Faculty of Science and Technology
Department of Physics and Technology

Investigations of ionospheric disturbances using coherent HF instrumentation

—

Artem Reznychenko

FYS-3900 Master thesis in Physics   May 2016

Investigations of ionospheric disturbances using coherent HF instrumentation

A Thesis Submitted by:

Artem Reznychenko

Master in Physics

Faculty of Science and Technology

UiT The Arctic University of Norway

Supervised by:

Prof. Cesar La Hoz

UiT The Arctic University of Norway

PhD, Associate professor

Oleksandr Koloskov

Institute of Radio Astronomy

of the National Academy of Sciences of Ukraine

Acknowledgements

I would like to thank my advisor Professor Cesar La Hoz for great help in writing this paper, for useful advices, instructions and explanations during this work. I would like to thank for his cooperation and efforts to understand my ideas and to implement them in life.

From the Ukrainian side I would like to thank my second supervisor Oleksandr Koloskov for their invaluable help and support me during the writing of this thesis.

I would like to thank my family, because they have always believed in me and supported me.

I would like to thank the Tromso University for a wonderful pastime, useful and interesting knowledge, new friends. Special thanks to the support staff for their work, for purity, light, heat and high-speed internet.

Abstract

This work is based on long term monitoring the HF signal data at 9.996 MHz at mid-latitudes from the short radio-path Moscow-Kharkiv and at high-latitudes from the longer radio-path Moscow-Tromso for 2013. The HF signal transmitter is the radio station RWM in Moscow, Russia. The receiver stations are located in Tromso (Norway) and Kharkiv (Ukraine). Using intensity and the Doppler frequency shift of the HF received signals we evaluated the effects of ionospheric disturbances. For the interpretation of the results additional information was also used about Total electron content (TEC) above Tromso, Kiruna (Sweden), Joensuu (Finland, mid-point of radio path Moscow – Tromso), Kharkiv; the critical frequency data F0F2 above Tromso and Moscow; the local magnetic K-indices in Tromso and Kharkiv; the planetary Kp-indices; the solar flux $F_{10.7}$, Sun spot number, X-ray background flux. The focus of the work was to write a program for processing the HF signals using the MatLab language. We also wrote the programs to visualize all data. The work confirmed the standard behavior of the HF signals. Effects, such as an increase in intensity of signal during the day and decrease at night; the positive Doppler frequency shift at sunrise and the negative at sunset were observed. We found some new features in the behavior of the HF signals in quiet conditions and during ionospheric disturbances. These include an increase of HF signal intensity after sunset in summer, a decrease of the intensity in the afternoon in summer and reduced intensity and unsteady behavior during magnetic storms.

Contents

1 Introduction

The parameters of the ionosphere are changed according to different factors (changes of solar activity, level of geomagnetic activity, precipitation of energetic particles etc.). Along with these changes, the critical frequency of F-layer (F0F2), absorption in the lower ionosphere and other factors affect the parameters of high frequency (HF) signals: the intensity, Doppler frequency shift and phase [1, 2]. The variations of the signal parameters can be used as an indicator of the ionospheric processes and disturbances [3, 4, 5].

To study relationships between ionospheric disturbances and the parameters of HF signal we performed a long term monitoring of coherent HF radiation with stable frequency 9.996 MHz, amplitude and phase from the RWM radio station (Russian exact time and frequency service) located near Moscow for radio paths Moscow-Tromso (Norway polar region) and Moscow-Kharkiv (Ukraine, Low Frequency Observatory (LFO), mid-latitude region). To record the signal we use the coherent HF receiver stations based on digital receivers WR-G313i. The receiver stations are located at LFO, Ukraine (49°56'01"N, 36°57'10"E) and Tromso, Norway (69°39'41" N, 18°56'30" E). In this work, we analyze the data for the year 2013.

The HF receiver station has the name of "coherent", because it includes the high stability reference oscillator that allows to measure parameters of coherent signal: exact frequency, phase and intensity. Certainly, the ionosphere affects the HF signal during propagation, it loses its coherence – its spectrum is widened and the amplitude and phase are changed. Receiving such HF signals, we evaluated the changes of intensity and Doppler frequency shift and we attempted to determine what processes and ionospheric disturbances caused the observed changes of the HF signal.

For the interpretation of the results we used the following additional information:
- changes of the time of day;
- changes of seasons;
- changes of ionospheric parameters: total electron content (TEC) above Tromso, Kiruna (Sweden), Joensuu (Finland, mid-point of radio path Moscow – Tromso), Kharkiv; critical frequency data F0F2 above Tromso and Moscow;
- magnetic activity data: the local magnetic K-indices in Tromso and Kharkiv; the planetary Kp-indices [6, 7];
- data about solar activity: the solar flux $F_{10.7}$, Sun spot number and X-ray background flux [7].

Most of the work on the thesis was devoted to the writing of special programs in the Matlab language. These programs allow to process the data from the receiving stations and determine the intensity and the Doppler shift of the received signal. Also we can plot the signal parameters, the ionospheric parameters, magnetic and solar activity. Through numerous figures an analysis of the impact of ionospheric disturbances on the parameters of the HF signal was investigated.

We confirmed the already known patterns of behavior of the HF signal parameters [2, 3, 8, 9] and identified new ones that also were discussed but they require a more detailed analysis in each case with additional data about the behavior of the ionosphere and solar activity.

In chapter 2 we provide an overview of the Earth's ionosphere and propagation of electromagnetic shortwaves. Section 2.1 describes the Earth's ionosphere and its features. Section 2.2 describes the reflection of electromagnetic waves from the ionosphere, which makes possible their propagation over long distances. Section 2.3 presents the main methods for the determination of the parameters and the state of the ionosphere using HF waves. Section 2.4 briefly describes the method of determining the total electron content using GPS signals.

Chapter 3 describes the basic tools that were used to obtain data for this thesis. The first section describes the sources of data and the geometry of the observations. Section 3.2 describes the coherent HF receiving equipment: their characteristics, parts, features and locations. Section 3.3 describes the transmitter of the coherent HF radio signal by the Moscow radio station RWM. Section 3.4 describes the modern ionosondes, which provided data on the critical frequency of the ionosphere. Section 3.5 provides brief information about the GPS receiver at the LFO station in Ukraine.

Chapter 4 describes the data structures and processing procedures. Section 4.1 describes the main program to control the receiver WR-G313i, the recording of the received signal and a description of the structure of the signal data. Section 4.2 describes the data processing for determining the signal to noise ratio of the received signal, the signal intensity and Doppler frequency shift. Section 4.3 describes the data structures for plotting intensity and Doppler frequency shift, the critical frequencies, TEC, the Earth's magnetic activity and solar activity.

Chapter 5 presents results and discussion of this thesis. We present here the annual distributions of ionospheric parameters such as TEC, the critical frequency of the F2 layer at the midpoints of the radio paths Moscow-Tromso and Moscow-LFO. We discuss the influence of solar and geomagnetic activity on the parameters of the ionosphere. We show the behavior and the results of the analysis of the radio signal intensities and Doppler frequency shifts in Tromso and LFO in the long-term and short-term periods. We describe the signatures of ionospheric disturbances that we found in the processed data.

In chapter 6 the conclusions and the aspects for future works are presented.

Finally Appendix A presents listings of MatLab programs for processing and plotting all used data.

2 Background

This chapter provides an overview of the Earth's ionosphere and propagation of electromagnetic shortwaves. The first section talks about the Earth's ionosphere and its features. Section 2.2 presents the possibility of the ionosphere to reflect electromagnetic waves, which ensures their propagation over long distances. Section 2.3 presents the main methods for the determination of the parameters and the state of the ionosphere using HF waves. The last section briefly describes the method of determining the total electron content using GPS signals.

2.1 The ionosphere

In this project, the ionosphere is treated as a region of the Earth's atmosphere, which starts at about 50 km and up, where there is a sufficient number of ionized particles to have a noticeable effect on the propagation of radio waves.

The character of radio propagation is determined by the spatial distribution of the refractive index. The refractive index determines the speed of radio wave propagation, the conditions of refraction and reflection, and depends on the electron density.

Figure 2.1 shows the altitude profile of the electron density.

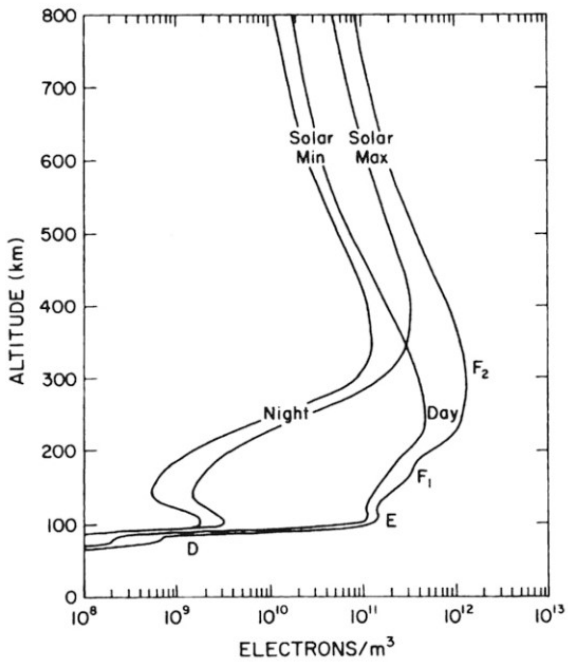

Figure 2.1: Altitude profile of the electron density for different solar activity [10].

We can see that the maximum electron density can vary in height from 200 to 450 km. It depends on many factors: the time of day, time of year, solar activity. At night, the electron density is lower than at day time, because there is no ionization of direct solar radiation.

The ionosphere is divided into regions D, E, F. Inside these regions can be intermediate layers of E1, E2, F1, F2.

2.1.1 The D region

The D region is from about 50 to 90 km. This region is very variable and impermanent. The additional layer C may appear below the D region due to the cosmic radiation [3]. The ionization and chemistry is very complex and is not fully understood [10]. Here are ions O_2^+, NO^+, N_2^+, metallic ions Mg^+, Fe^+, heavy bonds-ions $H^+(H_2O)$, $NO^+(H_2O)$, $NO^+(CO_2)$ and other [3]. The D region plays an important role in the propagation of long waves [11] - it absorbs significantly radiowaves due to frequent collisions between electrons and molecules of the atmosphere.

2.1.2 The E region

The E region (90 - 110 km) always exists in the daytime at all seasons across the globe and its behavior is highly dependent on solar radiation. The E2 layer (110 - 120km) is found only in some parts of the globe.

In the daytime, the E region is formed by the action of solar ultraviolet radiation ($800 < \lambda \leq 1028$Å) and soft X-ray ($10 < \lambda < 100$Å). The dominant neutrals are O_2 and N_2, and the dominant ions are O_2^+ and NO^+, created in the ion-molecular reactions ($O_2^+ + N_2 \rightarrow NO^+ + NO$; $O_2^+ + NO \rightarrow NO^+ + O_2$; $N_2^+ + O_2 \rightarrow N_2 + O_2^+$). The main mechanism of the disappearance of charged particles is dissociative recombination of ions and electrons ($NO^+ + e \rightarrow N + O$; $O_2^+ + e \rightarrow O + O$; $N_2^+ + e \rightarrow N + N$). At night time, the electron density is reduced and ionization in the E layer is maintained due to diffused ultraviolet solar radiation and corpuscular particles [12]. The E region affects the propagation of medium wavelengths - it causes refraction and causes them to follow the curvature of the Earth [11].

Sporadic E_s layers are impermanent layers consisting of enhanced regions of electrons with a horizontal length of the order of several tens of kilometers. Sporadic layers can reflect short and ultrashort waves at high values of ionization [11].

2.1.3 The F region

The F region divided into 2 regions: the F1 and the F2. In the daytime, there is both regions. Most of the data obtained from the electron density profiles indicate that the F1 and F2 layers have a weak stratification. At night, almost all the charged particles are concentrated at altitudes of 300-400 km [12].

The F1 region (120-200 km) is day-time layer and subject to a similar solar control as a the E layer. The dominant components for ionization are – N_2 and O (less O_2). The dominant ions are O_2^+ and NO^+, less O^+. Dominants reactions that create ions O_2^+ and NO^+ are $O^+ + N_2 \rightarrow NO^+ + O_2$; $O^+ + O_2 \rightarrow O_2^+ + O$; $N_2^+ + O_2 \rightarrow N_2 + O_2^+$; $N_2^+ + O \rightarrow NO^+ + N$; $O_2^+ + N_2 \rightarrow NO^+ + NO$; $O_2^+ + NO \rightarrow NO^+ + O_2$. The main mechanism of the disappearance of charged particles is dissociative recombination of ions and electrons ($NO^+ + e \rightarrow N + O$; $O_2^+ + e \rightarrow O + O$) [10, 13]. At sunset the electron density in the F1 layer is reduced. At night, the F1 layer disappears almost completely. In addition at night stops the process chemical formation of neutral atoms, so the process of ionization at night less than in the E region.

The F2 region (above 200 km) has a maximum of the electron density in the inospheric profile. The F2 region has a number of significant irregularities and don't have so simple dependence from zenith solar angle like in the E and F1 layers. Of course, at day time the solar radiation is main factor influence on electron density. However, at these altitudes, the rate of chemical recombination of ions is comparable to the rate of their diffusion and transfer mechanisms (ambipolar diffusion) begin to affect the value of the ion and electron density.

The magnetic field strongly influences the diffusion process because ambipolar diffusion occurs mainly along the direction of the geomagnetic field [3, 10].

For example, in the mid-latitudes the maximum electron concentration is greater in winter than in summer, the so-called winter anomaly. The winter anomaly is caused by an increase of neutral particles due to reduced diffusion which in turn is caused by colder winter temperatures. The increased availability of neutral atomic oxygen leads to enhanced plasma densities [12].

In the F2 region, the dominant component for ionization is atomic oxygen. The dominant ion is O^+. Radiative recombination of ions O^+ ($O^+ + e \rightarrow O + h\nu$) is very slow, so the main mechanism of their disappearance is a two-step process: first, reactions of O^+ with molecular O_2: $O^+ + O_2 \rightarrow O_2^+ + O$; and with molecular N_2: $O^+ + N_2 \rightarrow NO^+ + N$. The next step is fast dissociative recombination of molecular ions ($O_2^+ + e \rightarrow O + O$; $NO^+ + e \rightarrow N + O$) [10, 12, 13].

In additional, the horizontal neutral winds influence on the electron density variations. Due to the influence of the magnetic field, the charged particles can move in the vertical directions along the field and fall down, where the loss factor is large, which reduces the electron density [3].

At night, in the F2 region the electron density is slightly reduced but does not disappear. High level of the electron density is maintained by ionization diffused solar radiation, high-energy particle precipitation and inflow of charged particles from the higher altitudes [3].

Short waves refract in the F1 and F2 layers and return to Earth, often at great distances from the transmitting stations [11] depending on the incidence angle.

2.2 Short wave propagation

All data for processing and analysis in this work were obtained on a High Frequency (HF) band. HF wave is the band of 3 MHz to 30 MHz and wavelengths from 10 to 100m, denominated short waves (SW).

Short waves can propagate as terrestrial waves and ionospheric waves. Terrestrial waves are usually of short range – up to 50 km – because they are absorbed by damping of the Earth's surface. The absorption increases rapidly with increasing frequency [14].

However, the ionospheric waves can be transmitted over very long distances (up to the point antipode of about 20 000 km) by multiple reflections from the ionosphere and the Earth's surface. Since the ionosphere is non-uniform, the waves propagate along curved paths. In such conditions the radio waves undergo total internal reflection from the ionosphere and return to Earth.

2.2.1 Reflection conditions HF radio wave

To simplify, the ionosphere can be represented in the form of thin layers with different electron concentration but constant concentration within a single layer as shown in figure 2.2. The concentration increases to a maximum and then decreases again.

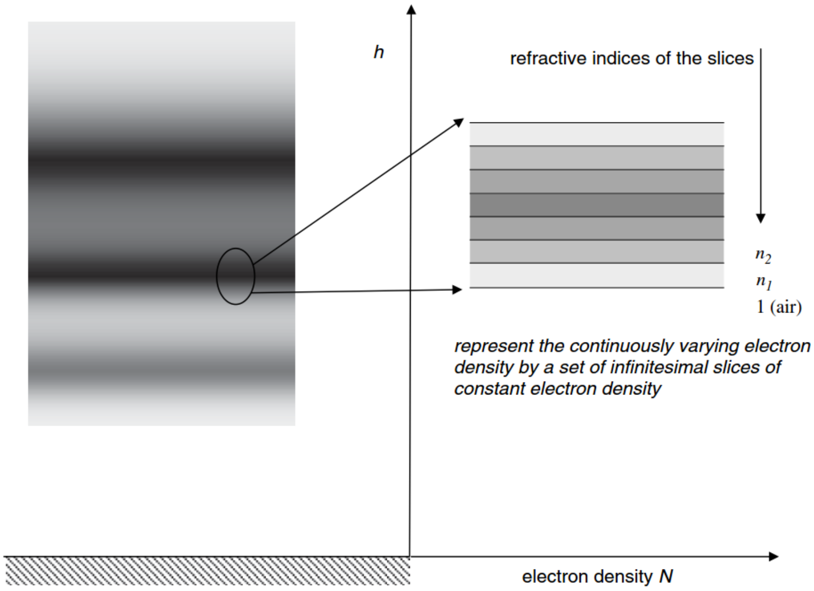

Figure 2.2: Presentation of the ionosphere in the form of thin layers [14].

The refractive index in each layer is determined by the formula [9, 14]:

$$n = \sqrt{1 - \frac{81 \cdot N}{f^2}} \tag{2.1}$$

where: N is electron density, f is frequency of the incident radio wave.

Thus, the refractive index will change from 1 up to some minimum value (at the ionization maximum) and then increase again.

Figure 2.3 shows the path of an electromagnetic wave through a layered ionosphere with angle of incidence φ.

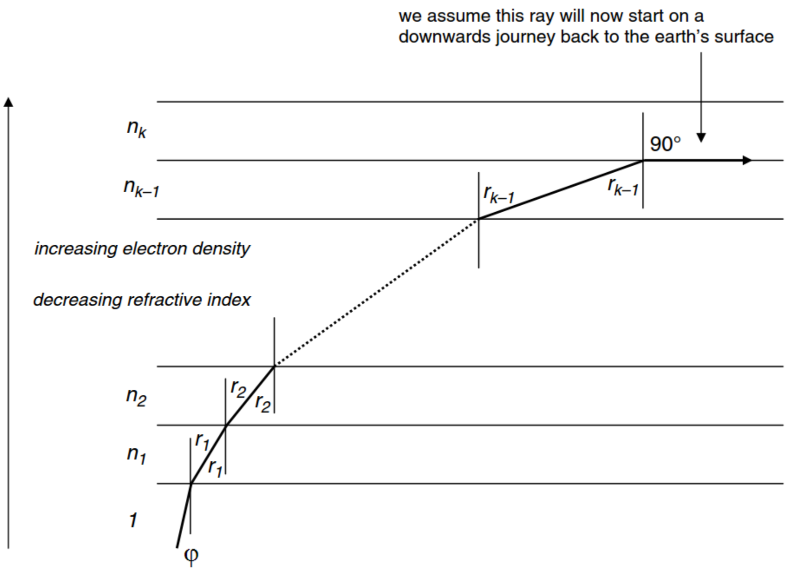

Figure 2.3: Refraction of radio wave [14].

After a sufficient number of refractions the angle of incidence at the k-th layer can reach 90º. This is the condition for reflection back to Earth [2, 14]. The condition can be expressed by applying the law of sines to each boundary between layers:

$$1 \cdot \sin\varphi = n_1 \cdot \sin r_1 = n_2 \cdot \sin r_2 = \cdots = n_{k-1} \cdot \sin r_{k-1} = n_k \cdot \sin 90$$

$$\sin\varphi = n_k \qquad (2.2)$$

or:

$$\sin\varphi = \sqrt{1 - \frac{81 \cdot N}{f^2}} \qquad (2.3)$$

Thus, when the angle of incidence φ = 90º then sin φ = 1. This means that the electron density level can be very little to reflect the wave [14].

When the incidence angle is reduced (increased elevation angle β) the wave penetrates deeper into the layers and the ionization level should be greater to achieve the desired refractive index n for the reflected wave back to Earth. If the maximum electron density does not satisfy the reflection condition then the wave will pass the ionosphere without reflection (escape rays) as shown in figure 2.4.

If the wave propagates vertically (i.e. φ = 0) the reflection condition for wave frequency f will look like this:

$$\sin\varphi = \sqrt{1 - \frac{81 \cdot N}{f^2}} = 0 \qquad (2.4)$$

$$f = 9 \cdot \sqrt{N} \qquad (2.5)$$

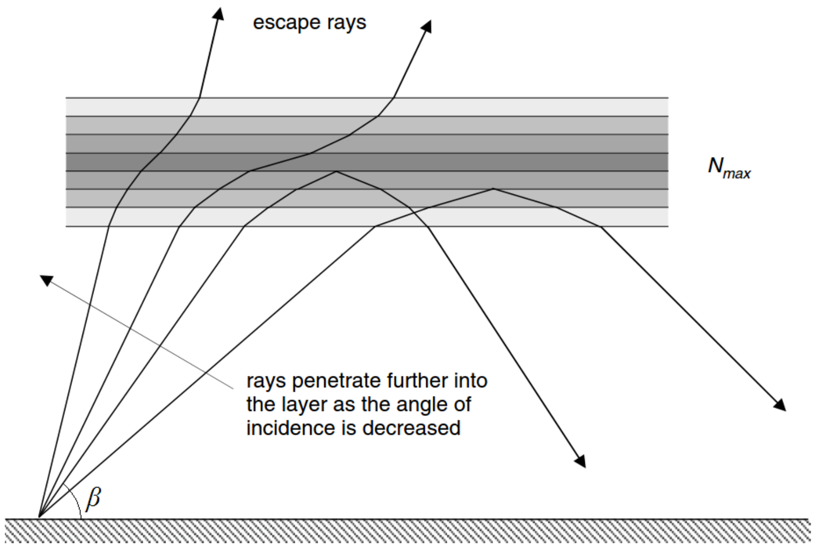

Figure 2.4: The behavior of waves at different angles of elevation β [14].

Using expressions 2.5, we can determine the maximum electron density in the ionosphere N_{max}. Then the frequency of the vertical incident wave and reflected back to Earth is called the critical frequency f_0 [9, 14]:

$$f_0 = 9 \cdot \sqrt{N_{max}} \qquad (2.6)$$

For oblique waves at incidence angle φ, the maximum frequency of the oblique wave (f_{ob}) which can be reflect from the same layer as the vertical wave (f_v), is given by secant law:

$$f_{ob} = f_v \cdot \sec\varphi \tag{2.7}$$

For example, the F layer has maximum electron density about 10^{12} electrons/m^3 and the maximum frequency of a wave that is reflected from the ionosphere at vertical incidence (φ = 0) will be: $f_0 = 9 \cdot \sqrt{10^{12}} = 9$ MHz.

2.2.2 Skip zones

As previously noted, the HF terrestrial waves suffer high absorption from the ground and do not propagate over long-distances. An ionospheric wave with elevation angle greater than the critical angle is not reflected but passes thru the top of the ionosphere (see figure 2.5). But if the nearest beam has the angle β less than the critical value, than this beam can be reflected from the ionosphere and get to the point C in figure 2.5. Rays with smaller angles reach, respectively, more distant point on the surface of the Earth [1].

At point B, the field strength of the ground wave has a threshold in the sense that the receiving of terrestrial waves at distances greater than point B is not possible. Then the region BC will represent the length of the skip zone, in which there is no reception of signals.

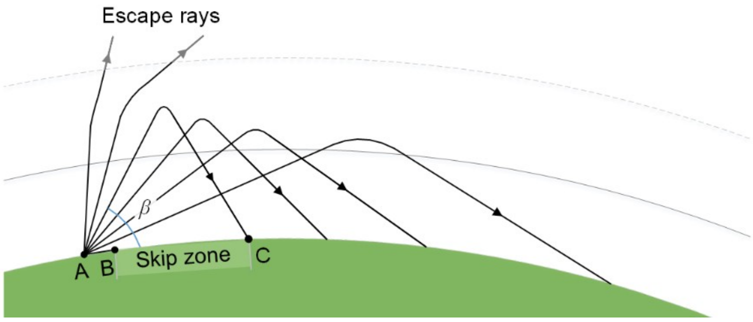

Figure 2.5: The skip zone.

The outer length of the skip zone is determined by the conditions of the ionospheric waves propagation: it depends on the time of day and frequency. The maximum frequency of the waves reflected from the ionosphere is determined by the formula 2.8 [2].

$$f_{max} = \sqrt{1 - \frac{81 \cdot N_{max}(1 + 2h/a)}{\sin^2\beta + 2h/a}} \tag{2.8}$$

where: a – the radius of Earth, h – the height of the transmitter point, N_{max} – maximum of electron density.

For a given frequency and decrease of the electron density due to the onset of darkness, as follows from 2.8, to save the constant values of frequency f_{max} the angle β should be reduced. Consequently, when it is dark the ionosphere can reflect only the rays with a lower elevation angle, which fall into more remote locations. By nightfall, the radius of the skip zone increases.

2.2.3 Features of propagation of short waves

Under normal conditions, for propagation of ionospheric waves, each of the regions of the ionosphere performs a well-defined function: the D and E layers are absorbing, the F2- layer is reflecting. Electron density of the E region is not sufficient to reflect short waves. Absorption of short waves in the layer F2 is much less than the absorption through the D and E regions (on the round path) [9].

However, during daytime for path lengths 2000-3000 km (meaning small angle β) often the E region is reflective. Under these conditions, the main absorption occurs in the D layer. Finally, the propagation of short radio waves can be disturbed in the event of sporadic E_s layer. Sporadic E_s layers can occur at any time of the day and has in some cases very large electron density and becomes a reflective layer. Sporadic E_s layers are usually located in the underside of the main E layer. Consequently, under conditions of high density E_s layers the waves do not reach the F layer.

2.3 Short wave method for observation of the ionosphere

2.3.1 The method of vertical sounding

Experimental research of the ionosphere began with the measurements of the vertical distribution of the electron density by vertical sounding (VS).

The ability of the ionosphere to reflect radio waves was used to determine the travel time to the point of reflection and back as a function of frequency. Sounding means the travel time depends on the frequency of the probing wave, which increases monotonically [13]. The time delay of a given frequency gives its altitude of reflection and the frequency gives the electron density at that altitude. The graph of this dependence is called ionogram. The height, determined by the delay time assumes that the wave travels at the speed of light. This is a very rough approximation, and the corresponding height is called virtual or effective height [15]. There are methods of converting virtual height to true height [16], for example, by the method of thin layers [2, 17]. In this method, the contribution of each thin layer in real height is determined by the average refractive index in the layer. The refractive index is calculated for all frequencies. The real height for frequency f is obtained by summing the thin layers up to the reflection height.

Regular ionospheric sounding, started over 80 years ago, is currently used in many places. VS was the only method of research of the ionosphere for a long time. The main bulk of information about the behavior the electron density at the altitudes of regions E (h ~ 90-140 km) and F1 (h ~ 140-200 km) up to a maximum layer F2 (h ~ 200- 400 km) was gained with this method.

The VS method, however, has its limitations, stimulating the search for other measuring tools. The VS method, for example, based on the use of short waves ($f \geq 1$ MHz) cannot be used to investigate the D region (h ~ 50-90 km), which reflect long and middle wave lengths [3, 11]. VS cannot measure in regions of decreasing electron density such as above the peak of the F layer. Another limitation of the method is that it gives information about the ionosphere only above the measuring station.

The standard equipment that works on these principles is called Ionosonde. The ionosonde is a unit consisting of a transmitter, a receiver, and control devices. The transmitter radiates sounding pulses of tens of microseconds with peak power of one to tens of kilowatts, the carrier frequency of the pulses increases during the measurement from 0.5 to about 25 MHz [8]. The time delay of the reflected signal is recorded as a function of frequency.

Currently more stations use modern Digital Ionosondes. Many of these ionosondes are portable and they have a much smaller sizes, weight, and power consumption than standard analog ionosondes. Digital Ionosondes provide much better ionograms [4]. Modern ionosondes have the ability to separate the ordinary from the extraordinary ray, can measure the Doppler shift signals coming from different directions, they make easier to remove discrete frequencies with noise, and to remove interfering signals from other services.

2.3.2 The oblique sounding method (OS)

The desire to receive information about horizontal distribution of ionospheric characteristics led to the creation a method of oblique sounding (OS). In the OS method, the transmitter and receiver are spatially separated and the radio wave, obliquely incident on the ionosphere, creates a connection between the transmitter and receiver. Figure 2.5 shows a diagram of oblique sounding with two alternative paths for the radio signal.

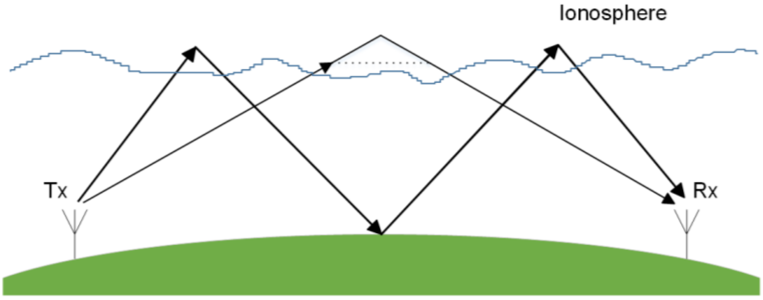

Figure 2.5: The diagram of oblique sounding [4].

This method can also provide ionograms from the region of reflection of the radio waves, so-called oblique ionograms [17]. These regions can be located over the oceans, marshes, where there is no possibility to carry out vertical sounding. The oblique ionograms can be converted into vertical ionograms by using the secant law (see equation 2.7) and the Martyn's equivalent path theorem [2, 18]. Figure 2.6 shows the geometry of the vertical and oblique sounders.

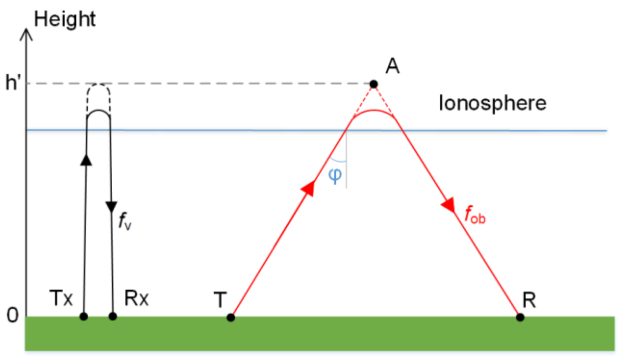

Figure 2.6: Applying the secant law and the Martyn's equivalent path theorem.

According to the secant law the frequency of the reflected oblique wave (f_{ob}) will be from the same height as the equivalent frequency of the vertical wave (f_v) (see figure 2.6). The equivalent frequency f_v gives information about the electron density at the reflection point

(A) of the ionosphere. According to Martyn's theorem the local virtual height (h′) of the reflection point of the oblique wave (f_{ob}) is equal to the height of the triangle TAR (see figure 2.6). The group path of the oblique wave (p′ = TA+AR) is determinated by the wave propagation time from the transmitter (point T) to the receiver (point R). The angle φ is obtained from: φ=arcsin(TR/p′). The local virtual height (h′) of the reflection point is:

$$h'(f_v) = \frac{1}{2} p'(f_{ob}) \cdot cos\,\varphi \qquad (2.9)$$

Oblique sounders employ frequencies ranging from 1 MHz to more than 20 MHz, since oblique waves can be reflected for higher frequencies compared to vertical sounders. The distant receiver receives the reflected wave [4]. To determine the exact signal delay the transmitter and the receiver must have accurate time synchronization. Typically, synchronization is achieved by using chirp techniques. The sounder receiver generates an internal chirped reference signal, which has the same form as the transmitter signal. This reference signal is mixed with the received signal. By carrying out a frequency analysis of the mixed signal we can calculate the relative delays of the different propagating frequencies [19]. Alternatively, synchronization can be achieved using GPS clocks.

For one-hop signal propagation the oblique ionogram shows the reflection frequency from the ionosphere above the mid-point of the radio path (from which electron densities can be obtained) [18, 20]. From this ionogram one can determine the maximum of reflection frequency also called the maximum usable frequency (MUF) for a given radio path.

For two-hop signal propagation, the oblique ionogram shows the reflected frequencies from the ionosphere in two regions that we can used determine the electron density profile and the usable frequencies for such a radio path. These frequencies will be lower than for the one-hop radio path according to the secant law (see expression 2.7) since the incident angle of the wave into the ionosphere is less. The delay for two-hop propagation will be greater as the signal travels a greater distance than the one-hop propagation [4, 17].

In the OS method, the placement of transmitter and receiver determines the position of the investigated region of the ionosphere, so it is fixed. Therefore, the OS method permits to determine the electron density profile above the fixed-point of the radio path and the usable frequencies for the propagation of radio waves for a given path.

2.3.3 The oblique backscatter sounding method

Like the oblique sounding method, the oblique backscatter sounding method allows to investigate the variations of the ionosphere along the horizontal direction. The essence of this method is as follows [3]: the transmitter emits the pulsed radio signal; the signal is forward reflected from the ionosphere and reaches the Earth's surface where it is scattered; a portion of the scattered signal returns to the transmitter/receiver via the same ionospheric path as shown in figure 2.6. The scattered signal parameters, such as signal intensity and time delay in relation to the probe pulse shape and distribution and their dependence on the direction of transmission and reception, enable us to obtain information about the distribution of the electron density of the ionosphere at different points. Continuous, systematic changes in the direction of transmission allow us to investigate the properties of the ionosphere and the Earth's surface over a large territory.

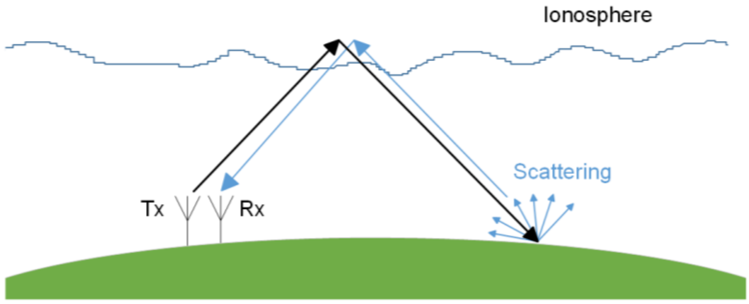

Figure 2.6: The diagram of oblique backscatter sounding.

2.3.4 Method sounding from space

The method of sounding from space is more effective for the study of horizontal variations [2, 5]. It had a great influence on the development of ionospheric science. This method is implemented by ionosondes on-board satellites. It provides a picture of the distribution of the ionospheric plasma below the satellite's orbit.

Ionosondes can give information about the ionosphere only along paths in which the electron density increases with distance from the measuring station. The ground ionosonde station explores the ionosphere only below the main peak the electron density, the satellite – above the same peak. The maximum is a common point, which is registered by the satellite and a ground station, and can be used to match the ground and satellite ionograms [3, 4].

2.3.5 The method of short-wave Doppler measurements

This method is very effective for the study of small perturbations of the reflective layer. Typically, the transmitter and receiver are located in one place and sounding is carried out vertically. If the reflective layer is moving upward with velocity V, the frequency of the reflected signal experiences a Doppler shift $\Delta f = 2fV/c$, where: f — the transmitter frequency, c — propagation velocity of radio the wave. Consequently, $V = 2\lambda\Delta f$, where λ — wave length. For example, if $\lambda = 100$ m and $\Delta f = 0.5$ Hz, then V=100 m/s. The short-wave Doppler system uses continuous radiation. The received signal is mixed with a reference signal at the transmitter frequency and the result of the beats is measured. Thus, we can estimate the vertical velocity of the reflecting layers and relative changes of the height of reflection [15]. The absolute height of the reflection is usually determined by an ionosonde.

2.3.6 The method of partial reflection

The partial reflection method is useful for investigations the lower ionosphere inaccessible to most of the usual methods (the sounding by mirror reflected waves, the measurements from satellites). It uses partial reflection of radio waves in the HF-range from refractive index irregularities in the region D of the ionosphere [21] generated by turbulent motions in the atmosphere. Partial reflection is weak, its amplitude is only 10^{-3}–10^{-5} weaker than the amplitude of total reflection, so, for sounding it is necessary to use powerful transmitters (up to 100 kW) and large antenna systems [15].

