Linking Single Event In-Orbit Data, Space Weather and Satellite Operations

Masters Thesis

Patrick Hornung

Delft University of Technology

Delft University of Technology

Challenge the future

Linking Single Event In-Orbit Data, Space Weather and Satellite Operations

Masters Thesis

by

Patrick Hornung

Student number:	468 11 50
Supervisor:	Dr. Alessandra Menicucci
Submission date:	May 31, 2019

Cover page:
Aurora Australis observed from the International Space Station
© Earth Science and Remote Sensing Unit, NASA Johnson Space Center

TU Delft
Delft University of Technology

Summary

This thesis studies the relationship between single event in-orbit occurrences, space weather and satellite operations through single event counter data from the satellites Sentinel 2A and 2B.

When ionizing radiation passes an electronic device, single events can occur. These are the deviations of device behavior from nominal behavior, caused by the charge deposited in the device due to a single particle. A prominent example of single events are bit flips in memories, but they can also have consequences that ultimately result in the loss of the satellite. The consequences and likelihood of single events depend on the particle environment and the characteristics and type of the device under consideration. Aboard satellites, single event counters register the occurrence of single events in some devices.

The Sentinel 2 satellites orbit in Lower Earth Orbit (LEO). There, single events are most likely caused by protons, which are particularly abundant in a region called the South-Atlantic Anomaly (SAA). In this region the inner radiation belt reaches altitudes much closer to the Earth's surface and as low as 200 km.

Space weather describes the modulation of space particle environments due to solar activity, and as such represents timely changes in radiation environments. Satellite operations influence the temperature of electronic components, which in turn influences their susceptibility to single events. A device's single event susceptibility can also change over time through degradation.

Based on the hypothesis that space weather, satellite operation and device degradation result in variable increases in single event counters, this thesis determines in what ways these three influences can be found in single event data. In doing so, it follows the goal of (1) judging the space weather sensing capability of single event counters, (2) learning about relevant influences from satellite operations and (3) how single event rate changes on the long term due to device degradation.

First, the theoretical background of single events and single event rate prediction is presented. Also, theory on space weather, relevant with the orbit of the Sentinel 2 satellites, as well as information about the satellites themselves and their orbit is provided. With the description of the satellites, an emphasis is put on single event drivers to determine how their changing can affect single event rate. Finally, a list of all identified drivers for single event rate with the Sentinel 2 satellites is given. This list, in combination with the random nature of single events, allows to make a distinction in the subsequent study of single event rate drivers: short-term effects need to be treated differently than the long-term evolution.

To facilitate the implementation of the analyses, existing Python libraries for data analysis are explored. After pre-processing the data and evaluating the results, one of the single event counters is selected for further analysis on the basis of event rate, time coverage and plausibility of event occurrences registered by the respective counter. The selected counter belongs to an Static Random Access Memory (SRAM), so that single event drivers specific to this type of component need to be treated.

The long-term analysis is performed on single event rate and position data and comprises an explanation of methodology, its implementation in Python, the obtained results and a discussion thereof. On the basis of time series decomposition, a moving-average filter is defined for data filtering and a window length of 20 days is determined exploratively. The filtered event rate data shows a behavior that in its trend is attributed to the solar cycle and device degradation and is similar among the two satellites. Future data will help to further distinguish the two factors.

Also, timely variations in single event rate are visible, which are hypothesized to be caused by the seasonal behaviour of the SAA, geomagnetic storms and hot spots in the device. Data from times of higher geomagnetic activity, simulations of radiation belt storm responses and data on the memory addresses affected by single events will help to distinguish these influences further.

Further, there is no correlation of event rate to relevant space weather variables, which means that single event counters can at best be used to detect long-term space weather phenomena as the ones mentioned previously.

Data on single event positions is subjected to a similar long-term analysis. The resulting scatter plots of filtered event position track the position of the SAA, but show peculiar and currently unexplained differences between Sentinel 2A and 2B. These are also attributed to memory hot spots. The linearized event position drift is similar to that of the SAA as reported in literature.

Next, short-term effects on single event rate are studied through Lomb-Scargle and box-fitting spectral methods. Again, methodology and implementation are given, before the different spectra are presented in the results section. The results show that the periodicity of SAA passes is the strongest influence and is represented twofold: through spectral peaks at the orbital period and at a 12-hour period. These results also indicate that no further influence on single event rate has been overlooked. A comparison between event rate in and out of eclipse provides ambiguous results.

The conclusion synthesizes the findings of the two analyses, defines a way forward to clear up the discovered ambiguities, and offers some secondary research ideas. This research is novel in that it provides quantitative links between solar cycle, device degradation and event rate and resolves the SAA position with higher temporal accuracy than another previous single event based approach. Lastly, a vision on the general study of single event data is presented.

Acknowledgements

My deep gratitude goes to Alessandra Menicucci of TU Delft for supervising this thesis, sharing knowledge with me and for providing valuable feedback and suggestions. Many thanks also for offering this fascinating and challenging research subject to begin with and for establishing the contact to the Copernicus team at European Space Agency (ESA).

Further, I want to thank Gianluca Furano, Giuseppe Mandorlo and Marek Prochazka of ESA for answering questions on the Sentinel 2 satellites, discussing research results and for providing the data set studied in this thesis.

Finally, I want to thank my parents and my brother for supporting me throughout my education. Katharina, thank you for encouraging me during my masters studies and this thesis in particular.

Contents

List of Figures

List of Tables

List of Abbreviations

AEX	Autumn Equinox
API	Application Programming Interface
BLS	Box Least Squares
CIR	Co-Rotation Interaction Region
CME	Coronal Mass Ejection
CRAND	Cosmic Ray Albedo Neutron Decay
CSV	Comma-Separated Values
DST	Disturbance Storm Time Index
EDAC	Error Detection and Correction
EOL	End of Life
ESA	European Space Agency
GCR	Galactic Cosmic Ray
GEO	Geostationary Orbit
GMES	Global Monitoring for Environment and Security
IGRF	International Geomagnetic Reference Field
LEO	Lower Earth Orbit
LET	Linear Energy Transfer
LS	Lomb-Scargle
MA	Moving Average
MMFU	Mass Memory & Formatting Unit
MSS	Memory System Supervisor
PDIC	Payload Data Interface Controller
RPP	Rectangular Parallelepiped
S2A	Sentinel 2A
S2A+B	Sentinel 2A and 2B
S2B	Sentinel 2B
SAA	South-Atlantic Anomaly
SRAM	Static Random Access Memory
TFG	Transfer Frame Generator
TID	Total Ionizing Dose
VEX	Vernal Equinox

List of Listings

1 Introduction

There are many obstacles in space flight, affecting both crewed and uncrewed missions. Extreme temperatures and lack of air are two of them, but the one probably least tangible is space particle radiation. It causes humans in space to see light flashes [19] and has severe health risks with interplanetary travel [23]. Up to now, no universal protection from space particle radiation exists, and so human space exploration stays confined to Earth's vicinity. For machines, space particles can lead to malfunctions, in ground based systems like voting machines[1], but more prominently in satellites.

The passing of such particles through an electronic component is called single event, and can have various consequences with satellites, ranging from small and recoverable issues like a flip of a memory cell up to the non-recoverable failure of components or even the entire system. The severe consequences that single events can have make prevention and mitigation measures self-evident. Mitigation methods to single events are, among others, Error Detection and Correction (EDAC), which works by storing a check-sum of several memory cells to later find a changed bit by comparing the check-sum to the content of the memory cells, redundancy, which works by storing data in more than one memory to detect differences later on, or shielding, which means that particles are prevented from reaching the sensitive component to begin with [25].

In what way a satellite experiences particle radiation, and therefore what consequences it causes, depends to a large degree on the environment in which it flies. The Earth-observation satellites Sentinel 2A (S2A) and Sentinel 2B (S2B) fly in LEO and where satellites experience a radiation environment characterized by trapped particles in the inner radiation belt, solar energetic particles and cosmic rays. The purpose of the satellites is capture every point of Earth's landmass within in multiple frequency spectra every 5 days, and by doing so, to monitor the health of wild and agricultural vegetation, gather information to disaster relief and provide information for many other prospects. [37, 48]

During the satellite design phase, the rate of single event occurrence is predicted to know about the measures needed for prevention and mitigation. Such predictions are made on the basis of the orbit of the satellite, models of space radiation environment and the characteristics of the device and shall be as accurate as possible: the rate of single events shall not be overestimated, because their prevention and mitigation requires resources like mass and electrical power [56], but the risk shall also not be underestimated, to not hamper device functioning. During a mission, the occurrence of single events can be measured by counters that are increased every time the manifestation of a single event has been detected. For instance, such a counter increases, whenever a changed bit has been detected in a memory through EDAC or redundancy. This is being done for several components and single event manifestations with the Sentinel 2 satellites.

If the flux of particles in a radiation environment changes, or when the energy distribution of these particles changes, single event rate changes as well. Particle fluxes and energies in different space environments are not constant over time, but are instead dynamic and this gives rise to space weather, which can be defined as "the conditions on the sun and in the solar wind, magnetosphere, ionosphere, and thermosphere that can influence the performance and reliability of spaceborne and ground-based technological systems" (US National Space Weather Program, through [14]). Next to that, the operation of the satellite should be

[1] `https://phys.org/news/2017-02-particles-outer-space-wreaking-low-grade.html`, retrieved 26.04.2019

taken into account since it has an influence on the radiation environment experienced at component level. Operation entails the orbit of the satellite, because different orbits pass different radiation environments that vary in severity. Next to that, operation can entail the attitude of the satellite, which results in a varying protection of electrical components from particles. Thus, satellite operation changes the rate of single events as well. Finally, the characteristics of a device influence the rate of single events. Device properties can change over time, which means that their intrinsic resistance to single events changes.

This thesis deals with the analysis of the single event counters aboard Sentinel 2A and 2B and puts it into the context of the introduced drivers of single event rate. In general, in-orbit single event data has received little attention in research so far, but as will be established in the course of this thesis, there is good reason to believe that its full potential has not unfolded yet, and that it can, to some extent, be used as sensors for those influences listed previously. Error Detection and Correction (EDAC) is a standard feature of all the modern digital data processing systems in space, and SEE counters originating from its operations are available in standard housekeeping telemetry. Nevertheless, they are normally used only for specific failure investigation and never before a comprehensive time series analysis and correlation has been attempted.

1.1. Motivation

Figure 1.1 shows the typical process of single event treatment in space missions and is also a primer to the theory underlying this research.

In the preparation and execution of space missions, the mitigation of single events begins with the prediction of the single event rate, which is made for each satellite component individually using the available radiation models, which can reproduce the radiation environment of the mission under analysis, and the radiation test data, which gives indications of the device's response to particle radiation. Single events in-orbit are determined by the actual radiation environment encountered during the mission and the behaviour of the device in that specific conditions. This highlights that there are discrepancies between the radiation environment and the radiation model as well as between the device's actual behaviour and the behavior seen in testing, because both employ assumptions, simplifications, measurements and tests.

The radiation environment in turn is determined by space weather. The satellite's operation should be taken into account since a different orientation of the satellite can determine a different radiation environment at component level. Space weather is here seen as the timely change in the radiation environment due to solar activity. Currently, there is only very limited consideration of varying space weather conditions in single event rate prediction, as for example by looking into different worst-case conditions [70]. Satellite operation comprises the orbit, which causes the satellite to pass positions of different radiation environment, but possibly also to vary in shielding. Operation also influences the device itself, by bringing it to different temperatures.

The single events in orbit create single event data, that is the main object of study in this thesis. Single event data has a temporal component, which gives the rate of single events over time, and a spatial component, that gives the position in orbit where they occurred.

