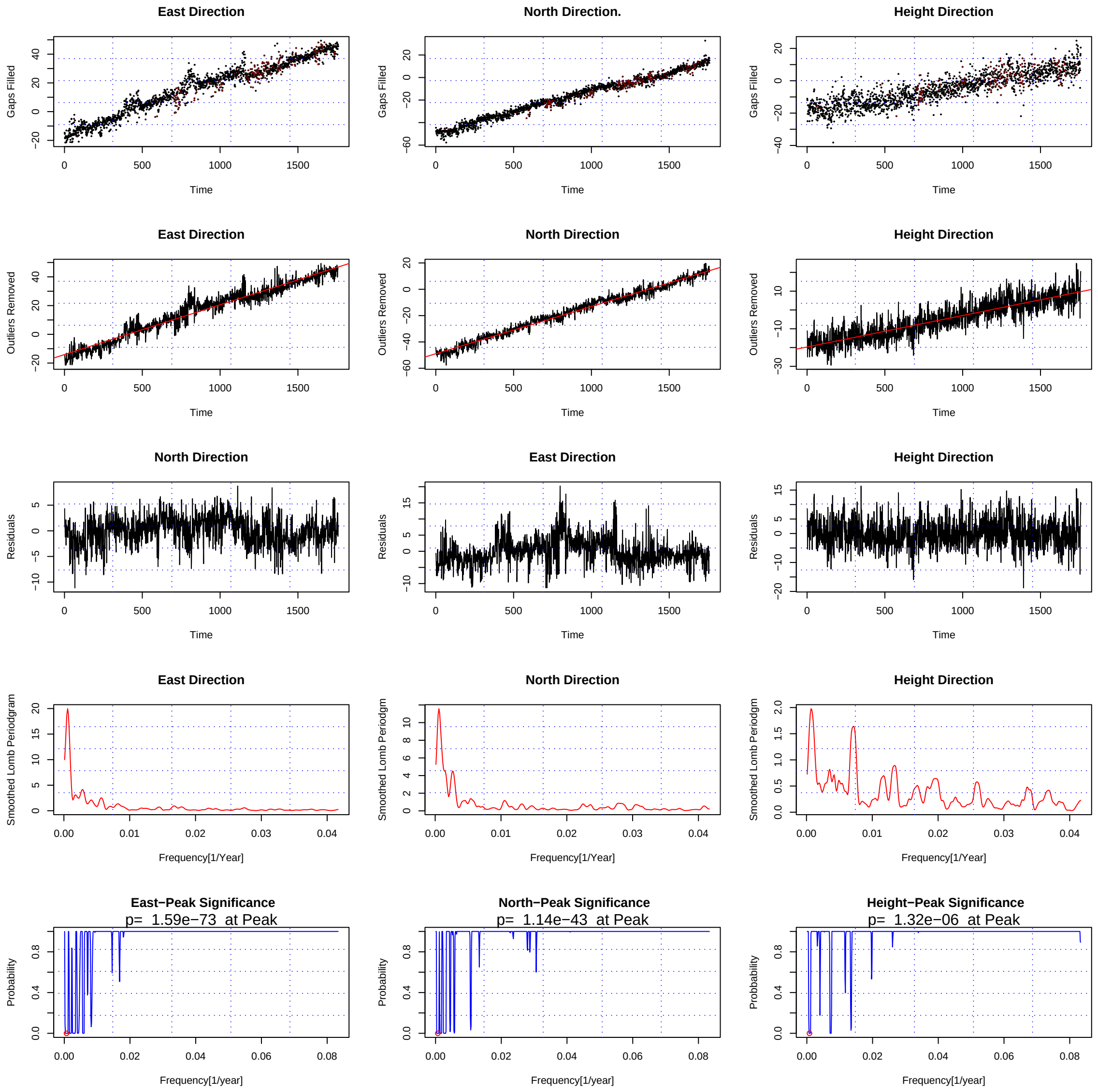


Figure A.11: *Site TRO1*