2.4 The method of total electron content

The total electron content (TEC) is one of the parameters of the Earth's ionosphere and is defined as the total number of electrons in a column of ionosphere of unit area from the ground to a height well above the peak of ionization, to at least 1000 km. TEC usually is described as a vertical column of one square meter in area. One TEC Unit or TECU is equal to 10^{16} electron/m^2

With the development of GPS satellite navigation systems it has become possible to determine the TEC between a satellite and a ground-based GPS receiver [22].

The method of obtaining TEC from GPS is as follows [23]. Each GPS satellite transmits information for positioning at two signals: L1 and L2, with carrier frequencies 1.57542 and 1.22760 GHz respectively. Main observed GPS parameters are so-called the pseudorange (P) and carrier phase. These parameters are used for slant range measurement between a GPS- receiver and a GPS-satellite. The ionosphere is a dispersive environment, which affects radio signals: refracts them, absorbs and scatters. This influence is evident in the phase shift and delay of signals. The difference between L1 and L2, allows to extract information about TEC from two-frequency GPS data. The phase data provides relative change of TEC by comparing the phase delays of the L1 and L2 signals. The 1 ns of differential time delay corresponds to 2.852 TECU. However, the phase delays do not give the level of TEC because the initial phase is unknown. The level of TEC is derived from the corresponding pseudorange difference (P2 - P1) for each satellite-receiver pair.

TEC depends on geographic location, local time, season, sun spot number, solar EUV (extreme ultraviolet) radiation and magnetic activity [24]. TEC is one of the key parameters in determining the impact of the ionosphere on the propagation of radio signals between satellites and ground stations. TEC determines the quality of numerous communication links, navigation and surveillance systems [25]. Global monitoring of TEC and extensive database also allows to predict such terrible events as earthquakes [26]. Physical mechanism of the seismo-ionospheric precursors appearance is mainly based on the hypothesis of electric fields of seismogenic origin related to vertical turbulent transportation of the injected aerosols and radioactive particles (radon isotopes). The increase of the atmospheric radioactivity level during the earthquake preparation leads to the enlargement of the ionization and electric conductivity of the near-ground atmosphere.

3 Instrumentations

This chapter describes the basic tools that were used to obtain data for this thesis. The first section describes the sources of data and the geometry of the observations. Section 3.2 describes the coherent HF receiving equipment: their characteristics, parts, features and locations. Section 3.3 describes the transmitter of the coherent HF radio signal by the Moscow radio station RWM. Section 3.4 describes the modern ionosondes, which provided data on the critical frequency of the ionosphere. Section 3.5 provides brief information about the GPS receiver at the LFO station in Ukraine.

3.1 The data sources and the geometry of observations

To study the relationships between ionospheric and HF signal parameters a long term monitoring of 2013 year was performed of the coherent HF signal at 9.996 MHz from the Moscow RWM radio station (Russian exact time and frequency service) for radio paths Moscow-Tromso (Norway polar region) and Moscow-Kharkiv (Ukraine, Low Frequency Observatory, mid-latitude region). The geometry of observations is shown in figure 3.7.

The recording of the HF signal was done at coherent receiver stations in Tromso, Norway, and at the Low Frequency Observatory (LFO) in Ukraine.

To compare the characteristics of HF signals with ionospheric, solar and geomagnetic parameters the data from other external sources were used. These sources and parameters are as follows:

- EISCAT (Ramfjordmoen – near Tromso, 69.6°N, 19.2°E) and IZMIRAN (Moscow, 55.47°N, 37.3°E) ionosondes (critical frequency of F2 layer F0F2);
- LFO (49°56'01"N, 36°57'10"E, local magnetic K-indices, total electron content);
- Joensuu, Finland (62.3911° N, 30.0961° E, GPS data for TEC [27]);
- Kiruna, Sweden (67.8°N 21°E, GPS data for TEC [28]);
- Tromso (69.66°N, 18.93°E, GPS data for TEC [29]);
- Tromso Geophysical Observatory (69.67°N, 18.95°E, local magnetic K-indices) [6];
- Space Weather Prediction Center national oceanic and atmospheric administration (Planetary Kp geomagnetic index, Global solar index - $F_{10.7}$, Sun sports number, X-ray background flux within 1-8 Å wavelength (GOES satellite data)) [7].

For the path Moscow-Tromso (Norway polar region) the point of reflection HF waves is located approximately over Joensuu, Finland during the long-term monitoring of 2013 year. In addition, the second region of the ionosphere is located between Moscow and Kharkiv (LFO).

The processed TEC data was obtained from the database of the Institute of Radio Astronomy of Kharkiv. For shorter path Moscow-Kharkiv, the TEC data for the mid-point is not available, but because of the relatively small length of the path the data from LFO is enough. Also the F0F2 data for LFO is not available, so we used data from the Moscow ionosonde. We suggest that for a short radio path the F0F2 in both places, in general, have a similar response to disturbances in the ionosphere.

The Kiruna data gives additional information about the polar region and was useful when there was no data for Tromso.

Figure 3.1: Geometry of observations.

3.2 Coherent HF receiving equipment

For received radio waves reflected from the ionosphere we used coherent receiver stations (RS) in 2 locations:
- LFO, Ukraine (49°56'0.5" N, 36°57'10" E);
- Tromso, Norway (69°39'41" N, 18°56'30" E).

The receiver station has name of “coherent”, because it includes the high stability reference oscillator that allows measure parameters of coherent signal: exact frequency, phase, intensity. In this work we don’t use measure the phase.

These receiver stations are permanently connected to the Internet allowing remote control and data transfer to the central server in Kharkiv.

The hardware part of the receiver stations includes:
1) The digital HF receiver module WR-G313i;
2) Desktop personal computer;
3) The oscillator reference frequency 10 MHz;
4) The active frame HF antenna.

All modules of the receiver stations, except the receiving antenna, are mounted inside the computer unit providing compact units.

3.2.1 The digital HF receiver module WR-G313i

The central element of the hardware in the receiver stations is the digital receiver, model: WR-G313i from WiNRADiO Communications company (Australia). The receiver module is built on a board with a PCI bus inserted in the PCI expansion slot of the motherboard of a personal computer (PC). The receiver module WR-G313i is shown in figures 3.2 and 3.3.

The receiver WR-G313i is fully software controlled and belongs to a class of equipment known as Software-defined radio (SDR). The receiver modules are controlled by the

WinRadioMetr program. This application is launched together with the operating system of PC, then initializes the receiving modules and puts the system in waiting mode for commands. The program reads the commands from the schedule files, which are locally produced or downloaded from the main server. More about the WinRadioMetr program is discussed later in section 5.1.

The number of devices installed in the PC are limited by the number of free PCI slots allowing us to make multi-channel receiving stations.

Figure 3.2: Digital HF receiver WR-G313i [30].

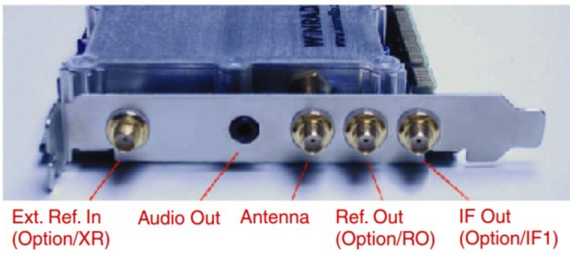

Figure 3.3: Connectors of digital HF receiver WR-G313i.

Main technical characteristics of the device are presented in table 3.1 [30].

Table 3.1: Main technical characteristics of WR-G313i

Receiver type	DSP-based SDR
Frequency range	9 kHz - 30 MHz
Tuning resolution	1 Hz
Spurious-free dynamic range	112 dB
Sensitivity	0.05 µV
Bandwidth	1 - 15000 Hz (adjustable in 1 Hz steps)
Intermediate frequencies	IF1: 45 MHz IF2: 16 kHz (variable 12-22 kHz)
Roofing filter	2 x 4-pole 15 kHz crystal filter
Antenna input	50 ohm (SMA connector)
Form factor	2/3 length PCI card (PCI 2.2 compliant)
Weight	330 g

WR-G313i functional scheme is shown in figure 3.4.

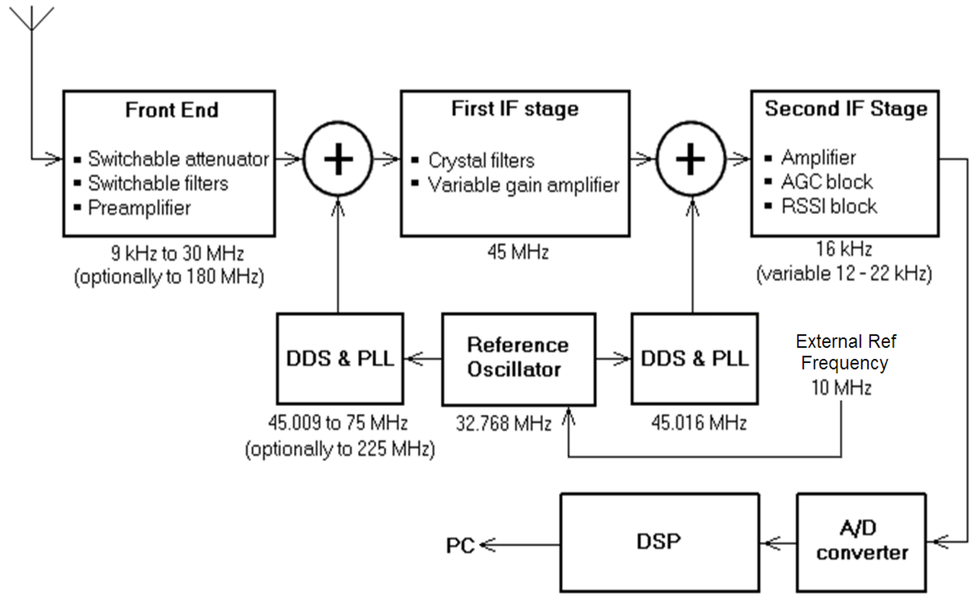

Figure 3.4: Functional scheme of the receiver module WR-G313i [31].

The WR-G313i receiver module is a superheterodyne receiver with two hardware intermediate frequencies (First IF and Second IF). The heterodyne reception is based on Direct Digital Synthesizers (DDS) with Phase Locked Loop (PLL) and Reference oscillator. A third frequency conversion is digital and is performed using the built-in digital signal processor (DSP), which is intended for in-memory transfers and demodulation of the waveform signal. This processing scheme does not used the computer resources and allows to build the multi-channel receiver using average performance PC.

In this work the signals from the Moscow station RWM were obtained using the following settings of the DSP: filter bandwidth of 1 kHz and sampling frequency of 2 kHz.

3.2.2 The oscillator reference frequency 10 MHz

The oscillator reference frequency 10 MHz is a thermally stabilized crystal oscillator OCXO 131-1000 (firm Isotemp Research), which has a relative stability better than 10^{-8}. The device is equipped with an external panel (mounted on the rear of the PC case), which has connectors for output frequency signal of 10 MHz and supply voltage to the antenna amplifier. The oscillator is required for precise Doppler and phase measurements.

It should be noted that each receiver has a constant error in the measurement of the Doppler frequency. It arises from the fact that the reference oscillator is not exactly 10 MHz but it can be for example 9.999999 MHz or 10.000001MHz or some other constant value. This error is defined in the installation and calibration of the receiver. This error was taken into account during the processing of data in the MatLab code.

3.2.3 The active frame HF antenna

The frame HF antenna with its antenna amplifier is shown in figure 3.5. The antenna consists of the frame (1) with a diameter of 70 cm and the wideband antenna amplifier (2) with a gain of ~ 10 dB. The antenna pattern (AP) has the form of an eight with a maximum in the frame plane. The width of the main lobe of the AP at the 0.5 level of power is at least 90°.

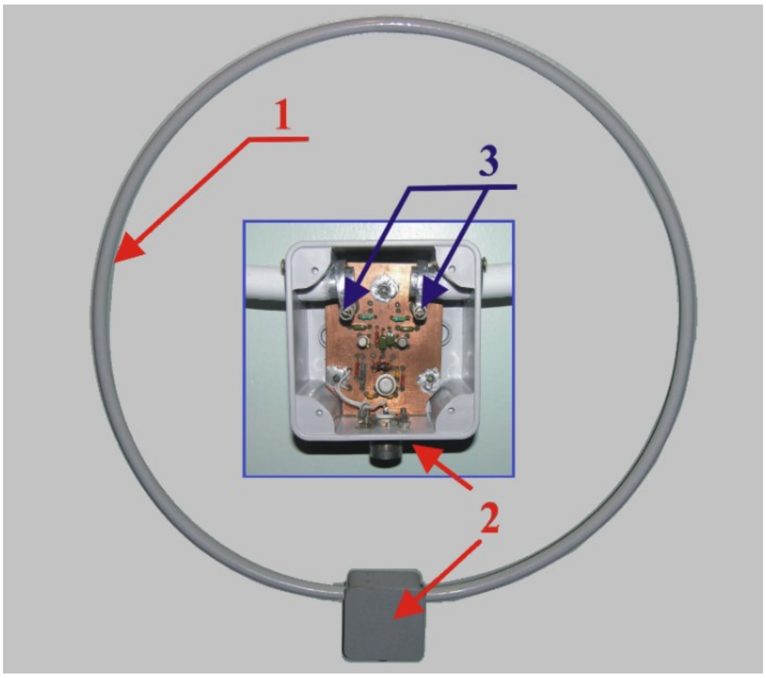

Figure 3.5: The active frame antenna: 1 – the frame; 2 – the antenna amplifier; 3 – the mounting screws.

3.2.4 The network of receivers

The receiving points at LFO and Tromso are part of a network of the receiver stations for Doppler ionospheric sounding [32]. All of them are presented in table 3.2.

Thanks to the extensive network of receivers, the behavior of the spectra of high frequency (HF) signals on the radio paths of different lengths and orientations, including long-range and ultra-long (more than 10,000 km) [33], was investigated. This allowed studying the characteristics of various ionospheric processes, for example the dynamic and statistical parameters of traveling ionospheric disturbances [34], the influence of the eclipse [35], the decaying artificial field-aligned irregularities [36].

The method of multi-position ionospheric Doppler measurements, which was developed in the Institute of Radio Astronomy (IRA) of Kharkiv, allows use to diagnose the numerous ordinary HF broadcast stations [32]. According to the International Telecommunication Union more than 5000 broadcast transmitters operate in the world. This makes it possible to perform Doppler measurements in almost any region of the globe using existing transmitters.

The locations the receiver stations are shown in figure 3.6

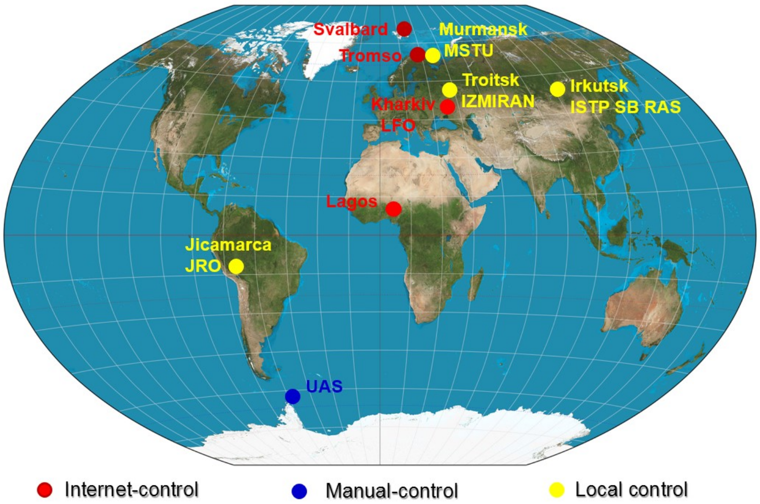

Figure 3.6: Map of location receiver stations.

Table 3.2: The network of receiver stations

Locations of the receiver stations	Coordinates	The characteristics of the RS
The Kjell Henriksen Observatory (KHO), the University Centre in Svalbard (UNIS)	78°08′53 N, 16°02′35″ E, Svalbard, Longyearbyen, Norway	double-channel, Internet control
Faculty of Science and Technology, the Arctic University of Norway (UiT)	69°39′41″ N, 18°56′30″ E, Tromso, Norway	double-channel, Internet control
Low Frequency Observatory (LFO), Institute of Radio Astronomy (IRA)	49°56′01″ N, 36°57′10″ E Kharkiv, Ukraine	one-channel, Internet control
University of Lagos (UNILAG)	06°26′ N, 03°25′ E, Lagos, Nigeria	one-channel, Internet control
Ukrainian Antarctic Station, Vernadsky Research Base (UAS)	65°14′44″ S, 64°15′29″ W, Galindez, Antarctic	one-channel, manual control
The Jicamarca Radio Observatory (JRO)	11°57′05″S, 76°52′27″W, Lima, Peru	one-channel, local control
Institute of Solar-Terrestrial Physics Siberian Branch of the Russian Academy of Sciences (ISTP SB RAS)	52°45′ N, 103°49′ E, Irkutsk, Russia	one-channel, local control
Institute of Terrestrial Magnetism, Ionosphere and Radio Wave Propagation of the Russian Academy of Sciences (IZMIRAN)	55°28′39″N, 37°18′53″ E, Troitsk, Russia	double-channel, local control
Murmansk State Technical University (MSTU)	68°57′17″ N, 33°03′44″ E, Murmansk, Russia	double-channel, local control

3.3 HF transmitter in Moscow

The source of coherent radio signal with stable frequency, intensity and phase which is received in Tromso and LFO is the HF radio station – called “Standard frequency and time signal station on shortwave” with call signature RWM. The station is located in Moscow, Russia (55°44'N, 38°12'E). The transmitter frequency is 9.996 MHz and the transmitter power is 8 kW [37]. The relative stability of frequency is better than $5 \cdot 10^{-12}$, the initial phase is zero [38].

The structure of the emitted signal is: from 0 to 7 minutes 55 seconds each half an hour - continuous unmodulated signal with stable frequency and amplitude, followed by 65 seconds without transmission, and the pulse signal during the remaining time from the 9th to the 30th minute. The pulse signal contains the identification station information in Morse code, the information about exact time and difference between Local Time (Moscow) and Universal Time. For the processing in this work we used data only in continuous operation mode.

3.4 Ionosondes

The critical frequency of the ionosphere data were taken from the ionosonde at the EISCAT station in Tromso and the ionosonde of IZMIRAN near Moscow.

At EISCAT Tromso there is a modern digital ionosonde – a Dynasonde [39]. Its characteristics are given in table 3.3 [40].

Table 3.3: Technical characteristic of Dynasonde EISCAT Tromso

Location	Ramfjordmoen, 69.6° N, 19.2°E
Antenna type	Transmit: vertex-down log-periodic; Receive: 6-dipoles
Pulse repetition frequency	100 Hz, programmable
Pulse type	Single Gaussian, 64 μs duration
Peak power (normal operation)	10 kW
Continuous power	200-300W
Range resolution	0.1 km (from stationary phase)
Frequency coverage	Typically 1-12.0 MHz, but programmable
Frequency resolution	0.4%, programmable
Sweep time (typical frequency range)	3 min, programmable

The Dynasonde is a very powerful and flexible tool for the investigation of the ionosphere. The instrument measures the time-of-flight, amplitude and phase of pulsed radio signals reflected back from the ionosphere [40]. The use of two parallel receivers and a small array of spaced receiving antennas allow the echo phase measurements to be processed to yield estimates of the polarization, Doppler shift and angle of arrival of the echoes for each transmitted frequency. The angles of arrival combined with the time-of-flight give the originating location in the sky for each echo. These echo locations enable the formation of a sky map for each ionogram. The Dynasonde Digital Ionogram is a collective assembly of all the echoes with their physical properties. Ionograms are recorded at regular intervals, typically every 12 min, during the year.

The second instrument is the Digisonde DPS-4, which is located near Moscow, Russia. Its characteristics [41] are presented in table 3.4.

Table 3.4: Characteristic of Digisonde DPS-4

Location	Moscow, Russia (55.47°N, 37.3°E)
Frequency Range	0.5 - 30 MHz (all modes of operation)
Antenna type	Transmit: turnstile Delta – 2 orthogonal radiating elements, 30 m tower; Receive: active crossed loops – turnstile antennas (1.5m diameters), 4 antennas in 60 m triangle with central antenna
Pulse repetition frequency	100Hz and 200 Hz
Pulse type	16 bit Complementary code, duration 533 μs (16 chips of 33 μs)
Peak pulse power	2 channels by 150 W each
Range resolution	2.5 km sample spacing, 500 m using differential phase technique
Frequency resolution	step size selectable to 1 kHz
Ionogram scan time	2-200 sec, programmable

The Digisonde is a portable vertical sounding system [41]. It is able to measure seven parameters of the echoes received from the ionosphere such as the frequency, the height of reflection, the amplitude, the phase, Doppler shift, the angle of arrival and polarization of the wave.

For the purposes of portability, the Digisonde has a peak output power 300W. For high-energy needs a long pulse is used, but then we cannot get a good height resolution. However, thanks to the use of 16 bit Complementary code with phase modulation we can get great height resolution with long sounding pulse (533 μs) [41]. Yet such a decision imposes a restriction on the lower limit of the measurement height 80 km.

3.5 GPS receiver

Data about the total electron content over Ukraine was obtained using a dual frequency receiver GPS Ashtech-MD12. The receiver is located at LFO (49°56'01"N, 36°57'10"E). The method by the Institute of Radio Astronomy got TEC by GPS data is briefly described in Section 2.4. Signals from the GPS satellites are recieved in 12 parallel channels. Measure parameters: the code and phase of the carrier signal L1 at frequency 1.57542 GHz and the carrier signal L2 at frequency 1.22760 GHz. The receiver is connected to the computer. The rate of registration of primary data is every five seconds.

Figure 3.7 shows GPS receiver Ashtech-MD12.

Figure 3.7: GPS receiver Ashtech-MD12 [42].

4 Data processing

This chapter describes the data structures and processing procedures. Section 4.1 describes the main program to control the receiver WR-G313i, the recording of the received signal and a description of the structure of the signal data. Section 4.2 describes the data processing for determining the signal to noise ratio of the received signal, the signal intensity and Doppler frequency shift. Section 4.3 describes the data structures for plotting intensity and Doppler frequency shift, the critical frequencies, TEC, the Earth's magnetic activity, solar activity. Data processing in this work was performed by programs written in MatLab.

4.1 WinRadioMetr

WinRadioMetr is the name of the main program to control the digital receiver WR-G313i and writing the receiver data to the PC hard drive. The application WinRadioMetr has been developed by the Institute of Radio Astronomy (IRA). The application supports independent work up to 3 receiving modules WR-G313i simultaneously.

WinRadioMetr has the following functionality:

- frequency tuning of the received HF signal (tuning step 1 Hz);
- control of the input attenuator (18 dB attenuation);
- tuning of the gain factor (range 0 to 90 dB in 10 dB steps);
- control of the bandwidth of the output digital filter (0.5 to 22 kHz);
- setting of the signal digitizing frequency (allowable digitizing rate 1, 2, 4, 8, 16, 48 kHz);
- generation and recording of output data files and log files of setting changes.

The main window of the program with an example of receiving module settings, the form of the spectrum and the signal are shown in figure 4.1.

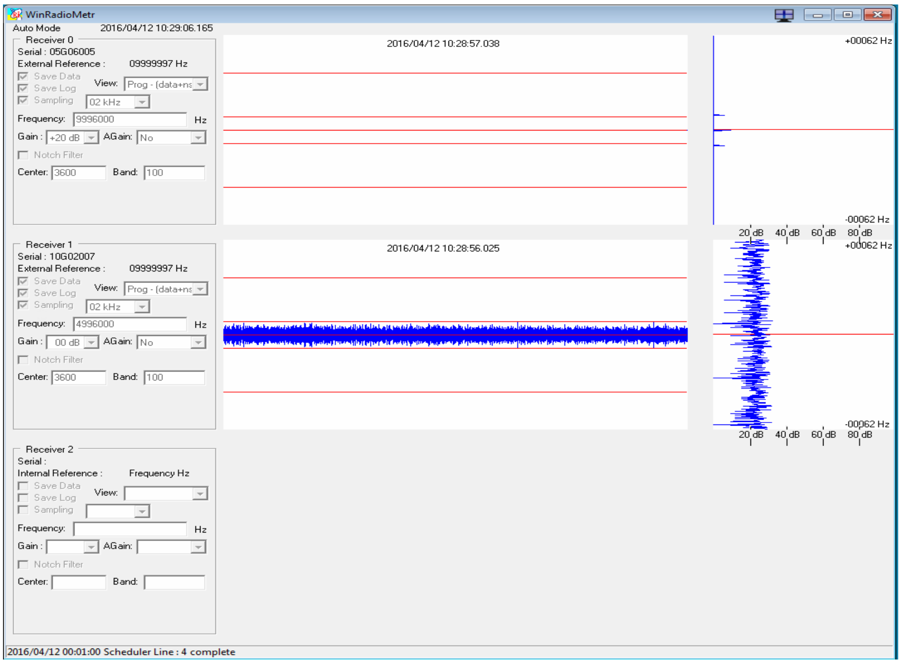

Figure 4.1: The main window of the WinRadioMetr program.

To control the receiver modules, the WinRadioMetr program uses special schedule files. The program get this files from the main server. For receiver stations without Internet access the schedule files are created locally. The schedule files contain a schedule of settings for each receiver module.

WinRadioMetr program generates several types of output files. The main files are the log files of setting changes and the data files that are created during the recording session. These files can be transmitted to a remote server via Internet.

The log files of setting changes contain the history of setting changes of the receiver modules. The data files are divided into two types – signal files containing the received digitalized signals and compact files for visualization containing average spectra and the signals in low resolution.

The WinRadioMetr program is the first step in the processing of the receiver signals. The output of the WinRadioMetr program is an input for further processing.

Signal files contain digitized receiver signals. After three frequency transformations in the receiver, the spectrum of the input signal at 9.996 MHz is mixed down to 500 Hz with a bandwidth 1kHz. This output signal is digitized with the sampling frequency of 2 kHz or equivalently one sample every 1/2000Hz = 500 microseconds recorded in int16 format. Format int16 can contain integer values from -32767 to 32767. The magnitude of these values corresponds to the voltage induced in the receiving antenna, taking into account the gain of the antenna amplifier, the gain or attenuation by hardware components of the receiver module. This single value is 2 bytes. Thus, signals file consists of 2-byte digitized signal values per sample for all recording time, usually one day. Simplified diagram of the signal digitization is shown in figure 4.2.

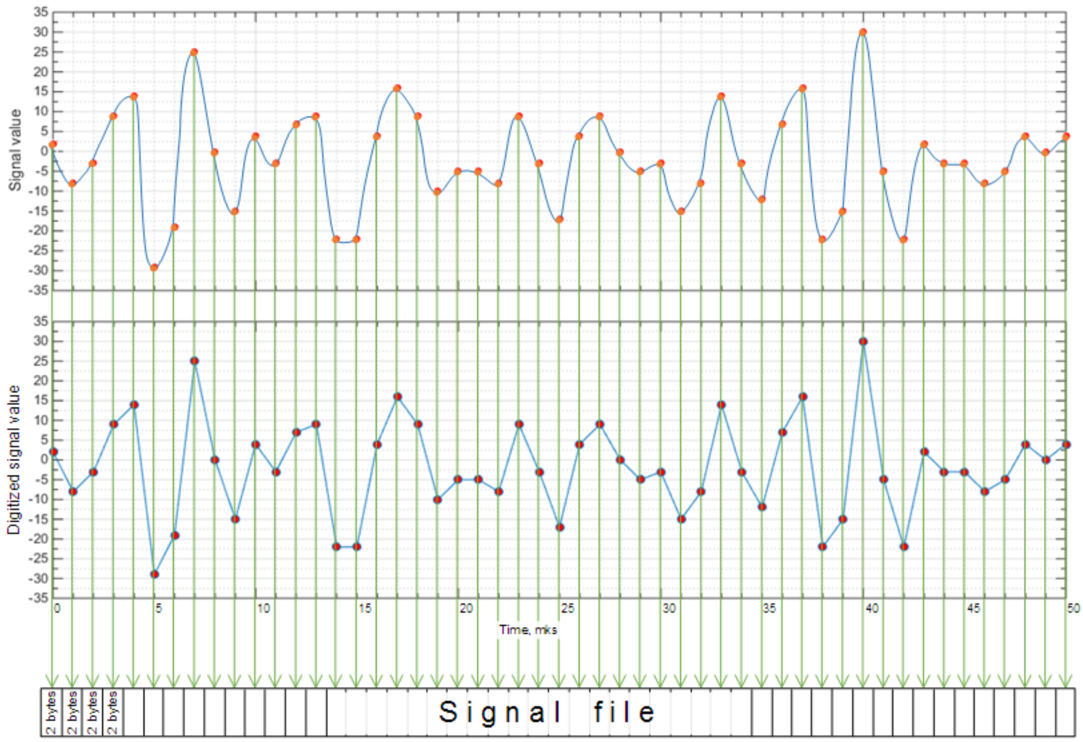

Figure 4.2: Illustration of signal digitization and data record pointed by green arrows.

The name of the signal file has the format: «yearmmddHHMMSS_xxxxxxxx.f02» and contains the date (yearmmdd) and recording start time (HHMMSS), the code name of the receiver module (xxxxxxxx) and the sampling frequency (f02 = 2000 Hz).

Log files of setting changes contain information about an exact time of all setting changes of the receiver module and values of these settings. The name of log files has the following format: «yearmmddHHMMSS_xxxxxxxx.inf».

4.2 Main program to process the signal files

We had written a special MatLab program "Run_RWM_processing" with the main function "Proc_RWM_CW_mod" for processing the signal files and log files. The listing is presented in Appendix A.

The program is able to process data one month at a time. The program calculates the intensity and the Doppler frequency shift of the signal (frequency 9.996 MHz) as a function of time of day based on information in the signal files and in the log files for each day of the month.

The processing program uses the data contained in the first 475 seconds of every half hour of the Moscow RWM station transmissions which is in the continuous mode of operation during that period (the remaining 1325 seconds operates in pulse mode). Thus, the repetition period of the transmissions is half hour or 1800 seconds, therefore it is also the repetion period of the continuous mode.

The program performs the following tests for the correct extraction of data from the signal files:

- comparison of the number of signal files and log files;
- comparison of the names of signal files and log files;
- determination of the presence of recorded data inside signal files;
- determination of the respective date inside the log file and in the file name;
- determination of the presence of the signal at the frequency of 9.996 MHz inside signal files according to data from log files.

When the test for a file fails, the file is not used.

For each date in the month (extracted from the file names) the program creates an array that contains 24 hours of data. This interval of 24 hours is divided into 1-minute segments (24*60=1440 segments per 24 hours day). Each 1-minnute segment has its own timestamp label corresponding to the middle of the minute. For each 1-minute segment the program detects the presence of the signal frequency and attenuation according to the log file.

Data from the signal file are extracted byte by byte. When data is not available from the beginning of the day (00:00:00) or until the end of the day (24:00:00) then the corresponding 1-min. missing segments are filled with zeros. The log file contains the information where are the first and last non-zero 1-minute segments. Also, zeros fill segments for the times when the receiver was not tuned to 9.996 MHz.

To simplify the attenuation correction procedure only 1-min segments are recorded that correspond to a selected value of attenuation, and other data is ignored. Typically, the total time of the signal with other values of attenuation is less than 30 seconds per day, so that significant loss of information does not occur during such attenuation processing.

Because we processed only the continuous signal, then every half hour from 0 to 475 seconds (from 0 to 7m 55 sec) the data are saved only for an integral number of 1-min segments (475/60 = 7.9, so 7 segments). The remaining incomplete segments vanish.

To exclude the possibility of synchronization loss between the RWM station and the processing PC each 1-min segment skips 5 seconds of data at the start and end. After this correction, each half an hour contains seven 1-minute segments each with 50 seconds of data.

The program calculates spectra to determine the signal Doppler frequency shift (DFS) and the signal intensity. For each 50 second segment the power spectrum is calculated. The spectra are calculated by the method of Welch. In this method the final spectrum results from averaging the spectrum of consecutive segments with 50% overlap. The individual spectra are computed by Fourier transform of the signal segments. The duration of each segment is 10 sec.

The spectrum is calculated for the entire signal bandwidth (1 kHz) after shifting the spectra to 0 Hz. An example of the spectrum for the 1-minute segment (50 sec data) is shown in figure 4.3. Doppler frequency shift is calculated in the range from -5 Hz to 5Hz (blue color on figure) (taken into account the error in determination DFS due to reference oscillator). In the band of -2 Hz to 2 Hz (red color) relative to Doppler shift, the signal intensity is calculated as the integral of the spectral components. Here is where the main part of energy of the signal is concentrated.

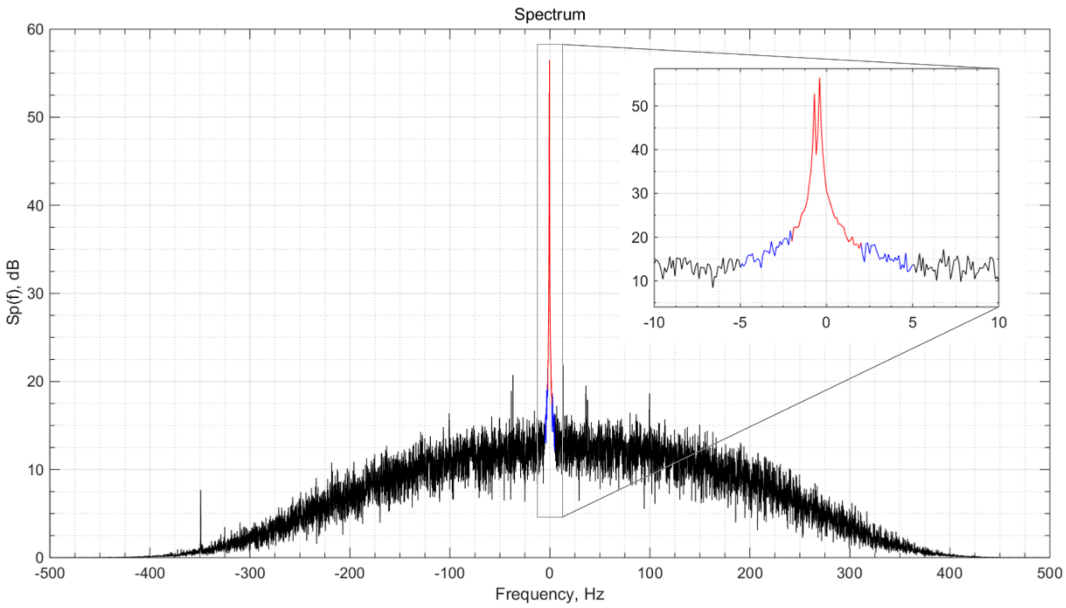

Figure 4.3: Spectrum for one segment and enlarged central fragment.

The noise intensity is determined like for the signal, but away from the center of the spectrum with the condition that it does not coincide with harmonics of the signal at 1*N and 10*N Hz where N is an integer 1, 2, 3 etc.. If the signal-to-noise ratio (signal intensity / noise intensity) is small, the Doppler shift is not reliable for given spectrum and further calculation was not involved.

Doppler frequency shift is calculated by two methods. The first is the determination of the maximum of the spectrum and DFS is the shift from this maximum to the central frequency. This method works well when we can distinguish a clear and single maximum of the spectrum. But it often gives an error when there are more random fluctuations near the true center of the spectrum. The second method – the method of moments [43] is a more accurate. It is based on the determination of the center mass of the spectrum. DFS is the shift of the center of mass from the central frequency. The center of mass of the spectrum is calculated by the formula 4.1.

$$f_{c.m.} = \frac{\int_{f1}^{f2} f \cdot Sp(f)}{\int_{f1}^{f2} Sp(f)} \quad (4.1)$$

where: $f_{c.m.}$ is the location of the center of mass of the spectrum in Hz, f_1 = -5Hz – start frequency for determination DFS, f_2 = +5Hz – end frequency for determination DFS, f – frequency range from f_1 to f_2, $Sp(f)$ is the power spectral density in the frequency range from f_1 to f_2.

Thus, by processing the spectra of the 1-minute segments we got 7 points in the beginning of each half an hour for the signal intensity and the Doppler frequency shift. Each point has a timestamp label equal to the middle of the 1-minute segment. Points with no continuous signal are filled with zeros. At a given stage the total number of processed points is 1440 for each day.