Using Figure 1.1, the potential uses of single event counters data, and therefore the motivation for this thesis, is explained. This list of potentials is not exhaustive, and only the major ones were identified here. This aspect will be discussed further in the conclusion chapter of this thesis. The three motivational goals here build on giving feedback to parts of the single event treatment process, using single event data. This is tantamount to treating single events as the outputs of a dedicated sensor. The potentials are shown by the red indicators in Figure 1.1.

1. The first potential of single event data is to use it to find links to space weather. Magnetic fields, such as that of Earth, are affected by space weather and modulate particle fluxes as well, and therefore go hand-in-hand with space weather influences, because in the end both factors affect the radiation environment. If links between these and single event data can be found, one might be able to gain insight about the solar activity, magnetic fields and the satellites' radiation environment at the given

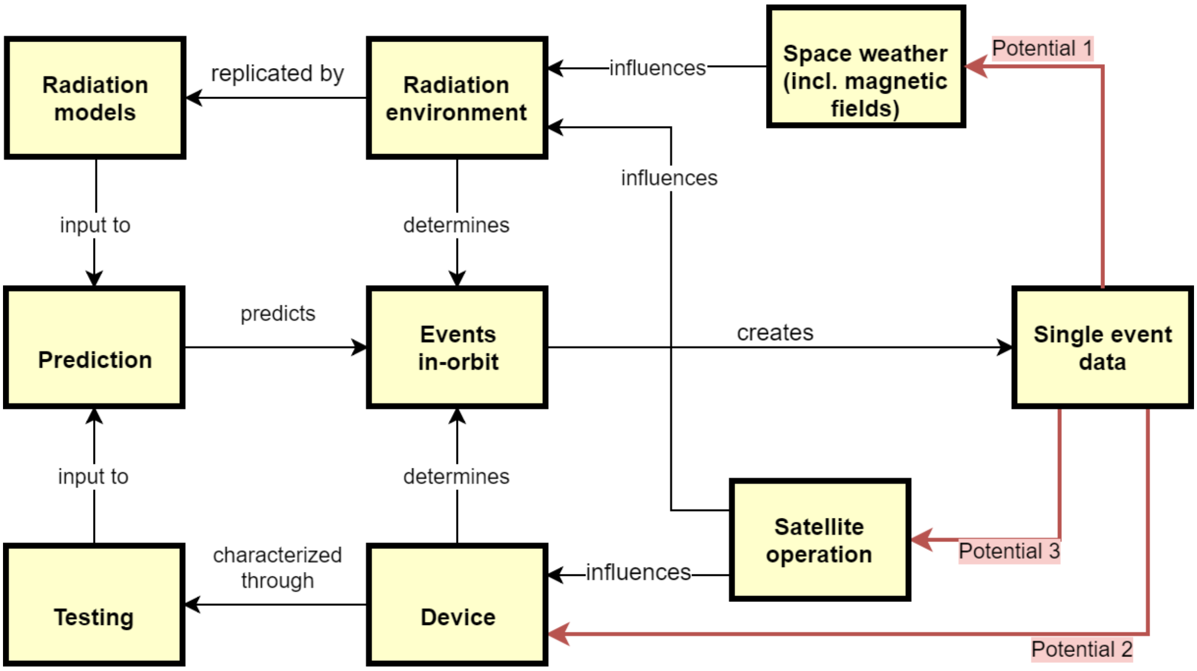


Figure 1.1: Single event treatment in space missions

time. Also, in case space weather influences to single event rate can be quantified, as a future step, their relevance in single event rate prediction can be pointed out.

2. Device characteristics also influence single event rate. They may change over time by degrading, meaning that more (or less) events might occur with mission time. So far, ground tests of devices subjected to aging and total dose testing have been used to determine this influence on single event rate [4]. Single event in-orbit data can allow to study the reasons for in-situ single event rate changes, this being a novel area of research not much explored so far.

3. The third potential is learning what single event data tells about the operation of the satellite, which in turn allows to identify those aspects of satellite operation that influence single event rate and possibly to quantify them. As a second step, this can make single event prediction more accurate by knowing which operation characteristics have a meaningful influence on single event rate.

These goals can be seen as the requirements for the research at hand. Afterwards, it will be verified to what extent it was possible to fulfill them. All three aspects have to be considered and separated in analyzing single event counter data: space weather, device changes and satellite operation. If one aspect is ignored, it will work as a confounder on the remaining data.

1.2. Problem Statement

The objective of this research is to generate knowledge about **the links between single events and space weather**, **the links between single events and satellite operation** and **single events and device long-term changes**, by employing in-orbit data, with the aim of **evaluating the use of single event data for space weather sensing** and **contributing to the topic of degradation induced single event rate changes**.

Now, research questions and sub-questions are derived from the mentioned potentials of using single event data for space weather monitoring, satellite operation monitoring and device monitoring. For each of these aspects, the questions is raised how it shall be analyzed and what the results and implication of the analysis are. Also, supplementary questions on the implementation of the research and its theoretical framework are formed. All of these questions shall focus on the situation with the Sentinel 2, which means that there is the need to look into the characteristics of these satellites specifically. Where possible, the questions shall be answered using both temporal and spatial single event data.

1. **What are the theoretical influences to single events?**
    - (a) How do single events occur?
    - (b) What space weather influences on single event rate are there in theory?
    - (c) What operation influences on single event rate are there in theory?
    - (d) To what extent does device degradation affect single event rate in theory?
    - (e) What needs to be known about the Sentinel 2 satellites?
2. **What infrastructure is needed to work with the given Sentinel 2 single event data?**
    - (a) What tools are needed for handling, analyzing and visualizing of single event data?
    - (b) What needs to be done for pre-processing of single event data?
    - (c) Which subset(s) of the data is(are) most meaningful for the research?
3. **To what extent is space weather visible in single event data?**
    - (a) How can one analyze the link between space weather and single event rate?
    - (b) What findings are there?
    - (c) What are the consequences of these findings?
4. **To what extent is satellite operation visible in single event data?**
    - (a) How can one analyze the link between satellite operation and single event rate?
    - (b) What findings are there?
    - (c) What are the consequences of these findings?
5. **To what extent are device changes visible in the data?**
    - (a) How can one analyze the link between device changes and single event rate?
    - (b) What findings are there?
    - (c) What are the consequences of these findings?

The verification of research question answers is given in Table 1.1. Three methods of answering the questions are used:

- Theory: The research question is answered by studying literature and summarizing the relevant information.
- Analysis: The research question is answered through an analysis, that consists of an implementation in programming and the provision of analysis outputs.
- Discussion: The research question is answered by bringing together insight from literature and insight from an analysis.

ID	Question Summary	Method	Verification Reference
1	Theoretical Framework	Theory	by sub-questions
1a	Single Event Theory	Theory	section 2.1
1b	Space Weather Theory	Theory	section 2.2
1c	Satellite Operation Theory	Theory	section 2.1
1d	Device Degradation Theory	Theory	section 2.1
1e	Sentinel 2	Theory	section 2.3
2	Infrastructure	Theory	by subquestions
2a	Tools/Implementation	Theory	chapter 3
2b	Pre-Processing	Theory+Analysis	section 4.1
2c	Data Selection	Analysis	section 4.2
3	Space Weather Links	Analysis	by subquestions
3a	Analysis Definition	Theory	section 2.4, section 2.5
3b	Results	Analysis	section 5.3
3c	Interpretation	Discussion	section 5.4
4	Satellite Operation Links	Analysis	by subquestions
4a	Analysis Definition	Theory	section 2.4, section 2.5
4b	Results	Analysis	section 6.3
4c	Interpretation	Discussion	section 6.4
5	Device Degradation	Analysis	by subquestions
5a	Analysis Definition	Theory	section 2.4, section 2.5
5b	Results	Analysis	subsection 5.3.1
5c	Interpretation	Discussion	subsection 5.4.1

Table 1.1: Verification of research questions

1.3. Thesis Structure

The thesis begins with a chapter on single events, space weather and Sentinel 2 theory. An emphasis lies on single event rate prediction, because here the relationships between single event influence factors and single event rate can be studied. The implications of satellite operation to single event data is given, so that all information is available to compile an extensive list of single event rate influences with these particular satellites and to define the data analysis. The definition of the analysis shows, that because of the nature of single events, different analyses need to be done for long-term influences and short-term influences. Also, the usage of spatial and temporal single event data is defined.

The subsequent chapter is an excursion into using Python for data analysis. Libraries for data handling, data storage, data processing and data visualization are explored so that a go-to solution is available when needed with the subsequent analyses.

Before the analyses can be conducted, a the data needs to be pre-processed and suitable data subsets have to be identified and selected. This is the focus of the next chapter and, as a result, all single event data is available in a convenient format, and the most promising single event counter has been selected. Information on the respective counter is provided.

Then, the methodology and implementation of the first analysis analysis, which focuses on those influences to single event rate that go over long time frames, is presented. This is entails an analysis of space weather (and magnetic field) influences to single events and degradation influences, and considers temporal and spatial dimension of single event data. The results are then presented and discussed.

The next analysis is that of short-term influences to single event rate. Again, methods and results are given, following a discussion. Due to its focus on short-term influences, this chapter deals with operational influences on single event rate, and thus, this chapter answers questions 4b and 4c.

In the final chapter, the conclusion, results of the analyses are put in comparison with the goals of section 1.1, resulting in several recommendations for making single event data useful in studying space weather, operation influences and device degradation. As a side effect of the previous analyses, understanding about what single event influences are the most relevant has been gained and is also presented. The impact of the results is explained, possible side-lines of research are presented and a critical reflection with the research is made.

2 Background

The following chapter provides the theoretical background needed for the analysis of the S2A+B single event counters and develops a preliminary definition of the analysis. To begin with, the physical working of single events are explained and a summary of their prediction is given. Factors causing them to vary over time, called drivers, are also looked into. Space weather is one of these, which is why a theoretical introduction to it follows in the subsequent section. After that, the characteristics of S2A+B are introduced and the theoretical framework for interpreting the data of the counters is finalized.

Since the Sentinel 2 satellites are flying in LEO [48], there is a focus on the occurrence of single events and the role of space weather for that particular environment. Also, this theory chapter does not go into detail about the methodology of the subsequent data analysis, because the data analysis itself is defined only at the end of section 2.3. Instead, theory on the analysis is introduced along with the analysis itself and immediately linked to its implementation.

2.1. Single Events

The following explanation of single event physics and rate prediction is based on [35] unless otherwise noted.

2.1.1. Physics and Prediction

Single events are caused by charge being deposited in an electronic device due to particle radiation. The charge is picked up by the device and leads to permanent or temporary unintended behaviour in the device, as for example the change of one or several memory cells statuses, rupture of transistor gates or functional disturbances [26]. Charge can be deposited directly from particle radiation passing the device, or from secondaries that are created through nuclear reactions between the passing particle and the material of the electrical device, which is usually silicone. However, charge has to be deposited in a part of the device that is actually sensitive to single events. For instance, with a memory any of its individual cells is sensitive to single events. This part of the device is called the sensitive volume. If a particle is very energetic, it can cause multiple events, for example by flipping multiple memory cells. Such events are called multiple events.

Heavy-Ion Events
Single events by direct charge deposition are often caused by GCRs, which are highly-energetic particles coming from outside of the solar system. Because of the composition of GCRs, these single events are also called heavy-ion events.

In predicting the rate of such events in a device, the energy that a single particle loses per distance traversed through the device is assumed to be linear and is called Linear Energy Transfer (LET). This energy loss is further assumed to be turned into charge entirely, so that through the ionization energy of the target material the energy loss of the particle can be related to a quantity of free charge being created.

Space environments contain diverse particles, both in type and energy, so that the distribution of LET in all particles in a space environment has to be defined by a LET spectrum, which relates the LET of particles to their abundance. This spectrum can be obtained through a radiation environment model and is usually altered to account for the shielding offered by the satellite.