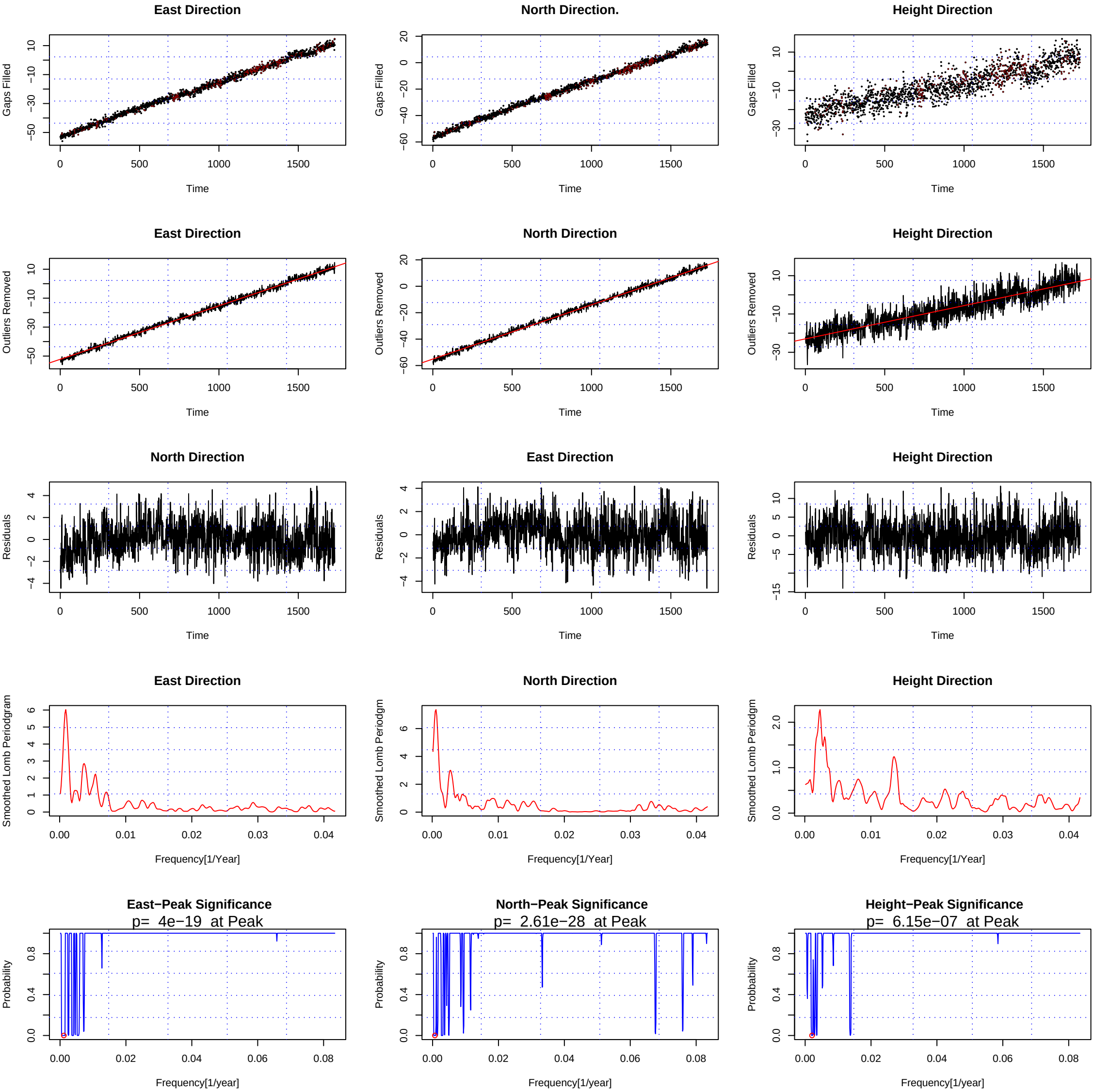


Figure A.12: *Site TRON*

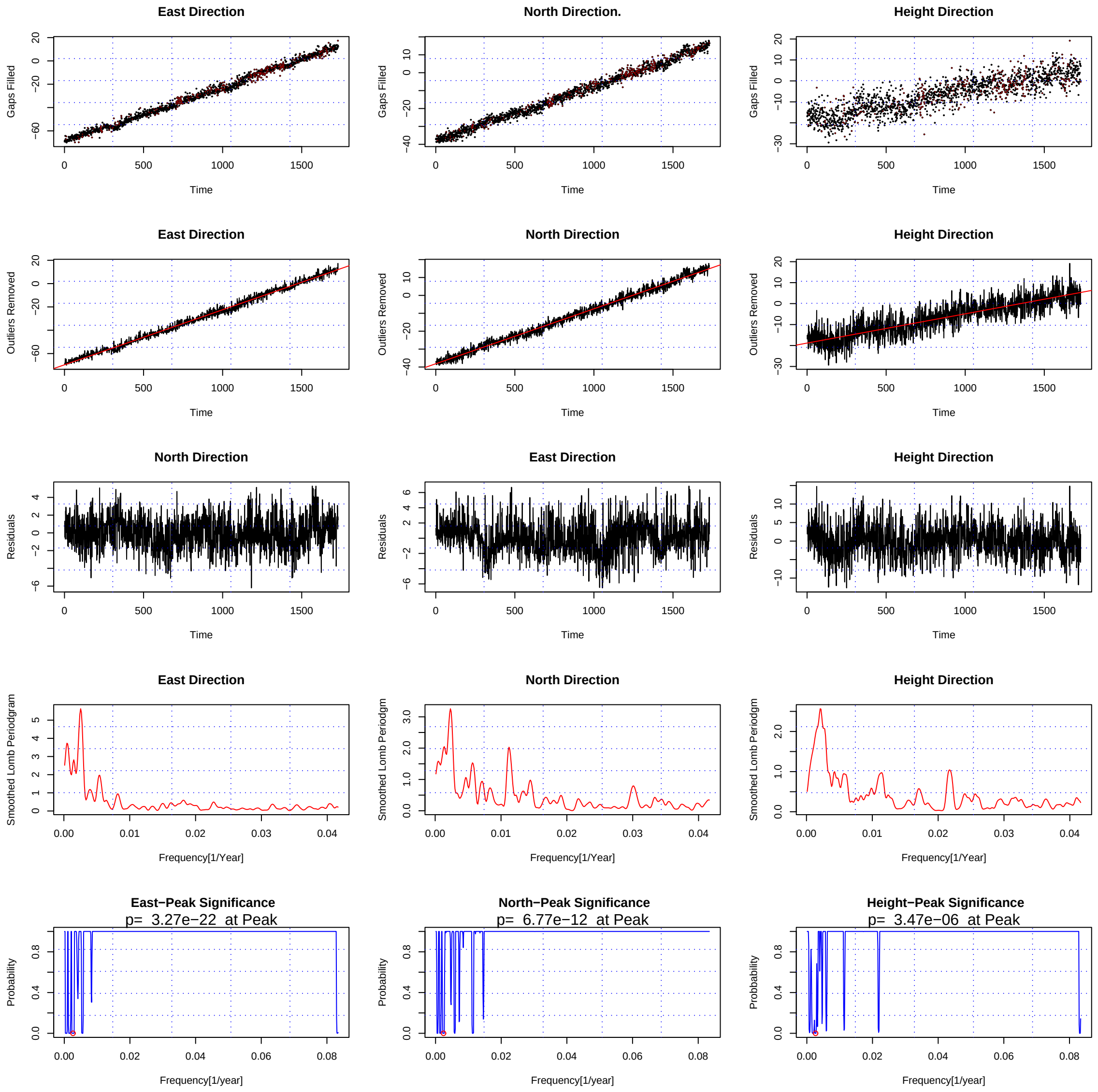