Next stage is the averaging. Diurnal arrays of the signal intensities and DFS are averaged over one month. For example for the intensities: the first point of the monthly averaged array of the intensities contains the averaging of the first points of all days of the intensities. Similarly, the second point of the averaged array is the average of all second points of each day. The blank points are ignored and do not affect the result of averaging. All the same for the monthly averaged array of DFS. The resulting averaged monthly arrays of the intensity and DFS contain 1440 points each. Figure 4.4 shows the mechanism for obtaining the averaged monthly arrays. The painted cells correspond to the presence of data.

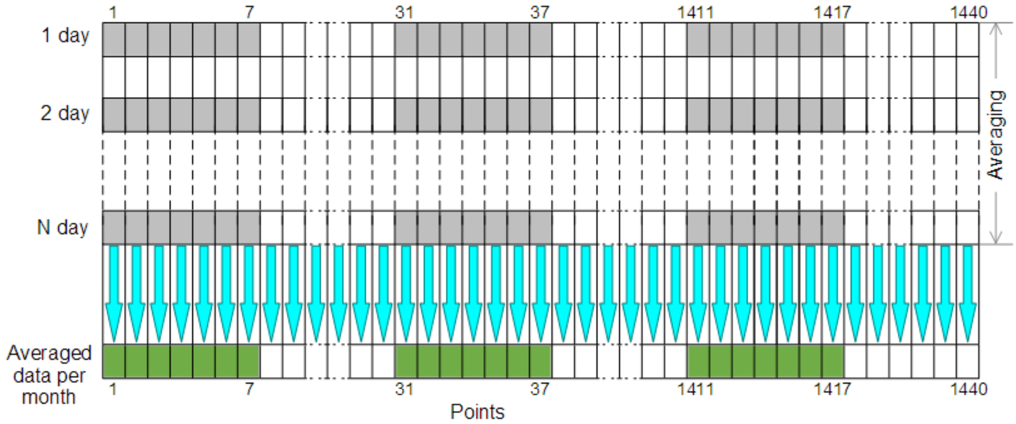

Figure 4.4: The mechanism for obtaining the averaged monthly arrays.

The next step is the averaging by interval of 600 seconds for each day. Monthly averaged arrays also are averaged. Each point of intensities and DFS was obtained by averaging the intervals of 10 points (the blank points ignored and do not affect the result of averaging). From blank points one obtains respectively blank points. Total averaged array for each day contains 144 points. The mechanism of averaging and the structure of averaged array for one day are shown in figure 4.5. Each new point has its timestamp label: data points (green cell) have a time stamp equal to the middle of the original 7-point data; blank points have a timestamp label equal to the middle of 10-minute interval.

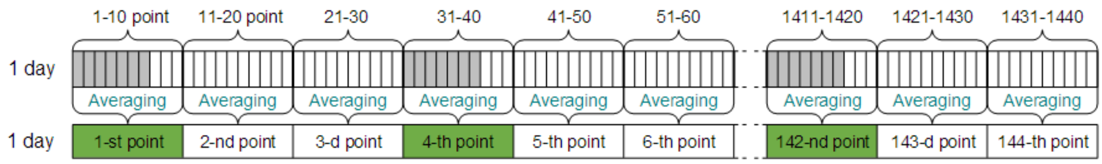

Figure 4.5: The mechanism of averaging by 600 s and the structure of averaged array.

Since averaging over 600 seconds arrays contains blank points, these points are removed. Thus the beginning of each half an hour contains only one averaged point. The number of points in each day is 48 (24*60/30=48). Each point has a time stamp corresponding to the middle of the original 7-minute interval at the beginning of each half an hour. The daily arrays of the intensities and DFS and the monthly averaged arrays have temporal resolution of 30 minutes. For processing and storage convenience, all arrays are collected into the annual arrays.

Of course averaging loses the instantaneous values of the intensity and the DFS, but to observe the overall picture of the behavior of these signal parameters during the day, month and year, that is enough.

4.3 Plotting of the graphs

The graphs of the intensities and DFS of the radio signal for the two receiver stations (Tromso and LFO) were prepared on the basis data arrays from the program "Proc_RWM_CW_mod". For further analysis, we have built graphs of monthly, daily and annual distributions of HF signal parameters. The programs for array processing and plotting of monthly distributions of intensities and DFS are presented in Appendix A. An example of the signal intensity in the LFO is shown in figure 4.6 for April 2013.

The blue lines show the changes in the signal intensity. As mentioned above, each day contains 48 data points with a time resolution of 30 minutes. For example, for 30 days of April, the total number of points is 1440 (48*30 = 1440). The gray dashed line is built according to monthly averaged array (for April) wich contains 48 points for one day. The DFS also has 1440 points for this month.

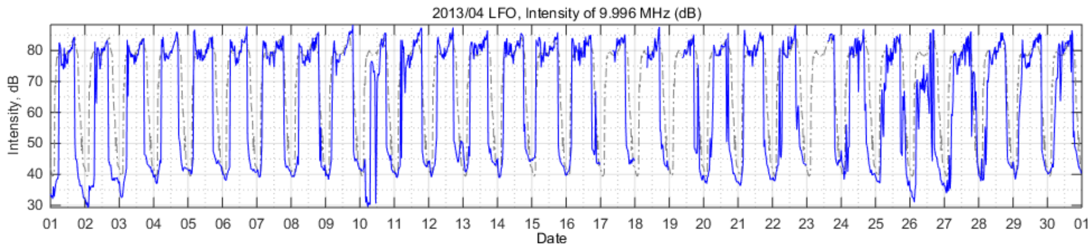

Figure 4.6: Intensity for LFO during April 2013.

Similarly, we have plotted ionospheric parameters: TEC for Tromso, Kiruna, Joensuu and LFO, critical frequency F0F2 for Tromso and LFO. TEC data format is 288 points (1 TECU=10^{16} electron/m^2) per day with a time resolution of 5 minutes. Average monthly distributions of TEC were obtained similarly as for intensities and DFS. The format of the averaged TEC is the same - 288 points with a time resolution of 5 minutes. An example of TEC graphic is shown in figure 4.7. The red and blue lines show the diurnal TEC variations and the gray line the averaged TEC.

The data format of Ionosonde Tromso: 720 points (points have dimension of MHz) per day with a time resolution of 2 minutes. The data format for Moscow ionosonde: 96 points per day with a time resolution of 15 minutes. The example of graphic for the critical frequency is shown in figure 4.8. The red line shows the diurnal variations of the critical frequency. The gray line is plotted by averaging the data for all the days of the month, according to the principle of averaging the intensities. The gray line includes 720 points for one day.

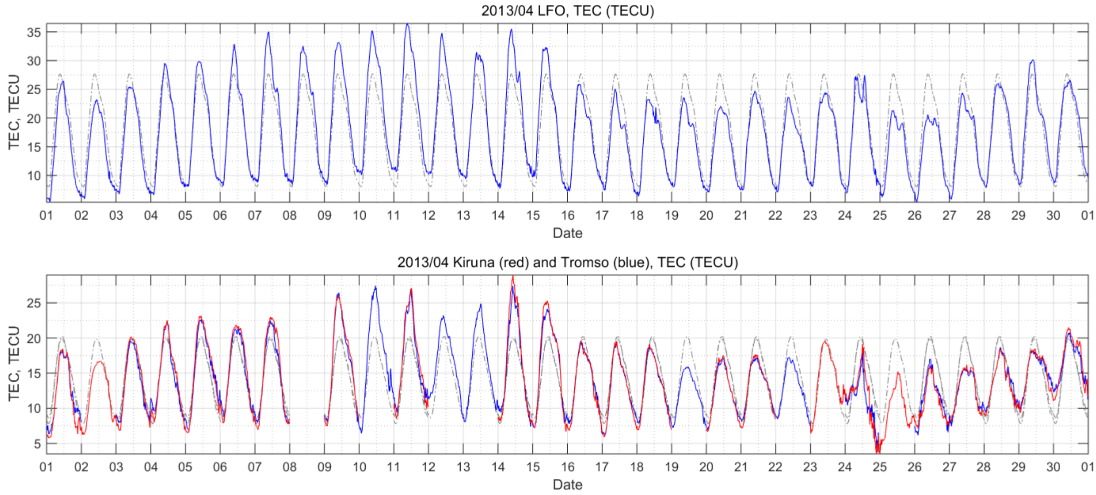

Figure 4.7: Example of graphic of TEC for LFO, Tromso and Kiruna during April 2013.

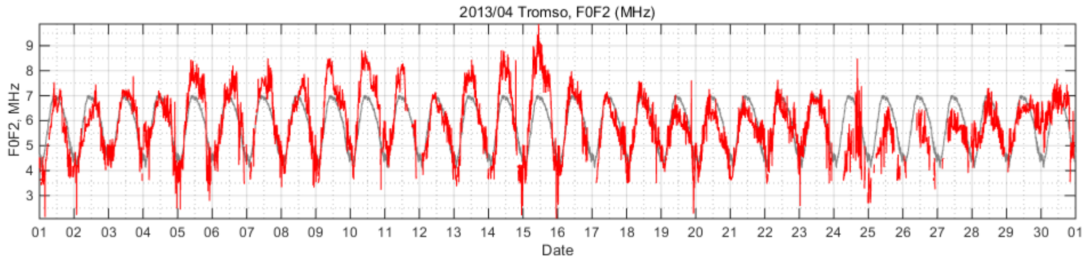

Figure 4.8: Example of graphic of F0F2 for Tromso during April 2013.

In addition, we have plotted the annual distributions for all parameters. Figure 4.9 shows an example of such distribution for the TEC. Vertical white gaps indicate the absence of data in these days. The relative colors of the graph show the value of the TEC. Each of the 288 points in a day has the color assigned in accordance with the value of this point. The points are arranged from below upward in accordance with their timestamps in a vertical line. In this way are constructed 365 days during the year.

Similarly, we have built the annual distribution of intensities and DFS signal, the critical frequencies F0F2, magnetic K-indices.

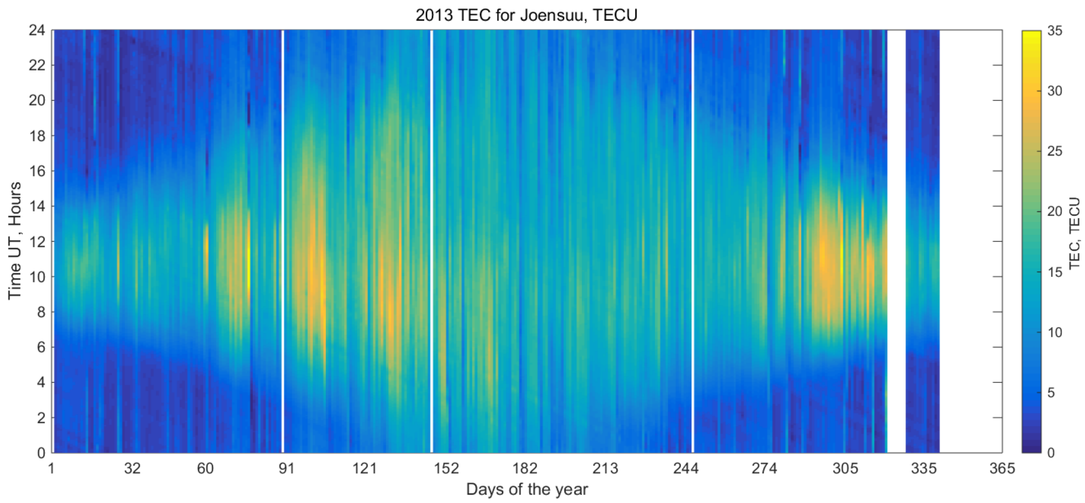

Figure 4.9: Example of graphic of yearly distribution TEC for Joensuu during 2013.

The monthly distributions of magnetic local K-indices and global Kp-index are plotted in the form of bar graphs. The indices data format: 8 points per day with a time resolution of 3 hours. Figure 4.10 shows an example of K indices. The magnetic indices are not averaged.

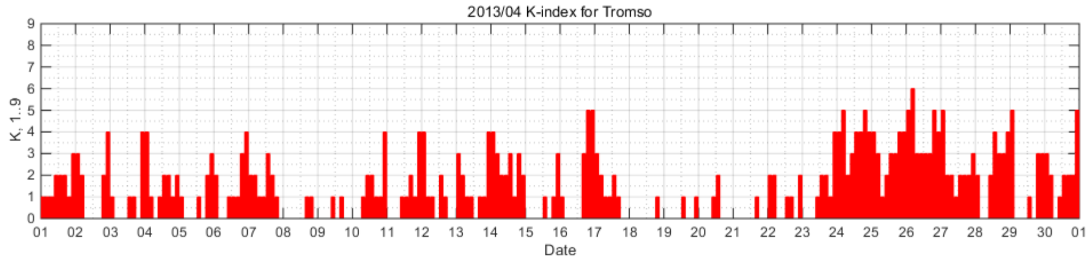

Figure 4.10: Example of graphic of K-indices for Tromso during April 2013.

We also have plotted in the form of bar graphs the indices $F_{10.7}$ solar activity (1 unit $*10e^{-22}$ $W/m^2/Hz$), Sun Spot Numbers (SSN) and the power of X-rays at ground level (1 unit$*10^{-7}$ W/m^2) in the spectral lines from 1 to 8Å. The data format: 1 point per day. An example is shown in figure 4.11.

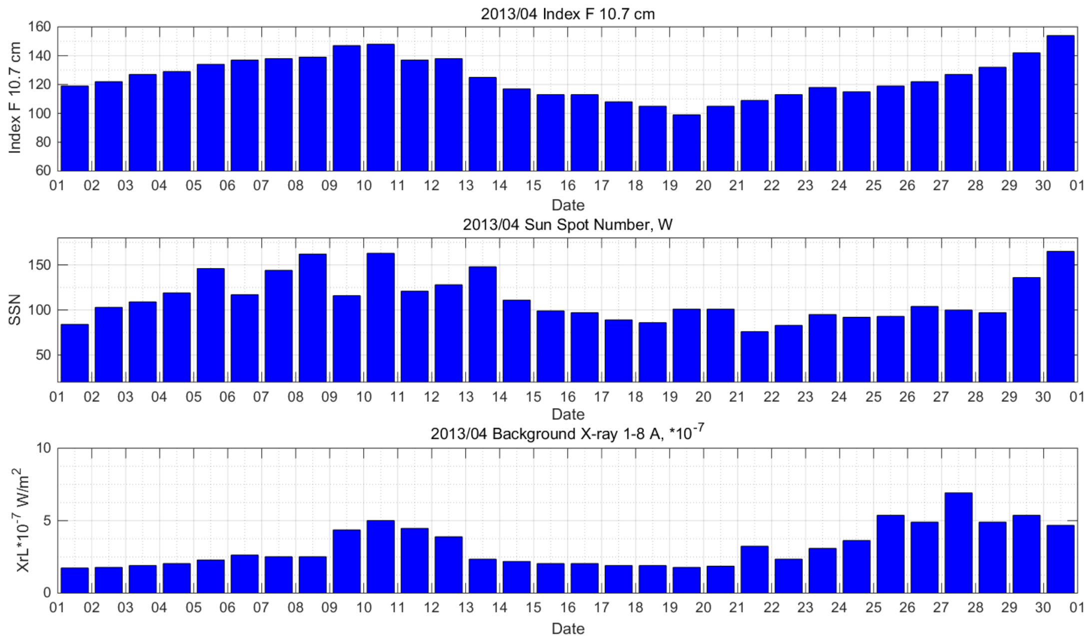

Figure 4.11: Example of graphic of solar activity during April 2013.

In addition to the bar graph of solar activity, the annual distribution of $F_{10.7}$ and SSN index was plotted by the same data – figure 4.12.

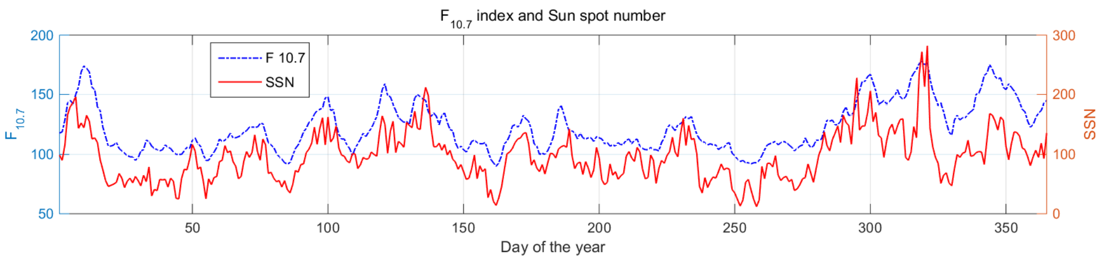

Figure 4.12: Annual distribution solar index $F_{10.7}$ and Sun spot number.

5 Results

Analysis of the relationships between the parameters estimated in chapter 4 and their geophysical implications will be discussed in this chapter. We present here the annual distributions of ionospheric parameters such as TEC, the critical frequency of the F2 layer at the midpoints of the radio paths Moscow-Tromso and Moscow-LFO. We discuss the influence of solar and geomagnetic activity on the parameters of the ionosphere. We show the behavior and the results of the analysis of the radio signal intensities and Doppler frequency shifts in Tromso and LFO in the long-term and short-term periods. We describe the signatures of ionospheric disturbances that we found in the processed data. We use all the data obtained during 2013.

5.1 Long-term behavior of the ionosphere and HF signals parameters at polar and mid-latitude radio paths

This section presents the results of seasonal variations of HF signals in the polar regions and midlatitude.

The distribution of the TEC, critical frequency F0F2, intensity and Doppler frequency shift is presented in the form of diurnal-annual dependencies. In these figures the vertical axis represents the time of day, the horizontal axis the day of the year, and the parameter value is represented in a color scale.

Figures 5.1 and 5.2 shows the diurnal-annual ***distribution of the critical frequency*** F0F2 for Tromso and LFO and figure 5.3 the behavior of the solar $F_{10.7}$ index and Sun Spot Number (SSN). The F0F2 data has been obtained from the Tromso Geophysical Observatory and Moscow ionosonde. (F0F2 data for LFO is not available so we used data from Moscow ionosonde. We suggest that for the short radio path, the F0F2 in both places, in general, may have similar responses to disturbances in the ionosphere).

The gaps in figure 5.1 are probably due to increased absorption in the lower layers of the ionosphere. The gaps that cover all 24 hours around days 140 and 220 are due to the ionosonde in Tromso being out of operation.

On figure 5.2 we can see some decrease of F0F2 during summer in the afternoon and increase in the evening. This behavior can be explained by the influence of neutral winds [3, 44, 45]. The winds drag causing an ionospheric drift, however charged particles move along magnetic field lines downwards where losses of electrons are greater and this causes a decrease in daytime critical frequency F0F2. In the evening, the direction of the wind changes and charged particles move along magnetic field lines upwards, where losses smaller (slow recombination) and this increases the critical frequency F0F2.

Figures 5.4 and 5.5 shows the diurnal-annual ***distribution of TEC*** for Joensuu, Finland (the point of reflection HF waves between Moscow and Tromso) and LFO. The TEC was calculated from GPS data at the receiving positions and at the mid-point of the radio path from Moscow to Tromso. For the mid-point of the shorter path Moscow-Kharkiv TEC data is not available, but because of the relatively shorter path length, the LFO data is sufficient. The summer afternoon effect of F0F2 seen in figure 5.2 is hard to detect in figure 5.4 for TEC above Finland but is slightly clearer in figure 5.5 above Ukraine. Figure 5.6 shows the behavior of the solar $F_{10.7}$ index (1 unit*$10e^{-22}$ $W/m^2/Hz$) and Sun Spot Number.

It is seen that changes of F0F2 and TEC to a large extent depend on solar indices during the year. The solar indices and the ionospheric parameters show several maxima, the greatest of which are located near days 70, 100, 130, 300. Thus, the annual cycle of critical frequencies and the TEC is largely determined by the level of solar activity, on which the regular annual variation is superimposed.

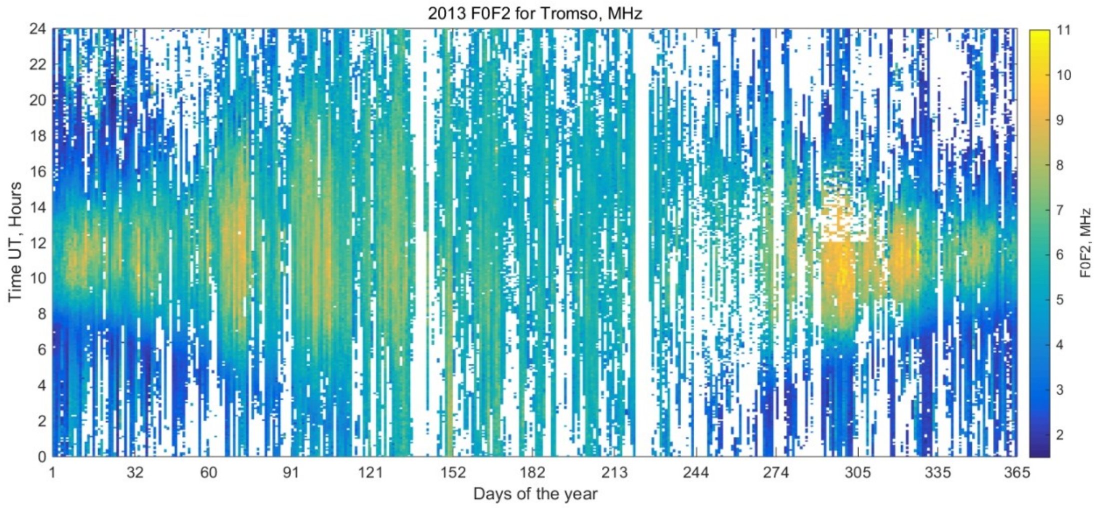

Figure 5.1: Annual distribution critical frequency of F2 layer F0F2 for Tromso.

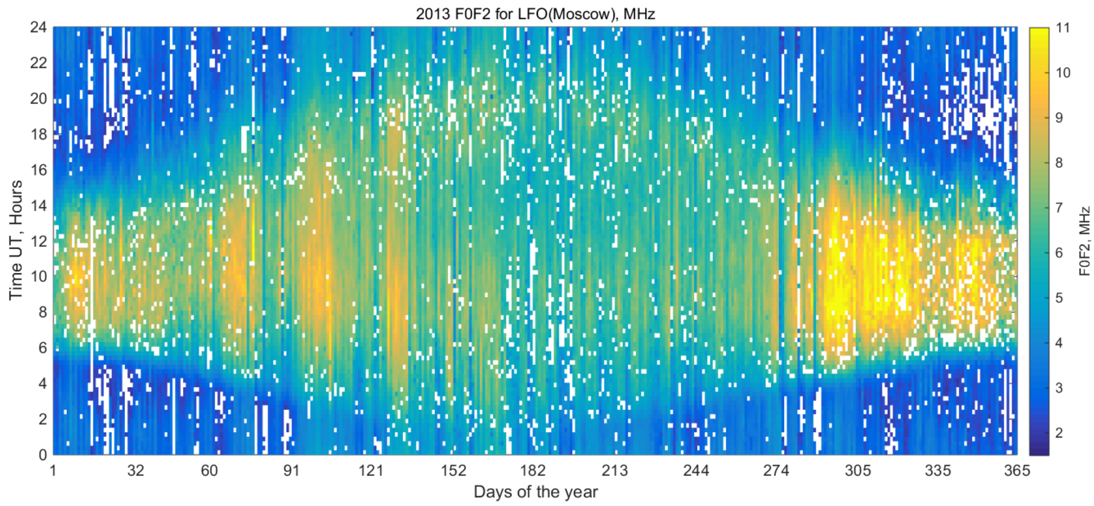

Figure 5.2: Annual distribution critical frequency of F2 layer F0F2 for LFO (Ukraine).

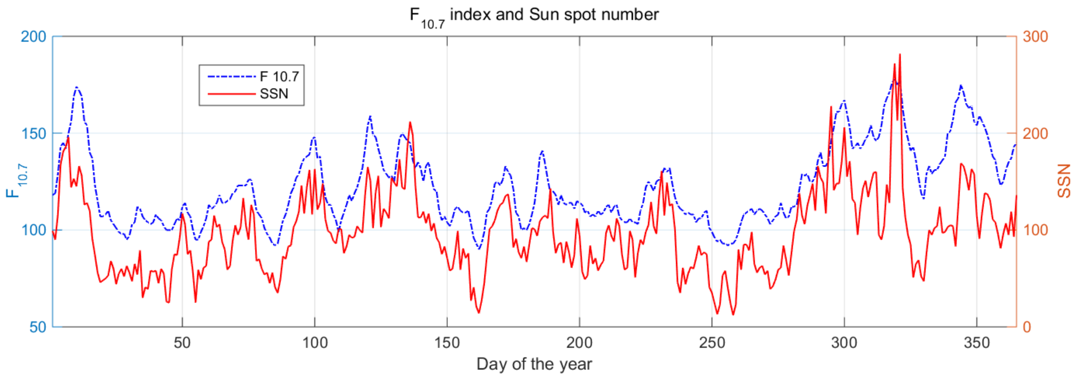

Figure 5.3: Annual distribution index $F_{10.7}$ and Sun spot number.

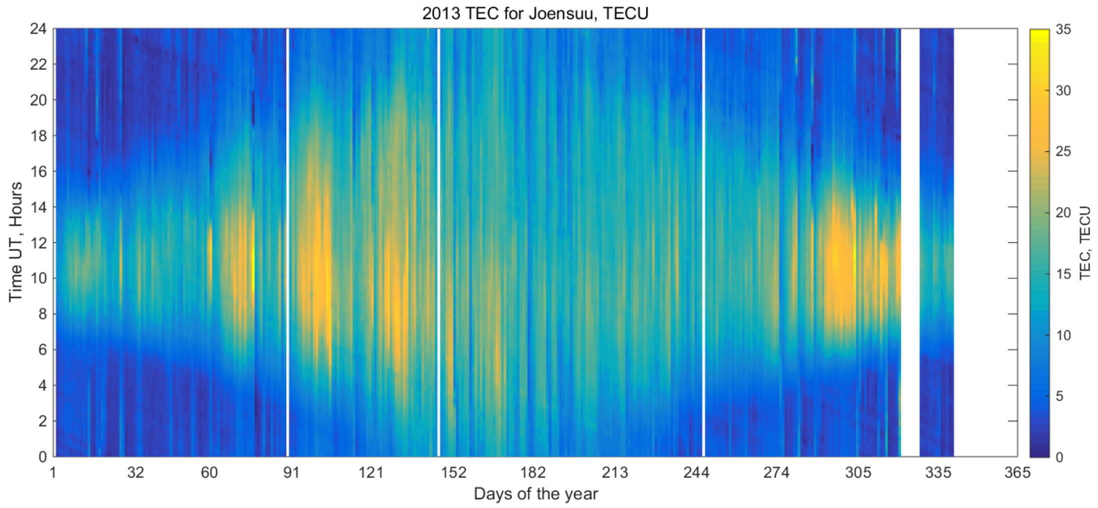

Figure 5.4: Total electron content according to GPS data in Joensuu (Finland).

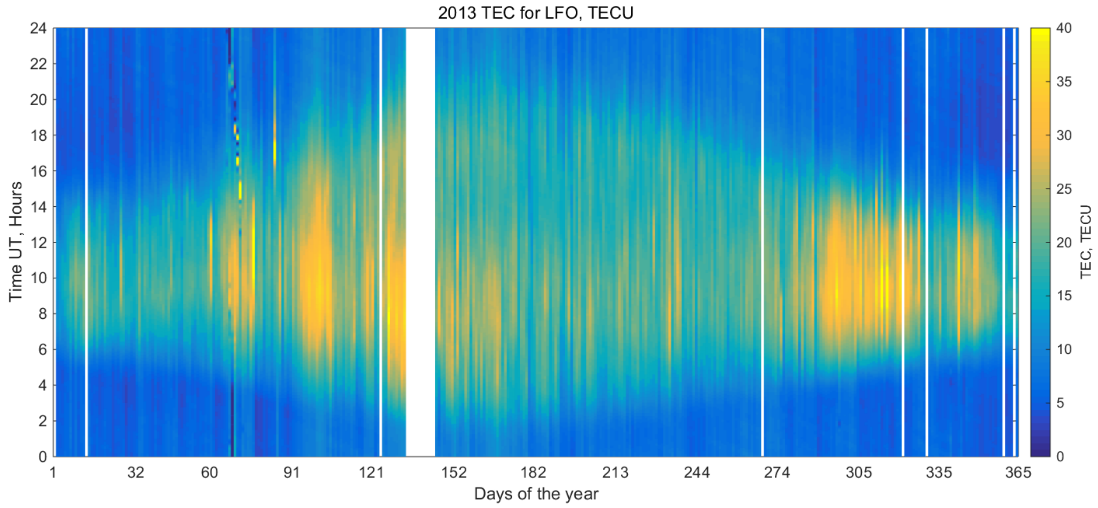

Figure 5.5: Total electron content according to GPS data in LFO (Ukraine).

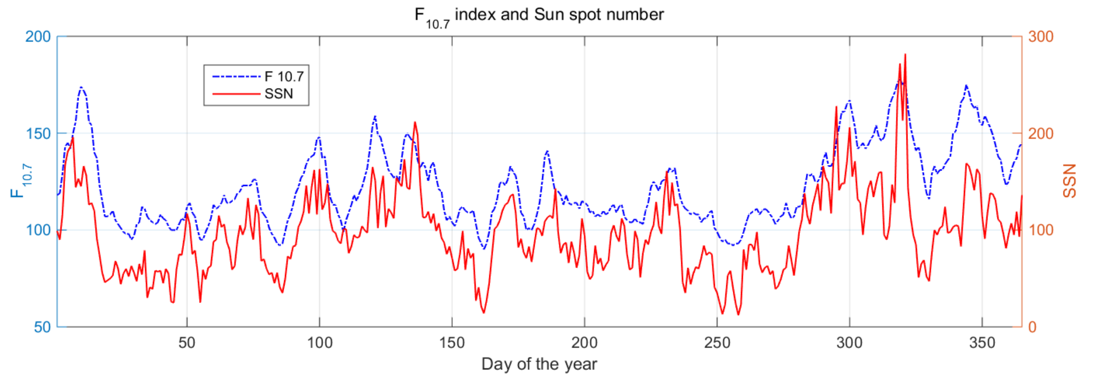

Figure 5.6: Annual distribution index $F_{10.7}$ and Sun spot number.

In addition, we can see that the behavior of the TEC depends on geomagnetic disturbances. Figures 5.7 and 5.8 show the TEC for Joensuu and magnetic K-indices in Tromso.

The TEC behavior is very individual during magnetic disturbances. Usually, TEC increases in the first 24 hours of a storm. In the second 24 hours or so the content decreases below that of the undisturbed ionosphere. The disturbance variation in TEC disappears by the third day or earlier [46].

TEC enhancements are due to ionization of the thermosphere by energetic particle precipitation from the magnetosphere. But TEC reductions also occur due to chemical changes in the neutral thermosphere (80-600 km). The enhanced auroral electrojet during magnetic storms in the polar regions heats the lower part of the thermosphere. This leads to new atmospheric circulation and plasma drift patterns which lead to a redistribution of concentrations of N2 and O2, which influences the loss rate of electrons [3, 47].

In the E region the dominant neutrals are O_2 and N_2, and the dominant ions are O_2^+ and NO^+, created in the ion-molecular reactions ($O_2^+ + N_2 \rightarrow NO^+ + NO$; $O_2^+ + NO \rightarrow NO^+ + O_2$; $N_2^+ + O_2 \rightarrow N_2 + O_2^+$). The main mechanism of the disappearance of charged particles is dissociative recombination of ions and electrons ($NO^+ + e \rightarrow N + O$; $O_2^+ + e \rightarrow O + O$) [10].

In the F1 region the dominant components for ionization are – N_2 and O (less O_2). The dominant ions are O_2^+ and NO^+, less O^+. Dominants reactions that create ions O_2^+ and NO^+ are $O^+ + N_2 \rightarrow NO^+ + O_2$; $O^+ + O_2 \rightarrow O_2^+ + O$; $N_2^+ + O_2 \rightarrow N_2 + O_2^+$; $N_2^+ + O \rightarrow NO^+ + N$; $O_2^+ + N_2 \rightarrow NO^+ + NO$; $O_2^+ + NO \rightarrow NO^+ + O_2$. The main mechanism of the disappearance of charged particles is dissociative recombination of ions and electrons.

In the F2 region the dominant component for ionization is atomic oxygen. The dominant ion is O^+. Radiative recombination of ions O^+ ($O^+ + e \rightarrow O + h\nu$) is very slow, so the main mechanism of their disappearance is a two-step process: first, reactions of O+ with molecular O_2: $O^+ + O_2 \rightarrow O_2^+ + O$; and with molecular N_2: $O^+ + N_2 \rightarrow NO^+ + N$. The next step is fast dissociative recombination of molecular ions ($O_2^+ + e \rightarrow O + O$; $NO^+ + e \rightarrow N + O$) [3].

Thus, the neutrals N2 and O2 lead to the creation of new ions which have fast recombination rates, especially in the F2 region.

Also during magnetic storms large-scale electric fields are generated that lead to plasma drift. These drifts influence the electron density.

Changes of TEC in Joensuu are greater than in LFO because Joensuu is located closer to the polar region.

Thus, the behavior of TEC is very complex during magnetic disturbances. Of course, magnetic disturbances produce changes in the critical frequency F0F2.

Although we don't have K-indices data in Joensuu, the K-indices in Tromso show this tendency.

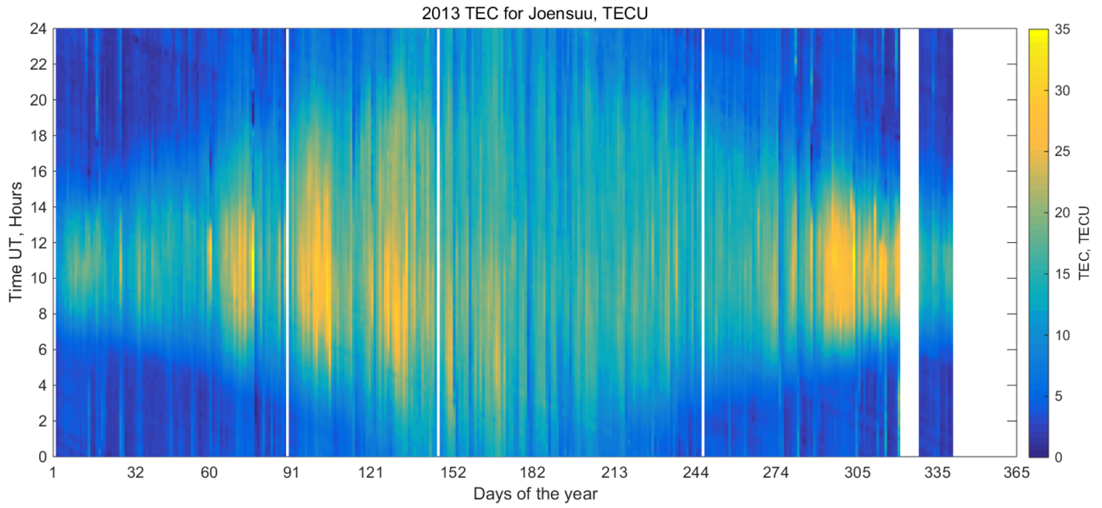

Figure 5.7: Total electron content according to GPS data in Joensuu (Finland).

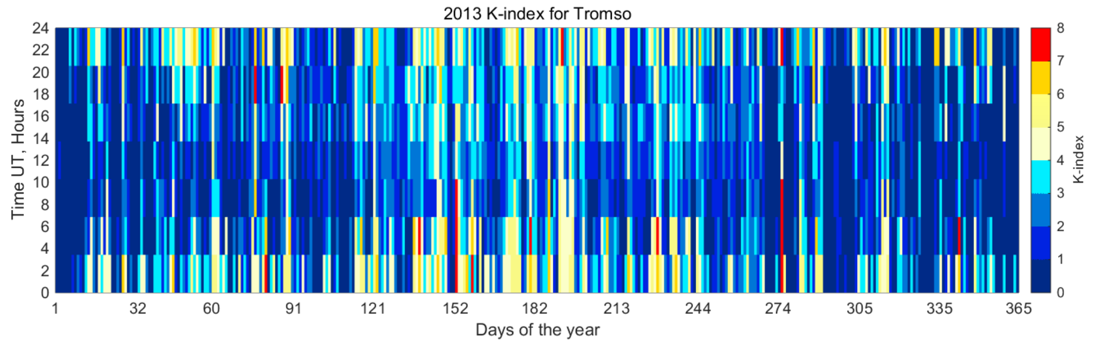

Figure 5.8: Local K-indices in Tromso.