If the accumulated free charge in the sensitive volume exceeds a certain threshold, called critical charge, a single event occurs. The critical charge depends on the device under consideration and the effect the device is prone to. Different types devices need different quantities of charge before their behaviour deviates from normal, and different kinds of misbehaviour also require different amounts of charge to be triggered. Yet, the concept of critical charge is actually a simplification, because not every part of a sensitive volume is equally sensitive to charge. This notion can be captured by a cross-section curve, which relates particle energy and probability of single event manifestation. Cross-section curves are determined by testing the device with particle accelerators, radionuclides or lasers [10].

The energy lost by the particle, and as such the charge being deposited, depends on the length of path being traversed by the particle. Often, sensitive volumina can be approximated as Rectangular Parallelepipeds (RPPs). Obviously, a particle passing an RPP from surface to surface will travel a different path length than a particle crossing from corner to corner. The probability distribution of path lengths, assuming a particle to come from a random direction in space, is called chord distribution.

From LET spectrum, critical charge or cross-section, and chord distribution, the heavy-ion event rate can be predicted using a mathematical formula to link them [26]:

$$N = \frac{S}{4} \int_{D_{min}}^{D_{max}} \Phi(< \mathsf{LET}(D)) \cdot \frac{dP_{CL}}{dD}(D) \;\; dD \tag{2.1}$$

with:

S	Surface area of the RPP		
$\Phi(< \mathsf{LET}(D))$	Integral particle flux spectrum	dP_{CL}/dD	Differential chord distribution

Proton Events
In single events caused by secondaries, the causing particle is mostly a proton, which is why they are also called proton events.

To predict the rate of proton events occurring in a device, the probability of the device upsetting from protons of a certain energy has to be determined, through testing or simulation. The function describing this probability is called cross section as well. The likelihood of proton upsets shows only negligible dependency on path length, because nuclear reactions are not influenced by the path length the particle traverses, but only by an atom of the material being hit or not. Therefore, chord distribution is not considered in the prediction of proton events and the concepts of LET and LET spectrum are no longer meaningful, but only the energy spectrum of protons in the respective space environment. Still, this spectrum, taken from a radiation model, needs to be adjusted for the satellite's shielding to obtain the actual energy spectrum seen by the device within the satellite.

From cross section curve and the energy distribution of protons in the expected environment, the rate of single events of this type can be predicted. The equation that relates these is [27]:

$$R = \int\limits_{E_{min}}^{E_{max}} \frac{d\Phi}{dE}\, \sigma_{SEU}(E)\; dE \tag{2.2}$$

with [27]:

E_{min}	Lower energy spectrum boundary	E_{max}	Upper energy spectrum boundary
$d\Phi/dE$	Differential particle flux spectrum	$\sigma_{SEU}(E)$	Cross section

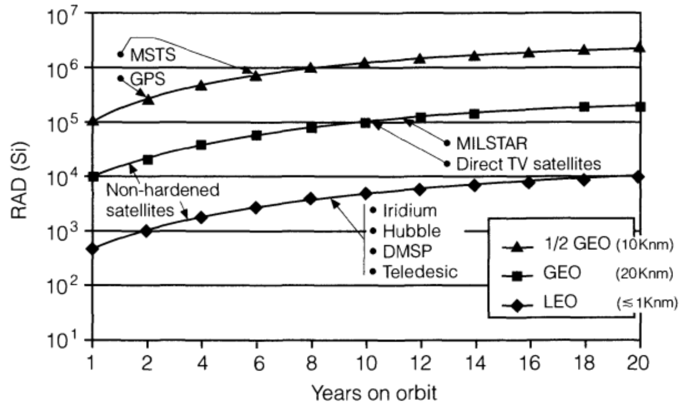


Figure 2.1: TID estimates for different environments, taken from [10]

2.1.2. Drivers

Equation 2.1 and Equation 2.2 show that there are two deciding inputs to single event rate calculation: particle fluxes and cross section. When these vary over time, single event rate varies as well.

Particle Fluxes

The right-hand sides of Equation 2.1 and Equation 2.2 are convolution operations, which means that associativity with scalar multiplication holds. This in turn allows to derive the relationship that by increasing the amount of particles in the environment by a scalar results in an increase of single event rate by the same scalar, under the assumption that all particle energies are affected equally. This is regardless of integral or differential fluxes being considered, because these are related linearly as well. So mathematically speaking:

$$R \approx \frac{d\Phi}{dE} \approx d\Phi(E) \tag{2.3}$$

This relationship will later be important when estimating the effects of space weather on single event rate. Space weather influences particle environments, and if a space weather influence on particle quantities can be estimated, that same estimation will hold for the influence of space weather on single event rate.

Cross Section

Cross-section of devices is influenced by temperature and degradation, which in turn can be caused by TID and aging. Changes in cross-section do not have an influence on single event rate that is as straightforward as that of changes in flux. Mathematically, a change in cross section by a scalar results in a change in event rate by the same scalar, but again only assuming cross section is changed equally over all energies. A reduction in onset threshold, being the energy at which a device begins to respond, results in a change in event rate that depends on the particle spectrum.

Increased temperatures are generally believed to increase cross section. To determine single event sensitivity with very insensitive devices it has early on been proposed to perform characterization at elevated temperatures [6]. Also, this tendency is generalizable for any type of device. Maximum cross section however does not change with temperature, because it is limited by device geometry [47].

For judging the effects of TID with devices aboard the Sentinel 2 satellites, the quantity of TID has to be known. Messenger and Ash provides an estimate of radiation dose received by satellites in different environments, shown in Figure 2.1. For the Sentinel 2 satellites, it can be estimated that they will reach 1 krad of equivalent dose after 2 years and will have received approximately 5 krad by end of mission (7.25 years duration). It has to be kept in mind that these values differ between devices due to device characteristics and shielding, so these numbers are a very rough estimate. [10, 37]

Many different consequences of TID on electronic devices' single event sensitivity have been reported in literature, so giving a clear relationship is hardly possible. There exist differences between different device types, device implementations and device operation.

Messenger and Ash mention that, because TID is in any case deteriorating a device, it generally results in an increase in single event rate. However, Messenger and Ash acknowledges, that there is yet little theoretical knowledge on the issue and also mentions conflicting experimental results. [10]

Salvy et al. on the other hand report that after subjecting devices to more than 100 krad of equivalent Si dose, there were no changes in the single event sensitivity of a tested type of analog-digital converter, digital-analog converter, flash memory and SRAM, neither with single-bit upsets nor with multiple-bit upsets.

Campbell and Stapor reports a cross section increase for different types of SRAM by a factor of 2 for the radiation doses accumulated at Sentinel 2 End of Life (EOL).

Schwank et al. experimentally find a general increase in single event rate after subjecting SRAMs to TID. The magnitude of increase depends on the static memory pattern that was written to the device before accumulating TID. Their conclusion is that for a true worst-case estimation of single event rate, both worst temperature and worst dose of the system under consideration must be taken into account.

As can be seen from the previous foray into literature, there is no clear TID dependence for single event rate, especially when considering radiation exposure at different temperatures with dynamic memory patterns. Most experiments do show an increase in cross section, which is why this behaviour is assumed for the devices of S2A+B. Aside from this, it is not disputed that TID results in an increase in stuck bits in memories [10, 53].

But not only dose results in a cross section change, pure storage is enough. In a study on SRAMs, Rousselin et al. show through modelling that cross section and single event rate as a consequence of aging. This increase is in the order of 25% for the worst configuration considered.

2.2. Space Weather

Space weather, as already defined in chapter 1, actually encompasses more than only the behaviour of space particle environments. However, for the purpose of this thesis, this is the only aspect relevant, so that the following section only looks at this part of space weather.

First, an introduction to space radiation environments in general is given and then it is explained what the relevant space weather effects are for S2A+B and how these space weather effects are linked to variables of solar and magnetic activity. Further, features of Earth's magnetic field and their interaction with space weather are explained.

2.2.1. Space Particle Environments

The LEO particle environment contains three types of particles, being trapped particles, Galactic Cosmic Ray (GCR) and solar particles. In quantitative terms, trapped particles have the by far highest abundance in the orbit of S2A+B. GCRs and solar particles appear in much lower quantity. This is also the reason why in many devices, proton events are much more abundant with LEO satellites than GCR or solar particle events. [22]

Trapped particles

Earth is surrounded by two radiation belts, that trap different types of particles. While the inner belt contains protons and electrons, the outer belt contains only electrons. The belts are separated by a slot region in which under certain conditions particles are trapped as well. The trapping of particle radiation in these regions is due to Earth's magnetic field, that catches particles and forces them to spiral around field lines between North to South over and over again. [21, 22]

Galactic Cosmic Rays

Galactic cosmic rays are particles of high energy that originate outside of the solar system. They appear almost isotropically when reaching Earth and have energies strong enough to enter Earth's magnetic field practically unaffected, which is why they also appear in environments that are normally protected from weak particles through strong fields. Their abundance is modulated by solar activity, since higher solar activity leads to their influx into the solar system being reduced due to the pressure of solar particles moving outwards of the solar system. [22]

Solar particles

Sun emits a permanent stream of particles, called solar wind. The solar wind particles, which are almost only protons and α-particles, are emitted radially into any direction, so they appear also in the vicinity of Earth. Generally, they do not have as much energy as GCRs, which is why only the strongest of them make it deeper into the Earth's magnetic field and therefore to lower altitudes. In a LEO environment, they are more abundant close to the poles, because there the Earth's magnetic field offers less protection from particles. From time to time, spikes in solar wind density occur due to holes in the corona, Sun's outer layer. The increase in particle flux due to such a coronal hole is called a Coronal Mass Ejection (CME) and is often accompanied by variations in Earth's magnetic field strength. [22]

2.2.2. Solar Activity

Solar activity is the main modulator in space weather and undergoes an 11 year cycle. End of 2018 marks the minimum of solar cycle 24, and due to the Sentinel 2 satellites having been launched in 2015 and 2017 [48], it can be approximated that their mission only took place at solar minimum so far.

Next to its 11 year variation, solar activity also shows spontaneous short-term activities: solar flares, CMEs and Co-Rotation Interaction Regions (CIRs). They all are solar particle events.