Figure A.13: *Site VARD*

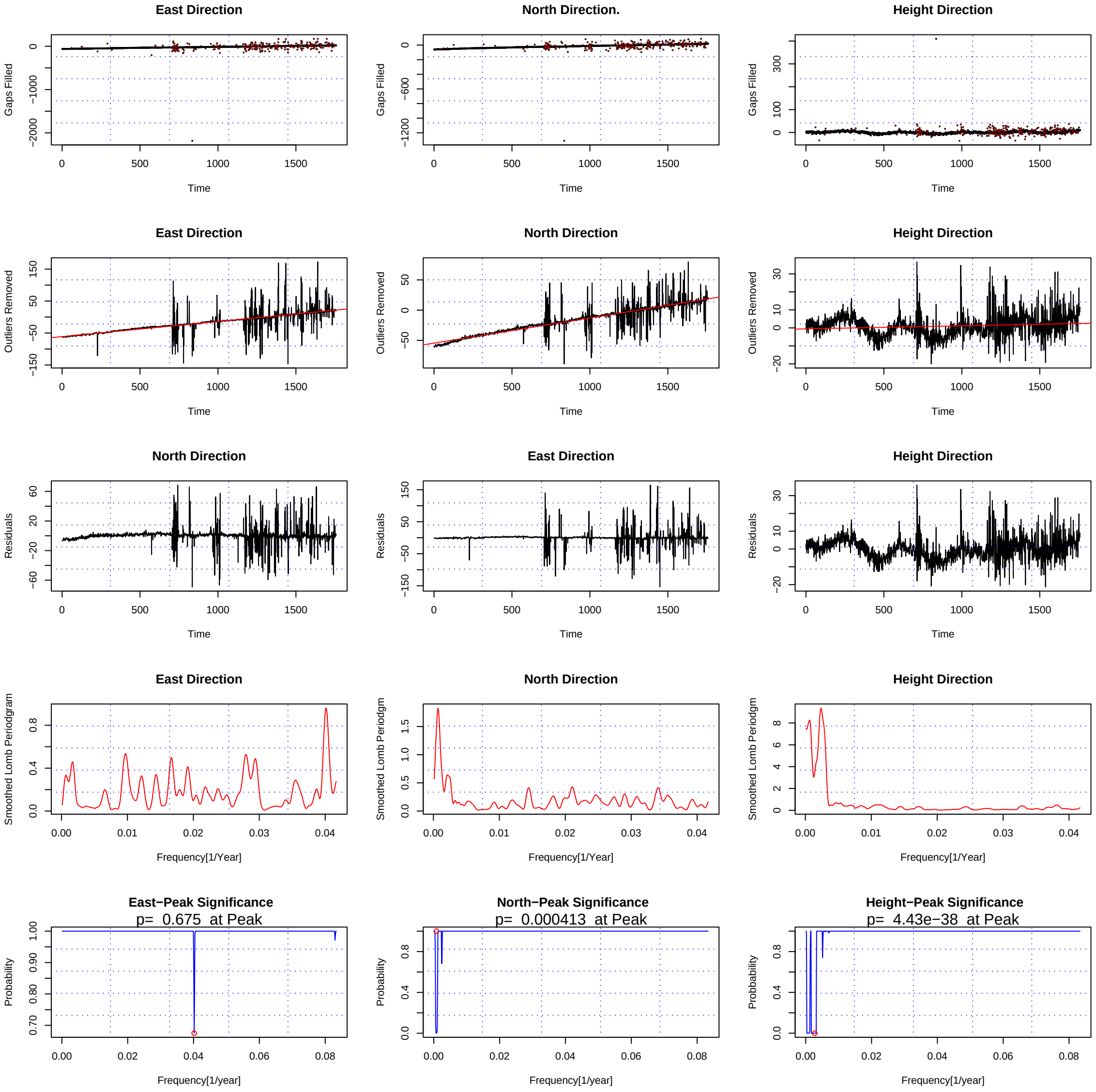


Figure A.14: *Site WSRT*

Appendix B

Matrix Theories

This appendix introduces matrix theories .