The duration of daily variation of TEC and F0F2 depends on the season. Here of decisive significance is the duration of daylight hours when solar radiation is capable of supporting the process of ionization [3, 48]. This is clearly seen in figures 5.9 and 5.10 where the TEC in Joensuu and the critical frequency in LFO (Moscow) are shown together with red lines at sunrise (the lower curve) and blue at sunset (the upper curve). The black vertical lines separate the seasons with the red arrows showing the corresponding spring (SEq) and autumn (AEq) equinoxes and summer (SS) and winter (WS) solstices.

The increasing trend of F0F2 and TEC corresponds to the time of sunrise, the maximum is reached closer to noon at Joensuu and close to 13:00 LT at Moscow, and the decrease in these parameters occurs at the time of sunset. Since the line of sunrise and sunset drawn are for the Earth's surface, the Sun rises earlier and sets later as altitude increases. So, solar ionization starts earlier and ends later than on the marked lines. Joensuu in summer is characterized by the constant presence of the Sun at ionospheric heights from about a month before and one month after the SS, although the Sun is not visible on the Earth's surface. Therefore, the ionization continues at night, albeit at a lower level due to the low solar elevation angle.

After sunset, the increased electron density occurs due to the action of neutral winds [45] and plasma drift. In this case, the charged particles rise along the lines of magnetic field in the area where recombination is slower and electrons live longer. It is clearly seen in the summer at middle latitudes in figure 5.10.

During spring, summer and autumn at night the electron concentration is supported by the descent of charged particles in the F2 region from the plasmasphere. The electrons accumulate in the plasmasphere during a long period of daily solar ionization but in winter the accumulation is less. At night in the F2 region the concentration decreases faster and hence the critical frequency is less.

In addition, in figure 5.10 we can see the rise of critical frequencies F0F2 in the winter. Perhaps this is a manifestation of the winter anomaly, when the electron density at this time is higher than in summer. The diffusion is reduced due to colder temperatures that lead to higher concentration of main component for ionization on F region – neutral oxygen atoms and electron density increases [47].

Additionally the rate of recombination is reduced leading additional increase in the electron density. Why the winter anomaly does not appear in Tromso and Joensuu is not clear.

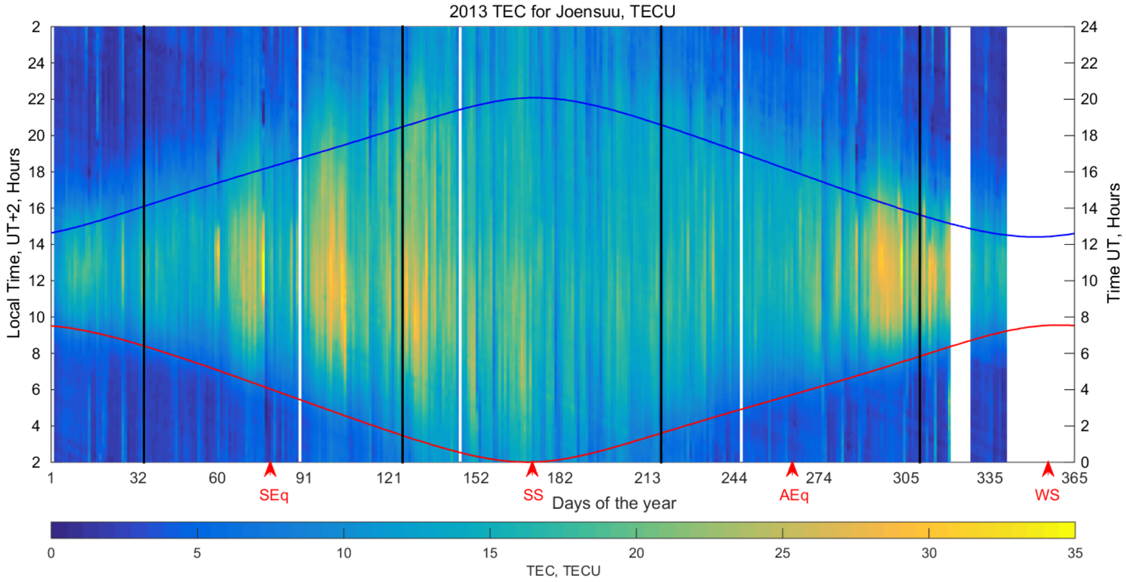

Figure 5.9: Total electron content according to GPS data in Joensuu (Finland) with lines sunrise (red – the lower curve) and sunset (blue – the upper curve).

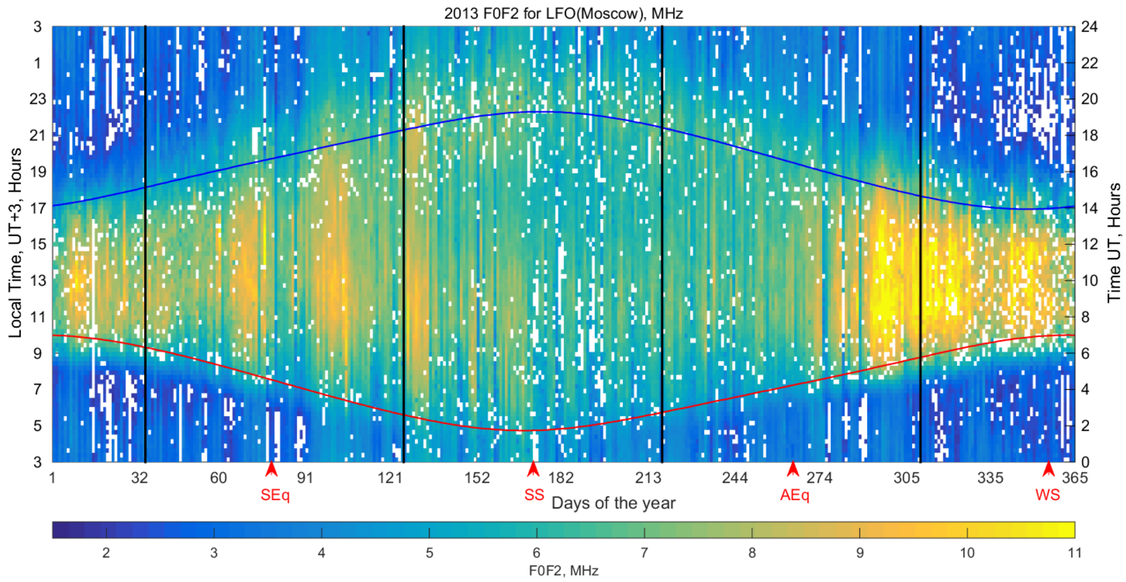

Figure 5.10: The critical frequency of F2 layer F0F2 according to ionosonde in Moscow with lines sunrise (the lower curve) and sunset (the upper curve) in Moscow.

Figure 5.11 shows the annual change in Doppler frequency shift (DFS). The DPS is non-zero only around sunrise, when the ionosphere is dropping, and around sunset, when the ionosphere is raising. They have opposite signs. The remaining periods of the day the ionosphere is mostly stationary. The blue and red lines are fittings to the general trends around sunset and sunrise respectively.

Figure 5.12 shows the change in signal intensity at LFO together with the curves of increase/decrease of DFS, defined by figure 5.11. It can be seen that positive Doppler frequency shift near sunrise corresponds to the beginning of the appearance of a sharp increase of signal intensity reflected from the ionosphere. The negative Doppler frequency shift near sunset corresponds to a decrease in signal intensity. (Matlab program for these figures is not presented because the most operations were performed manually).

Note that the afternoon dip in the HF signal intensity appears more strongly than for F0F2 or TEC. The afternoon dip is caused by stronger signal absorption due to increased ionospheric ionization in the afternoon period along the propagation path. To a lesser extent it is associated with a local minimum of the critical frequency.

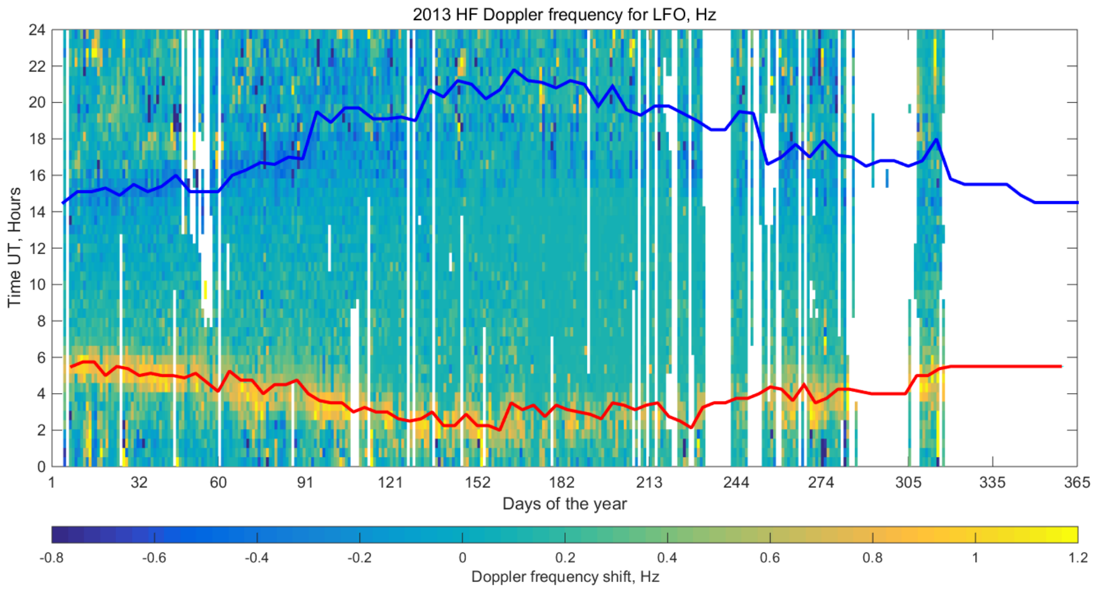

Figure 5.11: Annual Doppler frequency shifts in LFO with interpolations lines (lower curve – positive morning DFS and upper curve – negative evening DFS).

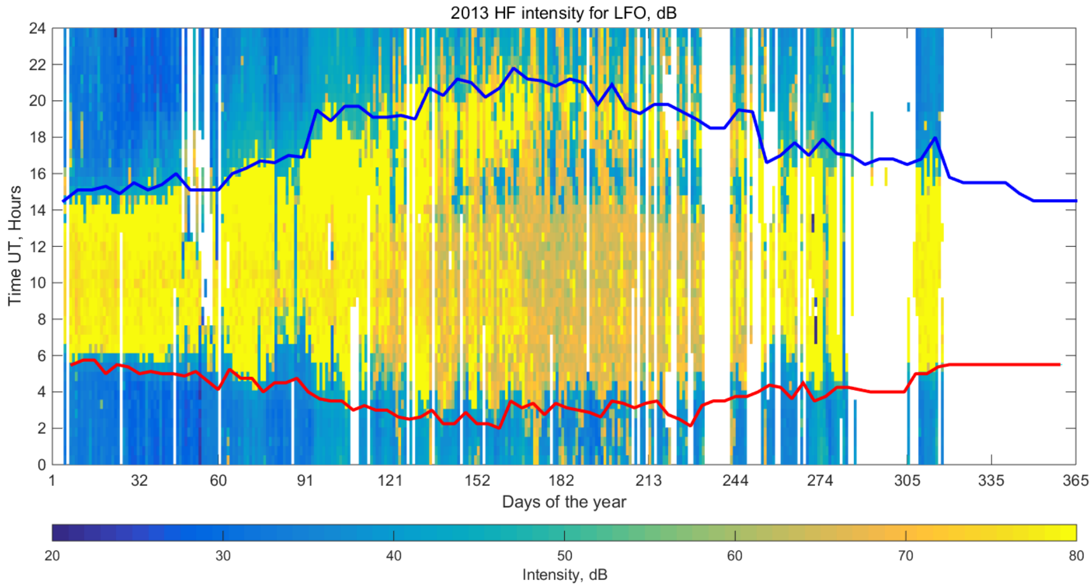

Figure 5.12: Intensity in dB in LFO with lines of morning – positive (lower curve) and evening – negative DFS (upper curve).

Figures 5.13, 5.14 for LFO and figures 5.15, 5.16 for Tromso show diurnal-annual changes in Doppler frequency shift and intensity, together with the lines of sunrise (red line – lower curve) and sunset (blue line – upper curve) at the midpoints of the radio paths.

Changes in DFS and intensity during the day can be explained on the basis of classical ideas of the propagation of short radio waves [2, 3, 9]. So, at sunrise the critical frequency begins to grow around the middle point of the radio path. At a certain moment it leads to "radio-growth" which is characterized by a sharp increase in signal intensity. The effective height of reflection of the signal from Moscow decreases and the corresponding DFS increases, meaning the layer is moving downward. At this time, the morning peak of DFS is registered and after that the DFS value stabilizes around zero. After sunset the critical frequency begins to decrease and the effective height of reflection of the signal increases and the DFS decreases.

We can see that the intensity of the signal for some time in the evening remains sufficiently high. This is caused by night processes that support ionization (neutral wind, electrons from the plasmasphere, etc.). Areas of non-zero Doppler shift in figures 5.13 and 5.15 at night may indicate movement of various ionized structures in the ionosphere or due to solar flares [2] or it might be due to noise.

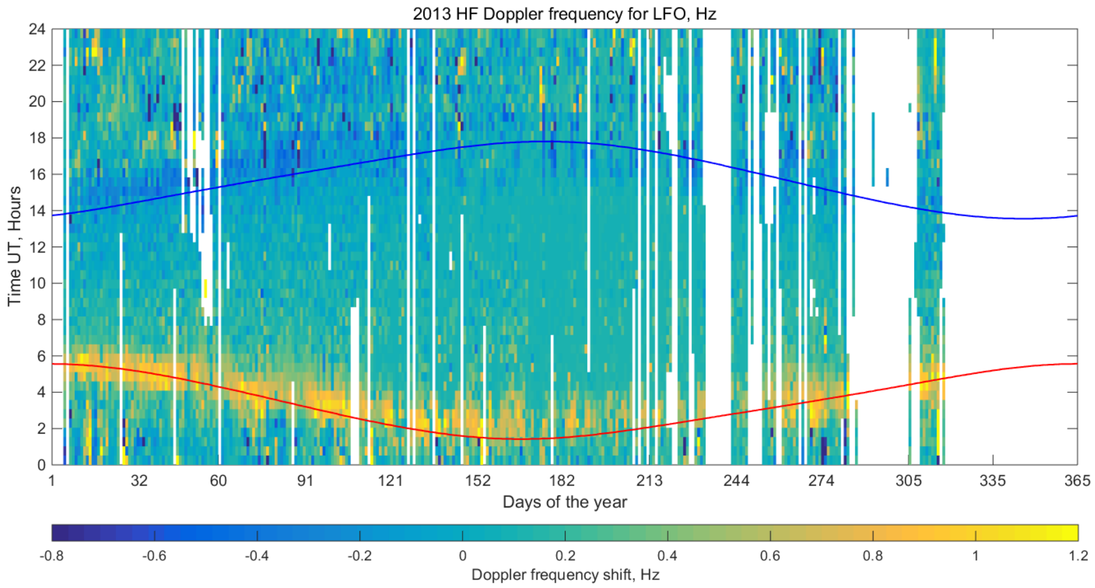

Figure 5.13: Doppler frequency shift of RWM 9.996 MHz HF signal recorded at LFO (Ukraine). Lower line corresponds to sunrise time and upper line one to sunset in mid-point of radio path.

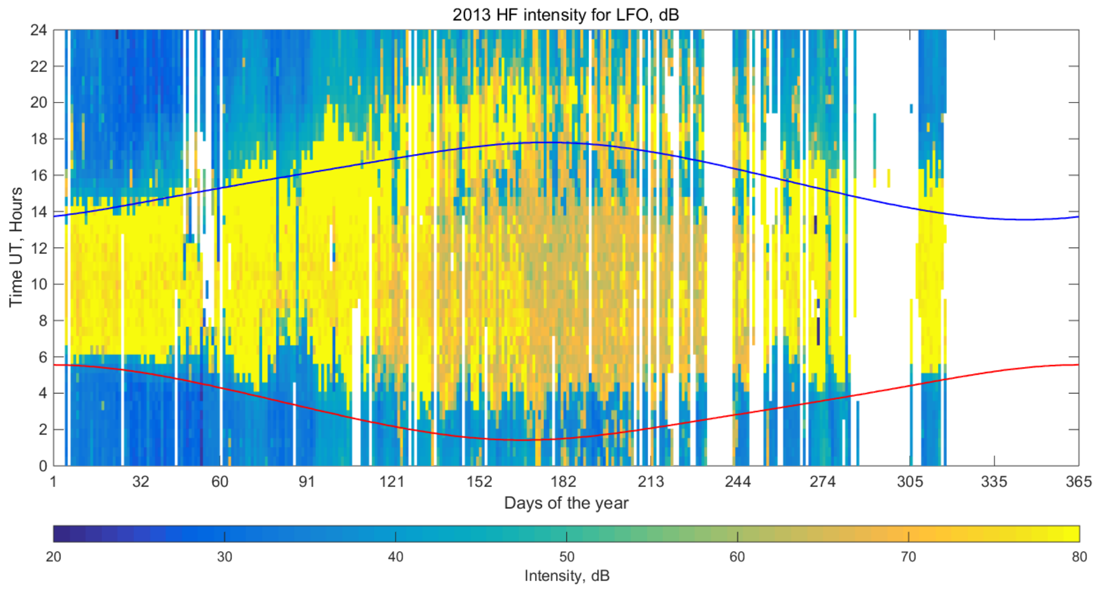

Figure 5.14: Intensity in dB of RWM 9.996 MHz HF signal recorded at LFO (Ukraine). Lower line corresponds to sunrise time and upper line one to sunset in mid-point of radio path.

An interesting feature is the decrease of signal intensity in the afternoon on the long and high-latitude radio-path from Moscow to Tromso (figure 5.16). The figure shows that the maximum intensity is observed in the morning and evening, and in the afternoon it decreases to very low values. An additional contribution to the intensity reduction at daytime is due to the polar day conditions at the receiving station and auroral absorption leading to additional absorption [8]. At LFO the afternoon minimum is much smaller than at Tromso. Seasonal changes in sunrise and sunset time to a greater extent affect the time variation of the signal intensity and the DFS. The solar and magnetic storms have slight effects.

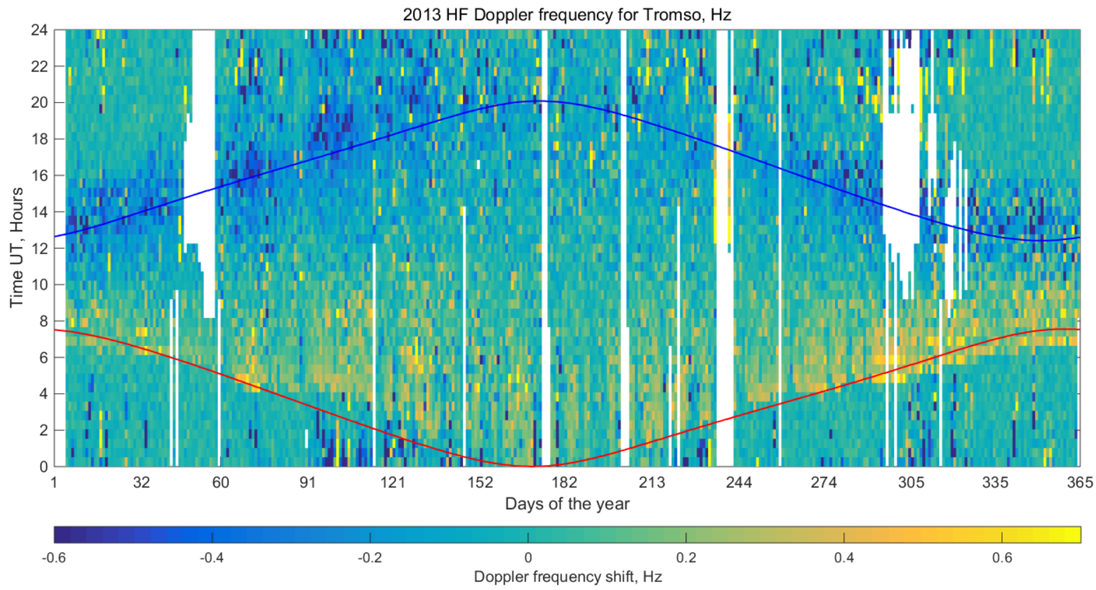

Figure 5.15: Doppler frequency shift of RWM 9.996 MHz HF signal recorded in Tromso. Lower line corresponds to sunrise time and upper line to sunset in Joensuu.

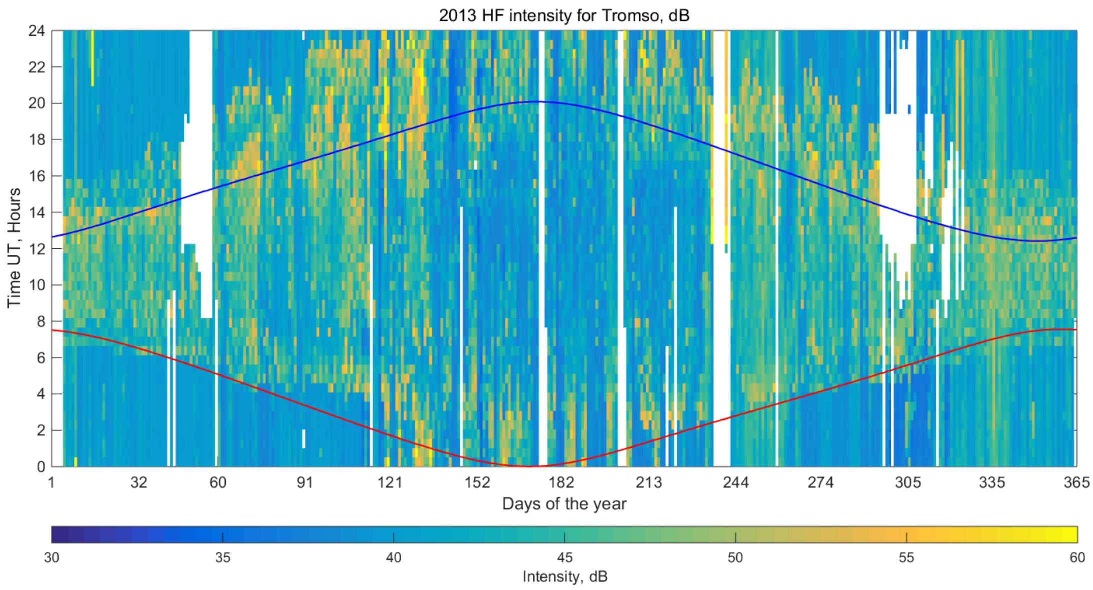

Figure 5.16: Intensity in dB of RWM 9.996 MHz HF signal recorded in Tromso. Lower line corresponds to sunrise time and upper line one to sunset in Joensuu (mid-point of radio path).

5.2 Short-term behavior of the ionosphere and HF signals parameters under influence of powerful solar and geomagnetic events

The following figures show the main characteristics of the ionosphere (F0F2, TEC) and HF signals (intensity, DFS), as a function of solar disturbances (X-rays, $F_{10.7}$ index, the Sun spot number) and magnetic activity (local K-indices and planetary Kp-indices). The monthly dependencies of ionospheric and HF parameters on the indices of solar and geomagnetic activity in various combinations were analysed. Examples of such dependencies are shown in figures 5.17, 5.18 and 5.19.

Figure 5.17 shows indices of magnetic and solar activity.

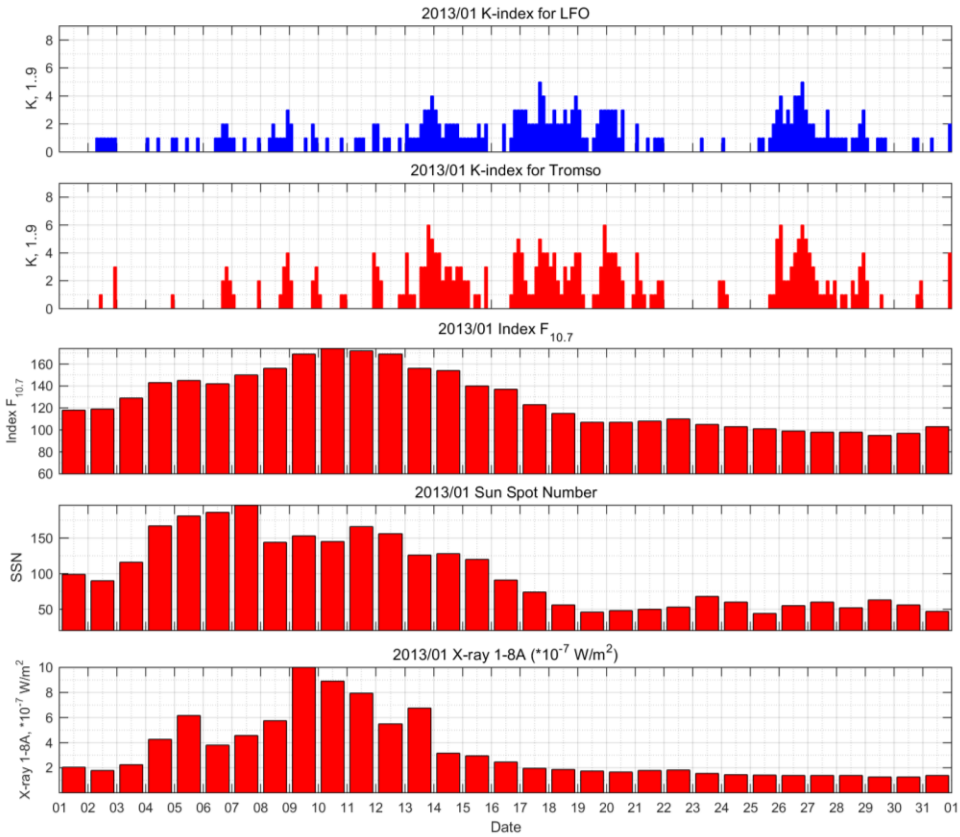

Figure 5.17: From top to bottom monthly (January, 2013) variations: local K-indices in LFO, local K-indices in Tromso, Solar index $F_{10.7}$, Sun spot number, X-ray background flux in wave band 1-8Å.

Figure 5.18 for January 2013 shows the variations of total electron content in the ionosphere above the receiving stations – LFO, Tromso and Kiruna – and changes in the critical frequency F0F2 of the ionosphere over Tromso. The Kiruna data gives additional information about the polar region and was useful when there was no data in Tromso. The continuous colored lines show the change from day to day. The gray dashed curve shows the diurnal variation averaged per month calculated with a running average from month to month using all data. It is useful to compare the daily deviations from the average behavior. It can be seen that the critical frequency and the total electron content quite often deviate from the average monthly values.

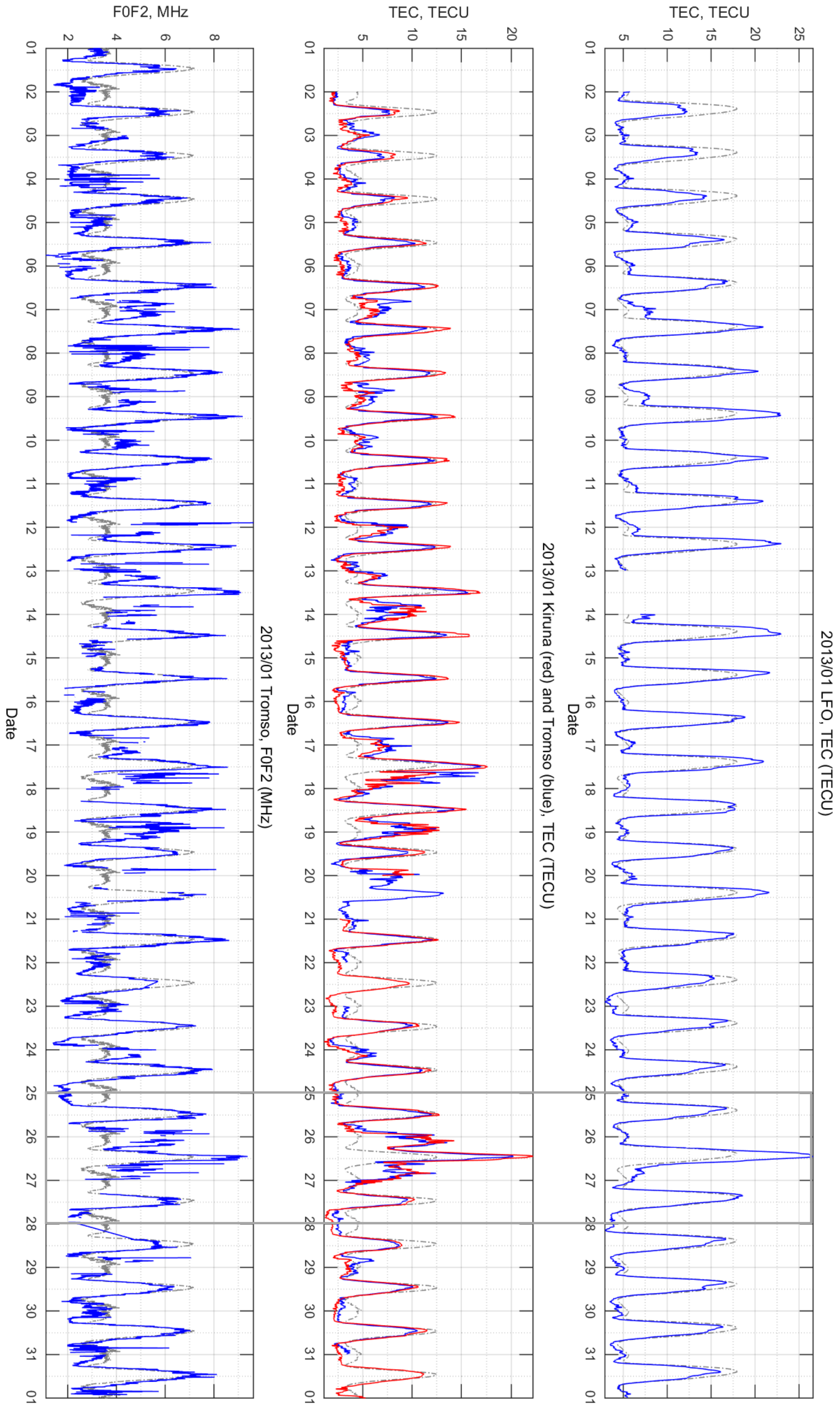

Figure 5.18: Monthly (January, 2013) variations of TEC in LFO (right), Tromso and Kiruna (center) and F0F2 in Tromso (Left). The daily curves are plotted in red and blue (continuous lines). Monthly averaged daily variation is presented in grey color (dashed curves). The disturbed period is framed by gray rectangle.

Comparing figures 5.17 and 5.18 shows that the deviations of TEC and F0F2 are associated with disturbances of the magnetic field. In contrast, the deviations depend weakly on solar disturbances.

Significant increases in the indices of magnetic activity result in clear changes in ionospheric parameters. For example, in the period January 25 to 28 there is an increase of local K-indices at LFO and Tromso. This results in an increase in TEC and F0F2 at both stations enclosed by a gray rectangle in figure 5.18.

Figure 5.19 for the same month shows the intensity variation and DFS signals at both stations. The gray dashed curve shows the daily variation, averaged per month calculated in the same way as for figure 5.18. It can be seen that the relationship of solar and geomagnetic disturbances with the characteristics of the HF signal is even more complicated. This can be explained by the fact that under conditions of geomagnetic disturbances the changes of the ionospheric parameters occur around the deflection region and in the lower part of the ionosphere. Both ionospheric changes affect the characteristics of the HF signals.

For example, in the disturbed period (indicated in figure 5.19 by the gray rectangle) there is the decrease of signal intensity in the polar region (Tromso) relative to the average level, but in the mid-latitude station (LFO) the observed signal intensity is not different from the monthly average. However, the changes of Doppler frequencies in the mid-latitude station were more pronounced.

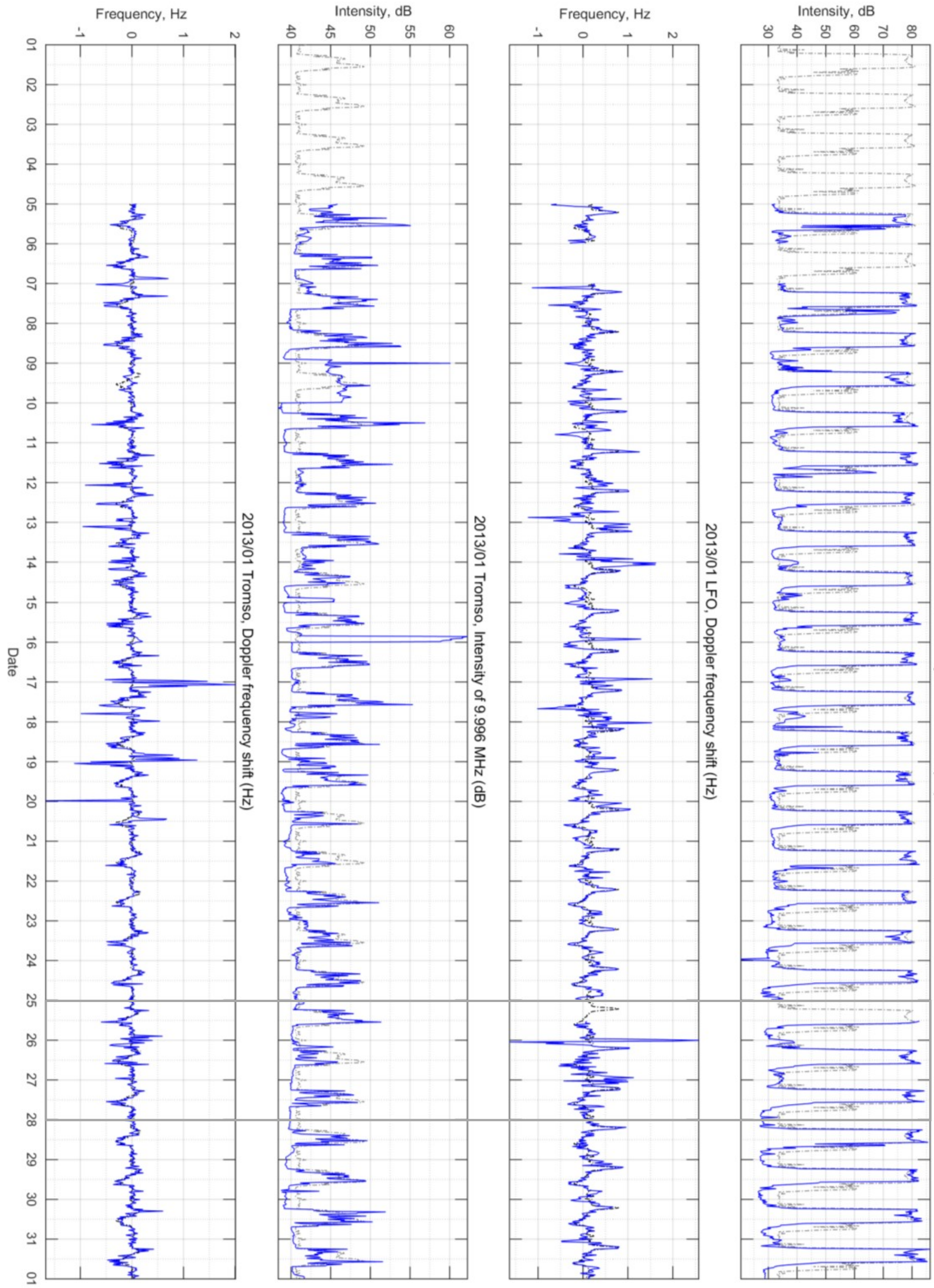

Figure 5.19: From right to left - monthly (April 2013) variations of HF signal intensity and DFS in LFO and in Tromso. The daily curves are plotted in blue (continuous lines). Monthly averaged daily variation is presented in grey color (dashed curves). The disturbed period is framed by gray rectangle.