A solar flare is the spontaneous strong emission of electromagnetic radiation, primarily in X-Ray and UV wavelengths, and heavy-ions. The probability for solar flares is increased at solar maximum and the years following it, while fewer and weaker solar flares occur around solar minimum. A solar flare can occur in conjunction with a Coronal Mass Ejection (CME). [22, 40]

A CME is a the ejection of a large amount of material from Sun's outer layer, the corona. CMEs add to the solar wind, but their particles, mostly protons, have a much higher velocity. CMEs occur more than once per day in times of solar maximum, but only about once per week during solar minimum [40]. CMEs can cause geomagnetic storms, but not all CMEs emitted by Sun are directed towards Earth, and not all of those directed towards Earth are strong enough to have noticeable effects. [13, 21]

A third, but less relevant, type of solar particle events are CIRs. These occur when a fast wave of solar wind encounters a slow wave of solar wind, resulting in compression and acceleration of particles. CIRs are more likely to occur near solar minimum and can also cause geomagnetic storms. [21]

Following a solar particle event, a decrease in GCR fluxes by up to 15 % can be seen, because the increased solar activity due to the solar flare inhibits the flux of GCRs into the solar system. This effect lats for several days and is called Forbush decrease. [3, 13]

There are several proxies to solar activity, for instance the number of sunspots seen from Earth or the solar radio flux, named $F_{10.7}$ after the wavelength at which it is measured. Both are strongly correlated. Figure 2.2 gives the $F_{10.7}$ for solar cycle 24. It can be seen, that the around 2018, the solar cycle is at a minimum with an $F_{10.7}$ of less than 75 solar flux units. In times of solar maximum, $F_{10.7}$ exceeds 150 solar flux units on a regular basis. [13]

Another time scale for the effects of solar activity to vary is the Sun's rotation period. Sun rotates faster at its equator than at its poles, ranging from 24.9 days to 31.5 days, so that on this time scale long lasting features of solar activity, bound to Sun's surface, repeat their particle emission towards Earth. Due to this, a 27-day periodicity can be found in the electron fluxes of the outer radiation belt [46]. Sun's magnetic field also shows a variation on that time scale [46], so that such a variation is also thinkable in the fluxes of inner radiation belt particles. [5]

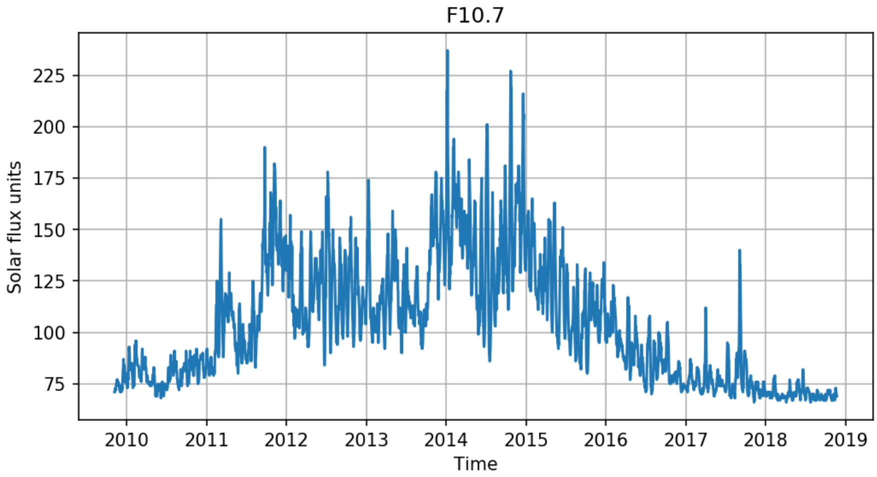
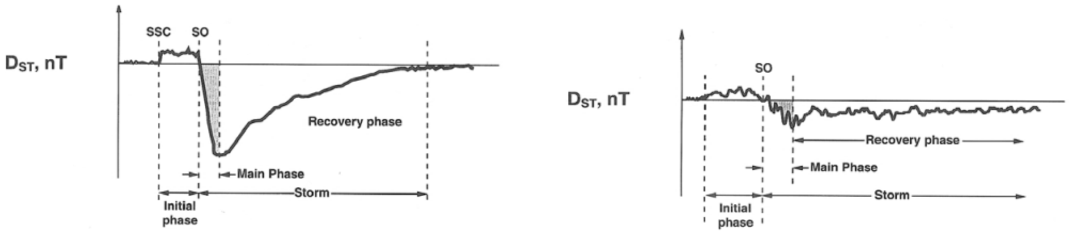


Figure 2.2: Solar $F_{10.7}$ radio flux

2.2.3. Earth's Magnetic Field

Earth's magnetic field changes both in general strength and shape of field, and does so on the long-term and short-term. Long-term changes are caused by internal processes and result in an overall weakening of the field. [11, 18]



(a) Storm caused by CME

(b) Storm caused by CIR

Figure 2.3: Course of DST during a geomagnetic storm, taken from [21]

Strong short-term changes, also referred to as geomagnetic storms, can be caused by solar wind enhancements such as CMEs and CIRs. They cause a quick change in Earth's magnetic field by depositing a large amount of particles in its vicinity. The current flow caused by these particles causes a magnetic field superimposed on Earth's otherwise undisturbed magnetic field, with the effect of CMEs on Earth's magnetic field is usually being stronger than that of CIRs. The strength of a geomagnetic storm can be quantified by its minimum Disturbance Storm Time Index (DST), which gives the strength of the ring current of particles about Earth, and thus the Sun induced change in the Earth's magnetic field. [5, 21]

As can be seen in Figure 2.3, there are three phases to a geomagnetic storm caused by a Coronal Mass Ejection (CME): (1) Sudden storm commencement is the fast increase in DST at the beginning of a geomagnetic storm. It is caused by the manipulation of Earth's magnetic field due to the transient in solar particle flux. (2) The main phase begins when the CME particles arrive at Earth and weaken Earth's magnetic field, resulting in DST to reduce. The strength of a CME can be quantified by its minimum DST. Lastly, (3) recovery begins, with DST returning to normal over a time of hours to few days. Geomagnetic storms due to CIRs are more subtle: there is no instantaneous reaction in DST and the minimum DST reached is generally not as reduced as with CMEs. [21]

2.2.4. Trapped Particles

The inner radiation belt extends from approximately 200 km to 1000 km above Earth's surface, while the outer radiation belt extends from 3 earth radii to geostationary orbit [21, 32]. Therefore, the radiation environment of S2A+B is only affected by the inner belt only, which is why no further consideration of the outer radiation belt is made. Also, electrons are not a common cause to single events [26], so only the protons of the inner belt need to be considered.

The protons in the inner radiation belt accumulate due to two processes: Cosmic Ray Albedo Neutron Decay (CRAND), which is the dominant source of inner belt protons, and the trapping of solar radiation, which is of lesser importance [16].

In CRAND, GCRs enter the Earth's magnetic field and atmosphere and release neutrons in nuclear reactions with atmospheric atoms. The resulting neutrons leave the atmosphere again directly, or indirectly after collisions. In any case, each neutron will decay into a proton, an electron and a neutrino. Depending on velocity and its direction of movement, the resulting proton gets captivated in Earth's inner radiation belt. [16, 21]

As already mentioned, solar activity regulates the flux of GCRs. In times of higher solar activity, there are fewer GCRs reaching Earth and thus fewer protons are created to be trapped in the radiation belts. As a consequence, there is also a link between solar activity and particle fluxes in the inner radiation belt. [21]

Next to that, solar particle events affect the particle fluxes of the inner radiation belt. CMEs can lead to injection of particles into the inner radiation belt, for those CMEs accompanied by little minimal DST, as well as particle loss, with those CMEs accompanied by stronger minimum DST. These effects arises from magnetic field line bending due to geomagnetic storms. It was for example observed, that a strong geomagnetic storm in 2003 completely purged parts of the inner radiation belt's protons and that it took months to recover [15]. However, the deeper a field line resides within the magnetic field, the less likely it is that it can be distorted strongly enough to release particles [50]. Particles in the inner radiation belt diffuse inwards, so that changes in the outer parts will over time also be visible in the inner parts of the inner radiation belt. Further, geomagnetic disturbances, having their cause in solar activity, can cause an accelerated transport of particles through the inner radiation belt. [5, 31]

Despite field line bending being able to free particles, the major particle loss mechanism in the inner belt is the contact of particles with the atmosphere. In times of higher solar activity, the outer atmosphere extends to higher altitudes due to stronger warming, allowing it to slow down larger quantities of trapped particles. It was also found, that CMEs can heat up the thermosphere, one of the outer layers of Earth's atmosphere. With an approximate height of 380 km, the thermosphere is below S2A+B, but it is plausible that a similar effect can be seen at higher altitudes, too. It is therefore thinkable, that CMEs also slightly reduce the particle population of the inner radiation belt in this way. [21, 58]

Therefore, there is a two-fold link between solar activity and inner radiation belt particles: as mentioned before, high solar activity results in less GCRs reaching Earth and thus less protons being created by CRAND, while at the same time, the existing particle population is subject to stronger particle removal because of the atmosphere. As a consequence, the inner radiation belt holds less particles around solar maximum, than it does around solar minimum. This means that the inner radiation belt, which is mainly determining S2A+B's radiation environment, is close to its maximum activity currently (2019) and therefore, the radiation environment seen by S2A+B is harshest. [21]

Particles in the inner radiation belt are subject to the east-west effect. Because of how the magnetic field of Earth is shaped, particles are more likely to come from west than from right [35]. Modelling the inner radiation belt is essential to predicting single event rates in LEO. Such models are created from space-based particle measurement data [39].

2.2.5. South Atlantic Anomaly

An important feature of the inner radiation belt is the South-Atlantic Anomaly (SAA). Because the magnetic field of Earth is not a perfect dipole and does not perfectly align with the shape of Earth, there is a region in which the Earth's magnetic field is weaker. Particles are repelled by magnetic fields, so that in a weak region, which means that particles of the inner radiation belt can approach Earth's surface especially close in that weak region, resulting in particularly high particle fluxes [18].

Figure 2.4 shows the Earth's magnetic field strength, as given by the International Geomagnetic Reference Field (IGRF) for the altitude of S2A+B's orbit. There, the SAA can be seen centering above south Brazil, close to the Atlantic ocean, and extending from the Pacific coast of South America to almost the Atlantic coast of South Africa. For LEO satellite missions, and therefore S2A+B, the SAA is the biggest persistent threat in terms of single events because of its comparatively high proton fluxes. However, the definition of SAA position by magnetic field depression is only one of three possibilities. It can also be found by the position of maximum particle fluxes and by the position at which particles are mirrored in a dipole approximation of Earth's magnetic field [11].

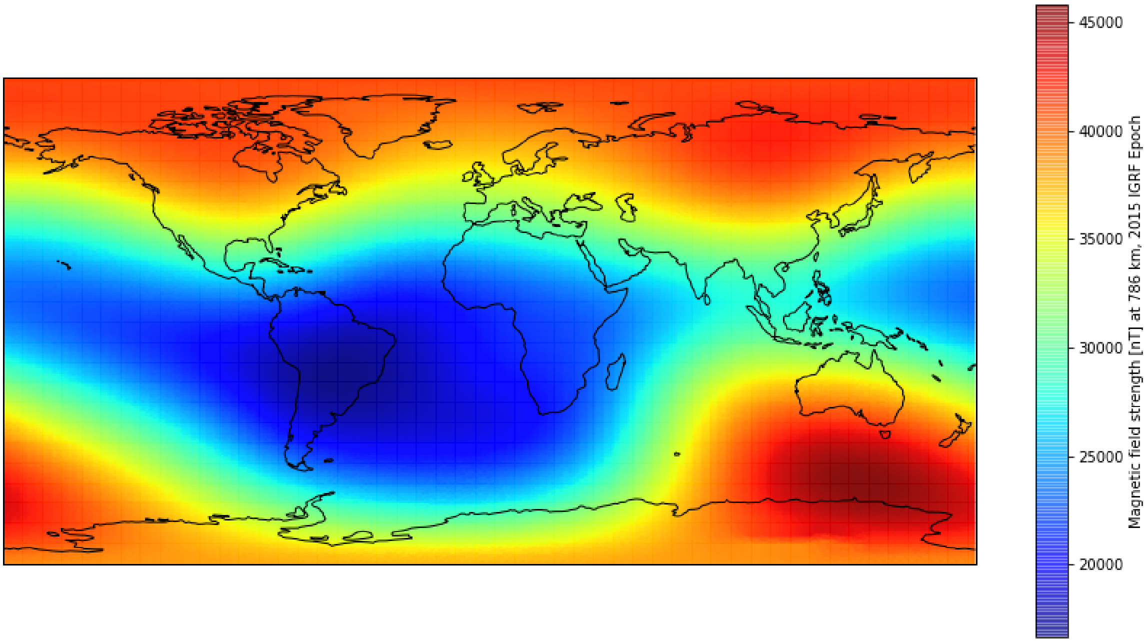


Figure 2.4: Magnetic field strength at 786 km

The solar F10.7 activity index is known to be anti-correlated to the particle flux specifically in the SAA, because solar activity modulates the flux of GCRs into the solar system which are the origin of the particles in the SAA [54]. Another anti-correlation peak is reported between present particle flux and F10.7 approximately 685 days in the past. This is attributed to the life-time of a particle in the inner radiation belt, so that when the particle flux increase at one point in time, it will decrease again 685 days later. [42]

Next, both the activity of the SAA and its spatial extent react to geomagnetic storms. It is found that both the area covered by the SAA, as obtained from particle count measurements, and the maximum particle count rate within the SAA are directly linked to DST. When DST decreases, both area and maximum particle count rate decrease instantaneously. These decreases are both found to be within a magnitude of 10-15% and there is a good correlation between SAA size and peak proton flux. The changes in peak flux and area recover over the same time span as changes in DST. [46]

Also, there is a semi-annual variation in the flux of particles in the SAA, with an amplitude of about

10 % of the average proton flux rate. According to [54], the theory behind this component is not entirely established yet, but the same semi-annual components can be seen in GCR count rates, that give rise to SAA particles through CRAND, and in the variation of DST. In addition, proton fluxes in the SAA were found to vary with the Sun's rotation and with an annual period. However, the effect of Solar cycle variation is much weaker than the semi-annual variation. The magnitude of annual proton flux variation is not specified any further by Schaefer et al., but it can be inferred from the reported figures that it is significantly weaker than the semi-annual cycle. [54]

For satellite data analysis, Schaefer et al. [54] determine a SAA signal of particle counts by using those particle counts, that have been made within:

$$SAA_{lon} = [-115\deg \text{ to } 30\deg] \tag{2.4}$$

$$SAA_{lat} = [-60\deg \text{ to } 10\deg] \tag{2.5}$$

For the later analysis of single event counters, this definition can be used to differentiate SAA events from non-SAA events.