B.1 Matrix theories

Definition B.1.1. ***Matrix Addition:***
If A and B are both $n \times m$ matrices, then the sum of A and B, denoted by $A + B$, is the $n \times m$ matrix whose entries are $a_{ij} + b_{ij}$ for each $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, m$.

Definition B.1.2. ***Matrix Multiplication:***
Let A be $n \times m$ and B be $m \times p$ matrix, The matrix product of A and B, denoted by $A.B$, is the $n \times p$ matrix C whose entries C_{ij} are given by

$$\begin{aligned} C_{ij} &= \sum_{k=1}^{n} a_{ik} b_{kj} \\ &= a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{im} b_{mj} \end{aligned} \tag{B.1}$$

for each $i = 1, 2, ..., n$ and $j = 1, 2, 3, ..., p$

Definition B.1.3. ***Matrix Transpose:***
The transpose of an $n \times m$ matrix $A = (a_{ij})$ is the $m \times n$ matrix $A^T = (a_{ji})$. A square matrix A is said to be symetric if $A = A^T$

Definition B.1.4. ***Square Matrix :***
The square matrix has the same numbers of row and columns. A diagonal matrix is a square matrix $D = (d_{ij)}$ The identty matrix is a square matrix.

Definition B.1.5. ***Upper triangular :***
An upper-triangular $n \times m$ matrix $U = (u_{ij)}$ has for each $j = 1, 2, \cdots, p$, the entries

$$u_{ij} \quad = \quad 0 \qquad \text{for each } i = j+1, j+2, \cdots, n$$

and for lower-triangular matrix $L = (l_{ij)}$ has for each $j = 1, 2, \cdots, n$ the entries

$$l_{ij} \quad = \quad 0 \qquad \text{for each } i = 1, 2, \cdots, j-1$$

Definition B.1.6. ***nonsingular :***
An $n \times n$ matyrix A is said to be nonsingular if an $n \times n$ matrix A^{-1} exists with $AA^{-1} = A^{-1}A = I$. The $A^{-1}$is called the inverse of A. The matrix without inverse ia called singular.

Definition B.1.7. ***Square Matrix :***
The square matrix has the same numbers of row and columns. A diagonal matrix is a square matrix $D = (d_{ij)}$ The identty matrix is a square matrix.

Definition B.1.8. ***Positiv defined:***
Suppose that Σ is a symetric pxp matrix. We say that Sigma is non-negative definite if

$$x^T \Sigma x \geq 0 \; for \; all \; p - vectors \; x$$

and its positive definite if

$$x^T \Sigma x > 0 \; for \; all \; p - vectors \; x \neq 0$$

Appendix C

Computer Programs

This appendix gives all implemented programs.

Table C.1: *R-Programs*

Name	***Description***
maSmoother.R	*Weigted Moving Average Filter*
medianSmoother	*Median Smoother*
sarimaForcast.R	*Prediction from SARIMA model*
gapWithSim.R	*Fill gaps by simulation*
gainFactorOutAlg.R	*Gain Factor Outlier Algorithm*
drsoAlgorithm.R	*Remove outliers by simulation*
robustGrubbsAlg.R	*Remove outliers by robust Grubb outlier detection*
estPeriodComp.R	*Estimation of Periodic components in a TS*
estLombMagnitude.R	*Least Square Method for estim. coeff. in Lomb Perio*
offsetDetection.R	*Detect and correct for jumps in TS*
lombPeriodogram.R	*Perform Least Square of frequence analysis*
mainPart1.R	*Main program of PART I of this thesis*
principalComp.R	*Principal Component analysis(PART II)*
factorAnalysis.R	*Factor Analysis (PART II)*
mainPart3.R	*Filter Implementation SARIMA (PART III)*
klobucharAlg.R	*Estimation of ionospher delay -Klobuchar Algorithm*
testDRSOAlgorithm.R	*Test Klobuchar algorithm*

Continued on next Page...