Periods with geomagnetic perturbations were chosen from the year 2013 and they have been analyzed in more detail at both stations. It was found that increased magnetic disturbances may lead to both increases and reductions of the ionospheric parameters and intensity the HF signals. Both of these two cases are considered in more detail and shown in figures 5.20 and 5.21.

The noted perturbation on January 25-27 is shown again in more detail in figure 5.20. The left pane shows data relating to the mid-latitude region (LFO) and the right pane to high-latitude (Tromso). The top panel shows the changes of the RWM station signal intensity at frequency 9.996 MHz, the second top panel – TEC variations. On the second panel from the bottom the curves show the changes in F0F2 critical frequency and finally the local K-indices are shown on the bottom panel (colored bars) for each station and a global index of Kp (black bars).

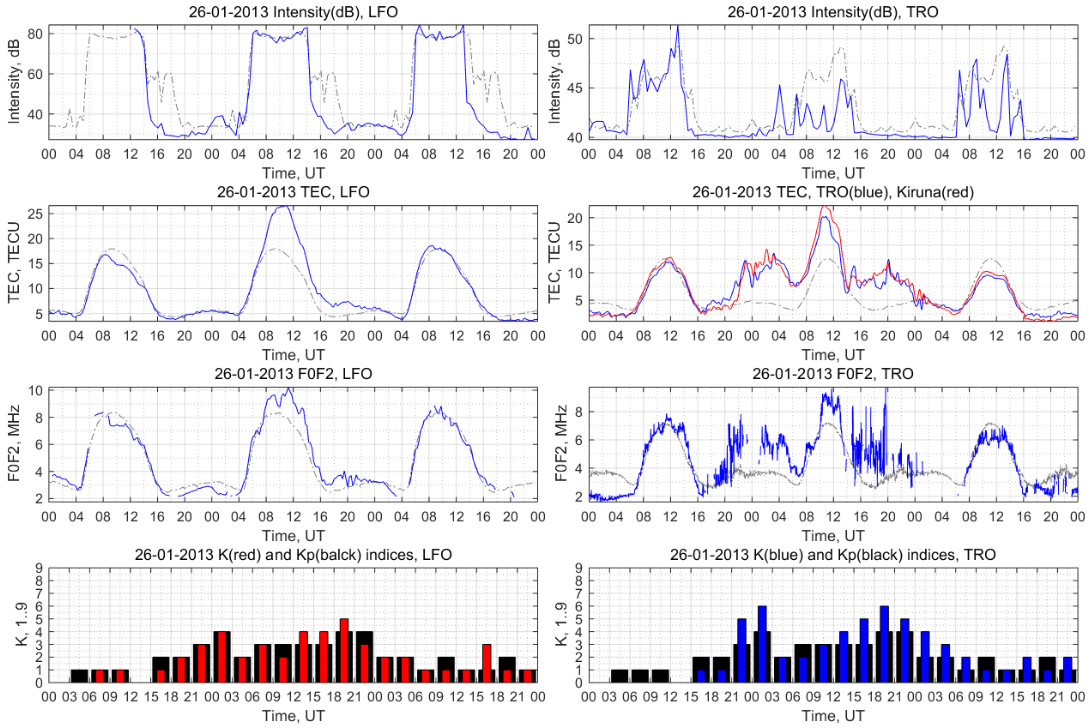

Figure 5.20: 25-27 January 2013. From top to bottom, variations of: HF signal intensity; TEC above the observation points; F0F2; local K (colored bars) and planetary Kp (black bars) magnetic indices. Left panels corresponds to LFO, right to Tromso.

Figure 5.20 clearly shows that both the total electron content of the ionosphere and the critical frequency increased at both stations. At the same time, the intensity of the HF test signal decreased only at high latitudes, and it does not change at mid-latitudes relative to the mean daily variation. Apparently, this is due to the fact that changes in ionospheric parameters during this magnetic storm are more pronounced in the high latitude region. First, the K-indices in Tromso were higher than at LFO. Second, changes in TEC and F0F2 at mid-latitudes are visible only on January 26 during the day while in high latitudes are clearly evident in the nights from 25 to 26 and from 26 to 27 January. In addition, the absence of part of the critical frequency data in Tromso is indirect evidence of the existence of sporadic ionospheric layers and a large amount of absorption, which apparently was the reason for the reduction of the intensity of the HF signal.

Figure 5.21 shows the changes of the same parameters during the powerful geomagnetic storm on 1-3 October 2013. Here, the intensity of the HF signal has decreased in both stations. At the same time, with approximately the same values of the K-indices, the reaction of the ionosphere was different. The electron density and the critical frequency declined in both locations. Apparently, absorption in the lower ionosphere in the polar region was even higher than in the previous case, because from noon October 1 to noon October 3 the F2 layer was not observed by the ionosonde (not any data).

Thus we can conclude that each magnetic storm is very individual and requires a careful analysis in each case study.

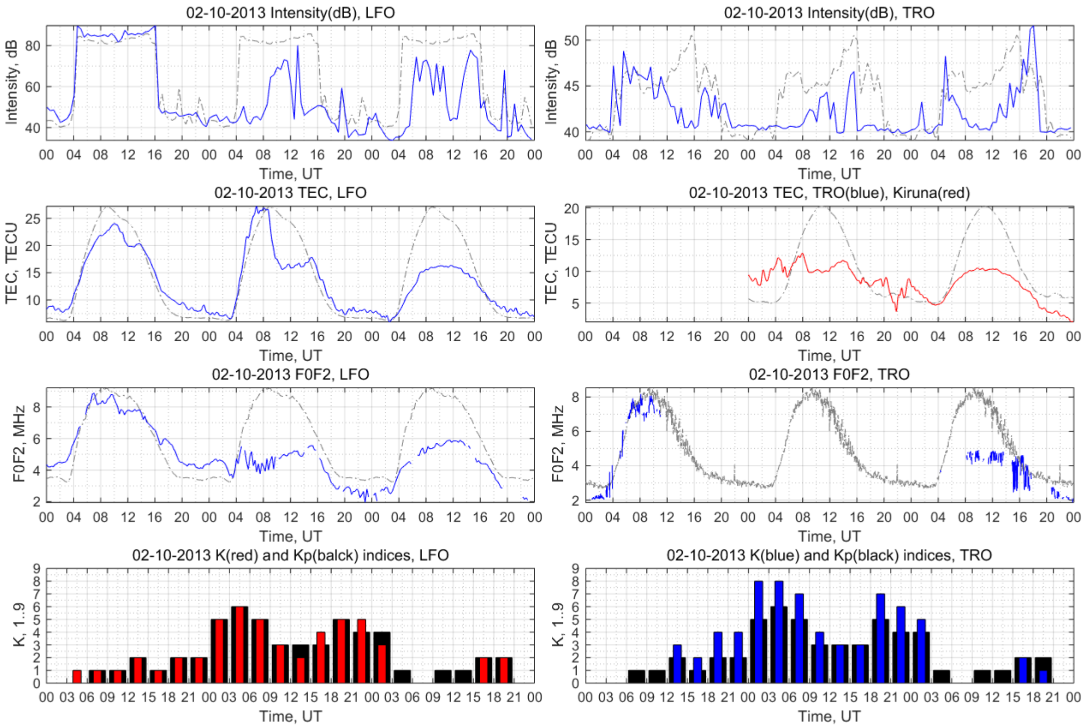

Figure 5.21: 01-03 October 2013. From top to bottom, variations of: HF signal intensity; TEC above the observation points; F0F2; local K (colored bars) and planetary Kp (black bars) magnetic indices. Left panels corresponds to LFO, right to Tromso.

6 Conclusions and future works

6.1 Conclusions

We presented in this work large-scale data processing and analysis of the behavior of HF signal parameters (intensity and Doppler frequency shift) of measurements carried out during 2013 for two radio paths: Moscow - Tromso (Norway, high latitude region) and Moscow - LFO (Kharkiv, Ukraine, middle latitude region). The data from the receiving stations at Tromso and LFO were processed by programs written in MatLab. In addition, the programs were written to visualize the HF signal parameters and ionospheric parameters (TEC, critical frequency F0F2), and geomagnetic and solar indices.

The following behaviors of HF signal parameters were detected:

- A reduction in the intensity of received signal at night and an increase of the intensity during the day. This is explained by the presence of solar ionization in the ionosphere and high electron density in the ionospheric layers, which reflect the HF signal [2, 9, 47].
- A reduction of intensity during the day due to the increase in the absorption of HF waves in the lower ionosphere. An additional contribution to the intensity reduction in the polar region (Tromso) is due to the polar day conditions and auroral absorption leading to additional absorption [8].
- An increase in signal intensity after sunset time may be due to the action of the neutral winds that contribute to the movement of charged particles above the F2 maximum where the losses of electrons are less [44, 45].
- A positive Doppler shift at sunrise shows that the electron density in the ionosphere increases and the reflection height of HF signal is reduced. In the absence of growth the electron density the Doppler shift is zero.
- A negative Doppler shift at sunset indicates that the reflection occurs out at higher altitudes with decreasing electron density and height of the reflection grows.
- Reduction of signal intensity during strong magnetic disturbances occur, since there is a change in TEC and F0F2, which leads to changes in the radio path compared to quiet conditions. Since the intensity of the signal is an integral parameter influenced by the many factors, that there is no a clear relationship with the ionospheric disturbances. The Doppler frequency shift is related to the parameters of ionospheric disturbances a more complicated manner, since during the disturbances the ionosphere dynamic may be very complex. The detailed analysis of each individual perturbations requires more time and more detailed data.

During analysis of HF signal parameters the behavior of the ionospheric parameters TEC and F0F2 were reviewed and the following features were detected:

- The standard behavior of TEC and F0F2 as a function of time of day - the increasing at daytime and the decreasing at night [48];
- The impact of the solar activity on the change TEC and F0F2 - when the flux $F_{10.7}$ increases then increases of TEC and F0F2 were observed (this is clearly seen in the annual scale).
- An increasing of the electron density after sunset time is due to the action of the neutral winds that contribute to transfer electrons to regions of the ionosphere with low recombination rates [44, 45].
- The TEC behavior during magnetic perturbations is ambiguous and may either increase or decrease, but the general trend is that during the first day of the magnetic storm the TEC increases, on the second day TEC decreases and in the following days TEC is restored to previous values [46].

6.2 Future works

In the future, we propose to make the following steps:
- Improve the MatLab code, to make it a friendlier to a user; improve the algorithm to prevent synchronization loss between the RWM station and the processing PC and to add processing of pulsed HF signal. The pulsed HF signal processing will improve the temporal resolution of the data, allow to see a more detailed behavior of the intensities and the Doppler frequency shift.
- Carry out a detailed analysis of the days with ionospheric disturbances involving additional information, such as a detailed flux of X-rays data, precipitation of energetic particles from the solar wind, auroral activity data.
- Make attempts to search for signatures of polar ionospheric inhomogeneities in the parameters of HF signals.

A MatLab Code

A.1 MatLab scripts for data processing

MatLab script "Run_RWM_processing" with the main function "Proc_RWM_CW_mod" for processing the signal files and log files, are used to the calculation of the intensities and DFS of HF received signals at LFO and Tromso. The next scripts are used to creating of the arrays with TEC data, F0F2 data, local magnetic K-indices and planetary Kp-index, the indices of Solar activity. Eventually, all data arrays are combined to one total .mat data file: 'All_parameters_tot.mat'. All scripts must be run consecutively for the correct filling of the intermediate .mat files and creating the total .mat data file. More information about the scripts are given inside the code by comments (identified by the starting "%" symbol and green colored).

A.1.1 Script "Run_RWM_processing.m"

This script is used to calling the main processing function "Proc_RWM_CW_mod" for both LFO and Tromso receivers during 2013 year. The description of the function is presented inside MatLab code by comments at section A.1.2.

```
% Script 'Run_RWM_processing' for calling function 'Proc_RWM_CW_mod'
% 'Proc_RWM_CW_mod' processes f02 and inf files from receivers,
% created arrays of DFS and intensities HF signal for one month
% and saved them to .mat files.

% Calling function for processing of data for Tromso:
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\01', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\02', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\03', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\04', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\05', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\06', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\07', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\08', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\09', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\10', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\11', -0.4, 9996000, 60, 10, 1, 30, [0 475]);
Proc_RWM_CW_mod('TRO', 'H:\HF\TRO\2013\12', -0.4, 9996000, 60, 10, 1, 30, [0 475]);

% Calling function for processing of data for LFO:
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\01', 0, 9996000, 60, -18, 1, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\02', 0, 9996000, 60, -18, 1, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\03', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\04', 0, 9996000, 60, -28, 2, 1000, [0 360]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\05', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\06', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\07', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\08', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\09', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\10', 0, 9996000, 60, -28, 2, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\11', 0, 9996000, 60, -38, 3, 1000, [0 475]);
Proc_RWM_CW_mod('LFO', 'H:\HF\LFO\2013\12', 0, 9996000, 60, -38, 3, 1000, [0 475]);
```

A.1.2 Function "Proc_RWM_CW_mod.m"

This function performs main processing of the signal and log files from receivers. The function extracts data from these files and calculates intensity and Doppler frequency shift of the HF signal for each day during one month. All variables are saved to .mat files. Inside the code

"CW" means the Continuous Wave – continuous mode operation of RWM station; the "f02" means the signal file; "inf" – the log file.

```
function [Type_inf_err] = Proc_RWM_CW_mod(sSite, data_fold_name, FD0, HF_frequency,...
                                     DT, AttRef, CnvAttRef, S2N_signal_Fr, T_lim_CW)
%
% Input:
%   sSite - observation point,(text parameter): 'TRO' or 'LFO';
%   data_fold_name - path folder where is .inf and .f02 files;
%   FD0 - correction the ZERO of Doppler for given receiver;
%   HF_frequency = 9996000 [Hz], HF frequency for processing;
%   DT = 60 [sec], chosen time resolution/ Can be 10,20,30,40,60,120,150,200,300...
%          (the duration each segment of data during day and the time interval...
%          for calculation mean spectrum for intensity and DFS);
%   AttRef - attenuation, only for this attenuation the data will be processed;
%   CnvAttRef - coefficient of attenuation, we need to multiply our data by
%               this coefficient for getting the original data;
%   S2N_signal_Fr -the criterion of existence of signal over noise for
%                    determination frequency for all modes;
%   T_lim_CW - band in sec when is CW mode, default [0 475];
% Output:
%   Type_inf_err - the possible errors after checking inf files:
%           0 - no errors, all files is correct;
%           1 - no lines at all;
%           2 - no first line;
%           3 - only first line;
%           4 - only first and last lines and no frequency lines;
%           5 - not exist chosen HF_frequency;
%           6 - not correct start date.
%
%   Also the function saves all workspace into MatLab current folder.
%===========================================================================

% Initial parameters:
T_cycle = 1800; % RWM cycle in sec for chosen mode (each 30 min is CW mode);
FatalError = 0; % initialization variable for Errors, which stop processing;
SampFreq = 2000; % Sampling frequency in Hz;

% SET PARAMETERS FOR DETERMINING EXISTANCE OF SIGNAL
dF_Mode = -54.5; % [Hz] frequency shift to determine level of noise;
% we must not get to harmonic of our signal on 1n and 10n Hz, n =1,2,3..
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Skipping interval for begin and end:
dTskip = [5 5]; % skip in seconds;
% Spectral intervals for Doppler frequency calculation:
dF_Fd = [-5 5]; % [from -Hz to +Hz] relative to central freq.;
% Spectral intervals for intensity calculation:
dF_Int = [-2 2]; % [from -Hz to +Hz] in this band-width is the main part of energy
% Length of individual realization for PWelch spectrum calculation:
dT_Sp = 10; % [sec], the time interval for calculation individual spectrum;
dF_Sp = 1/dT_Sp; % spectral resolution in Hz;
NP_Sp = dT_Sp*SampFreq; % calculate Number of points for one individual spectrum;

% Coefficients for converting amplitudes:
CnvMode = 1; % conversion amplitude for CW mode;

% Recalculate initial parameters:
NPT = round(86400/DT);% number of time intervals per day (1day=86400sec);
DATA = zeros(DT*SampFreq,1); % initialization variable for each full DT data segment;
SArr = (dTskip(1)*SampFreq+1):((DT-dTskip(2))*SampFreq); % positions of points in...
                                                     % each segment for processing;
DATAS = zeros((DT-dTskip(1)-dTskip(2))*SampFreq,1); % for truncated data segment;
[Sp,f] = pwelch(DATAS,boxcar(NP_Sp),NP_Sp/2,NP_Sp,SampFreq); % initialization length...
                                     % of Power Spectra Density =Sp and Frequency =f;
f = f-SampFreq/4-dF_Sp-FD0; % defenition array of the real frequencies;
[~,dFpntFd(1)] = min(abs(dF_Fd(1)-f)); % low position in f array for DFS calculation;
[~,dFpntFd(2)] = min(abs(dF_Fd(2)-f)); % high position in f array for DFS calculation;
%%
% Find f02 files:
f02_files = dir([data_fold_name,'/*.f02']); % list of names .f02 files(data files);
Nf02 = size(f02_files,1); % quantity of data files;
```

```
% Find inf files:
inf_files = dir([data_fold_name,'/*.inf']); % list of names .inf files(log files);
Ninf = size(inf_files,1); % quantity of inf files;

% Initialization indicator parameters:
N_lines_inf = zeros(Ninf,1); % the variable for quantity lines in .inf files;
ContinueWorking = zeros(Ninf,1); % the variable for flag(indicator) to continue...
                                  % processing data file or not;
Type_inf_err = zeros(Ninf,1); % the variable  for type of the error in inf file;

% Initialization arrays for output parameters:
Fd_max_all = zeros(NPT,1,Nf02); % zeros array for Doppler by maximum method;
Fd_moment_all = zeros(NPT,1,Nf02); % zeros array for Doppler by moment method;
Int_max_all = zeros(NPT,1,Nf02); % zeros array for Intensity by absolute maximum;
Int_enrg_all = zeros(NPT,1,Nf02); % zeros array for Intensity in range dF_Int;
S2N = zeros(NPT,1,Nf02); % zeros array for Signal to Noise ratio;
Date_Arr_all = zeros(1,Nf02); % zeros array for all dates of processing files;

% Initialization arrays for other parameters:
Arr_Att = zeros(NPT,Nf02); % zeros array for daily distributions of attenuation:
                           % Value is the number of dB,...
                           % 999 - is no signal;
Arr_Freq = zeros(NPT,Nf02);% zeros array for indicators of daily distributions...
               % HF_frequency:
                  % 0 - is no HF_frequency
                  % 1 - HF_frequency is all time
                  % 2 - HF_frequency with changing frequency or attenuation
Arr_FulnsModes = zeros(NPT,1); % zeros array for indicators of fullness modes values:
                          % 0 - OFF, 1 - ON,
                          % 2 - ON but stop 5 sec before END;

% Reading date from f02 data file names:
Year = str2double(inf_files(1).name(1:4)); % year of processing files;
Month = str2double(inf_files(1).name(5:6)); % month of processing files;

% Array of times beginning each DT segment for data processing:
Arr_T0_b = linspace(DT,NPT*DT,NPT)-DT;
% Array of times mid point each DT segment:
Arr_T0 = Arr_T0_b+DT/2;

%% Determination of the fullness each DT segment in files during CW mode:
for k=1:NPT, % k - index of time interval in day
  t_cycle = mod(DT*(k-1)+DT/2,T_cycle);% time of the middle each DT segment in each...
                                         % CW intervals (T_cycle)
  if (t_cycle>T_lim_CW(1)) && (t_cycle<T_lim_CW(2)),
     Arr_FulnsModes(k,1) = 1;
     if T_lim_CW(2)-t_cycle < DT/2, %if last DT segment don't fit in boundary T_lim_CW;
         Arr_FulnsModes(k,1) = 2; % then this segment get flag 2;
     end;
  end;
end;

%% Test that files are OK and have the correct HF carrier frequency:
if Nf02~=Ninf, % check that quantity of data files(Nf02)=quantity of switching log...
                                                                   % files (Ninf)
    disp('Number of f02 not correspond to number of inf -> exit');
    FatalError = 1;
    if Nf02>Ninf
       disp('Number of f02 more then number of inf');
    elseif Nf02<Ninf
       disp('Number of f02 less then number of inf');
    end
else
    % Test each file:
    for k=1:Ninf, % k - index of file;
        % Check that file names f02 and inf are correct:
        if strcmp(f02_files(k).name(1:23),inf_files(k).name(1:23))

          % Check that correct HF_frequency exist in inf file and calc:
            % Open inf file and get first line:
            finf = fopen([data_fold_name,'\',inf_files(k).name],'r');
            inf_line = fgetl(finf); % read 1-st line in inf file;
```

```
    % Get other lines until end of file:
    if inf_line~=-1, % if line is full
        % Get Date from first line:
        splited_str = strsplit(inf_line,' ');
        sDATE_inf = cell2mat(splited_str(1));
        sDATE = [sDATE_inf(1:4),sDATE_inf(6:7),sDATE_inf(9:10)];

        HF_frequency_inf = cell2mat(splited_str(3));

  % Check correct start date:
        if strcmp(sDATE,inf_files(k).name(1:8))~=1,
          disp(sprintf('file (%d) %s has not correct start date %s',...
                                          k,inf_files(k).name,sDATE));
          Type_inf_err(k) = 6;
          ContinueWorking(k) = 0;
        end;

  % Check format of the first line of inf file:
        if strcmp(HF_frequency_inf,'LogData_Started')
            N_lines_inf(k) = 1; % we have already ONE line in file at least;
        else
            disp(sprintf('file (%d) %s has not correspond first line', ...
                                        k,inf_files(k).name));
            Type_inf_err(k) = 2;
            inf_line=-1;
        end;
  % Check other lines of inf file:
        while inf_line~=-1, % while line is full;
            inf_line = fgetl(finf); % read line;
                if inf_line~=-1, % if line is full;
                    N_lines_inf(k) =  N_lines_inf(k)+1; % increase the...
                                                          % counter of lines;
                    splited_str = strsplit(inf_line,' ');
                    HF_frequency_inf = cell2mat(splited_str(3));% read HF freq;
                    if strcmp(HF_frequency_inf,'LogClosed')% if line is the...
                                                           % END of inf file;
                        N_lines_inf(k) =  N_lines_inf(k)-1; % decrease the...
                                                           % counter of lines;
                        if (ContinueWorking(k)==0) && (N_lines_inf(k)==1),
                            disp(sprintf('file (%d) %s has first and last...
                                    lines but no frequency line',...
                                       k,inf_files(k).name));
                            Type_inf_err(k) = 4;
                        end;
                    else % if line is not the END and has Frequency value;
      % Check frequency in line correspond to frequency from first line:
                        if HF_frequency==str2double(HF_frequency_inf),
                            ContinueWorking(k) = 1; % else: ContinWorking = 0;
                        end;
                    end;
                else
                    if N_lines_inf(k)<2,
                        if strcmp(HF_frequency_inf,'LogClosed')~=1,
                            disp(sprintf('file (%d) %s has only first line...
                                and no frequency lines',k,inf_files(k).name));
                            Type_inf_err(k) = 3;
                        end;
                    else
                        if ContinueWorking(k)==0,
                            disp(sprintf('file (%d) %s has no lines with...
                          frequency %d Hz',k,inf_files(k).name,HF_frequency));
                            Type_inf_err(k) = 5;
                        end;
                    end;
                end;
        end; % the end of checking other lines of inf file.
    else % if current inf file don't has lines:
        N_lines_inf(k) = 0;
        disp(sprintf('file (%d) %s has no lines',k,inf_files(k).name));
        sDATE = inf_files(k).name(1:8);
        Type_inf_err(k) = 1;
    end; % the end of checking of the content current inf file;
    fclose(finf);
```

```
        else % if strcmp(f02_files(k).name(1:23),inf_files(k).name(1:23))...
             % if file names are not correct;
        disp(sprintf('files (%d) %s not correspond to %s - exit',...
                                    k,f02_files(k).name,inf_files(k).name));
        FatalError = 1;
        end;
    end; % the end of checking of names f02 and inf files and inf file content;
end;% the end of checking of quantities and names f02 and inf files and...
                                                                  % inf file content;
%% Exit if fatal error:
if FatalError, return;
end;
if all(ContinueWorking),
    disp(sprintf('All files in %s are OK',data_fold_name));
else % if
    disp(sprintf('Files in %s are OK, but some of them will not use.',data_fold_name));
    for k=1:Ninf,
        if ContinueWorking(k)==0,
            disp(sprintf('file (%d) %s will not use in calculations', ...
                                                    k,inf_files(k).name));
        end;
    end;
end;

%=====================================================================================
%% Processing f02 files:
% Determination daily dependencies for signal parameters.
% Main loop for all files.

for i_f02=1:Nf02, % i_f02 - index of the data files, start processing f02 files;
    if ContinueWorking(i_f02)==1, % if inf files are OK after previous checking;
    % Filling data array:
        Date_Arr_all(i_f02) = str2double(inf_files(k).name(7:8));
    % Parsing correct inf file
            % Initialization arrays for coorections f02 data:
            inf_time= zeros((N_lines_inf(i_f02)-1),1); % initialization array for...
                                                          % times from inf file;
            inf_freq= zeros((N_lines_inf(i_f02)-1),1); % initialization array for...
                                                    % frequencies from inf file;
            inf_att = zeros((N_lines_inf(i_f02)-1),1); % initialization array for...
                                                    % attenuations from inf file;

            finf = fopen([data_fold_name,'\',inf_files(i_f02).name],'r');
            inf_line = fgetl(finf); % get 1-st line from inf file;
                for j=2:N_lines_inf(i_f02), % j - index of line in inf file,
                    inf_line = fgetl(finf); % get from 2-nd line to end in inf file;
                    splited_str = strsplit(inf_line,' ');
                    sTIME = cell2mat(splited_str(2));
                    inf_time((j-1),1) = 3600*str2double(sTIME(1:2))+ ...
                  60*str2double(sTIME(4:5))+str2double(sTIME(7:12)); % the time in sec;
                    inf_freq((j-1),1) = str2double(cell2mat(splited_str(3)));% value...
                                                                  % of frequency
                    inf_att((j-1),1) = str2double(cell2mat(splited_str(4)));% value...
                                                                % of attenuation
                end;
            fclose(finf);
            clear j;

    % Calculation distributions of attenuation and correct HF frequency:
        for j=1:(N_lines_inf(i_f02)-1), % j - index of line in inf file;
            Nind = sum((inf_time(j,1)-Arr_T0_b)>=0); % Nind -index of time interval...
                                                        % in array Arr_T0_b;
            if inf_freq(j,1)==HF_frequency,%if freq in time interval is correct for us;
                Arr_Freq(Nind:NPT,i_f02) = 1;
            else
                Arr_Freq(Nind,i_f02) = 2;% if freq in time interval is not correct;
            end;
            Arr_Att(Nind:NPT,i_f02) = inf_att(j,1);
        end;
```

```
 % Set time boundaries corresponding to f02 file size:
    Nmin = ceil(inf_time(1,1)/DT);
    if Nmin==0,
        Nmin = 1;
    end;
    Nmax = floor((inf_time(1,1)+f02_files(i_f02).bytes/2/SampFreq)/DT);
    if Nmax==0,
        Nmax = 1;
    end;
    Arr_Freq(1:Nmin,i_f02) = 0; % no freq. data outside time boundaries f02 file;
    Arr_Freq(Nmax:NPT,i_f02) = 0;
    Arr_Att(1:Nmin,i_f02) = 999; % no attenuat data outside time boundaries...
                                                                    % f02 file;
    Arr_Att(Nmax:NPT,i_f02) = 999;



%% Fill DATA segments for one day with corresponding to data from f02 file:
    disp(sprintf('Processing file %s (%d from %d)', ...
                                            f02_files(i_f02).name,i_f02,Ninf));
    ff02 = fopen([data_fold_name,'\',f02_files(i_f02).name],'r');

    for indDT=1:NPT, % indDT - index of DT segment inside Day
 % Quantity of bites for shift from start of day (00:00:00 UT) relative to first...
                                                              % time in inf file:
        dBytes = (indDT*DT-inf_time(1,1))*SampFreq*2;
        if (dBytes>0)&&(dBytes<(f02_files(i_f02).bytes-DT*SampFreq*2))
            fseek(ff02, dBytes, 'bof');
            DATA = double(fread(ff02,DT*SampFreq,'int16'));% fill of DATA segment;
        else %if dBytes<0 then data from receiver still empty,...
                                   % or dBytes > size of f02 file;
            DATA = zeros(DT*SampFreq,1);
        end;

 % Processing current DATA segment and fill corresponding data array:
   % Select segments with only HF_frequency and CW mode:
        if (Arr_Freq(indDT,i_f02)==1)&&(Arr_FulnsModes(indDT)==1),% without any...
                                                      % changes in current segment;
            FreqResp = 1; % flag of no changes in current segment;
        else
            FreqResp = 0;
        end;

   % Select segments with only AttRef and sellected coefficients for attenuation:
        if Arr_Att(indDT,i_f02)==AttRef,
            AttResp = CnvAttRef*CnvMode;
        else
            AttResp = 0;
        end;

   % Convert DATA to truncated size and make amplitude conversion:
        DATAS = DATA(SArr)*FreqResp*AttResp;

   % Make spectral processing if DATAS is not zero:
        if all(DATAS==0)

        else % when DATAS~=0

   % Calculate spectrum by Welch method for nonzero data:
            [Sp,f] = pwelch(DATAS,boxcar(NP_Sp),NP_Sp/2,NP_Sp,SampFreq);
            f = f-SampFreq/4-dF_Sp-FD0; % frequency from -Sp...0... to +Sp range;

   % Calculate Doppler frequensy and Intensity in Doppler Maximum:
            [Int_max_all(indDT,1,i_f02),FDop_ind] = max(Sp(dFpntFd(1):dFpntFd(2)));
                                         % [the max of spectrum,index of this max];
            Int_max_all(indDT,1,i_f02) = Int_max_all(indDT,1,i_f02)*dF_Sp;
                                                % intensity in point max spectrum;
            Fd_max_all(indDT,1,i_f02) = f(FDop_ind+dFpntFd(1)-1);% DFS (frequency...
                                                              % of point max);
   % Doppler frequency calculation by the method of moments:
            Fd_moment_all(indDT,1,i_f02)= sum(Sp(dFpntFd(1):dFpntFd(2)).*...
                         f(dFpntFd(1):dFpntFd(2)))/ sum(Sp(dFpntFd(1):dFpntFd(2)));
```

```
        % Calculate range for signal intensity calculation:
                [~,dFpntInt(1)] = min(abs(dF_Int(1)+Fd_moment_all(indDT,1,i_f02)-f));
                [~,dFpntInt(2)] = min(abs(dF_Int(2)+Fd_moment_all(indDT,1,i_f02)-f));

        % Calculate intensity in selected range:
                Int_enrg_all(indDT,1,i_f02) = ...
            trapz(f(dFpntInt(1):dFpntInt(2)),Sp(dFpntInt(1):dFpntInt(2))); % Intensity;

        % Calculate range for noise intensity calculation:
             [~,dFpntInt(1)]=min(abs(dF_Int(1)+Fd_moment_all(indDT,1,i_f02)+dF_Mode-f));
             [~,dFpntInt(2)]=min(abs(dF_Int(2)+Fd_moment_all(indDT,1,i_f02)+dF_Mode-f));
                Int_enrg_noise = ...
trapz(f(dFpntInt(1):dFpntInt(2)),Sp(dFpntInt(1):dFpntInt(2)));% intensity of noise;

        % Calculate signal to noise ratio:
                S2N(indDT,1,i_f02) = Int_enrg_all(indDT,1,i_f02)/Int_enrg_noise;

        % Determine existence of signal and make additional tuning if signal is not...
                                                                             % exist:
              % if signal not detected make DFS equal to zero
                if S2N(indDT,1,i_f02)<S2N_signal_Fr,
                    Fd_max_all(indDT,1,i_f02) = 0;
                    Fd_moment_all(indDT,1,i_f02) = 0;
                 end;
             end; % end for else (when DATAS~=0);

        end; % end of the loop of filling DATA segments;
        fclose(ff02);

    % Skiping not correct inf file:
    else % if ContinueWorking(i_f02)==0, if inf file is not OK after checking;
        Arr_Att(:,i_f02) = 999; % no data;
        Arr_Freq(:,i_f02) = 0; % no data;
    end;
end; % the end processing f02 files;
%%
% Put array to one set with complete days:
% (the connecting of data if we have data for one day inside two different files)
NDay = 1; % days for arrays with unique days
NDay_all = 1; % days for all arrays

% Set first values:
Fd_max(:,:,NDay) = Fd_max_all(:,:,NDay_all);
Fd_moment(:,:,NDay) = Fd_moment_all(:,:,NDay_all);
Int_max(:,:,NDay) = Int_max_all(:,:,NDay_all);
Int_enrg(:,:,NDay) = Int_enrg_all(:,:,NDay_all);
Date_Arr(NDay) = Date_Arr_all(NDay_all);

% Calculate arrays for all days:
while NDay_all<Nf02,
    NDay_all = NDay_all+1;
    if Date_Arr_all(NDay_all)==Date_Arr_all(NDay_all-1),%if we have data for one day...
                                                          % inside two different files
        Fd_max(:,:,NDay) = Fd_max(:,:,NDay)+Fd_max_all(:,:,NDay_all);
        Fd_moment(:,:,NDay) = Fd_moment(:,:,NDay)+Fd_moment_all(:,:,NDay_all);
        Int_max(:,:,NDay) = Int_max(:,:,NDay)+Int_max_all(:,:,NDay_all);
        Int_enrg(:,:,NDay) = Int_enrg(:,:,NDay)+Int_enrg_all(:,:,NDay_all);
    else
        NDay = NDay+1;
        Fd_max(:,:,NDay) = Fd_max_all(:,:,NDay_all);
        Fd_moment(:,:,NDay) = Fd_moment_all(:,:,NDay_all);
        Int_max(:,:,NDay) = Int_max_all(:,:,NDay_all);
        Int_enrg(:,:,NDay) = Int_enrg_all(:,:,NDay_all);
        Date_Arr(NDay) = Date_Arr_all(NDay_all);
    end;

end;

% Calculate mean arrays throw all days for one month:
Fd_max_mean = mean(Fd_max,3); % 3 it is dimensions of NDay;
Fd_moment_mean = mean(Fd_moment,3);
Int_max_mean = mean(Int_max,3);
Int_enrg_mean = mean(Int_enrg,3);
```

```
% Calculate arrays with NaN instead of zeros:
% Initialization arrays:
Fd_max_NaN = Fd_max;
Fd_moment_NaN = Fd_moment;
Int_max_NaN = Int_max;
Int_enrg_NaN = Int_enrg;
Fd_max_mean_NaN = Fd_max_mean;
Fd_moment_mean_NaN = Fd_moment_mean;
Int_max_mean_NaN = Int_max_mean;
Int_enrg_mean_NaN = Int_enrg_mean;

% Convert zeros to NaN:
Fd_max_NaN(find(~Fd_max_NaN)) = NaN;
Fd_moment_NaN(find(~Fd_moment_NaN)) = NaN;
Int_max_NaN(find(~Int_max_NaN)) = NaN;
Int_enrg_NaN(find(~Int_enrg_NaN)) = NaN;
Fd_max_mean_NaN(find(~Fd_max_mean_NaN)) = NaN;
Fd_moment_mean_NaN(find(~Fd_moment_mean_NaN)) = NaN;
Int_max_mean_NaN(find(~Int_max_mean_NaN)) = NaN;
Int_enrg_mean_NaN(find(~Int_enrg_mean_NaN)) = NaN;

% Preparation for calculation of mean values for 10-minites intervals
NP600s = 144; % quantity of the 10-min intervals per day;
DT600s = 600; % quantity of seconds in 10-min interval;
% Array of exact time of 10-min intervals (not equally spaced);
for k=1:3:144,
    Arr_T600s(k) = 210+(k-1)/3*1800; % the points of time of data segments;
    Arr_T600s(k+1) = 870+(k-1)/3*1800; % the points of time of seconds NaN segments;
    Arr_T600s(k+2) = 1470+(k-1)/3*1800;% the points of time of thirds NaN segments;
end

% Equally spaced time array of 10-min intervals:
Arr_T = linspace(270,86130,144);
%
% Initialization mean values for 10-min intervals:
Fd_ma_NaN = zeros(NP600s,1,NDay)*NaN;
Fd_mo_NaN = zeros(NP600s,1,NDay)*NaN;
Int_ma_NaN = zeros(NP600s,1,NDay)*NaN;
Int_en_NaN = zeros(NP600s,1,NDay)*NaN;
Fd_ma_mean_NaN = zeros(NP600s,1)*NaN;
Fd_mo_mean_NaN = zeros(NP600s,1)*NaN;
Int_ma_mean_NaN = zeros(NP600s,1)*NaN;
Int_en_mean_NaN = zeros(NP600s,1)*NaN;

% Calculate mean values:
NP_mean = round(DT600s/DT);
for k=1:NP600s, % k - index of each 10 min interval;
    NP_600s(:,:) = (1+NP_mean*(k-1)):(NP_mean*k); % indices of DT segments for...
                                              % averaging and getting points of DT600s;
    Fd_ma_NaN(k,:,:) = nanmean(Fd_max_NaN(NP_600s,:,:),1);
    Fd_mo_NaN(k,:,:) = nanmean(Fd_moment_NaN(NP_600s,:,:),1);
    Int_ma_NaN(k,:,:) = nanmean(Int_max_NaN(NP_600s,:,:),1);
    Int_en_NaN(k,:,:) = nanmean(Int_enrg_NaN(NP_600s,:,:),1);
    Fd_ma_mean_NaN(k,:) = nanmean(Fd_max_mean_NaN(NP_600s,:),1);
    Fd_mo_mean_NaN(k,:) = nanmean(Fd_moment_mean_NaN(NP_600s,:),1);
    Int_ma_mean_NaN(k,:) = nanmean(Int_max_mean_NaN(NP_600s,:),1);
    Int_en_mean_NaN(k,:) = nanmean(Int_enrg_mean_NaN(NP_600s,:),1);
end;

% Folder name and path for saving of .mat files:
    mat_folder_name = [pwd,'\',sSite]; % in current folder of Matlab;
    mkdir(mat_folder_name);
    % Save workspace:
    save(sprintf('%s\\%04d%02d_%08dHz_%s_%04dsec_dF',mat_folder_name,Year,Month,...
                                                        HF_frequency,sSite,DT);

end % the end of function.
```