The SAA is known to be drifting towards north-west. Drift rates documented for this phenomenon vary strongly. Westward drifts in the range of 0.28 to 0.43 degrees per year and northward drifts in the range of 0.08 to 0.18 degrees per year are reported [54]. For accurate prediction of particle fluxes, models need to know where the SAA is [9]. Therefore, permanent revision of its position within models is needed. Predictions for the year 2100 come to the conclusion that the SAA will increase strongly in size due to the decline in Earth's magnetic field strength. Measurements of magnetic field strength in the current SAA find a field strength decrease of about $-25\ nT\dot{y}^{-1}$ and even higher rates elsewhere on Earth. A second anomalous weak spot, besides the SAA, could form as well by 2100, south west of Africa. [11]

2.3. Sentinel 2

From the specifics of the satellites and their mission, it is derived what influences to single event rate there are, so that finally an extensive list of influences can be defined. This allows to derive the setup of the single event counter analysis.

2.3.1. Purpose

The Sentinel 2 satellites are part of Global Monitoring for Environment and Security (GMES), an earth observation program funded by the European Union. GMES has the overarching goals of providing information for environmental lawmaking and for valorization by private enterprises. This makes many sub-services necessary, such as atmosphere monitoring, emergency management, land and maritime monitoring, climate change monitoring and security services. The GMES space segment comprises several satellites using different observation methods and qualities, such as multi-spectral optical sensors, altimeters, thermal radiometers and radar. [36]

2.3.2. Satellites and Orbit

The Sentinel 2 formation consists of the identical satellites Sentinel 2A and Sentinel 2B. Sentinel 2A marked the begin of the formation and was launched in 2015. Sentinel 2B followed in 2017. Their purpose is to provide imagery of Earth's land mass at high resolution in 13 spectral bands at a swath width of 290 km. This includes all continental land mass, but also island larger than $100\ km^2$ and any island belonging to the European Union as well as inland waters. The Sentinel 2 satellites contribute to GMES goals in detecting land cover changes. Overall, the two satellites produce 1.6 TByte of payload data daily, which is transmitted to ground either in one of two daily transmission windows to one of the four ground stations or via optical communication through relay satellites in geostationary orbit. [37]

The satellites are flying in a sun-synchronous orbit at 786 km mean altitude and 98° inclination and are designed for a mission duration of 7.25 years. The orbit allows the satellite's payloads to achieve global coverage within 10 days, and therefore, its ground track repeats after 10 days. This is the representative ground track duration of the satellites. In their orbit, there is a distance of half an orbital period between the satellites. With respect to their ground tracks, this translates to a delay in points visited of 5 days, which means that the combination of the two satellites achieves global coverage within 5 days. For the continuous provision of services offered by Sentinel 2, the satellites Sentinel 2C and 2D shall take over the tasks of S2A+B, once S2A+B reach their lifetime. [36, 37]

Given the mean orbital height of 786 km, here approximated from the mean altitude provided in [37] and a spherical Earth of radius $6371\ km$ [40], the orbital period of the satellites is [45]:

$$T = 2 \cdot \pi \cdot \sqrt{\frac{a^3}{\mu_{Earth}}} = 6025.7\ s = 100.43\ min \tag{2.6}$$

Because the orbit has a high inclination, each revolution of the satellite around Earth consists of a sunlit part and an eclipsed part. This impacts the general structure temperature (although isolation is provided by reflecting coating), because it rises through solar irradiation during the sunlit phase and falls during eclipse by radiative heat transfer.

From a technical point of view, the two satellites are identical. With respect to single events, this means that each component's shielding is identical as well as the radiation environment seen by each of the satellites, except for the phase delay of 5 days. Further, the satellites are operated in the same way, so that their attitudes do not differ and the level of effective shielding is the same throughout the mission.

2.4. Implications

The previous chapters have shown that timely varying single event rates depend on particle flux variations and cross section variations. In the orbit of S2A+B, the SAA is the the largest threat in terms of proton events, and GCRs and solar particle events are the largest threat in terms of heavy-ion events. The flux of heavy ions is however much lower than that of those protons in the SAA and heavy-ions cause less events than SAA protons. Therefore, a single event counter for a proton sensitive device in LEO will document radiation environment changes more sensitively than one that is sensitive to heavy ions, and is therefore the better choice for studying links to space weather, respectively for meeting the objectives of this research.

It was shown, that the flux of protons in the SAA is controlled by space weather. Solar activity defines the semi-annual and long-term development of particle fluxes in the SAA, and geomagnetic storms, caused by CMEs and CIRs can cause abrupt changes to the particle population. The link between particle fluxes and space weather must be similar for single event counters: if space weather leads to high proton fluxes, a higher event rate must be present and vice-versa. But, because sensitive volumina are small, electric devices prone to single event do not see particles clearly, but only the ones that accidentally strike them. As a consequence, single event counters dependent on coincidence, which in turn means that either much noise will be found in their signals or there will be too few events to make an observation at all. Yet, the underlying component of an event-rich counter must be witness of space weather.

Because outside of the SAA, proton fluxes are negligible in the orbit of S2A+B, in rough approximation, single events with proton sensitive devices occur exclusively within the SAA. It was also shown previously, that the SAA is drifting. Therefore, the long-term trend of single event positions is expected to follow the long-term movement of the SAA, again distorted by noise. By studying the underlying component, another link between single event counters and space weather, as well as internal processes of Earth's magnetic field, can be established.

The previous chapter also explored, that with respect to particle rates in the SAA, the driving phenomena (solar activity, geomagnetic storms) can be described by the two variables $F_{10.7}$ and DST. Therefore, there must be a link between these variables and the space weather component in single event signals.

On the basis of the previous theoretical considerations, an extensive list of influence parameters to single event rate with S2A+B can be made. These shall be studied for their significance towards single event rate. All of these influences have periodic behavior, because the operation of the S2A+B satellites follows a 10d-pattern and because of space weather's overarching modulation through the periodic solar cycle. Given the short mission time of the satellites, compared to the duration of the solar cycle, not all of the signals are periodic with the available data.

The list of periodic influences to single event rate contains:

- **Orbital period:** The orbital period of 6026 seconds is possibly visible in the single event rate signal.

  With those orbits passing the SAA, the satellites repetitively pass regions of high and low particle fluxes on this time scale, thus the same should apply to single events. Since not all revolutions pass the SAA, this effect does not strictly apply to all orbital periods.

  Also, the components could see different event rates at eclipse than during the time spent sunlit. It is thinkable, that due to the difference in atmospheric density between day and night side, a difference in the radiation environment exists and therefore one in single event rate. Another way by which eclipsing could influence event rates is by causing the temperature of a component to vary, in consequence changing cross section and event rate.

- **Time spent in SAA:** In the 10d representative ground track, there are several passes of the SAA with differing path length traversed through the SAA. Each time the SAA is passed, single event rate is expected to increase for proton sensitive devices, resulting in visibility in the singe event signal for the duration of the SAA pass.

- **Semi-diurnal component:** Another way to look at the orbit is, that the satellite's are flying in a trajectory about Earth's barycenter, while Earth's radiation belts, bound to Earth itself, are rotating through the orbit of the satellites with a period of one day. Because the orbit, inclined by 98°, passes both sunlit and eclipsed SAA, the diurnal rotation of Earth leads to semi-diurnal revisits of the SAA. Therefore, there could be a 12 hours component in the single event signal.

- **Solar rotation:** The particle flux emitted by Sun is not homogeneous throughout its surface, so that during the 25-31 day solar rotation period changes in particle flux towards Earth might occur. For example, when a long-lasting coronal holes is repeatedly rotated towards Earth, this is the period by which the radiation environment changes, and thus, the period by which single event rate changes. But as already pointed out, this influence is of dubious relevance for particles in the SAA, because strong magnetic field changes are necessary to inject or remove particles.

- **Semi-annual SAA variation:** As explained earlier, the SAA undergoes a semi-annual variation in its strength, that leads to higher particle flux rates around the equinoxes than around the solstices and lower fluxes around the solstices (July and December). This behaviour might be visible in single event rate.

- **Annual period:** In one year, the distance between Sun and Earth undergoes one period.

  This variation in distance might influence atmospheric density through varying irradiation, comparable to the seasons experienced on Earth's surface. With the atmosphere being the main sink for inner radiation belt particles, thus modulating the particle fluxes in the SAA, this is possible to appear in single event rate.

  Another possibility is, that this periodicity results in varying temperatures of a device throughout one year. That in turn changes cross-section and single event rate.

- **Solar cycle:** During the 11 year solar cycle, particle content of the inner radiation belt goes from minimum to maximum and back again, making it a clear candidate for what should be visible in the single event signal. There is a energy dependent delay in particle fluxes, so that the SAA reaches its peak activity later than the solar cycle [42].

Given that S2A, the earlier satellite, just started in 2015, and the available data at maximum goes until September 2018, there is not enough time covered for the data to reflect on an entire solar cycle, because it is much longer. Also, annual and semi-annual variations will only undergo 2, respectively 4, cycles in the available time frame.

2.5. Analysis Definition

Previous theoretical study is now synthesized to obtain the concrete analysis steps for the single event counters. Because of the available data and the different time scales of influences that could be present, different paradigms have to be applied:

- **Time-Series Analysis**

  The first analysis to be done, is that of long-term single event rate and single event position development. Because no or only few periods of these long-term influences have been seen by the satellites, a visual inspection of single event rate appears more worthwhile than spectral analysis. The goal is to find and explain connections between space weather and single event counters. Noise is expected to be present in the single event signal, so it will have to be removed for this analysis. Additionally, the correlation between the filtered single event signal and solar and magnetic activity proxies will be analyzed.

  It was also mentioned that there should be an association between SAA drift and single event positions, because the SAA moves over longer time scales. To clarify this, single event positions are also subjected to a long-term behaviour analysis, which will deepen the understanding between single event counters and space weather.

- **Spectral Analysis**

  Short-term behavior, despite being inhibited by noise, supposedly can be studied through spectral analysis because of the many times a short-term repetitive phenomenon has appeared.