Table C.1 –Continued

Name	***Description***
testLombPeriodogram.R	*Test Lomb Periodogram*
testOffsetDetection.R	*Test Offset Detection Algorithm*
testGainFactorOutlAlg.R	*Test Gain Factor Outlier Algorithm*
testPeriodogrmoothing.R	*Test the periodogram smoothing algorithm*
testGapWithSim.R	*Test Gaps by simulation*
testRobustGrubbOutlier.R	*Test Robust Brubb Outlier algorithm*
testRobustOutlier.R	*Test Robust Outlier Algorithm*

Bibliography

[A01] Peter J. Brockwell and Richard A. Davis, ” Introduction to time series and Forecasting ”*, 2002 Springer*

[A02] Robert H. Shumway, David S. Stoffer , ” Time Series Analysis and Its Applications ”*, 2000 Springer*

[A03] B. W. Silverman , ” Density Estimation for Statistics and Data Analysis” *, CHAPMAN & HALL*

[A04] Michele Basseville, Igor V.Nikiforov , ” Detection of abrupt Changes, Theory and Application ” *, Prentice-hall, inc*

[A05] M.P. Wand, M.C. Jones ” Kernel Smoothing ”*, CHAPMAN & HALL*

[A06] Wendy L. Martinez, Angel R. Martinez, ” Computational Statistics Handbook with MATLAB ”*. 2002 by Chapman & Hall/CRC*

[A07] Christian P. Robert, Geogre Casella, ” Monte Carlo Statistical Methods, Second Edition ” *. ISBN 0-387-21239-6, 2004 Springer*

[A08] Richard A. Johnson, Dean W. Wichern, ” Apppied Multivariate Statistical Analysis, Fifth Edition ” *ISBN 0-13-121973-1, 2002 by Prentice-Hall, inc.*

[A09] Lennart Ljung, ” System Edentification Theory for the User, Second Edition ”*. ISBN 0-13-656695-2, Prentice-Hall, inc.*

[A10] Mohinder S. Grewal, Lawrence R. Weill and Angus P. Andrews. ” Global Positioning Systems, Inertial Navigation, and Integration ” *, WILEY*

[A11] Itay Hen, Anat Sakov, Neri Kafkafi, Ilan Golani, Yoav Benjamini "The dynamics of spatial behavior: How can robust smoothing techniques help ?", *Journal of Neuroscience Methods 133(2004)161-172, ELSEVIER.*

[A12] B. Hofmann-Wellenhof, H.Lichtenegger, and J. Collins . "GPS Theory and Practices, Third, Revised Edition" *. ISBN 3-12-82591-6 Springer-Verlag Wien New York.*

[A13] Alfred Leick "GPS Satellite Surveying , third edition"*. ISBN 0-371-05930-7, Copyright 2004 by John Wiley & Sons, Inc*

[A14] Gilbert Strang, Kai Borre " Linear Algebra, Geodesy, and GPS" *. ISBN 0-9614088-6-3, WELLESLEY-CAMBRIDGE PRESS& Sons, Inc*

[A15] "JOURNAL OF TIME SERIES ANALYSIS, VOLUME 28, NO.2, March 2007"*. ISBN 0143-9782, Blackwell Publishing*

[A16] Simon D. P. Williams " Offsets in Global Positioning System time series " *JOURNAL OF GEOPHISICAL RESEARCH, VOL. 108, NO.B6, published 21 June 2003.*

[A17] John A. Klobuchar"Ionospheric Time-Delay Algorithm for Single-Frequancy GPS Users"*. IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, Vol. AES-23, No.3, May 1987*

[A18] Niranjan Prasad, A.D. Sarma " Ionospheric Time Delay Estimation using IDW Grid Model for GAGAN " *J.Ind. Geophys. Union (October 2004), Vol.8, No.4, pp.319-327.*