A.1.3 Script "Set_yearly_time_HF_param13.m"

This script calculates the time parameters for intensities and DFS: time of data points in seconds for each day, in MatLab date time for one year, in days number during year, in month number during year. Also here are initialized the yearly arrays for intensities and DFS. These parameters are saved to MatLab file: 'All_parameters_1.mat'.

```
% Script "Set_yearly_time_HF_param13" for initialization of yearly HF parameters
% (2013 year) for CW mode: time arrays and initialization arrays for intensities
%  and Doppler frequency shift for LFO and TRO.

% Set initial parameters:
Year = 2013; % Year;
Nt = 48; % Number of data points during one day;
Nd = 365; % Number of days in one year;

% Initialization Time parameters:
TdayHF_sec = zeros(Nt,1); % array for the time of HF data points in 'seconds' for...
                                                                          % one day;
TyearHF_dt13 = zeros(Nt,Nd);% array for the time of HF data points in...
                                                     % 'MatLab DateTime' during one year;
TyearHF_nd13 = zeros(Nt,Nd); % array for the time of HF data points in...
                                                         % 'number of day' during one year;
TyearHF_y13 = zeros(Nt,Nd); % array for the time of HF data points in 'Year';
TyearHF_m13 = zeros(Nt,Nd); % array for the time of HF data points in 'Months'...
                                                                      % during one year;
TyearHF_d13 = zeros(Nt,Nd); % array for the time of HF data points in 'Days in month'
                                                                       % during one year;

% Set initial HF parameters:
for k=1:Nt, % k - index of data points during one day;
    TdayHF_sec(k,1) = 240+1800*(k-1); % daily Time in sec;
end;
for j=1:Nd, % j -index of days during one month;
    for k=1:Nt, % k - index of data points during one day;
      TyearHF_dt13(k,j) = datenum(Year,1,1)+(TdayHF_sec(k,1)+86400.*(j-1))/86400.;
                                                       % yearly T in MatLab datetime;
      TyearHF_nd13(k,j) = j; % yearly days mumber;
      [TyearHF_y13(k,j),TyearHF_m13(k,j),TyearHF_d13(k,j)]=datevec(TyearHF_dt13(k,j));
                                                                     % yearly [Y M D];
    end;
end;

% Initialization HF parameters -
% arrays for Doppler frequency shift (DFS) and Intensity (Int):

for Np=1:2 % Np is Stations number: 1 - LFO; 2 - TRO;
  FmoN13 = zeros(Nt,Nd,Np)*NaN; %array for regular DFS calculated by moments;
  FmomN13 = zeros(Nt,Nd,Np)*NaN;%array for monthly averaged DFS calculated by moments;
  FmaN13 = zeros(Nt,Nd,Np)*NaN; %array for regular DFS calculated by maximum;
  FmamN13 = zeros(Nt,Nd,Np)*NaN;%array for monthly averaged DFS calculated by maximum;
  IenN13 = zeros(Nt,Nd,Np)*NaN; %array for regular Int calculated by energy...
                                                                     % (+-2Hz from max);
  IenmN13 = zeros(Nt,Nd,Np)*NaN; %array for monthly averaged Int calculated by...
                                                              % energy (+-2Hz from max);
  ImaN13 = zeros(Nt,Nd,Np)*NaN; %array for regular Int calculated by maximum;
  ImamN13 = zeros(Nt,Nd,Np)*NaN;%array for monthly averaged Int calculated by maximum;
end;

clear('j','k','Np');

save('All_parameters_1.mat'); % save all parameters to mat file;
```

A.1.4 Script “Calc_daily_HF_CW13.m”

The following script is used for creating 48 points of the intensity and DFS of the HF signal for each day during year from arrays of function "Proc_RWM_CW_mod". Also this script combines all days to yearly arrays. This script must be run only after previous scripts.

```
% Script "Calc_daily_HF_CW13" calculate daily HF variations (2013 year) CW mode
% Script converts 144 data points to 48 points per day for each day and
% connected all months into the yearly arrays. The monthly data is arrays from
% function 'Proc_RWM_CW_mod';
% The missing days are NaN.
%
load('All_parameters_1.mat');
%
for StN = 1:2; % station number (1 - LFO; 2 - TRO);
  for ind_month=1:12
    if StN==1
        Dialog_title = sprintf('Select .mat data file for LFO %d month',ind_month);
    elseif StN==2
        Dialog_title = sprintf('Select .mat data file for TRO %d month',ind_month);
    end;

 % Select the monthly data file to loading from dialog window:
    [MatFileName,MatFilePth]=uigetfile('2013*.mat',Dialog_title);

    if isequal(MatFileName,0) % if no filename or selected 'Cancel' in dialog window;
        return; % stop loop;
    else
        load([MatFilePth,MatFileName]); % load data from the monthly mat file;
        List_var=whos('-file',[MatFilePth,MatFileName]); % list of variables from...
                                                           % monthly data file;
        list_names = cell(1,length(List_var));% initialization variable for the names;

    % List of the variable names which is deleted before loading new...
                                                        % month data array:
        for i=1:length(List_var)
            list_names{1,i}=List_var(i).name; % list of names variables from...
                                              % monthly data file;
        end;
    end;

  Npnt = length(Date_Arr);% numbers of processed dates for current month;

    % Set values HF parameters arrays:
    for k=1:Npnt, % k - day numbers wich were processed in current month;
      N_day = day(datetime([Year Month Date_Arr(k)]),'dayofyear'); % day number...
      % according to Year, Month and day from Date_Arr from monthly data array;
      FmoN13(:,N_day,StN) = Fd_mo_NaN(1:3:144,1,k);
      FmomN13(:,N_day,StN) = Fd_mo_mean_NaN(1:3:144,1);
      FmaN13(:,N_day,StN) = Fd_ma_NaN(1:3:144,1,k);
      FmamN13(:,N_day,StN) = Fd_ma_mean_NaN(1:3:144,1);
      IenN13(:,N_day,StN) = Int_en_NaN(1:3:144,1,k);
      IenmN13(:,N_day,StN) = Int_en_mean_NaN(1:3:144,1);
      ImaN13(:,N_day,StN) = Int_ma_NaN(1:3:144,1,k);
      ImamN13(:,N_day,StN) = Int_ma_mean_NaN(1:3:144,1);
    end;

  clear(list_names{:}); % clear variable before loading new month data array:
  clear('StN','ind_month','Dialog_title','List_var','list_names','i','Npnt');
  end;

end;

save('All_parameters_2.mat'); % save all HF parameters to .mat file;
```

A.1.5 Script "Calc_m_index.m"

This script creates the indices of each day (day number) during 2013 year and indices of the begin day and the end day of each month in year. The script must be run only after previous script ("Calc_daily_HF_CW13"). The array m_indxs is very important and will be used for averaging TEC and F0F2 data in next scripts. The array "mind" will be used for plotting figures in next scripts. The indices are saved to individual .mat file: 'm_index.mat' and are added to .mat file 'All_parameters_2.mat'.

```
% Script "Calc_m_index" sets indices parameters for monthly averaging TEC and F0F2,
% and indices (begin day and end day of each month) for plotting figures...
% in the next scripts.

% Set array for indices begin day and end day of each month:
mind = zeros(12,2)*NaN; % this array will be used for plotting figures;

%% Set parameters for monthly averaging:
m_indxs=cell(12,1); % initialization array for day numbers for each month;

m_indxs{1,1} = 1:31; % m01, day numbers in January;
m_indxs{2,1} = 32:59; % m02, day numbers in February;
m_indxs{3,1} = 60:90; % m03, day numbers in Marth;
m_indxs{4,1} = 91:120; % m04, day numbers in April;
m_indxs{5,1} = 121:151;% m05, day numbers in May;
m_indxs{6,1} = 152:181;% m06, day numbers in June;
m_indxs{7,1} = 182:212;% m07, day numbers in July;
m_indxs{8,1} = 213:243;% m08, day numbers in August;
m_indxs{9,1} = 244:273;% m09, day numbers in September;
m_indxs{10,1} = 274:304;% m10, day numbers in October;
m_indxs{11,1} = 305:334;% m11, day numbers in November;
m_indxs{12,1} = 335:365;% m12, day numbers in December;

for i=1:12, % i - month in year;
    m=m_indxs{i,1}; % range of day numbers for current month;
    mind(i,1) = m(1); % first day in month;
    mind(i,2) = m(end); % last day in month;
end;
clear('i','m');
save('m_index.mat'); % save month indices in .mat file;
save('All_parameters_2.mat','-append'); % add 'm_index' and 'mind' to HF prameters;
```

A.1.6 Script "Calc_TEC_13.m"

This script reads special diurnal TEC data files (for LFO, Kiruna, Joensuu and Tromso) and creates the yearly arrays with format 288x365, where 288 – number of points per one day, 365 – days in 2013 year. In additional, here are created the time arrays for TEC data points. This script must be run only after previous script ("Calc_m_index"). TEC parameters are saved to .mat file: 'All_parameters_3.mat' (together with HF parameters).

```
% Script "Calc_TEC_13" calculates yearly TEC parameters (2013 year)
load('All_parameters_2.mat'); % after script 'Calc_m_index_13';
% Set initial parameters:
Year = 2013; % Year
Nt = 288; % Number of points during day;
Nd = 365; % Number of days in year;
Np = 4;  % Stations number: % 1 - LFO
                             % 2 - Kiruna
                             % 3 - Joensuu
                             % 4 - Tromso
TdayTEC_sec = zeros(Nt,1); % array for the time of TEC data points in sec during day;
TyearTEC_dt13 = zeros(Nt,Nd);% array for the time of TEC data points in...
                                  % 'MatLab DateTime' during one year;
```

```
TyearTEC_nd13 = zeros(Nt,Nd);% array for the time of TEC data points in...
                                % 'day number' during one year;
TyearTEC_y13 = zeros(Nt,Nd);% array for the time of TEC data points in 'Year';
TyearTEC_m13 = zeros(Nt,Nd);% array for the time of TEC data points in...
                                              % 'Months' during one year;
TyearTEC_d13 = zeros(Nt,Nd);% array for the time of TEC data points in...
                                                 % 'Days in month' during one year;
%% Calculate Time TEC parameters:
for k=1:Nt,
    TdayTEC_sec(k,1) = 0+300*(k-1); % array of time of TEC data points in sec...
                                                                  % during day;
end;
for j=1:Nd,
    for k=1:Nt,
        TyearTEC_dt13(k,j) = datenum(Year,1,1)+(TdayTEC_sec(k,1)+...
                        86400.*(j-1))/86400.; % yearly Time in datetime;
        TyearTEC_nd13(k,j) = j; % yearly days number;
        [TyearTEC_y13(k,j), TyearTEC_m13(k,j), TyearTEC_d13(k,j)] =...
                                            datevec(TyearTEC_dt13(k,j));
                                                        % yearly [Y M D];
    end;
end;

% Set initial TEC parameters:
TECN13 = zeros(Nt,Nd,Np)*NaN; % array for TEC data;
TECmN13 = zeros(Nt,Nd,Np)*NaN; % array for monthly averaged TEC data;
TECe13 = zeros(Nd,Np); % array for flags of existence of TEC data for current day;

% Chouse TEC array for LFO:
khar_files = dir('khar*.13A'); % names of khar (LFO) - TEC data files;
Nkhar = size(khar_files,1); % number of LFO TEC data files;
% Calc TEC for LFO:
for k=1:Nkhar, % k - index of files (days) during year;
    TECcur = dlmread(khar_files(k).name);% read data from current file;
    % TECcur - temp array, 1st column - time, 2nd - TEC values;
    if all(abs(TECcur(:,1)-TdayTEC_sec(:,1))<100), % check of the existence...
                                                     % of all times during day;
        NDcur = str2num(khar_files(k).name(8:10)); % day number current TEC file;
        TECN13(:,NDcur,1) = TECcur(:,2); % write TEC data to the early array;
        TECe13(NDcur,1) = 1; % flag of existence data for current day;
    end;
end;

% Chouse TEC array for Kiruna:
kir0_files = dir('kir0*.13A'); % names of Kiruna - TEC data files;
Nkir0 = size(kir0_files,1); % number of Kiruna TEC data files;
% Calc TEC for Kiruna:
for k=1:Nkir0, % k - index of files (days) during year;
    TECcur = dlmread(kir0_files(k).name); % read data from current file;
    if all(abs(TECcur(:,1)-TdayTEC_sec(:,1))<100), % check of the existence...
                                                     % of all times during day;
        NDcur = str2num(kir0_files(k).name(8:10)); % day number current TEC file;
        TECN13(:,NDcur,2) = TECcur(:,2); % write TEC data to the early array;
        TECe13(NDcur,2) = 1; % flag of existence data for current day;
    end;
end;

% Chouse TEC array for Joensuu:
joen_files = dir('joen*.13A'); % names of Joen - TEC data files;
Njoen = size(joen_files,1); % number of Joen data files;
% Calc TEC for Joensuu:
for k=1:Nkir0, % k - index of files (days) during year;
    TECcur = dlmread(joen_files(k).name); % read data from current file;
    if all(abs(TECcur(:,1)-TdayTEC_sec(:,1))<100), % check of the existence...
                                                     % of all times during day;
        NDcur = str2num(joen_files(k).name(8:10)); % day number current TEC file;
        TECN13(:,NDcur,3) = TECcur(:,2); % write TEC data to the early array;
        TECe13(NDcur,3) = 1; % flag of existence data for current day;
    end;
end;

% Chouse TEC array for Tromso:
tro1_files = dir('tro1*.13A'); % names of Tro - TEC data files;
```

```
Ntro1 = size(tro1_files,1); % number of Tro data files;

% Calc TEC for Tromso:
for k=1:Ntro1,
    TECcur = dlmread(tro1_files(k).name); % read data from current file;
    if all(abs(TECcur(:,1)-TdayTEC_sec(:,1))<100), % check of the existence...
                                                   % of all times during day;
        NDcur = str2num(tro1_files(k).name(8:10)); % day number current TEC file;
        TECN13(:,NDcur,4) = TECcur(:,2); % write TEC data to the early array;
        TECe13(NDcur,4) = 1; % flag of existence data for current day;

    end;
end;

%% Mothly averaging:
for k=1:Np, % k - station number;
    for i=1:12, % i - month in year;
        m=m_indxs{i,1}; % range of day numbers for current month;
        for j=m,
            if TECe13(j,k),
                TECmN13(:,j,k) = nanmean(TECN13(:,m,k),2);
            end;
        end; clear('m');
    end;
end;

clear('j','k','NDcur','TECcur','i');

save('All_parameters_3.mat');% save all TEC parameters together with HF parameters;
```

A.1.7 Script “Calc_F0F2_TRO_13.m”

Following script extracts the critical frequency F0F2 data from txt data files of the Tromso Dynasonde. The txt files contains values of F0F2 with time resolution of 2 minutes. The script creates the yearly array of F0F2 data (720x365 points) and the arrays of corresponding times. This script must be run only after previous script (“Calc_TEC_CW13”). F0F2 parameters are saved to .mat file: 'All_parameters_4.mat' (together with TEC and HF parameters).

```
% Script "Calc_F0F2_TRO_13" calculates yearly F0F2 parameters (2013 year) from Tromso
% Dynasonde;
% (need indices (m_indxs) for monthly averaging);

load('All_parameters_3.mat'); % after script 'Calc_yearly_TEC_13';
% Set initial parameters:
Year = 2013; % Year;
Nt = 720; % Number of points during day (1day=24*60=1440/2=720 points);
Nd = 365; % Number of days in year;
t_res=2*60; % Time resolutions in sec in data file;

%%
TdayF_TRO_sec = zeros(Nt,1); % array of time of F0F2 data points in sec during day;
TyearF_TRO_dt13 = zeros(Nt,Nd); % array for the time of F0F2 data points in...
                                 % 'MatLab DateTime' during one year;
TyearF_TRO_nd13 = zeros(Nt,Nd); % array for the time of F0F2 data points in...
                                % 'day number' during one year;
TyearF_TRO_y13 = zeros(Nt,Nd);% array for the time of F0F2 data points in 'Year';
TyearF_TRO_m13 = zeros(Nt,Nd);% array for the time of F0F2 data points in...
                                             % 'Months' during one year;
TyearF_TRO_d13 = zeros(Nt,Nd);% array for the time of F0F2 data points in...
                                           % 'Days in month' during one year;

% Set initial F0F2 parameters:
FN13_TRO = zeros(Nt,Nd)*NaN; % array for F0F2 data Tromso;
FmN13_TRO = zeros(Nt,Nd)*NaN; % array for monthly averaged F0F2 data Tromso;
F0F2_line_TRO=zeros((Nd*Nt),1); % line yearly array for F0F2 data;

%% Open and read txt data file:
[name_dat_f0f2,path_f0f2]=uigetfile('*','Open f0f2 data files for Tromso');
fid=fopen([path_f0f2,'\',name_dat_f0f2],'r');
```

```
data = textscan(fid,'%*s %*s %f %f %*[^\n]','commentStyle','%');% array of...
                                  % times and F0F2 data from txt file;
fclose(fid);
clear('fid');
%%
F0F2_line_TRO=data{1,2}(:); % separete F0F2 data and time data;
Dday=(data{1,1}(:));% array of the times from txt file;

% Substitute zeros on NaN:
for i=1:length(F0F2_line_TRO)
    if (F0F2_line_TRO(i))==0
        F0F2_line_TRO(i)=NaN;
    end;
end;
%%
j=0; % initialization index for data which will be change inside loop and...
%       data will be larger then number elements in Dday;
num_elem =0;
NewDday = Dday;
n_point = Nt*Nd; % quantity points should be in F0F2 data in one year;

% Correction F0F2 data (append zeros instead of absence data):
for i=1:length(Dday)-1
    dif=Dday(i+1)-Dday(i); % difference between two neighboring elements;
    j=j+1;
    if dif>0.0018 % if between two neighboring data points is larger time...
        % it is mean that we don't have data between these points;
        num_elem = floor(dif/0.0014); % define quantity of absence points;
        temp_Dday = NewDday((j+1):end); % save time data part from...
                                        % second point to the end into temp;
        temp_f=F0F2_line_TRO((j+1):end); % save data part from second point to...
                                                       % the end into temp;
        for ind_nan =1:num_elem % fill NaN to absence points;
            F0F2_line_TRO(j+ind_nan)=NaN;
            NewDday(j+ind_nan)=NaN;
        end;
        j=j+num_elem; % shift to data after NaN region;
        num_elem =0; % clear quantity of absence points;
        NewDday((j+1):end)=[]; % delete time after new j index;
        F0F2_line_TRO((j+1):end)=[]; % delete elements after new j index;
        F0F2_line_TRO=[F0F2_line_TRO;temp_f]; % recovery array F0F2 data from temp;
        NewDday=[NewDday;temp_Dday];% recovery array time from temp;
    end;
    % Fill NaN the end of year if there are not data points:
    if (i==length(Dday)-1)&&(j+1)<n_point
        nun_end_points_TRO=n_point-j-1; % quantity elements to the end of year;
        end_points = zeros(nun_end_points_TRO,1)*NaN;
        F0F2_line_TRO=[F0F2_line_TRO;end_points];
    end;
end;
%%
% Calculate F0F2 parameters:
for j=1:Nd, % j - day in year;
    for k=1:Nt, % k - point in day;
        TdayF_TRO_sec(k,1) = 60+(t_res)*(k-1); % daily T of each point in sec
        TyearF_TRO_dt13(k,j) = datenum(Year,1,1)+(TdayF_TRO_sec(k,1)+...
                                   86400.*(j-1))/86400.; % yearly T in MatLab time;
        TyearF_TRO_nd13(k,j) = j; % yearly days number;
        [TyearF_TRO_y13(k,j), TyearF_TRO_m13(k,j), TyearF_TRO_d13(k,j)] =...
                                datevec(TyearF_TRO_dt13(k,j)); % yearly [Y M D];
        FN13_TRO(k,j) = F0F2_line_TRO((j-1)*Nt+k); % create F0F2 yearly array...
                                                          % for each day;
    end;
end;

%% Monthly averaging:
for i=1:12, % i - month in year;
    m=m_indxs{i,1}; % range of day numbers for current month;
    for j=m,
        FmN13_TRO(:,j) = nanmean(FN13_TRO(:,m),2);
    end; clear('m');
end;
```

```
clear('t_res','dif','','Dday','num_elem','NewDday',...
     'name_dat_f0f2','path_f0f2','temp_Dday','temp_f','end_points',...
     'j','k','NDcur','TECcur','i');

save('All_parameters_4.mat'); % save F0F2 Tromso parameters together with HF and TEC;
```

A.1.8 Script "Calc_F0F2_LFO_13.m"

Following script extracts the critical frequency F0F2 data from txt data files of the Moscow Digisonde. The txt files contains values of F0F2 with time resolution of 15 minutes. Since the txt file contains a small number of points, then to complete the txt file was corrected manually. The script creates the yearly array of F0F2 data (96x365 points) and the arrays of corresponding times. This script must be run only after previous script ("Calc_F0F2_TRO_13"). The F0F2 parameters are saved to .mat file: 'All_parameters_4.mat' (together with F0F2 Tromso, TEC and HF parameters).

```
% "Calc_F0F2_LFO_13" calculates yearly F0F2 parameters (2013 year) from Moscow
% Digisonde
% But is used name LFO: for example F0F2_LFO;
% (need indecies (m_indxs) for monthly averaging);
% txt data file must contains 96*365=35040 points.

load('m_index.mat'); %load data for month averaging(after script 'Calc_m_index);

% Set initial parameters:
Year = 2013; % Year;
Nt = 96; % Number of points during day (1day=24*60=1440/15=96 points);
Nd = 365; % Number of days in year;
t_res=15*60; % Time resolutions in sec in data file;

TdayF_LFO_sec = zeros(Nt,1); % array of time of F0F2 data points in sec during day;
TyearF_LFO_dt13 = zeros(Nt,Nd); % array for the time of F0F2 data points in...
                                 % 'MatLab DateTime' during one year;
TyearF_LFO_nd13 = zeros(Nt,Nd); % array for the time of F0F2 data points in...
                                % 'day number' during one year;
TyearF_LFO_y13 = zeros(Nt,Nd);% array for the time of F0F2 data points in 'Year';
TyearF_LFO_m13 = zeros(Nt,Nd);% array for the time of F0F2 data points in...
                                              % 'Months' during one year;
TyearF_LFO_d13 = zeros(Nt,Nd);% array for the time of F0F2 data points in...
                                              % 'Days in month' during one year;
% Set initial F0F2 parameters:
FN13_LFO = zeros(Nt,Nd)*NaN; % array for F0F2 data LFO;
FmN13_LFO = zeros(Nt,Nd)*NaN; % array for monthly averaged F0F2 data LFO;
F0F2_line_LFO=zeros((Nd*Nt),1); % line yearly array for F0F2 data;

%% Open and read txt data file:
[name_dat_f0f2,path_f0f2]=uigetfile('*','Open f0f2 data files for Moscow');
fid=fopen([path_f0f2,'\',name_dat_f0f2],'r');
data = textscan(fid,'%*s %f %*s %*s %s %*[^\n]','commentStyle','%'); % array of...
                                                    % F0F2 data from txt file;

fclose(fid);
clear('fid');

%% Fill F0F2_line_LFO array from txt file:
for i=1:length(data{1,2}(:))
    if strcmp((data{1,2}(i)),'---')
        F0F2_line_LFO(i)=NaN; % substitute empty points on NaN;
    else
        F0F2_line_LFO(i)=str2double(data{1,2}(i)); % array of the F0F2 from txt file;
    end
end
clear ('data');
%%
% Calc F0F2 parameters:
for j=1:Nd, % j - day in year;
    for k=1:Nt, % k - point in day;
        TdayF_LFO_sec(k,1) = 60+(t_res)*(k-1); % daily T of each point in sec;
```

```
        TyearF_dt13(k,j) = datenum(Year,1,1)+(TdayF_LFO_sec(k,1)+...
                           86400.*(j-1))/86400.; % yearly T in MatLab time;
        TyearF_nd13(k,j) = j; % yearly days number;
        [TyearF_y13(k,j), TyearF_m13(k,j), TyearF_d13(k,j)] =...
                                 datevec(TyearF_dt13(k,j)); % yearly [Y M D];
        FN13_LFO(k,j) = F0F2_line_LFO((j-1)*Nt+k); % create F0F2 yearly array for...
                                                                     % each day;
    end;
end;

%% Monthly averaging:
for i=1:12, % i - month in year;
    m=m_indxs{i,1}; % range of day numbers for current month;
    for j=m,
        FmN13_LFO(:,j) = nanmean(FN13_LFO(:,m),2);
    end; clear('m');
end;

clear('t_res','name_dat_f0f2','path_f0f2','j','k','i','m');
% Add F0F2 LFO parameters to F0F2 Tromso, HF and TEC parameters:
save('All_parameters_4','-append');
```

A.1.9 Script “Calc_K_13.m”

Following script creates the time arrays for magnetic K and Kp-indices and extracts the magnetic indices data from txt files to the data arrays. Initial txt files has different format, because are taken from different resources. Arrays of times and magnetic indices are saved in individual .mat file: `'K_13.mat'`. In additional the script saves K, Kp, HF, TEC and F0F2 parameters together to mat file: `'All_parameters_5.mat'`. This script must be run only after previous script (“`Calc_F0F2_LFO_13`”).

```
% Script "Calc_K_13" sets yearly K and Kp parameters (2013 year)
% Calculate time parameters for K (LFO and Tromso) and Kp magnetic indices,
% and extracts indices data from txt files to the arrays of local K (LFO and Tromso)
% and planetary Kp-indices;

% Set initial parameters:
Year = 2013; % Year;
Nt = 8; % Number of points during day, time resolution of 3 hour;
Nd = 365; % Number of days in year;

TdayK_sec = zeros(Nt,1); % array of time of K,Kp-indices in sec during day;
TyearK_dt13 = zeros(Nt,Nd); % array for the time of K,Kp-indices in...
                                  % 'MatLab DateTime' during one year;
TyearK_nd13 = zeros(Nt,Nd); % array for the time of K,Kp-indices in...
                                 % 'day number' during one year;
TyearK_y13 = zeros(Nt,Nd);% array for the time of K,Kp-indices in 'Year';
TyearK_m13 = zeros(Nt,Nd);% array for the time of K,Kp-indices in...
                                                 % 'Months' during one year;
TyearK_d13 = zeros(Nt,Nd);% array for the time of K,Kp-indices in...
                                              %'Days in month' during one year;

KLFO13 = zeros(Nt,Nd)*NaN; % array for local magnetic K-indices for LFO;
KTRO13 = zeros(Nt,Nd)*NaN; % array for local magnetic K-indices for Tromso;
Kp13 = zeros(Nt,Nd)*NaN; % array for planetary magnetic Kp-indices;

%% Set time K parameters: (the same for Kp)
for j=1:Nd, % j - day in year;
    for k=1:Nt, % k - point in day;
        TdayK_sec(k,1) = 5400+10800*(k-1); % daily T in sec
        TyearK_dt13(k,j) = datenum(Year,1,1)+(TdayK_sec(k,1)+86400.*...
                              (j-1))/86400.; % yearly T in MatLab datetime;
        TyearK_nd13(k,j) = j; % yearly days number;
        [TyearK_y13(k,j), TyearK_m13(k,j), TyearK_d13(k,j)] =...
                                datevec(TyearK_dt13(k,j)); % yearly [Y M D];
    end;
end;
```

```
%% Open and read txt K data file for LFO:
[name_dat_K_LFO,path_K_LFO]=uigetfile('*','Open magnetic data file for LFO');
fid=fopen([path_K_LFO,'\',name_dat_K_LFO],'r');
K_LFO = textscan(fid,'%f %f %f %f %f %f %f %f %*[^\n]','commentStyle','%');%...
                              % cell array of K-indices from txt file for LFO;
fclose(fid);

for i=1:8, % i - data point in day;
    KLFO13(i,:) = K_LFO{1,i}; % fill rows in array KLFO13;
end;
clear('fid','K_LFO','i');

%% Open and read txt K data file for Tromso:
[name_dat_K_TRO,path_K_TRO]=uigetfile('*','Open magnetic data file for Tromso');
fid=fopen([path_K_TRO,'\',name_dat_K_TRO],'r');
K_TRO = textscan(fid,'%*s %s %s %*[^\n]','commentStyle','#');% cell array of...
                                   % K-indices from txt file for Tromso,
                      % two column of cells by 4 character in string format;
fclose(fid);
%


% Extract K-indices from string cells:
for j=1:Nd, % j - day in year;
    KTRO13(1,j)=str2double(K_TRO{1,1}{j}(1)); % extract: Xxxx xxxx;
    KTRO13(2,j)=str2double(K_TRO{1,1}{j}(2)); % extract: xXxx xxxx;
    KTRO13(3,j)=str2double(K_TRO{1,1}{j}(3)); % extract: xxXx xxxx;
    KTRO13(4,j)=str2double(K_TRO{1,1}{j}(4)); % extract: xxxX xxxx;

    KTRO13(5,j)=str2double(K_TRO{1,2}{j}(1)); % extract: xxxx Xxxx;
    KTRO13(6,j)=str2double(K_TRO{1,2}{j}(2)); % extract: xxxx xXxx;
    KTRO13(7,j)=str2double(K_TRO{1,2}{j}(3)); % extract: xxxx xxXx;
    KTRO13(8,j)=str2double(K_TRO{1,2}{j}(4)); % extract: xxxx xxxX;
end;
clear('fid','K_TRO','i','j');

%% Open and read txt Kp data file:
[name_dat_Kp,path_Kp]=uigetfile('*','Open planetary magnetic index data file');
Kp13=(dlmread([path_Kp,name_dat_Kp]))'; % array of Kp-indices;
%
clear('k');
save('K_13.mat'); % save magnetic indices to individual mat file;
load('All_parameters_4.mat'); % load all data: HF, TEC and F0F2;
save('All_parameters_5.mat'); % save K, Kp, HF, TEC, F0F2 together;
```

A.1.10 Script "Calc_SolarIndices_13.m"

This script extracts Solar Indices (Solar radio flux $F_{10.7}$, Sun spot area, Sun Spot number, X-ray Background Flux) from txt data files to the yearly arrays. The arrays have dimensions 1 point per day during 365 day of year. Also the script creates the time arrays for solar indices. This script must be run only after previous script ("Calc_K_13"). All data (Solar indices together with F0F2 LFO, F0F2 Tromso, HF, TEC, K, Kp parameters) are saved to mat file: 'All_parameters_tot.mat'.

```
% Script "Calc_SolarIndices_13" sets yearly Solar Indexes (SI) parameters (2013 year)

% Set initial parameters:
Year = 2013; % Year;
Nt = 1; % Number of points during day;
Nd = 365; % Number of days in month;

TdaySI_sec = zeros(Nt,1); % array of time of SI in sec during day;
TyearSI_dt13 = zeros(Nt,Nd);% array for the time of SI in...
                                            % 'MatLab DateTime' during one year;
TyearSI_nd13 = zeros(Nt,Nd); % array for the time of SI in...
                                                 % 'day number' during one year;
TyearSI_y13 = zeros(Nt,Nd);% array for the time of SI in 'Year';
TyearSI_m13 = zeros(Nt,Nd);% array for the time of SI in 'Months' during one year;
```

```
TyearSI_d13 = zeros(Nt,Nd);% array for the time of SI in 'Days in month' during year;

SolarIndex2013 = zeros(Nd,14)*NaN; % array for all SI;
% Description for SolarIndex2013 (for columns)
% 01 - Year
% 02 - Month
% 03 - Date
% 04 - F10.7 cm,(*10e-22 W/m^2/Hz);
% 05 - Sunspot Number
% 06 - Sunspot Area
% 07 - POW of XrL background (10e-7 W/m2);

F10p7_13 = zeros(Nt,Nd)*NaN; % array for Solar radio flux F10.7 cm data;
SSA13 = zeros(Nt,Nd)*NaN; % array for Sun Spot Area in 10e-6 of the...
                                                     % solar hemisphere;
SSN13 = zeros(Nt,Nd)*NaN; % array for Sun Spot Numbers;
XrLb13 = zeros(Nt,Nd)*NaN;% array for X-ray Background Flux data,(10^-7 W/m2);

% Set initial Solar Indicies (SI) parameters:
for j=1:Nd,
    for k=1:Nt,
        TdaySI_sec(k,1) = 43200; % daily time in sec during one day;
        TyearSI_dt13(k,j) = datenum(Year,1,1)+(TdaySI_sec(k,1)+...
                        86400.*(j-1))/86400.; % yearly T in MatLab datetime;
        TyearSI_nd13(k,j) = j; % yearly days mumber;
        [TyearSI_y13(k,j), TyearSI_m13(k,j), TyearSI_d13(k,j)] =...
                                datevec(TyearSI_dt13(k,j));% yearly [Y M D];
    end;
end;
%
%% Open and read txt data file:
[name_dat_SI,path_SI]=uigetfile('*','Open Solar data file');
SolarIndex2013=dlmread([path_SI,name_dat_SI]); % array of data SI;

%%
F10p7_13 = (SolarIndex2013(:,4))'; % Solar radio flux F10.7 cm,(*10e-22 W/m^2/Hz);
SSN13 = (SolarIndex2013(:,5))'; % Sun Spot Numbers;
SSA13 = (SolarIndex2013(:,6))'; % Sun Spot Area in 10e-6 of the...
                                                     % solar hemisphere;
XrLb13 = (SolarIndex2013(:,7))'; % X-ray Background Flux,(10^-7 W/m2);

clear('j','k');

% Save Solar Indices and F0F2 LFO, F0F2 Tromso, HF, TEC and K, Kp parameters:
load('All_parameters_5.mat'); % load K, Kp, HF, TEC, F0F2 together;
save('All_parameters_tot.mat'); % save all parameters in total mat file.
```

A.2 MatLab scripts for plotting data

The following scripts are used for plotting different figures according to data from total .mat file: `'All_parameters_tot.mat'`. Below are present the scripts for plotting the intensities and DFS HF signal, the TEC, the local magnetic K and planetary Kp-indices, the solar indices.