The next chapter establishes the analysis' infrastructure to meet the goals of the analysis itself and for data pre-processing. Finally, the pre-processing is documented and a choice on the single event counters to be analyzed is made, because not all of them will be sensitive to protons and not all of them will provide the same quality of data. The subsequent chapter gives the time domain analysis and the chapter thereafter gives the frequency domain analysis, which determines what periodicities are present and if they are relevant to single event rate. For both analyses, method, implementation, results and discussion are documented.

3

Python for Data Analysis

3.1. Python

For meaningful results and a well-structured analysis, a closer consideration of implementation methods and analysis infrastructure is needed. After a trade-off in [66], a choice was made for Python as platform for the implementation of the analysis. The main reasons for choosing Python were it being open-source and its large ecosystem, that hold libraries available for plentiful tasks. Other advantages of Python in research are its native support of scientific computing, extensibility, support of many input and output options and strength in automating tasks [67].

Using Python, respectively open source solutions in research, allows to reduce the effort in implementing the methodology and to put more emphasis on interpreting the results. So far, redundant implementations of simulation and analysis tools caused unnecessary workload, which is avoided through open source software becoming more widespread [72]. Yet, it comes at the cost of having less reliable tools than when using proprietary software. The advantage of being able to review the source code of an open source tool only comes to play, when it is actually done, which is not necessarily the case.

3.2. Libraries

The core functions needed when working with large amounts of data are data handling, data processing, data storage and data visualization. In each of these perimeters, the needs of the data at hand need to be considered.

- Data Handling and Storage: Both aspects must respect the time series nature of the data and have to be able to deal with missing data and uneven sampling. Because large amounts of data are worked with, processing speed, storage speed and retrieval speed are also important.
- Data Processing and Modelling: The time series analysis will require the application of filters, while the spectral analysis requires periodogram functions.
- Data Visualization: The data visualization solution for the thesis needs to provide general plotting functions, and additionally shall provide functions for geospatial plotting.

The following sections introduce Python libraries these functions which are available through the Python Package Index [1]. Purpose of this section is to have a go-to Python library during the analysis of the data and to screen it for the desired functionality. Next to that, the section also helps to define the subsequent data analysis more clearly, because it shows what analyses can be done by which library and what effort.

[1] `www.pypi.org`, accessed 03.11.2018

3.2.1. Data Handling

In the following, Python modules for handling of data are introduced, which are Pandas, Numpy and Artic.

Pandas (short for panel data) is a Python library which provides functionality for working with structured data sets, like tabular data or time series data. As such, is a natural match to Comma-Separated Values (CSV) data and for single event counter data. Pandas provides special time series functionality, such as automatic alignment time series data from different sources, shifting, resampling, slicing, handling of missing data, and grouping and conversion to manifold data structures. It performs faster than R's data.frame class, which substantiates its choice for the infrastructure of this thesis. For these reasons, it is chosen as the main data handling library in the following research. [29, 34, 77]

Numpy is the Python ecosystem's library for mathematical functions and provides corresponding data structures as well. Thus, it is used in both data handling and data analysis. Its mathematical functions cover linear algebra, matrix operations, logic functions, polynomials and complex numbers. Numpy offers multi-dimensional data structures and is therefore suitable for working with matrices and vectors, as which time series can also be depicted. Numpy comes with functions for statistics, which will be useful in the further research. It also allows to work with missing data through the masked array data type, where missing values can be masked out for the mathematical operations applied to the data. The masking is propagated automatically, so that the result is also masked as far as it depends on missing data. [44, 74]

Arctic is a Python library for time series and tick series data storage on the basis of MongoDB. It has its basis in finance and allows for rapid data storage and retrieval, good compression of stored data and is capable to work directly with Pandas' dataframes. For time series data, arctic offers specialized storage methods to account for the nature of the data and how it its supposed to be retrieved. The VersionStore allows to keep track of multiple versions of a time series and to reproduce its development, ChunkStore specializes in fast data retrieval by storing it in pre-defined chunks (for example daily chunks) and TickStore is designed to work with continuously spaced time series data. Yet, Arctic is only the interface, and actual data storage takes place on MongoDB. To work with Arctic, a MongoDB database server is needed. A simple way of using MongoDB is by starting it from a portable application [2], which also ensures that the whole database can be ported for future work on the project. [63]

3.2.2. Data Analysis and Modeling

Relevant data analysis and modeling libraries are Numpy, which was already introduced among data handling solutions, Scipy, Statsmodels and Astropy.

Scipy is a library for scientific computing, built on Numpy. Among others, it offers functionality for interpolation, optimization, integration, linear algebra and statistics. Dedicated functions for statistics with missing data are provided as well, making it relevant for the research at hand. [1, 75]

The library Statsmodels is a library for statistics and also provides functions for time series analysis, modelling and filtering. Its focus is rather on statistical inference than on prediction. Given this project is based off time series data, statsmodels functionality could become relevant with the research. [68, 76]

Astropy provides functions for the analysis of astronomical data. Mention-worthy functions are different spectral analysis methods, such as Lomb-Scargle and Box-Fit spectral analysis. These methods are robust to missing data, which is why might become useful in the given research. [65, 73]

Another library for usage in data analysis is PySolar. PySolar is used to calculate the relative position of sun to a point on or above Earth, given time and position. This can be used to determine if the Sentinel satellites were under solar illumination at a certain time, or if they were eclipsed. [78]

3.2.3. Data Visualization

Data visualization again have to reflect on the characteristics of the given data. This results in looking into the libraries Matplotlib, Cartopy, Seaborn and Plot.ly

[2] `https://github.com/lightchpa/MongoDBPortable/releases/`, accessed 03.11.2018

Matplotlib is the most widespread Python library for creating publication-ready plots. Matplotlib offers an object oriented interface and a functional interface called pyplot, which is how Matplotlib is used mostly and also similar in the way Matlab handles plotting. Matplotlib's outputs can be made interactive through additional libraries, such as mplcursor, which provides hovering functions to graphs. [24]

Cartopy is a library that was developed by the UK Met Office and is built on matplotlib. It is used for plotting with different projections of the Earth, while automatically transforming the inputs to suit the respective projection. [2]

Seaborn is a high-level API to matplotlib that offers several pre-defined plotting methods for statistical analyses. It has higher pandas integration than matplotlib. [62]

Plot.ly is a library is particularly used for its 3D plots, which are useful for visualizing spatio-temporal point data. [43]

Finally, missingno [69] allows to visualize missing data and is based on Pandas, too.

3.2.4. Documentation

The Python ecosystem offers libraries for interactive data analysis, support in documenting the analysis and automated source code documentation.

IPython is an interactive console for Python, that allows to piece-wise execute code and to exploratively work with data. The addition IPyParallel allows to run IPython on a cluster. Therefore, also normal execution of Python scripts can be moved to a cluster or into parallel processes on the local machine, which allows for concurrency and thus faster execution. [80]

Jupyter, built upon IPython, integrates code-execution and documentation. In a Jupyter notebook, both the outputs of a computation, such as numbers and plots are shown, but also user defined text with formatting and support for mathematical formulas. This exceeds the possibilities offered by simple commenting and therefore is a good fit for scientific analyses. Jupyter notebooks can be exported to LATEXor published on their own.

Sphinx[3] is one of several source code documentation libraries that extract comments in the form of structured text from source code and automatically provide a documentation in the form of HTML or LATEX. Sphinx can be used to document the Application Programming Interface (API) of Python functions created in the scope of this research.

[3] `http://www.sphinx-doc.org/en/master/`, accessed 02.05.2019

4

Data Preparation and Selection

The single event data pack being studied in this thesis come from the housekeeping data of the satellites and contains their position, recorded in latitude and longitude, and 24 single event counters. 11 of these counters are Sentinel 2B counters, while 13 of the counters stem from Sentinel 2A. So for each Sentinel 2B counter, there is the matching Sentinel 2A counter, and additionally, 2 Sentinel 2A-only counters are part of the data pack. All data is provided in CSV files, that amount to 18 GByte in total. Each of the CSV files bundles the readings of several counters for the same time span.

Since the Sentinel 2 single event data pack exclusively contains single event data from memories, all of these have been found by a scrubber, that repetitively checks the integrity of several memory cells and corrects them if necessary. At the same time, a counter is increased that gives the total number of single events. The scrubbing interval is not the same for all counters and reduces the data's quality. Therefore, its impact on the analysis has to be considered. Also, a counter can be reset or can overrun, once the maximum recordable number of events has been recorded.

In this chapter, all counters are pre-processed, which means that event occurrences are extracted from satellite housekeeping data. Then, a set of counters shall be selected for further study.

4.1. Data Pre-Processing

The following section describes the pre-processing of data from the S2A+B data pack, using the previously introduced Python modules. The implementation is given in listing A.1. There is also pre-processing needed for supplementary data that will be used in some of the analyses, which is however not described here, but along with the corresponding analysis.

S2A+B position data needs to be prepared first, since an event is characterized by time and position. From the event counters, an event is found by its time occurrence, and for each event, the matching position of the satellite needs to be retrieved for the time of event detection.

When referring to position data, only latitude and longitude are meant. The S2A+B data pack comes without altitude, respectively distance to Earth's center of gravity. Where needed, altitude shall be assumed to be at constant 786 km above Earth's surface [48], which itself is assumed a perfect sphere.

4.1.1. General Pre-Processing

Both position and event data come in the same format, which is described in the following. Each file in the data pack consists of a header (example given in listing 4.1) and the data given line-wise (example given in listing 4.2). The following steps of pre-processing are the same for position and event counter and follow the process shown in Figure 4.1.

Listing 4.1: Sample header from CSV files of S2A+B data pack

```
1  # Exported Parameter(s) from GRAINS
2  # Parameter List:
3  MST00495 [SENTINEL2A],MST00495 [SENTINEL2B],MST00599 [SENTINEL2A],MST00599
      [SENTINEL2B],MST00601 [SENTINEL2A],MST00601 [SENTINEL2B],MST00505 [
     SENTINEL2A],MST00505 [SENTINEL2B]
4  # Time Window Start (Time Date (S2K DOY)):
5  2017.267.00.00.00
6  # Time Window End (Time Date (S2K DOY)):
7  2017.358.00.00.00
8  # Number of samples (per parameter and total):
9  1572381,1561863,1572366,1561830,1572366,1561830,0,0,9402636
10 # Number of parameters:
11 8
12 # Name,Unit,Description,Data,Type,Max,Min,Avg,StdDev
13 MST00495,null,MSSSINGLBITERR,UNSIGNED_INT
     ,1809.000000,1353.000000,1579.500000,129.350461
14 MST00495,null,MSSSINGLBITERR,UNSIGNED_INT
     ,1469.000000,1001.000000,1231.700000,131.170524
15 MST00599,null,FMM0_SBE_CNT,UNSIGNED_INT
     ,12.000000,10.000000,10.400000,0.778353
16 MST00599,null,FMM0_SBE_CNT,UNSIGNED_INT
     ,12.000000,10.000000,11.400000,0.925828
17 MST00601,null,FMM1_SBE_CNT,UNSIGNED_INT
     ,500.000000,356.000000,362.200000,27.348074
18 MST00601,null,FMM1_SBE_CNT,UNSIGNED_INT
     ,12.000000,10.000000,11.400000,0.925828
19 MST00505,null,PDICHSAERRCNT,UNSIGNED_INT, , , , ,
20 ST00505,null,PDICHSAERRCNT,UNSIGNED_INT, , , , ,
21 # DATE TIME,MST00495 [SENTINEL2A],MST00495 [SENTINEL2B],MST00599 [
     SENTINEL2A],MST00599 [SENTINEL2B],MST00601 [SENTINEL2A],MST00601 [
     SENTINEL2B],MST00505 [SENTINEL2A],MST00505 [SENTINEL2B]
```

The headers of the CSV files are needed for formatting the data. They provide general information about the file's content:

- Line 3 gives a list of all time series in the respective CSV file and the satellite from which they stem.
- Lines 5 and 7 provide start and stop of the time series given in the respective file. Since each data point is time stamped individually, the start and stop can be reproduced from the data, making this information irrelevant for further processing.
- Line 9 gives the number of samples for all the time series in the file. This information can be used later to check the processed data for its inclusion of all raw data.
- Line 11 gives the number of parameters and the lines following line 13 provide their description. As such, they provide general metrics of the time series (minimum, maximum, standard deviation) and the data type. In the end, all data considered here is numeric, and numeric data handling is done automatically by pandas. Therefore, no consideration needs to be made of the information given in these lines.
- In the line before the actual data starts, again all time series and their origin are given.

Thus, the first relevant information to be extracted from the header are the names of the time series and the satellite from which they originate. It can be seen that header length depends on the number of time series in the file, making it necessary to calculate the starting line of time series data before reading.

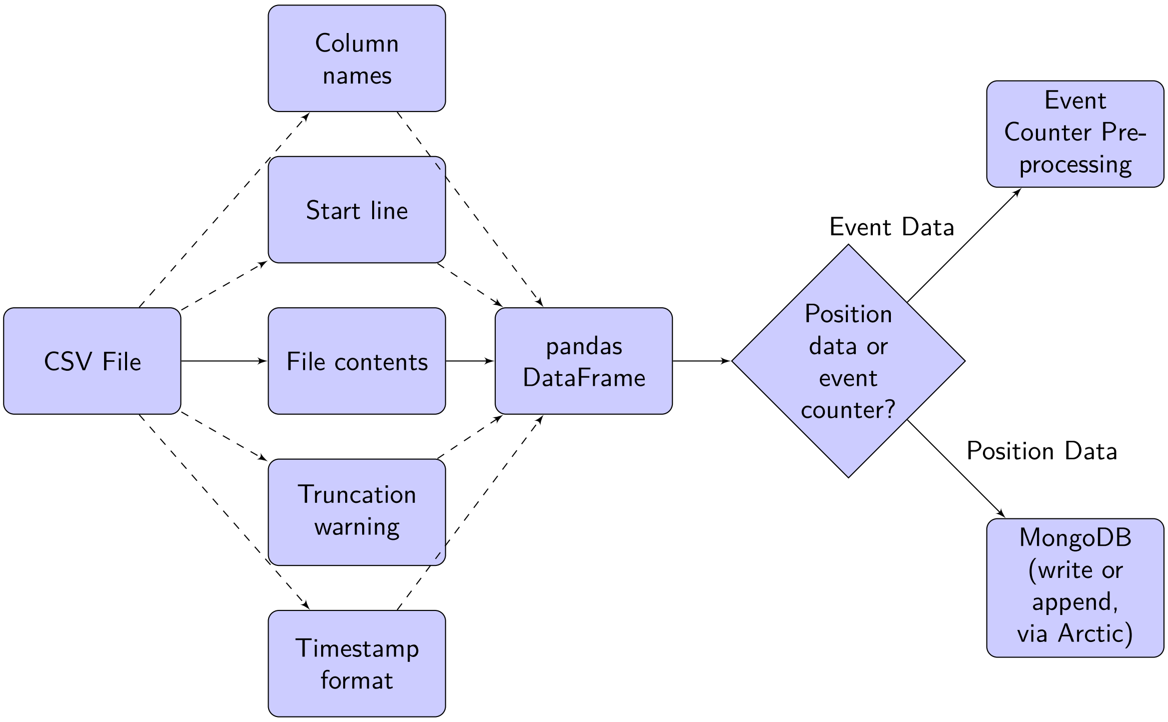


Figure 4.1: Data pre-processing

Further, the software used for the creation of S2A+B data pack limits file size to 10 million lines, which not necessarily contains all the data saved with the respective counter. When the user tries to export more than 10 million lines of data, the software adds a truncation warning at the end of the file. The implications are twofold: in pre-processing, the user shall be made aware of the existence of such warning, to know that further data might be available, and, to allow for further processing of the data, the warning needs to be removed because it does not fit into the scheme of CSV data.

Lastly, inspection of the files showed that the format of the time stamp differs depending on the accuracy by which the file is saved. It can be either seconds-precision or milliseconds-precision, which requires to use different formatters in time stamp harmonization.

Listing 4.2: Sample lines from CSV files of S2A+B data pack

```
2017.175.00.00.00,-24.2735812000000000,11.7467172000000000, ,
2017.175.00.00.01,-24.2870180000000000,11.8059664000000000, ,
2017.175.00.00.02,-24.3004589000000000,11.8652152000000000, ,
2017.175.00.00.03, , ,42,
2017.175.00.00.03,-24.3139038000000000,11.9244635000000000, ,
2017.175.00.00.04,-24.3273527000000000,11.9837113000000000, ,
2017.175.00.00.05,-24.3408057000000000,12.0429587000000000, ,
```

With this information available, a Pandas data frame can be created. The implementation of this is part of listing A.1 and the function provided for this task is described in the API documentation of section A.6. This preparation step has to be done individually for each time series in a file, given that not each time series has information at every time stamp recorded, as can be seen in listing 4.2. In case there is no information available at a certain time stamp, a space is given for the time series. These spaces need to be removed, leaving only the actual recordings. All files are read individually, before data belonging to the same time series is concatenated.

This finalizes pre-processing for position data, which is next saved in MongoDB through Arctic, where it is sorted and stored. With event data however, further steps need to be taken.

4.1.2. Event Counter Pre-Processing

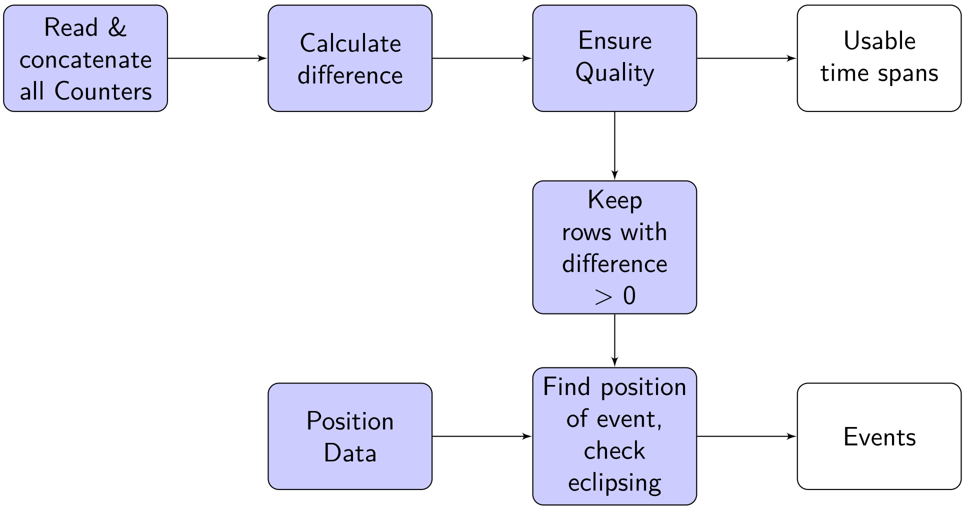

Read & concatenate all Counters
Calculate difference
Ensure Quality
Usable time spans
Keep rows with difference > 0
Position Data
Find position of event, check eclipsing
Events

Figure 4.2: Event data preparation

The process of event counter pre-processing is shown in Figure 4.2. It is implemented in listing **??** and documented in section A.6. After reading all data belonging to one counter and concatenating it, the counter needs to be differentiated. As a result, only counter changes remain, which are either events (counter increases) or counter resets (counter decreases).

Before counter decreases are removed, the quality of the data is checked by counting the number of recordings per time. With the counters at hand, the time distance between consecutive recordings is less than 60 seconds for nominal recordings, and higher than 60 seconds for data outages. Therefore, only those times where at least one sample per 60 seconds is available shall be considered good data quality. The begin and end of time spans of usable data is then written to file.

Next, only recordings with counter increases of more than 0 are kept in the differentiated time series. This removes counter resets as well as recordings that were quiet and did not contain an event occurrence. All that remains are those recordings that registered single or multiple events.

For each of these, the corresponding event position has to be read from the previously prepared position data. This requires the entire data to be retrieved from the database, which was possible in this analysis, but in the future can exceed system memory. Therefore, chunk-wise position data retrieval might become necessary. While matching event time and satellite position, the quality of the event position data is monitored by calculating the time delay between time of event occurrence and time of closest position recording. This could reduce the position data quality in case an event was recorded at time t, but position data is only available for times $t-k$ or $t+k$, making interpolation necessary. However, there was no event in any of the counters for which no position data at exactly the same time was available, so that no further consideration of this issue was needed.

Also, it is determined on the basis of its time stamp and position, the event has occurred while the satellite was in eclipse or while the satellite was sunlit. The procedure and necessity for this is described later in chapter 6. Lastly, events and their positions are written to CSV file.

Using the event file and the usable time spans file the counters can be reconstructed. Thus, there is no longer the need to process the large input files. The described procedure is repeated for each counter and both satellites.

4.1.3. Preparation Results

The data preparation has produced two results: the position data of the two satellites, stored in MongoDB using Arctic, and the event counters, stored by time of event occurrence and list of covered time spans.

The resulting event data is summarized in Table 4.1. In this scope, Table 4.1 differentiates between total counter increases, which is the number of events including multiple events, and individual counter increases, which counts multiple events as one event occurrence. Further, the event positions of all counters are plotted in Figure 4.5, Figure 4.6, Figure A.1, Figure A.2, Figure A.3 and Figure A.4.

In terms of event rate for both individual events and total events, counters CST01830 and CST02259 agree rather well for Sentinel 2A and 2B. For CST01835 and CST02261, Sentinel 2B shows an event rate that is around 30% higher than that of Sentinel 2A. Because of the low event count overall, these counters seem to be susceptible only to very strong particles.

4.2. Data Selection

Now that all counter have been pre-processed, a focus shall be set for the further analyses. This means, that one or multiple counters are selected based on:

4.2.1. Rationale

The criteria in selecting the best counter for analysis are:

- Coverage: data coverage (in days) is a general metric of data quality. The counter most worthwhile to analyze has the little gaps and good coverage. Also, better coverage is desirable, because it means that less provisions have to be taken for dealing with missing values and uneven sampling rate.
- Plausibility: based on experience, it can be analyzed from plots if the counters behave in an implausible manner. Since the goal of the thesis is to analyze normal single event counters, and not to analyze the behaviour of outliers, a counter shall be selected that complies with the general expectations towards a single event counter.
- Event rate: event rate is a proxy for information content. The higher the event rate of a counter is, the more it is likely that the counter has observed changes in its radiation environment or is able to provide insight about operation by showing repetitive behaviour. Therefore, a good counter has a high event rate, given with respect to the time it has covered.

To allow for the trade-off above, Figure 4.3 and Figure 4.4 have been prepared.

Figure 4.3 shows the quantity of single events and multiple events for all the counters. Single events are caused by particles of lower energy than multiple events, and abundance is higher for particles of less energy in any space environment. Therefore, with higher multiple event count, the number of observed multiple events must become lower.

Figure 4.4 shows the data coverage of the counters. The data of Figure 4.4 is sampled to one recording per day. Gaps in data of up to 10 minutes have been skipped, while any longer gap in the data results in an unusable day.

It is striking that the CSTMA counters do not follow the tendency of having less events at higher multiplicity. Generally they have a large count of multiple events with the highest amount being 255-fold multiple events. 255 is the capacity of an 8 bit register, and so it is likely, that there is a problem with the counter. As a consequence, CSTMA counters are not considered for analysis.