A.2.1 Script "Plot_monthly_Int_and_DFS_LFO_TRO.m"

The script `"Plot_monthly_Int_and_DFS_LFO_TRO.m"` plots figures of Intensities and Doppler frequency shift of the HF signal for each month at two receiver stations: LFO and Tromso. These figures are the main visual result of processing data from receivers. The figures allowed to see the diurnal and monthly behavior of HF signal parameters and to compare them with other ionospheric parameters for determining the correlations. One figure contains the Intensities and DFS for both receiver stations for one month. All figures are saved to MatLab figure files and to .png files (a raster graphics file format).

```
% Script "Plot_monthly_Int_and_DFS_LFO_TRO" plots the Intensity (Int) and
% Doppler shift (DFS) for LFO and Tromso for each month

% Set Font parameters:
set(0,'DefaultAxesFontSize',7,'DefaultAxesFontName','Helvetica');
set(0,'DefaultTextFontSize',8,'DefaultTextFontName','Helvetica');
%
load('All_parameters_tot.mat');% load data for plotting figures;
% Calculate linear indices for each month:
for j=1:12, % j - month in year;
    itHFb(j) = sub2ind(size(TyearHF_dt13),1,mind(j,1));% linear index of begin time;
    itHFe(j) = sub2ind(size(TyearHF_dt13),48,mind(j,2)); % linear index of end time;
    for k=1:2, % k - satation number: 1 - LFO, 2 - TRO;
        iHFb(j,k) = sub2ind(size(IenN13),1,mind(j,1),k);% linear index of begin Int;
        iHFe(j,k) = sub2ind(size(IenN13),48,mind(j,2),k);% linear index of end Int;
    end;
end;
%% Plot HF intensity for all months of the year:
for j=1:12, % j - month in year;
 %% Plot intensity for LFO:
    subplot(4,1,1);
h=plot(TyearHF_dt13(itHFb(j):itHFe(j)),10*log10(IenmN13(iHFb(j,1):iHFe(j,1))),'-.',...
        TyearHF_dt13(itHFb(j):itHFe(j)),10*log10(IenN13(iHFb(j,1):iHFe(j,1))),'-');
    ax = gca;
    % Set titles:
    t1=title(sprintf('%04d/%02d LFO, Intensity of 9.996 MHz (dB)',Year,j));
    t1.FontSize = 8; t1.FontWeight = 'normal'; % font for title;
    ylabel('Intensity, dB');
    % Set parameters for lines:
    h(1).Color = [0.5 0.5 0.5]; % averaged line;
    h(2).Color = 'b';
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([10*log10(IenN13(iHFb(j,1):iHFe(j,1))) ...
                                 10*log10(IenmN13(iHFb(j,1):iHFe(j,1)))]);
    YLimMAX = max([10*log10(IenN13(iHFb(j,1):iHFe(j,1))) ...
                                 10*log10(IenmN13(iHFb(j,1):iHFe(j,1)))]);
    % Set grid and Ticks
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = [];% turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
    % Set axes limits
    ax.XLim = [XLimMIN XLimMAX];
    if(~any(isnan([YLimMIN YLimMAX]))),
        ax.YLim = [YLimMIN YLimMAX];
    end;
    ax.Position=[0.13 0.763 0.8 0.19];
 %% Plot intensity for TRO(Tromso):
    subplot(4,1,3);
h=plot(TyearHF_dt13(itHFb(j):itHFe(j)),10*log10(IenmN13(iHFb(j,2):iHFe(j,2))),'-.',...
        TyearHF_dt13(itHFb(j):itHFe(j)),10*log10(IenN13(iHFb(j,2):iHFe(j,2))),'k-');
    ax = gca;
    % Set titles:
    t2=title(sprintf('%04d/%02d Tromso, Intensity of 9.996 MHz (dB)',Year,j));
    t2.FontSize = 8; t2.FontWeight = 'normal'; % font for title;
    %xlabel('Date');
    ylabel('Intensity, dB');
    % Set parameters for lines:
    h(1).Color = [0.5 0.5 0.5]; % averaged line;
    h(2).Color = 'b';
    % Set parameters for axes:
    % Calculate axes limits:
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([10*log10(IenN13(iHFb(j,2):iHFe(j,2))) ...
                                 10*log10(IenmN13(iHFb(j,2):iHFe(j,2)))]);
    YLimMAX = max([10*log10(IenN13(iHFb(j,2):iHFe(j,2))) ...
                                 10*log10(IenmN13(iHFb(j,2):iHFe(j,2)))]);
```

```
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on';  ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = [];% turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
    if(~any(isnan([YLimMIN YLimMAX]))),
        ax.YLim = [YLimMIN YLimMAX];
    end;
    ax.Position=[0.13 0.299 0.8 0.19];

%% Plot Doppler for LFO:
    subplot(4,1,2);
    h = plot(TyearHF_dt13(itHFb(j):itHFe(j)),FmomN13(iHFb(j,1):iHFe(j,1)),'-.',...
             TyearHF_dt13(itHFb(j):itHFe(j)),FmoN13(iHFb(j,1):iHFe(j,1)),'-');
    ax = gca;
    % Set titles:
    t3=title(sprintf('%04d/%02d LFO, Doppler frequency shift (Hz)',Year,j));
    t3.FontSize = 8;  t3.FontWeight = 'normal'; % font for title;
    %xlabel('Date');
    ylabel('Frequency, Hz');
    % Set parameters for lines
    h(1).Color = 'k'; % or gray color [0.5 0.5 0.5], averaged line;
    h(2).Color = 'b';
    % Set parameters for axes:
    % Calculate axes limits:
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([FmoN13(iHFb(j,1):iHFe(j,1)) FmomN13(iHFb(j,1):iHFe(j,1))]);
    YLimMAX = max([FmoN13(iHFb(j,1):iHFe(j,1)) FmomN13(iHFb(j,1):iHFe(j,1))]);
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = [];% turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
    % Set axes limits
    ax.XLim = [XLimMIN XLimMAX];
    if(~any(isnan([YLimMIN YLimMAX]))),
        ax.YLim = [YLimMIN YLimMAX];
    end;
    ax.Position=[0.13 0.531 0.8 0.19];

%% Plot Doppler for Tromso:
    subplot(4,1,4);
    h = plot(TyearHF_dt13(itHFb(j):itHFe(j)),FmomN13(iHFb(j,2):iHFe(j,2)),'-.',...
             TyearHF_dt13(itHFb(j):itHFe(j)),FmoN13(iHFb(j,2):iHFe(j,2)),'-');
    ax = gca;
    % Set titles
    t4=title(sprintf('%04d/%02d Tromso, Doppler frequency shift (Hz)',Year,j));
    t4.FontSize = 8; t4.FontWeight = 'normal'; % font for title;
    xlabel('Date');
    ylabel('Frequency, Hz');
    % Set parameters for lines
    h(1).Color = 'k';% or gray color [0.5 0.5 0.5], averaged line;
    h(2).Color = 'b';
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([FmoN13(iHFb(j,2):iHFe(j,2)) FmomN13(iHFb(j,1):iHFe(j,2))]);
    YLimMAX = max([FmoN13(iHFb(j,2):iHFe(j,2)) FmomN13(iHFb(j,1):iHFe(j,2))]);
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
```

```
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
    if(~any(isnan([YLimMIN YLimMAX]))),
        ax.YLim = [YLimMIN YLimMAX];
    end;
    ax.Position=[0.13 0.066 0.8 0.19];

    % Save file to fig and png files in current MatLab folder:
    sFigFileName = sprintf('2013%02d_hf_Intens&Doppler',j);
    fig = gcf;
    saveas(fig,sFigFileName,'fig'); % save figures in MatLab format;
    fig.PaperUnits = 'centimeters';
    fig.PaperPosition = [0 0 46.5 30];
    fig.PaperPositionMode = 'manual';
    print(sFigFileName,'-dpng','-r0'); % save figures in png format;
end;
```

A.2.2 Script "Plot_monthly_TEC_F0F2.m"

This script plots figures of the TEC for LFO, Tromso, Kiruna, Joensuu and the F0F2 for LFO and Tromso in different combinations according to the selection dialog in the Command Window of MatLab. The figures are plotted for each month during 2013 year. All figures are saved to MatLab figure files and to .png files. These figures allowed to compare different ionospheric parameters during each month with each other and with HF parameters from the previous script.

```
% Script "Plot_monthly_TEC_F0F2" plots TEC LFO, TEC Tromso and Kiruna,
% F0F2 LFO and Tromso on one figure in different combinations according to
% selection dialog.
%
load('All_parameters_tot'); % load data for plotting figures;

% Set Font parameters:
set(0,'DefaultAxesFontSize',7,'DefaultAxesFontName','Helvetica');
set(0,'DefaultTextFontSize',8,'DefaultTextFontName','Helvetica');

% Calculate linear indices for each month:
for j=1:12, % j - month in year;
    itTECb(j) = sub2ind(size(TyearTEC_dt13),1,mind(j,1)); % linear index of...
                                                    % begin time TEC in month;
    itTECe(j) = sub2ind(size(TyearTEC_dt13),288,mind(j,2));% linear index of...
                                                       % end time TEC in month;
    itF_LFOb(j) = sub2ind(size(TyearF_LFO_dt13),1,mind(j,1));% linear index of...
                                                % begin time F0F2_LFO in month;
    itF_LFOe(j) = sub2ind(size(TyearF_LFO_dt13),96,mind(j,2));% linear index of...
                                                % end time F0F2_LFO in month;
    itF_TROb(j) = sub2ind(size(TyearF_TRO_dt13),1,mind(j,1));% linear index of...
                                                % begin time F0F2_TRO in month;
    itF_TROe(j) = sub2ind(size(TyearF_TRO_dt13),720,mind(j,2)); % linear index of...
                                                % end time F0F2_TRO in month;

    for k=1:4, % k - station number: 1 - LFO, 2 - Kiruna, 3-JOEN, 4 - Tromso;
        iTECb(j,k) = sub2ind(size(TECN13),1,mind(j,1),k);% linear index of begin TEC;
        iTECe(j,k) = sub2ind(size(TECN13),288,mind(j,2),k);% linear index of end TEC;
    end;
end;

% Set initial empty values for parts of name for saving figure:
save_name1=[]; save_name2=[]; save_name3=[]; save_name4=[]; save_name5=[];

%% Dialog of selection parameters for figure:
txt1 = ('Input number for 1st, 2nd, 3d parameters\nwhich you want see on figure:\n');
txt2 = ('1-TEC LFO\n2-TEC TRO+Kiruna\n3-TEC JOEN\n4-F0F2 TRO\n5-F0F2 LFO\n');
txt3 = ('the 1st prameter:\n');
config1=input([txt1,txt2,txt3]);
config2=input('the 2nd parameters on figure:\n');
config3=input('the 3d parameters on figure:\n');
clear('txt1','txt2','txt3');
```

```matlab
%% Plot TEC for all months of the year

for j=1:12, % j - month in year;
    %% Plot TEC for LFO:
    % Check position for TEC LFO:
    if (config1==1) || (config2==1) || (config3==1),
        if config1==1,
            subplot(3,1,1); % 1st position on figure;
        elseif config2==1,
            subplot(3,1,2); % 2nd position on figure;
        elseif config3==1,
            subplot(3,1,3); % 3d position on figure;
        end;
 h = plot(TyearTEC_dt13(itTECb(j):itTECe(j)),TECmN13(iTECb(j,1):iTECe(j,1)),'-.',...
             TyearTEC_dt13(itTECb(j):itTECe(j)),TECN13(iTECb(j,1):iTECe(j,1)),'-');
    ax = gca;
    % Set titles:
    t1=title(sprintf('%04d/%02d LFO, TEC (TECU)',Year,j));
    t1.FontSize = 8; t1.FontWeight = 'normal';% font for title;
    save_name1 = ('TEC_LFO');% part of name for figure saving;
    xlabel('Date');
    ylabel('TEC, TECU');
    % Set parameters for lines:
    h(1).Color = [0.5 0.5 0.5]; % averaged line;
    h(2).Color = 'b';
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([TECN13(iTECb(j,1):iTECe(j,1)) TECmN13(iTECb(j,1):iTECe(j,1))]);
    YLimMAX = max([TECN13(iTECb(j,1):iTECe(j,1)) TECmN13(iTECb(j,1):iTECe(j,1))]);
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
        if(~any(isnan([YLimMIN YLimMAX]))),
            ax.YLim = [YLimMIN YLimMAX];
        end;

        % Exact position of subplot:
        if config1==1
            ax.Position=[0.13 0.714 0.725 0.21]; % 1st position;
        elseif config2==1
            ax.Position=[0.13 0.431 0.725 0.21]; % 2nd position;
        elseif config3==1
            ax.Position=[0.13 0.15 0.725 0.21]; % 3d position;
        end;
    end; % the end of check position TEC LFO;

    %% Plot TEC for Tromso+Kiruna:
    % Check position for TEC Kiruna+Tromso:
    if (config1==2) || (config2==2) || (config3==2),
        if config1==2,
            subplot(3,1,1); % 1st position on figure;
        elseif config2==2,
            subplot(3,1,2); % 2nd position on figure;
        elseif config3==2,
            subplot(3,1,3); % 3d position on figure;
        end;
  h = plot(TyearTEC_dt13(itTECb(j):itTECe(j)),TECmN13(iTECb(j,2):iTECe(j,2)),'-.',...
          TyearTEC_dt13(itTECb(j):itTECe(j)),TECN13(iTECb(j,4):iTECe(j,4)),'-',...
          TyearTEC_dt13(itTECb(j):itTECe(j)),TECN13(iTECb(j,2):iTECe(j,2)),'-');
    ax = gca;
    % Set titles:
    t2=title(sprintf('%04d/%02d Kiruna (red) and Tromso (blue), TEC (TECU)',Year,j));
    t2.FontSize = 8; t2.FontWeight = 'normal';% font for title;
    save_name2 = ('TEC_TRO');% part of name for figure saving;
    xlabel('Date');
    ylabel('TEC, TECU');
```

```
  % Set parameters for lines:
  h(1).Color = [0.5 0.5 0.5]; % averaged line;
  h(2).Color = 'b'; % line TEC for Tromso;
  h(3).Color = 'r'; % line TEC for Kiruna;
  % Set parameters for axes:
  % Calculate axes limits
  XLimMIN = datenum(Year,1,mind(j,1));
  XLimMAX = datenum(Year,1,mind(j,2))+1;
  YLimMIN = min([TECN13(iTECb(j,2):iTECe(j,2)) TECN13(iTECb(j,4):iTECe(j,4))...
                                        TECmN13(iTECb(j,2):iTECe(j,2))]);
  YLimMAX = max([TECN13(iTECb(j,2):iTECe(j,2)) TECN13(iTECb(j,4):iTECe(j,4)) ...
                                        TECmN13(iTECb(j,2):iTECe(j,2))]);
  % Set grid and Ticks:
  grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
  ax.MinorGridLineStyle = ':';
  NumTicks = round(XLimMAX-XLimMIN)+1;
  ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
  datetick('x','dd','keepticks');
  % Set axes limits
  ax.XLim = [XLimMIN XLimMAX];
      if(~any(isnan([YLimMIN YLimMAX]))),
          ax.YLim = [YLimMIN YLimMAX];
      end;
      % Exact position of subplot:
      if config1==2
          ax.Position=[0.13 0.714 0.725 0.21]; % 1st position;
      elseif config2==2
          ax.Position=[0.13 0.431 0.725 0.21]; % 2nd position;
      elseif config3==2
          ax.Position=[0.13 0.15 0.725 0.21]; % 3d position;
      end;
  end; % the end of check position TEC Kiruna+Tromso;

  %% Plot TEC for Joensuu (mid-point)
  % Check position for TEC Joensuu:
  if (config1==3) || (config2==3) || (config3==3),
      if config1==3,
          subplot(3,1,1); % 1st position on figure;
      elseif config2==3,
          subplot(3,1,2); % 2nd position on figure;
      elseif config3==3,
          subplot(3,1,3); % 3d position on figure;
      end;
h = plot(TyearTEC_dt13(itTECb(j):itTECe(j)),TECmN13(iTECb(j,3):iTECe(j,3)),'-',...
         TyearTEC_dt13(itTECb(j):itTECe(j)),TECN13(iTECb(j,3):iTECe(j,3)),'-');
  ax = gca;
  % Set titles:
  t3=title(sprintf('%04d/%02d Joensuu (mid point), TEC (TECU)',Year,j));
  t3.FontSize = 8; t3.FontWeight = 'normal';% font for title;
  save_name3 = ('TEC_JOEN');% part of name for figure saving;
  xlabel('Date');
  ylabel('TEC, TECU');
  % Set parameters for lines:
  h(1).Color = [0.5 0.5 0.5]; % averaged line;
  h(2).Color = 'r';
  % Set parameters for axes:
  % Calculate axes limits:
  XLimMIN = datenum(Year,1,mind(j,1));
  XLimMAX = datenum(Year,1,mind(j,2))+1;
  YLimMIN = min([TECN13(iTECb(j,3):iTECe(j,3)) TECmN13(iTECb(j,3):iTECe(j,3))]);
  YLimMAX = max([TECN13(iTECb(j,3):iTECe(j,3)) TECmN13(iTECb(j,3):iTECe(j,3))]);
  % Set grid and Ticks:
  grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
  ax.MinorGridLineStyle = ':';
  NumTicks = round(XLimMAX-XLimMIN)+1;
  ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
  datetick('x','dd','keepticks');
  % Set axes limits:
  ax.XLim = [XLimMIN XLimMAX];
      if(~any(isnan([YLimMIN YLimMAX]))),
          ax.YLim = [YLimMIN YLimMAX];
      end;
```

```
        % Exact position of subplot:
        if config1==3
            ax.Position=[0.13 0.714 0.725 0.21]; % 1st position;
        elseif config2==3
            ax.Position=[0.13 0.431 0.725 0.21]; % 2nd position;
        elseif config3==3
            ax.Position=[0.13 0.15 0.725 0.21]; % 3d position;
        end;
    end; % the end of check position TEC Joensuu;

    %% Plot F0F2 for Tromso:
    % Check position for F0F2 Tromso:
    if (config1==4) || (config2==4) || (config3==4),
        if config1==4,
            subplot(3,1,1); % 1st position on figure;
        elseif config2==4,
            subplot(3,1,2); % 2nd position on figure;
        elseif config3==4,
            subplot(3,1,3); % 3d position on figure;
        end;
    h=plot(TyearF_TRO_dt13(itF_TROb(j):itF_TROe(j)),...
               FmN13_TRO(itF_TROb(j):itF_TROe(j)),'-.',...
               TyearF_TRO_dt13(itF_TROb(j):itF_TROe(j)),...
               FN13_TRO(itF_TROb(j):itF_TROe(j)),'-');
    ax = gca;
    % Set titles:
    t4=title(sprintf('%04d/%02d Tromso, F0F2 (MHz)',Year,j));
    t4.FontSize = 8; t4.FontWeight = 'normal'; % font for title;
    save_name4 = ('F0F2_TRO');% part of name for figure saving;
    xlabel('Date');
    ylabel('F0F2, MHz');
    % Set parameters for lines:
    h(1).Color = [0.5 0.5 0.5]; % averaged line;
    h(2).Color = 'b';
    % Set parameters for axes:
    % Calculate axes limits:
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([FN13_TRO(itF_TROb(j):itF_TROe(j))...
                                  FmN13_TRO(itF_TROb(j):itF_TROe(j))]);
    YLimMAX = max([FN13_TRO(itF_TROb(j):itF_TROe(j))...
                                  FmN13_TRO(itF_TROb(j):itF_TROe(j))]);
    % Set grid and Ticks:
    grid on;  ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
        if(~any(isnan([YLimMIN YLimMAX]))),
            ax.YLim = [YLimMIN YLimMAX];
        end;
        % Exact position of subplot:
        if config1==4
            ax.Position=[0.13 0.714 0.725 0.21]; % 1st position;
        elseif config2==4
            ax.Position=[0.13 0.431 0.725 0.21]; % 2nd position;
        elseif config3==4
            ax.Position=[0.13 0.15 0.725 0.21]; % 3d position;
        end;
    end; % the end of check position F0F2 Tromso;

    %% Plot F0F2 for LFO:
    % Check position for F0F2 LFO:
    if (config1==5) || (config2==5) || (config3==5),
        if config1==5,
            subplot(3,1,1); % 1st position on figure;
        elseif config2==5,
            subplot(3,1,2); % 2nd position on figure;
        elseif config3==5,
            subplot(3,1,3); % 3d position on figure;
        end;
```

```
    h = plot(TyearF_LFO_dt13(itF_LFOb(j):itF_LFOe(j)),...
              FmN13_LFO(itF_LFOb(j):itF_LFOe(j)),'-',...
              TyearF_LFO_dt13(itF_LFOb(j):itF_LFOe(j)),...
                     FN13_LFO(itF_LFOb(j):itF_LFOe(j)),'-');
    ax = gca;
    % Set titles:
    t5=title(sprintf('%04d/%02d LFO(Mosc), F0F2 (MHz)',Year,j));
    t5.FontSize = 8; t5.FontWeight = 'normal'; % font for title;
    save_name5 = ('F0F2_LFO'); % part of name for figure saving;
    xlabel('Date');
    ylabel('F0F2, MHz');
    % Set parameters for lines:
    h(1).Color = [0.5 0.5 0.5]; % averaged line;
    h(2).Color = 'r';
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([FN13_LFO(itF_LFOb(j):itF_LFOe(j))...
                                 FmN13_LFO(itF_LFOb(j):itF_LFOe(j))]);
    YLimMAX = max([FN13_LFO(itF_LFOb(j):itF_LFOe(j))...
                                 FmN13_LFO(itF_LFOb(j):itF_LFOe(j))]);
    % Set grid and Ticks:
    grid on;  ax.XMinorGrid = 'on';  ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
        if(~any(isnan([YLimMIN YLimMAX]))),
            ax.YLim = [YLimMIN YLimMAX];
        end;
        % Exact position of subplot:
        if config1==5
            ax.Position=[0.13 0.714 0.725 0.21]; % 1st position;
        elseif config2==5
            ax.Position=[0.13 0.431 0.725 0.21]; % 2nd position;
        elseif config3==5
            ax.Position=[0.13 0.15 0.725 0.21]; % 3d position;
        end;
    end; % the end of check position F0F2 LFO;

    %% Save file to fig and png files:
    txt_filename=[save_name1,'&',save_name2,'&',save_name3,'&',...
                                                save_name4,'&',save_name5];
    sFigFileName = sprintf('2013_%02dm_%s',j,txt_filename);
    fig = gcf;
    saveas(fig,sFigFileName,'fig'); % save figures in MatLab format;
    fig.PaperUnits = 'centimeters';  fig.PaperPosition = [0 0 47 25];
    fig.PaperPositionMode = 'manual';
    print(sFigFileName,'-dpng','-r0'); %save figures in png format;

end;
```

A.2.3 Script "Plot_monthly_K.m"

This script is used for plotting the figures of the local magnetic K-indices for LFO and Tromso and the planetary Kp-indices. The figures are plotted for each month during 2013 year. All figures are saved to MatLab figure files and to .png files. These figures helped us to determinate the large geomagnetic disturbances.

```
% Script "Plot_monthly_K" plots monthly data for magnetic K and Kp-indices

% Set Font parameters:
set(0,'DefaultAxesFontSize',7,'DefaultAxesFontName','Helvetica');
set(0,'DefaultTextFontSize',8,'DefaultTextFontName','Helvetica');
%
load('All_parameters_tot.mat'); % load data for plotting figures;
```

```
% Calculate linear indices for each month:
for j=1:12, % j - month in year;
    iKb(j) = sub2ind(size(TyearK_dt13),1,mind(j,1)); % linear index of...
                                                    % begin K-index in month;
    iKe(j) = sub2ind(size(TyearK_dt13),8,mind(j,2)); % linear index of...
                                                     % end K-index in month;
end;

% Plot K for each month of the year:
for j=1:12, % j - month in year;
    % Plot data for LFO:
    subplot(3,1,1);
    h = bar(TyearK_dt13(iKb(j):iKe(j)),KLFO13(iKb(j):iKe(j)),'b');
    ax = gca;
    % Set titles:
    t1=title(sprintf('%04d/%02d K-index for LFO',Year,j));
    t1.FontSize = 8; t1.FontWeight = 'normal'; % font for title;
    xlabel('Date');
    ylabel('K, 1..9');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = 0; YLimMAX = 9;
    % Set grid and Ticks:
    grid on;  ax.XMinorGrid = 'on';  ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];

    % Plot data for Tromso:
    subplot(3,1,2);
    h = bar(TyearK_dt13(iKb(j):iKe(j)),KTRO13(iKb(j):iKe(j)),'r');
    ax = gca;
    % Set titles
    t2=title(sprintf('%04d/%02d K-index for Tromso',Year,j));
    t2.FontSize = 8; t2.FontWeight = 'normal'; % font for title;
    xlabel('Date');
    ylabel('K, 1..9');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = 0; YLimMAX = 9;
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];

    % Plot planetary Kp-index:
    subplot(3,1,3);
    h = bar(TyearK_dt13(iKb(j):iKe(j)),Kp13(iKb(j):iKe(j)),'k');
    ax = gca;
    % Set titles:
    t2=title(sprintf('%04d/%02d Kp-index',Year,j));
    t2.FontSize = 8; t2.FontWeight = 'normal'; % font for title;
    xlabel('Date');
    ylabel('K, 1..9');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = 0; YLimMAX = 9;
```

```
% Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];

    % Save file to fig and png files:
    sFigFileName = sprintf('2013%02d_K',j);
    fig = gcf;
    saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
    fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 47 25];
    fig.PaperPositionMode = 'manual';
    print(sFigFileName,'-dpng','-r0')% save figures in png format;
end;
```

A.2.4 Script "Plot_monthly_K_SI.m"

This script plots K-indices together with Solar Indices parameters ($F_{10.7}$, Sun spot number, X-ray Background Flux) on one figure for each month in year. All figures are saved to MatLab figure files and to .png files. The figures allowed to compare magnetic and solar indices and to see which parameter has greater influence on the behavior HF and ionospheric parameters.

```
% Script "Plot_monthly_K_SI" plots K-indices for LFO and TRO, F10.7, SSN,
% X-ray Background Flux together on the one figure.

% Set Font parameters:
set(0,'DefaultAxesFontSize',7,'DefaultAxesFontName','Helvetica');
set(0,'DefaultTextFontSize',8,'DefaultTextFontName','Helvetica');
load('All_parameters_tot.mat'); % load data for plotting figures;

% Calculate linear indices K for each month:
for j=1:12,
    iKb(j) = sub2ind(size(TyearK_dt13),1,mind(j,1)); % linear index of...
                                                     % begin K in month;
    iKe(j) = sub2ind(size(TyearK_dt13),8,mind(j,2)); % linear index of...
                                                     % end K in month;
    iSIb(j) = sub2ind(size(TyearSI_dt13),1,mind(j,1)); % linear index of...
                                                     % begin SI in month;
    iSIe(j) = sub2ind(size(TyearSI_dt13),1,mind(j,2)); % linear index of...
                                                     % end SI in month;
end;

% Plot K-indices, F10.7, SSN, X-ray for each month:
for j=1:12, % j - month in eyar;
 %% Plot K data for LFO:
    subplot(5,1,1);
    h = bar(TyearK_dt13(iKb(j):iKe(j)),KLFO13(iKb(j):iKe(j)),'b');
    ax = gca;
    % Set titles:
    t1=title(sprintf('%04d/%02d K-index for LFO',Year,j));
    t1.FontSize = 8; t1.FontWeight = 'normal'; % font for title;
    %xlabel('Date');
    ylabel('K, 1..9');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = 0; YLimMAX = 9;
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = []; % turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
```

```
    % Set axes limits
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];
    ax.Position=[0.13 0.801 0.8 0.12];

 %% Plot K data for Tromso:
    subplot(5,1,2);
    h = bar(TyearK_dt13(iKb(j):iKe(j)),KTRO13(iKb(j):iKe(j)),'r');
    ax = gca;
    % Set titles:
    t2=title(sprintf('%04d/%02d K-index for Tromso',Year,j));
    t2.FontSize = 8; t2.FontWeight = 'normal'; % font for title;
    %xlabel('Date');
    ylabel('K, 1..9');
    % Set parameters for axes
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = 0; YLimMAX = 9;
    % Set grid and Ticks
    grid on; ax.XMinorGrid = 'on';  ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = []; % turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
    % Set axes limits
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];
    ax.Position=[0.13 0.651 0.8 0.12];

%% Plot data for Solar radio flux F 10.7 cm (*10e-22 W/m^2/Hz);
    subplot(5,1,3);
    FLimMIN = 60; FLimMAX = 160;
    YLimMIN = min(F10p7_13(iSIb(j):iSIe(j)));
    YLimMAX = max(F10p7_13(iSIb(j):iSIe(j)));
    if (YLimMIN>FLimMIN) && (YLimMAX<FLimMAX),
        bar(TyearSI_dt13(iSIb(j):iSIe(j)),F10p7_13(iSIb(j):iSIe(j)),'b');
    else % when SSN changes are more than 160 units during month;
        bar(TyearSI_dt13(iSIb(j):iSIe(j)),F10p7_13(iSIb(j):iSIe(j)),'r');
    end;
    ax = gca;
    % Set titles:
    t3=title(sprintf('%04d/%02d Index F_1_0_._7',Year,j));
    t3.FontSize = 8; t3.FontWeight = 'normal'; % font for title;
    %xlabel('Date');
    ylabel('Index F_1_0_._7');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([YLimMIN FLimMIN]);
    YLimMAX = max([YLimMAX FLimMAX]);
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = []; % turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
    % Set axes limits
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];
    ax.Position=[0.13 0.493 0.8 0.12];

 %% Plot data for Sun Spot Number
    subplot(5,1,4);
    SLimMIN = 20; SLimMAX = 180;
    YLimMIN = min(SSN13(iSIb(j):iSIe(j)));
    YLimMAX = max(SSN13(iSIb(j):iSIe(j)));
    if (YLimMIN>SLimMIN) && (YLimMAX<SLimMAX),
        bar(TyearSI_dt13(iSIb(j):iSIe(j)),SSN13(iSIb(j):iSIe(j)),'b');
    else
```

```
        bar(TyearSI_dt13(iSIb(j):iSIe(j)),SSN13(iSIb(j):iSIe(j)),'r');
    end;
    ax = gca;
    % Set titles:
    t4=title(sprintf('%04d/%02d Sun Spot Number',Year,j));
    t4.FontSize = 8; t4.FontWeight = 'normal'; % font for title;
    %xlabel('Date');
    ylabel('SSN');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = min([YLimMIN SLimMIN]);
    YLimMAX = max([YLimMAX SLimMAX]);
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on'; ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    ax.XTickLabel = []; % turn off the tick Label of X axis;
    %datetick('x','dd','keepticks');
    % Set axes limits:
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];
    ax.Position=[0.13 0.343 0.8 0.12];

%% Plot data for X-Ray:
    subplot(5,1,5);
    YLimMAX = max(XrLb13(iSIb(j):iSIe(j)));
    bar(TyearSI_dt13(iSIb(j):iSIe(j)),XrLb13(iSIb(j):iSIe(j)),'r');
    ax = gca;
    % Set titles:
    t5=title(sprintf('%04d/%02d X-ray 1-8A (*10^-^7 W/m^2)',Year,j));
    t5.FontSize = 8; t5.FontWeight = 'normal'; % font for title;
    xlabel('Date');
    ylabel('X-ray 1-8A, *10^-^7 W/m^2');
    % Set parameters for axes:
    % Calculate axes limits
    XLimMIN = datenum(Year,1,mind(j,1));
    XLimMAX = datenum(Year,1,mind(j,2))+1;
    YLimMIN = 1e-7;
    % Set grid and Ticks:
    grid on; ax.XMinorGrid = 'on';  ax.YMinorGrid = 'on';
    ax.MinorGridLineStyle = ':';
    NumTicks = round(XLimMAX-XLimMIN)+1;
    ax.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
    %ax.XTickLabel = []; % turn off the tick Label of X axis;
    datetick('x','dd','keepticks');
    % Set axes limits
    ax.XLim = [XLimMIN XLimMAX];
    ax.YLim = [YLimMIN YLimMAX];
    ax.Position=[0.13 0.188 0.8 0.12];

 %% Save file to fig and png files:
    sFigFileName = sprintf('2013%02d_K_F107_SSN_XRay',j);
    fig = gcf;
    saveas(fig,sFigFileName,'fig'); % save figures in MatLab format;
    fig.PaperUnits = 'centimeters';
    fig.PaperPosition = [0 0 32 30];
    fig.PaperPositionMode = 'manual';
    print(sFigFileName,'-dpng','-r0'); % save figures in png format;
end;
```

A.2.5 Script "Plot_yearly_All_param.m"

Script "`Plot_yearly_All_param.m`" plots early distributions of all parameters: the intensity and DFS HF signal for LFO and Tromso, TEC in Kiruna, Joensuu, LFO, the Solar indices $F_{10.7}$ and Sun spot number, the magnetic K-indices for Tromso. All figures are plotted in pseudocolors (except Solar indices). All figures are saved to MatLab figure files and to .png files. These figures help visually estimate the yearly behavior different parameters during 2013 year.

```
% Script "Plot_yearly_All_param" plots yearly distributions Intensity and DFS
% for LFO and Tromso, TEC for LFO, Joensuu and Kiruna, F0F2 for Tromso and LFO;
% F 10.7 and San Spot Number, K-indices for Tromso.

% Set Font parameters:
set(0,'DefaultAxesFontSize',10,'DefaultAxesFontName','Helvetica');
set(0,'DefaultTextFontSize',11,'DefaultTextFontName','Helvetica');
load('All_parameters_tot.mat'); % load data for plotting figures;
% Set parameters for HF distributions:
Days365 = linspace(1,365,365); % day numbers for x axis;
tIen = linspace(0,24,48);% y values of times for plot HF parameters (Int, DFS);
xTickArr(1,1:12) = mind(:,1); % Tick for x axis;
xTickArr(1,13) = mind(12,2);
yTickArr(1:13) = linspace(0,24,13); % Tick for y axis;
% Set parameters for TEC distributions:
tTEC = linspace(0,24,288); % y values of times for plot TEC;
% Set parameters for F0F2 distributions:
tF_TRO = linspace(0,24,720); % y values of times for plot F0F2 Tromso;
tF_LFO = linspace(0,24,96); % y values of times for plot F0F2 LFO;
% Set parameters for K distributions:
tK = linspace(0,24,8); % y values of times for plot K-indices;

%% Plot distribution for HF intensity LFO
IenN13_LFO = IenN13(:,:,1);
hp = pcolor(Days365,tIen,10*log10(IenN13_LFO));
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([20 80]);
t1=title('2013 HF intensity for LFO, dB');
t1.FontSize = 11; t1.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'Intensity, dB'; hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_LFO_IntHF_dB';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for HF intensity Tromso:
IenN13_TRO = IenN13(:,:,2);
hp = pcolor(Days365,tIen,10*log10(IenN13_TRO));
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([30 60]);
t2=title('2013 HF intensity for Tromso, dB');
t2.FontSize = 11; t2.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'Intensity, dB';hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files
sFigFileName = '2013_TRO_IntHF_dB';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for HF Doppler frequency LFO:
FmoN13_LFO = FmoN13(:,:,1);
hp = pcolor(Days365,tIen,FmoN13_LFO);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([-0.8 1.2]);
t3=title('2013 HF doppler frequency for LFO, Hz');
t3.FontSize = 11; t3.FontWeight = 'normal'; % font for title;
```

```
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb=colorbar('horiz');hb.Label.String='Doppler frequency, Hz';hb.FontSize=10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files
sFigFileName = '2013_LFO_DFS_HF_Hz';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for HF Doppler frequency Tromso:
FmoN13_TRO = FmoN13(:,:,2);
hp = pcolor(Days365,tIen,FmoN13_TRO);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([-0.6 0.7]);
t4=title('2013 HF doppler frequency for Tromso, Hz');
t4.FontSize = 11; t4.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb=colorbar('horiz');hb.Label.String ='Doppler frequency, Hz';hb.FontSize=10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_TRO_DFS_HF_Hz';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for TEC LFO:
TECN13_LFO = TECN13(:,:,1);
hp = pcolor(Days365,tTEC,TECN13_LFO);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([0 40]);
t5=title('2013 TEC for LFO, TECU');
t5.FontSize = 11; t5.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'TEC, TECU';
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_LFO_TEC_TECU';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for TEC Joensuu (mid-point):
TECN13_Joen = TECN13(:,:,3);
hp = pcolor(Days365,tTEC,TECN13_Joen);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr;
caxis([0 35]);
t6=title('2013 TEC for Joen, TECU');
t6.FontSize = 11; t6.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb=colorbar('horiz'); hb.Label.String='TEC, TECU'; hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_JOEN_TEC_TECU';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;
```

```
%% Plot distribution for TEC Kiruna:
TECN13_Kir = TECN13(:,:,2);
hp = pcolor(Days365,tTEC,TECN13_Kir);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([0 40]);
t7=title('2013 TEC for Kiruna, TECU');
t7.FontSize = 11; t7.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'TEC, TECU'; hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_KIR_TEC_TECU';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 80 45];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for F0F2 Tromso:

hp = pcolor(Days365,tF_TRO,FN13_TRO);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; %set ticks for axes;
caxis([1.5 11]);
t8=title('2013 F0F2 for Tromso, MHz');
t8.FontSize = 11; t8.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'F0F2, MHz'; hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_TRO_F0F2_MHz';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 80 45];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for F0F2 LFO:

hp = pcolor(Days365,tF_LFO,FN13_LFO);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr;
caxis([1.5 11]);
t9=title('2013 F0F2 for LFO, MHz');
t9.FontSize = 8; t9.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'F0F2, MHz'; hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files
sFigFileName = '2013_LFO_F0F2_MHz';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for F 10.7 and Sun Spot Number:
[ax,h1,h2]=plotyy(Days365,F10p7_13,Days365,SSN13);
 grid on;
 h1.LineWidth = 1;  h2.LineWidth = 1;
 h1.Color = 'b';    h2.Color = 'r';
 h1.LineStyle = '-.';
 ylabel(ax(1),'F_1_0_._7') % left y-axis
 ylabel(ax(2),'SSN') % right y-axis
 ax(1).XLim=[1 365];  ax(2).XLim=[1 365];
 t10=title('F_1_0_._7 index and Sun spot number');
 t10.FontSize = 11; t10.FontWeight = 'normal'; % font for title;
 xlabel('Day of the year');
 l1=legend('F 10.7','SSN');
```

```
 l1.Position =[0.243 0.462 0.07 0.084]; l1.FontSize = 10; % set legend;
 ax(1).Position =[0.13 0.11 0.73 0.5]; ax(2).Position =[0.13 0.11 0.73 0.5];
sFigFileName = '2013_F107&SSN';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 10];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;

%% Plot distribution for K-index Tromso:
load('Mycolormap.mat');% load .mat file with a more contrast palette;
colormap(Mymap); % set a more contrast palette for figure;
hp = pcolor(Days365,tK,KTRO13);
ax = gca; ax.XTick = xTickArr; ax.YTick = yTickArr; % set ticks for axes;
caxis([0 8]);
t11=title('2013 K-index for Tromso');
t11.FontSize = 11; t11.FontWeight = 'normal'; % font for title;
xlabel('Days of the year'); ylabel('Time UT, Hours');
shading flat;
hb = colorbar('horiz'); hb.Label.String = 'K-index'; hb.FontSize = 10;
ax.Position=[0.13 0.2 0.775 0.6595]; % figure position;
hb.Position=[0.13 0.0844 0.775 0.025]; % bar position;
% Save file to fig and png files:
sFigFileName = '2013_TRO_K';
fig = gcf;
saveas(fig,sFigFileName,'fig');% save figures in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0');% save figures in png format;
```

A.2.6 Script “Plot_Sunrise_13.m”

This script plots the lines of the times sunrise and sunset during 2013 year for Joensuu, LFO and Moscow on the selected background yearly figure (from previous script “Plot_yearly_All_param”). The figures with lines of sunrise and sunset are saved to .fig file in MatLab format and to .png file. This figures allow to see the behavior different parameters (HF or ionospheric parameters) in relation to times of the sunrise and sunset.

```
% Script "Plot_Sunrise_13" plots Sunrise and Sunset time lines 2013 year for Joensuu,
% Moscow and LFO,  Joensuu time for TEC JOEN, Int Tromso, DFS Tromso;
% Moscow time for F0F2 LFO;
% LFO time for TEC LFO Int LFO, DFS LFO.

% Clock belt for local time:
time_zone = 3; % hour (Joensuu +2; LFO +2; Moscow +3);
% Date of change on sammer time:
t_b_summer = datenum('2013/03/31'); % 'yyyy/mm/dd' begin of summer time;
t_e_summer = datenum('2013/10/26'); % 'yyyy/mm/dd' the end of summer time;
flagsumtime = 0; % for Moscow in 2013 year is no summer time correction;
flagLT = 0; % add the y axis with Local Time to figure: 0 - not add, 1- add;
%
%% Extract the times Sunrise and times Sunset from txt data file:
if time_zone==2,
  txtopendialog=('Open Sunrise-set data for timezone UT+2');
elseif time_zone==3,
  txtopendialog=('Open Sunrise-set data for Moscow UT+3');
end;
[FileName,PathName]=uigetfile('*.txt',txtopendialog);
filename1=[PathName,'\',FileName];
fid1 = fopen(filename1);
Sun_riseset = textscan(fid1,'%s %s %s','CommentStyle','%');
fclose(fid1);
%%
Date_arr = Sun_riseset{1}; % cell array of dates;
Date_num_arr = zeros(length(Date_arr),1);% array for dates in MatLab form;
Sunrise_num = zeros(length(Date_arr),1);% array for Sunrise times in MatLab form;
Sunset_num = zeros(length(Date_arr),1);% array for Sunset times in MatLab form;
```

```
Sunrise_0_24 = zeros(length(Date_arr),1);% array for Sunrise times in part of day;
Sunset_0_24 = zeros(length(Date_arr),1);% array for Sunrise times in part of day;
n_t_b_summer=0; % initialization index of begin-day summer time;
n_t_e_summer=0; % initialization index of end-day summer time;
time_zone_num = time_zone*1/24; % convert time zone to time Matlab;

for i=1:length(Date_arr), % i - day in month;
   temp_date=char(Date_arr(i));
   Date_num_arr(i)=datenum(str2num(temp_date(7:10)),str2num(temp_date(4:5))...
     ,str2num(temp_date(1:2))); % transfer Dates and Times from cells...
                                % to date in format MatLab;
    % Calc Sunrise times:
    if strcmp((char(Sun_riseset{2}(i))),'NaN'),%find cell with NaN - NO SUN DAY
        Sunrise_num(i) = NaN;
        Sunrise_0_24(i) = NaN;
    else
        str_sunrise = [char(Sun_riseset{1}(i)),'.',char(Sun_riseset{2}(i))];
        Sunrise_num(i)= datenum(str_sunrise,'dd.mm.yyyy.HH:MM')-time_zone_num;
        Sunrise_0_24(i)=Sunrise_num(i)-datenum(char(Sun_riseset{1}(i)),'dd.mm.yyyy');
        if (Sunrise_0_24(i)<0),
            Sunrise_0_24(i)=0;
        end;
    end;
    % Calc Sunset times:
    if (strcmp((char(Sun_riseset{3}(i))),'NaN'))||(isempty(char(Sun_riseset{3}(i)))),
        Sunset_num(i)= NaN; % if in cell is NaN or cell is empty
        Sunset_0_24(i) = NaN;
    else
        str_sunset = [char(Sun_riseset{1}(i)),'.',char(Sun_riseset{3}(i))];
        Sunset_num(i)= datenum(str_sunset,'dd.mm.yyyy.HH:MM')-time_zone_num;
        Sunset_0_24(i) = Sunset_num(i)-datenum(char(Sun_riseset{1}(i)),'dd.mm.yyyy');
        if Sunset_0_24(i)<0,
            Sunset_0_24(i)=1;
        end;
    end;
end;
%%
[~,n_t_b_summer]=min(abs(t_b_summer-Date_num_arr));%day number of begin summer time;
[~,n_t_e_summer]=min(abs(t_e_summer-Date_num_arr));%day number of the end summer time;

i=0;
%% Correction summer time:
if flagsumtime==1; % if you need correction summer time;
   for i=1:length(Date_arr),% i - day in year;
      if (i>=n_t_b_summer)&&(i<=n_t_e_summer),% if current day is summer time;
        Sunrise_num(i)= Sunrise_num(i)-1/24; % minus 1 hour;
        Sunset_num(i)= Sunset_num(i)-1/24; % minus 1 hour;
        Sunrise_0_24(i)=Sunrise_0_24(i)-1/24;% minus 1 hour;
        Sunset_0_24(i)=Sunset_0_24(i)-1/24; % minus 1 hour;
      end;
   end;
end;
%% Open figure.fig files (yearly distributions):
% Sunrise and Sunset lines will be drawn on this figure;
[FileNameFig,PathNameFig]=uigetfile('*.fig','Open background figure');
figname = [PathNameFig,FileNameFig];
openfig(figname);
%% Plot Sunrise and Sunset lines:
Days=(1:length(Date_arr));
%figure;
hold on;
plot(Days,Sunrise_0_24*24,'r','LineWidth',1); % Sunrise line;
plot(Days,Sunset_0_24*24,'b','LineWidth',1); % Sunset line;
hold off;
%% Plot scale local time:
if flagLT ==1,
hold on;
[axL,h1,h2]=plotyy(Days,Sunrise_0_24*24,Days,Sunset_0_24*24);
h1.LineWidth = 1; % Sunrise line;
h2.LineWidth = 1; % Sunset line;
h1.Color = 'r'; % Sunrise line;
h2.Color = 'b'; % Sunset line;
label1 = sprintf('Local Time, UT+%d, Hours',time_zone);
```

```
ylabel(axL(1),label1); % left y-axis
ylabel(axL(2),'Time UT, Hours');  % right y-axis
axL(1).YTick = [0;2;4;6;8;10;12;14;16;18;20;22;24]; % left y axis;
if time_zone==2, % for Joensuu and LFO;
axL(1).YTickLabel = [2;4;6;8;10;12;14;16;18;20;22;24;2]; % for Joensuu
elseif time_zone==3, % for Moscow;
axL(1).YTickLabel = [03;5;7;9;11;13;15;17;19;21;23;01;03]; % for Moscow
end;
axL(2).YTick = [0;2;4;6;8;10;12;14;16;18;20;22;24]; % UT, right y axis;
axL(2).YTickLabel = [0;2;4;6;8;10;12;14;16;18;20;22;24];
axL(1).YColor = 'k'; axL(2).YColor = 'k';
axL(1).YLim =[0 24]; axL(2).YLim =[0 24];
hold off;
end;

%% Save plots in .fig and .png
sFigFileName=[(strtok(figname, '.')),'+sunriseset'];
% strtok - cut all name without '.fig'
fig = gcf;
saveas(fig,sFigFileName,'fig'); % save figure in MatLab format;
fig.PaperUnits = 'centimeters'; fig.PaperPosition = [0 0 30 15];
fig.PaperPositionMode = 'manual';
print(sFigFileName,'-dpng','-r0'); % save figure in png format;
```

A.2.7 Script "Plot_dayly_all.m"

This script plots the Intensities, TEC, F0F2 and magnetic K, Kp-indices for LFO and Tromso on one figure for 3 selected days (the 1 day is selected and 1 day before it and 1 day after it). Such figures allow studying one day in more detail. The figures are saved to MatLab figure files and to .png files.

```
% Script "Plot_dayly_all" plots TEC, Intensity, f0f2, Dopler, Kp,
% K for Tromso and LFO. (2013 year)

% Set Font parameters:
set(0,'DefaultAxesFontSize',8,'DefaultAxesFontName','Helvetica');
set(0,'DefaultTextFontSize',8.8,'DefaultTextFontName','Helvetica');
%
Date_need = '14-05-2013'; % The required date in format: dd-mm-yyyy;
%
load('All_parameters_tot.mat'); % load data for plotting figures;
beg_year_num=datenum(Year,01,01); % the begin of year in Matlab format;
N_day = datenum(Date_need,'dd-mm-yyyy')-datenum(beg_year_num)+1;%...
                                          % defifnition day number in year;
add_day = 1; % numbers of days witch add to N_day to right ==>
befor_day =1; % numbers of days witch add before to N_day to left <==

% Calculate linear indices:
itHFb = sub2ind(size(TyearHF_dt13),1,N_day-befor_day);% linear index of...
                                                    % the begin time of HF;
itHFe = sub2ind(size(TyearHF_dt13),48,N_day+add_day);% linear index of...
                                                     % the end time of HF;
% linear indices of the begin and the end of HF intensity:
for i=1:2, % k - satation number: 1 - LFO, 2 - TRO;
iHFb(i)= sub2ind(size(IenN13),1,N_day-befor_day,i);
iHFe(i)= sub2ind(size(IenN13),48,N_day+add_day,i);
end
% linear index of the begin time of F0F2 LFO:
itFb_LFO = sub2ind(size(TyearF_LFO_dt13),1,N_day-befor_day);
% linear index of the end time of F0F2 LFO:
itFe_LFO = sub2ind(size(TyearF_LFO_dt13),96,N_day+add_day);
% linear index of the begin time of F0F2 Tromso:
itFb_TRO = sub2ind(size(TyearF_TRO_dt13),1,N_day-befor_day);
% linear index of the end time of F0F2 Tromso:
itFe_TRO = sub2ind(size(TyearF_TRO_dt13),720,N_day+add_day);
% linear index of the begin time of TEC:
itTECb = sub2ind(size(TyearTEC_dt13),1,N_day-befor_day);
% linear index of the end time of TEC:
itTECe = sub2ind(size(TyearTEC_dt13),288,N_day+add_day);
```

```
% linear index of the begin F0F2 LFO:
iFb_LFO = sub2ind(size(FN13_LFO),1,N_day-befor_day);
% linear index of the end F0F2 LFO:
iFe_LFO = sub2ind(size(FN13_LFO),96,N_day+add_day);
% linear index of the begin F0F2 Tromso:
iFb_TRO = sub2ind(size(FN13_TRO),1,N_day-befor_day);
% linear index of the end F0F2 Tromso:
iFe_TRO = sub2ind(size(FN13_TRO),720,N_day+add_day);

for k=1:4, % k - station number: 1-LFO, 2-Kiruna, 3-Joensuu, 4-Tromso;
  iTECb(k) = sub2ind(size(TECN13),1,N_day-befor_day,k);% linear index of begin TEC;
  iTECe(k) = sub2ind(size(TECN13),288,N_day+add_day,k);% linear index of end TEC;
end;

iKb = sub2ind(size(TyearK_dt13),1,N_day-befor_day); %linear index of begin K and Kp;
iKe = sub2ind(size(TyearK_dt13),8,N_day+add_day); %linear index of end K and Kp;
%% Plot LFO
figure('Units', 'normalized', 'OuterPosition', [0 0 1 1]);
% Plot Intensity for LFO:
subplot(4,2,1);
h1=plot(TyearHF_dt13(itHFb:itHFe),10*log10(IenmN13(iHFb(1):iHFe(1))),'-.',...
    TyearHF_dt13(itHFb:itHFe),10*log10(IenN13(iHFb(1):iHFe(1))),'-');
ax1=gca;
t1=title(sprintf('%s Intensity(dB), LFO', Date_need));
t1.FontSize = 8.8; t1.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
ylabel('Intensity, dB');
% Set parameters for lines:
 h1(1).Color = [0.5 0.5 0.5]; % averaged line;
 h1(2).Color = 'b';
% Calculate axes limits:
 XLimMIN = datenum(Date_need,'dd-mm-yyyy')-befor_day;
 XLimMAX = datenum(Date_need,'dd-mm-yyyy')+add_day+1;
 YLimMIN = min([10*log10(IenN13(iHFb(1):iHFe(1))) 10*log10(IenmN13(iHFb(1):iHFe(1)))]);
 YLimMAX = max([10*log10(IenN13(iHFb(1):iHFe(1))) 10*log10(IenmN13(iHFb(1):iHFe(1)))]);
% Set grid and Ticks:
 grid on;  ax1.XMinorGrid = 'on';  ax1.YMinorGrid = 'on';
 ax1.MinorGridLineStyle = ':';
 NumTicks = 6*(add_day+1+befor_day)+1;
 ax1.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax1.XLim = [XLimMIN XLimMAX];
 if(~any(isnan([YLimMIN YLimMAX]))),% if not NaN or zero;
     ax1.YLim = [YLimMIN YLimMAX];
 end;
 ax1.Position=[0.13 0.78 0.335 0.139];

%%
% Plot TEC for LFO:
subplot(4,2,3);
h2=plot(TyearTEC_dt13(itTECb:itTECe),TECmN13(iTECb(1):iTECe(1)),'-.',...
             TyearTEC_dt13(itTECb:itTECe),TECN13(iTECb(1):iTECe(1)),'-');
ax2=gca;
t2=title(sprintf('%s TEC, LFO', Date_need));
t2.FontSize = 8.8; t2.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
ylabel('TEC, TECU');

% Set parameters for lines:
 h2(1).Color = [0.5 0.5 0.5]; % averaged line;
 h2(2).Color = 'b';
% Calculate axes limits:
 YLimMIN = min([(TECN13(iTECb(1):iTECe(1)))  (TECmN13(iTECb(1):iTECe(1)))]);
 YLimMAX = max([(TECN13(iTECb(1):iTECe(1)))  (TECmN13(iTECb(1):iTECe(1)))]);
% Set grid and Ticks:
 grid on; ax2.XMinorGrid = 'on'; ax2.YMinorGrid = 'on';
 ax2.MinorGridLineStyle = ':';
 NumTicks = 6*(add_day+1+befor_day)+1;
 ax2.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax2.XLim = [XLimMIN XLimMAX];
```

```
 if(~any(isnan([YLimMIN YLimMAX]))), % if not NaN or zero;
     ax2.YLim = [YLimMIN YLimMAX];
 end;
 ax2.Position=[0.13 0.561 0.335 0.139];

%%
%Plot f0f2 for LFO:
subplot(4,2,5);
h3=plot(TyearF_LFO_dt13(itFb_LFO:itFe_LFO),FmN13_LFO(iFb_LFO:iFe_LFO),'-.',...
            TyearF_LFO_dt13(itFb_LFO:itFe_LFO),FN13_LFO(iFb_LFO:iFe_LFO),'-');
ax3=gca;
t3=title(sprintf('%s F0F2, LFO', Date_need));
t3.FontSize = 8.8; t3.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
ylabel('F0F2, MHz');
% Set parameters for lines
 h3(1).Color = [0.5 0.5 0.5]; % averaged line;
 h3(2).Color = 'b';
% Calculate axes limits:
 YLimMIN = min([(FN13_LFO(iFb_LFO:iFe_LFO)) (FmN13_LFO(iFb_LFO:iFe_LFO))]);
 YLimMAX = max([(FN13_LFO(iFb_LFO:iFe_LFO)) (FmN13_LFO(iFb_LFO:iFe_LFO))]);
% Set grid and Ticks
 grid on; ax3.XMinorGrid = 'on'; ax3.YMinorGrid = 'on';
 ax3.MinorGridLineStyle = ':';
 NumTicks = 6*(add_day+1+befor_day)+1;
 ax3.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 datetick('x','hh','keepticks');
% Set axes limits
 ax3.XLim = [XLimMIN XLimMAX];
 if(~any(isnan([YLimMIN YLimMAX]))),
     ax3.YLim = [YLimMIN YLimMAX];
 end;
 ax3.Position=[0.13 0.342 0.335 0.139];
%%
%Plot Kp and K-indices for LFO:
subplot(4,2,7);
h4 = bar(TyearK_dt13(iKb:iKe),Kp13(iKb:iKe),0.8,'k'); % Kp-index;
hold on;
h5 = bar(TyearK_dt13(iKb:iKe),KLFO13(iKb:iKe),0.4,'r'); % K-inex LFO;
ax4=gca;
t4=title(sprintf('%s K(red) and Kp(balck) indices, LFO', Date_need));
t4.FontSize = 8.8; t4.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
ylabel('K, 1..9');

% Set grid and Ticks:
 grid on; ax4.XMinorGrid = 'on'; ax4.YMinorGrid = 'on';
 ax4.MinorGridLineStyle = ':';
 NumTicks = 8*(add_day+1+befor_day)+1;
 ax4.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 ax4.YTick = (0:9);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax4.XLim = [XLimMIN XLimMAX]; ax4.YLim = [0 9];
hold off;
ax4.Position=[0.13 0.123 0.335 0.139];

%% Plot TRO
% Plot Intensity for TRO:
subplot(4,2,2);
h6=plot(TyearHF_dt13(itHFb:itHFe),10*log10(IenmN13(iHFb(2):iHFe(2))),'-.',...
    TyearHF_dt13(itHFb:itHFe),10*log10(IenN13(iHFb(2):iHFe(2))),'-');
ax6=gca;
t6=title(sprintf('%s Intensity(dB), TRO', Date_need));
t6.FontSize = 8.8; t6.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
yl6=ylabel('Intensity, dB');
% Set parameters for lines:
 h6(1).Color = [0.5 0.5 0.5]; % averaged line;
 h6(2).Color = 'b';
% Calculate axes limits:
 YLimMIN = min([10*log10(IenN13(iHFb(2):iHFe(2)))...
                                10*log10(IenmN13(iHFb(2):iHFe(2)))]);
```

```
 YLimMAX = max([10*log10(IenN13(iHFb(2):iHFe(2)))...
                                10*log10(IenmN13(iHFb(2):iHFe(2)))]);
% Set grid and Ticks:
 grid on; ax6.XMinorGrid = 'on'; ax6.YMinorGrid = 'on';
 ax6.MinorGridLineStyle = ':';
 NumTicks = 6*(add_day+1+befor_day)+1;
 ax6.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax6.XLim = [XLimMIN XLimMAX];
 if(~any(isnan([YLimMIN YLimMAX]))),% if not NaN or zero;
     ax6.YLim = [YLimMIN YLimMAX];
 end;
 ax6.Position=[0.5 0.78 0.335 0.139];
%%
%Plot TEC for TRO:
subplot(4,2,4);
h7=plot(TyearTEC_dt13(itTECb:itTECe),TECmN13(iTECb(2):iTECe(2)),'-.',...
            TyearTEC_dt13(itTECb:itTECe),TECN13(iTECb(4):iTECe(4)),'-',...
            TyearTEC_dt13(itTECb:itTECe),TECN13(iTECb(2):iTECe(2)),'-');
ax7=gca;
t7=title(sprintf('%s TEC, TRO(blue), Kiruna(red)', Date_need));
t7.FontSize = 8.8; t7.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
yl7=ylabel('TEC, TECU');
% Set parameters for lines
 h7(1).Color = [0.5 0.5 0.5]; % averaged line;
 h7(2).Color = 'b'; % line color for TEC Tromso;
 h7(3).Color = 'r'; % line color for TEC Kiruna;
% Calculate axes limits:
 YLimMIN = min([(TECN13(iTECb(2):iTECe(2))) (TECN13(iTECb(4):iTECe(4)))...
                                        (TECmN13(iTECb(2):iTECe(2)))]);
 YLimMAX = max([(TECN13(iTECb(2):iTECe(2))) (TECN13(iTECb(4):iTECe(4)))...
                                        (TECmN13(iTECb(2):iTECe(2)))]);
% Set grid and Ticks:
 grid on; ax7.XMinorGrid = 'on'; ax7.YMinorGrid = 'on';
 ax7.MinorGridLineStyle = ':';
 NumTicks = 6*(add_day+1+befor_day)+1;
 ax7.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax7.XLim = [XLimMIN XLimMAX];
 if(~any(isnan([YLimMIN YLimMAX]))),% if not NaN or zero;
     ax7.YLim = [YLimMIN YLimMAX];
 end;
 ax7.Position=[0.5 0.561 0.335 0.139];
%%
% Plot f0f2 for TRO:
subplot(4,2,6);
h8=plot(TyearF_TRO_dt13(itFb_TRO:itFe_TRO),FmN13_TRO(itFb_TRO:itFe_TRO),'-.',...
            TyearF_TRO_dt13(itFb_TRO:itFe_TRO),FN13_TRO(itFb_TRO:itFe_TRO),'-');
ax8=gca;
t8=title(sprintf('%s F0F2, TRO', Date_need));
t8.FontSize = 8.8; t8.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
yl8=ylabel('F0F2, MHz');
% Set parameters for lines:
 h8(1).Color = [0.5 0.5 0.5]; % averaged line;
 h8(2).Color = 'b';
% Calculate axes limits:
 YLimMIN = min([(FN13_TRO(itFb_TRO:itFe_TRO)) (FmN13_TRO(itFb_TRO:itFe_TRO))]);
 YLimMAX = max([(FN13_TRO(itFb_TRO:itFe_TRO)) (FmN13_TRO(itFb_TRO:itFe_TRO))]);
% Set grid and Ticks:
 grid on; ax8.XMinorGrid = 'on'; ax8.YMinorGrid = 'on';
 ax8.MinorGridLineStyle = ':';
 NumTicks = 6*(add_day+1+befor_day)+1;
 ax8.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax8.XLim = [XLimMIN XLimMAX];
 if(~any(isnan([YLimMIN YLimMAX]))),% if not NaN or zero;
     ax8.YLim = [YLimMIN YLimMAX];
 end;  ax8.Position=[0.5 0.342 0.335 0.139];
```

```
%%
%Plot Kp and K-indices for TRO:
subplot(4,2,8);
h9 = bar(TyearK_dt13(iKb:iKe),Kp13(iKb:iKe),0.8,'k'); % Kp-indices;
hold on;
h10 = bar(TyearK_dt13(iKb:iKe),KTRO13(iKb:iKe),0.4,'b');% K-indices;
ax9=gca;
t9=title(sprintf('%s K(blue) and Kp(black) indices, TRO', Date_need));
t9.FontSize = 8.8; t9.FontWeight = 'normal'; % font for title;
xlabel('Time, UT');
yl9=ylabel('K, 1..9');
% Set grid and Ticks:
 grid on; ax9.XMinorGrid = 'on'; ax9.YMinorGrid = 'on';
 ax9.MinorGridLineStyle = ':';
 NumTicks = 8*(add_day+1+befor_day)+1;
 ax9.XTick = linspace(XLimMIN,XLimMAX,NumTicks);
 ax9.YTick = (0:9);
 datetick('x','hh','keepticks');
% Set axes limits:
 ax9.XLim = [XLimMIN XLimMAX];
 ax9.YLim = [0 9];
hold off;
 ax9.Position=[0.5 0.123 0.335 0.139];

%% Save file to fig and png files
 sFigFileName = sprintf('All_parameters_%s',Date_need);
 fig = gcf;
 saveas(fig,sFigFileName,'fig'); % save figures in MatLab format;
 fig.PaperUnits = 'centimeters';
 fig.PaperPosition = [0.635 6.35 30 17];
 fig.PaperPositionMode = 'manual';
 print(sFigFileName,'-dpng','-r0'); % save figures in png format;
```

References

[1] Rawer K. (1993), *Wave Propagation in the Ionosphere*, Springer, ISBN: 9789401736657, doi: 10.1007/978-94-017-3665-7.

[2] Davies K. (1973), *Ionospheric radio waves*, Moscow "Mir", UDC 551.510.535, (in Russian).

[3] Brunelli B.E., Namgaladze A.A. (1988), *Physics of the ionosphere*, Moscow "Nauka", ISBN: 5-02-000716-1.

[4] Hunsucker R.D. (1991), *Radio techniques for probing the terrestrial ionosphere*, Springer, ISBN: 3540534067.

[5] Zolesi B., Cander L.R. (2014), *Ionospheric Prediction and Forecasting*, Springer, ISBN: 978-3-642-38430-1, doi: 10.1007/978-3-642-38430-1.

[6] *Tromso Geophysical Observatory*: http://flux.phys.uit.no/geomag.html

[7] *Space weather prediction center, national oceanic and atmospheric administration*. URL: ftp://ftp.swpc.noaa.gov/pub/warehouse/2013/

[8] Hunsucker R.D, Hargreaves J.K., Alexander D.J. (2002), *High-Latitude Ionosphere and its Effects on Radio Propagation*, Cambridge University Press, ISBN: 9781139145534.

[9] Dolukhanov M.P. (1972), *Radio wave propagation*, Moscow "Svyaz", UDC 621.371, (in Russian).

[10] Brekke A. (2013), *Physics of the Upper Polar Atmosphere*, Springer, ISSN: 2194-5225, doi: 10.1007/978-3-642-27401-5.

[11] Izumov N.M., Linde D.P. (1983), *Fundamentals of radio engineering*, Moscow "Svyaz", UDC 621.37, (in Russian).

[12] Danilov A.D. (1989), *Popular aeronomy*, 2nd edition, Gidrometeoizdat, ISBN: 5-286-00171-8, (in Russian).

[13] Schunk R., Nagy A. (2009), *Ionospheres. Physics, Plasma Physics, and Chemistry*, Cambridge University Press, ISBN: 0521632374.

[14] John A. Richards (2008), *Radio Wave Propagation*, Springer, ISBN: 3-540-77124-7, doi: 10.1007/978-3-540-77125-8.

[15] Hargreaves J.K. (1982), *The Upper Atmosphere and Solar-Terrestrial Relations*, Leningrad "Gidrometeoizdat", UDC 551.510.5, (in Russian).

[16] Titheridge J.E. (1988), *The real height analysis of ionograms: A generalized formulation*, Radio Science, Vol.23(5), pp.831-849, doi: 10.1029/RS023i005p00831.

[17] Dr. Ruth Bamford (2000), *The oblique ionospheric sounder*, Project Final Report, Rutherford Appleton Laboratory, Chilton, UK, URL: http://www2.ofcom.org.uk/static/archive/ra/topics/research/rcru/project42/finalreport/RA_final_Report_oblique_Sounder1.pdf

[18] Ippolito A., Scotto C., Sabbagh D. (2016), *A procedure for the reliability improvement of the Oblique Ionograms Automatic Scaling Algorithm*, Radio Science, doi: 10.1002/2015RS005919.

[19] Arthur P.C., Cannon P.S. (1994), *ROSE: A high performance oblique ionosonde providing new opportunities for ionospheric research*, Annals of Geophysics, Vol.37(2), pp.135–143, ISSN: 1593-5213, doi: 10.4401/ag-4220.

[20] Chen Gang, Zhao ZhengYu, Zhang et al. (2012), *Application of the oblique ionogram as vertical ionogram*, Science China Technological Sciences, Vol.55(5), pp.1240-1244, doi: 10.1007/s11431-012-4795-8.

[21] Olsen R.O., Mott D.L. (1981), *Measurements of D-Region Electron Densities Utilizing the Partial Reflection Technique*, AIAA Journal, Vol.19(12), pp. 1558-1561, doi: 10.2514/3.60094.

[22] Dyrud L., Jovancevic A., Brown A. (2008), *Ionospheric measurement with GPS: Receiver techniques and methods*, Radio Science, Vol.43(6), ISSN: 0048-6604, doi: 10.1029/2007RS003770.

[23] Schaer S. (1999), *Mapping and Predicting the Earth's Ionosphere using the Global Positioning System*. Ph.D thesis. AIUB, University of Berne, Berne.

[24] Krypiak-Gregorczyk A., Wielgosz P. (2013), *Validation of approximation techniques for local total electron content mapping*, Acta Geodynamica et Geromaterialia., Vol.10(3), pp.275–283, ISSN: 1214-9705, doi: 10.13168/AGG.2013.0027.

[25] Kersley L., Malan D., Pryse S.E. (2004), *Total electron content – A key parameter in propagation: measurement and use in ionospheric imaging*, Annals of geophysics, Vol.47(2-3), pp.1067-1091, ISSN: 1593-5213.

[26] Namgaladze, A.A., Zolotov O.V. et al. (2009), *Ionospheric total electron content variations observed before earthquakes: Possible physical mechanism and modeling*, Proceedings of the MSTU, Vol.12(2), pp.308-315, ISSN: 1997-4736, UDC 550.347.2, URL: http://vestnik.mstu.edu.ru/v12_2_n35/articles/22_namgal.pdf

[27] *GPS receiver Joensuu.* URL: http://www.epncb.oma.be/_networkdata/siteinfo4onestation.php?station=JOEN00FIN

[28] *GPS receiver Kiruna*. URL: http://www.epncb.oma.be/_networkdata/siteinfo4onestation.php?station=KIR000SWE

[29] *GPS receiver Tromso*. URL: http://www.epncb.oma.be/_networkdata/siteinfo4onestation.php?station=TRO100NOR

[30] *Specification. WiNRADiO WR-G313i*, 2004, WiNRADiO Communications, Melbourne, URL: http://www.nakatoyo.com/WiNRADiO/WR-G313i.PDF

[31] Lorek C. (2005), *WiNRADiO WR-G313i PC-controlled receiver*, URL: http://www.winradio.com/pdf/g313i-review-radcom.pdf

[32] Koloskov A.V., Yampolski Y.M., Zalizovskii A.V., Galushko V.G. (2014), *Network of Internet-controlled HF receivers for ionospheric researches*, Radio Physics and Radio Astronomy,Vol.19(4), pp.324-335, ISSN:1027-9636, UDC 550.388.2, (in Russian).

[33] Zalizovskii A.V., Galushko V.G., Kashcheev A.S., Koloskov A.V. et al. (2007), *Doppler selection of HF radio signals on long paths*, Geomagnetism and Aeronomy, Vol.47(5), pp.674–684, ISSN: 0016-7932, doi: 10.1134/S001679320705012X.

[34] Beley V.S., Galushko V.G., Yampolski Yu. (1995), *Traveling ionospheric disturbances diagnostics using HF signal trajectory parameter variations*, Radio Science,Vol.30(6), pp.1739-1752, ISSN: 00486604, doi: 10.1029/95RS01992.

[35] Kashcheev S.B, Zalizovskii A.V., Koloskov A.V. et al. (2009), *Frequency Variations of HF Signals at Long-Range Radio Paths during a Solar Eclipse*, Radio Physics and Radio Astronomy, Vol.14(4), pp.353-366, ISSN: 1027-9636, UDC 621.396.967, (in Russian).

[36] Hysell D.L., Kelley M.C., Yampolski Y.M., Beley V.S., Koloskov A.V. (1996), *HF radar observations of decaying artificial field-aligned irregularities*, Journal of Geophysical Research, Vol.101(A12), doi: 10.1029/96JA02647.

[37] Betke K. (2009), *Standard frequency and time signal stations on longwave and shortwave*.

[38] The bulletin of the Russian State service of time and frequency (2013), *Standard signals of frequency and time*, URL: ftp://ftp.vniiftri.ru/BULLETINS/V

[39] *EISCAT Tromsø Dynasonde*, URL: http://www.eiscat.uit.no/dynasond.html

[40] Rietveld M., Wrightb J., Zabotinb N., Pittewayc M. (2008), *The Tromso dynasonde*, Polar Science,Vol.2(1), pp.55–71, ISSN: 1873-9652, doi:10.1016/j.polar.2008.02.001.
[41] Reinisch B. et al. (2007), *The digisonde portable sounder – DPS. Technical manual*, University of Massachusetts Lowell Center for Atmospheric Research, Version 4.3.
[42] *GPS receiver photo*, URL: http://geospace.com.ua/observatory/images/gps1.gif
[43] Akhmetianov V.R., Mishina O.A. (2010), *Processing of Wind Coherent Doppler Lidar Data on the Base of Gaussian Approximation Method*, Journal of Instrument Engineering, ISSN: 0021-3454, UDC 681.7; 681.518.26, (in Russian).
[44] Kohl H., King J.W. (1967), *Atmospheric winds between 100 and 700 km and their effects on the ionosphere*, Journal of Atmospheric and Terrestrial Physics, Vol.29(9), pp.1045-1062, doi:10.1016/0021-9169(67)90139-0.
[45] Kohl H., King J.W., Eccles D. (1968), *Some effects of neutral air winds on the ionospheric F-layer*, Journal of Atmospheric and Terrestrial Physics, Vol.30(10), pp.1733-1744, doi:10.1016/0021-9169(68)90094-9.
[46] Mendillo M. (2006), *Storms in the ionosphere: Patterns and processes for total electron content*, Reviews of Geophysics, Vol.44(4), RG4001 ISSN: 8755-1209, doi:10.1029/2005RG000193.
[47] Akasofu S.I., Chapman S. (1972), *Solar-Terrestrial physics*, part 2, Clarendon Press, ISBN: 0198512627.
[48] Perevalova N.P., Polyakova A.S., Zalizovski A.V. (2010), *Diurnal variations of the total electron content under quiet helio-geomagnetic conditions*, Journal of Atmospheric and Solar-Terrestrial Physics, Vol.72(13), pp.997-1007, doi:10.1016/j.jastp.2010.05.014.