Another plausibility issue arises on MST0599, which records only multiple events but no single events. It is assumed that this is a flaw in the event counter itself, making it unusable for further analysis.

Thus, MST00495 and the CST counters remain. The CST counters offer a higher number of events than CSTMA counters and a plausible ratio of single events to multiple events. However, their event count is lower than that of MST00495, both when considering the pure number of events and the rate of events given the time covered.

Thus, MST00495 for S2A+B remain. Their ratio of single to multiple events is plausible, and the counter does not show any unnatural behaviour so far. For a more detailed consideration of their data quality, the positions of events are plotted in Figure 4.5 and Figure 4.6. These show a plausible distribution with event clustering within the SAA and multiple events of high count occurring close to the poles, hinting at them having occurred due to strong particles of solar or galactic origin.

For completeness it shall be mentioned that a counter MST00601 has been pre-processed as well but did not turn out to have recorded any event. Therefore it is no longer part of the analysis.

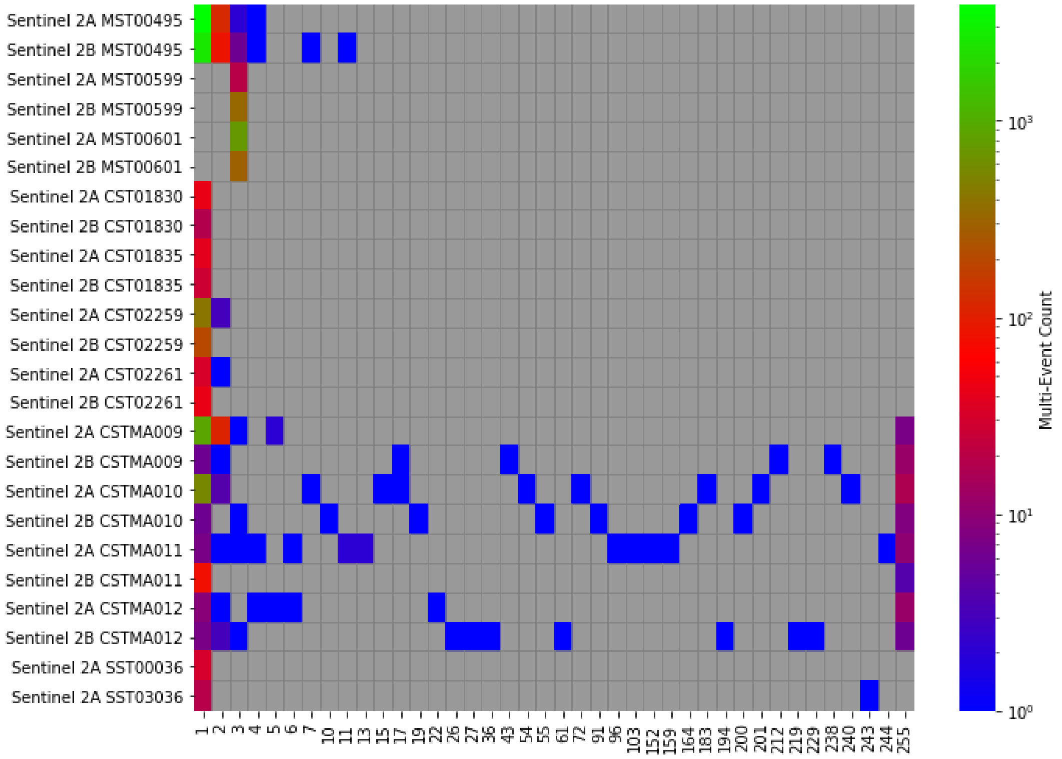


Figure 4.3: Multiple event quantities for all Counters

4.2.2. MSS SRAM

The selected counter, MST00495, is a counter that belongs to the Mass Memory & Formatting Unit (MMFU) of S2A+B. The MMFU is responsible for all satellite data that is deemed to be transferred to ground and sits on a deep-space facing side-wall of the satellites [37]. Since downlink is not possible permanently, the data needs to be buffered and for this purpose, the MMFU contains several Flash memory modules. Considering the flow of data, the memory modules sit between the Payload Data Interface Controllers (PDICs), that retrieve data from the instruments, and the Transfer Frame Generators (TFGs), that prepare data for downlink. The interplay of PDICs, TFG and memory modules is coordinated by the Memory System Supervisors (MSSs), which also writes in housekeeping data for transfer to ground to Flash memory. There are two MSSs per satellite and while they can run at the same time, only one of them is able to work with the other modules at a time. [61]

Among other components, each MSS contains a 30 MHz processor, that runs the software for MMFU management, as well as two banks of $512,000 \cdot 8bit$ EDAC protected SRAM memory, resulting in 4096 KBytes of SRAM in total. MST00495 is the single bit error counter of this SRAM memory. Because there is only one counter for both modules, it is assumed that this counter holds the combined count of single bit errors in both MSSs and there is no need to differentiate between times of different MSSs being active. [61]

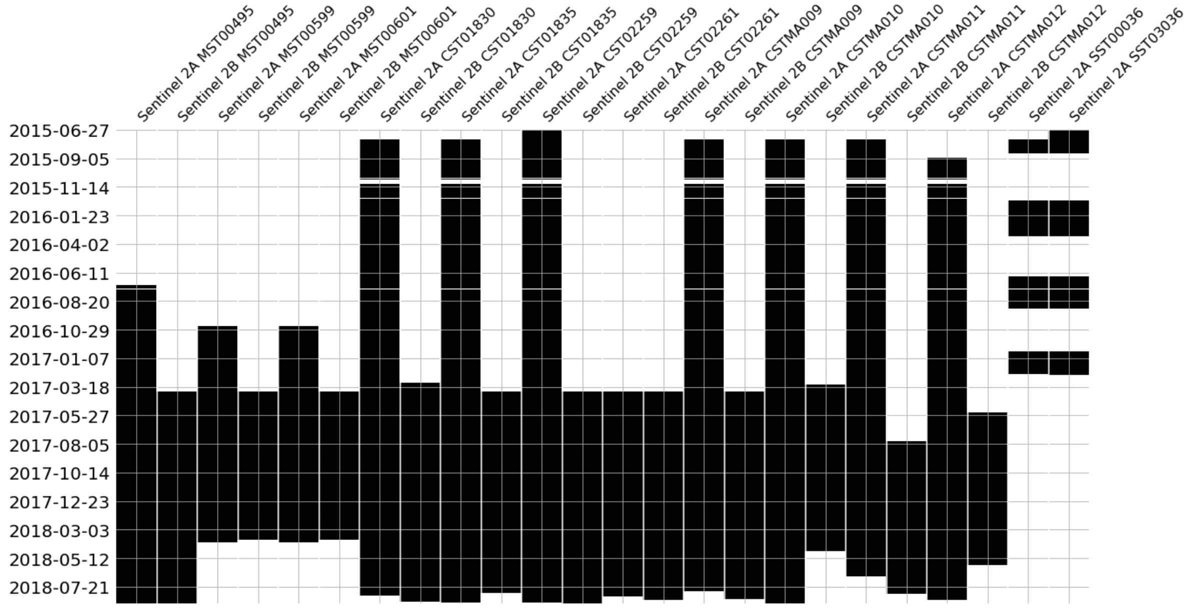


Figure 4.4: Data coverage, day-wise, 10 min. gap allowed

The recorded event time is not precisely the actual event time, because the occurrence of an event first has to be found through EDAC. The process of searching for events is called scrubbing, which can be done cyclically for the entire memory and on-demand when a memory cell is used. With the MSS SRAM scrubbing is done cyclically [61]; the time by which it is repeated is called scrubbing period. This period is 20 seconds [1] for the MST00495 counter. Given the clearly visible SAA in Figure 4.5 and Figure 4.6, this is plausible, because else, the SAA contour of single events would be stretched.

With SRAMs, temperature gradients among memory cells can occur due to the proximity of some memory cells to other functional units of the circuit that dissipate larger amounts of heat, or because of high quantities of read and write cycles in a small fraction of memory cells. Such areas of high temperature are called hot spots. Because temperature is a single event rate driver, hot spots are appended to the list of influence factors given in section 2.4. Their occurrence can be random, when caused by dynamic memory accesses, but can also be periodic when linked to SRAM utilization patterns. The MSS SRAM stores both dynamic data, about to be transferred to memory modules, and static data, that gives information about permanently corrupted memory blocks in the memory modules. The dynamic part of the memory is unlikely to result in hot spots, but the repeated access of the bad block table could result in a hot spot within the memory. With every single event contributing to MST00495, the address at which the error occurred is saved as well, so that the contribution of hot-spots to single event rate can be made observed as event clustering in physically neighbouring addresses, that have become particularly susceptible to single events because of elevated temperature. [55, 61]

In SRAMs, bit errors not only occur due to single events, but can also be caused by switching events of the circuit. Because the occurrence of these is purely random, such errors are not considered any further either. [59]

[1] Interview Marek Prochazka, ESA, 24.01.2019

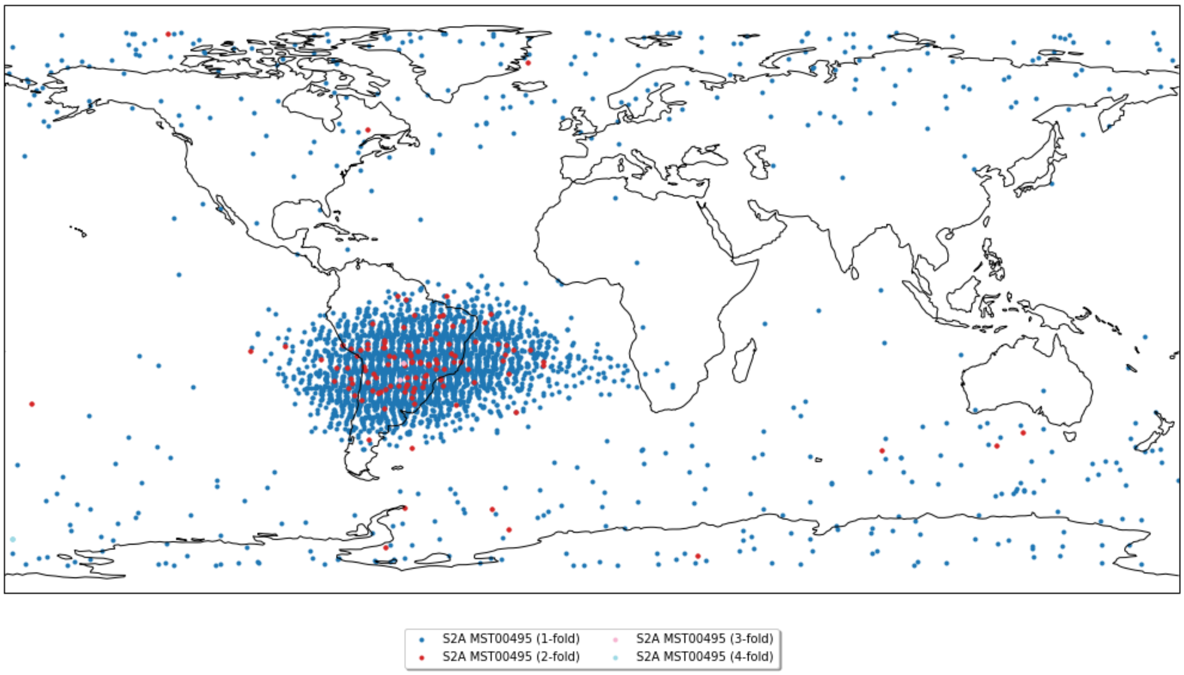


Figure 4.5: Map of Sentinel 2A MST00495 events

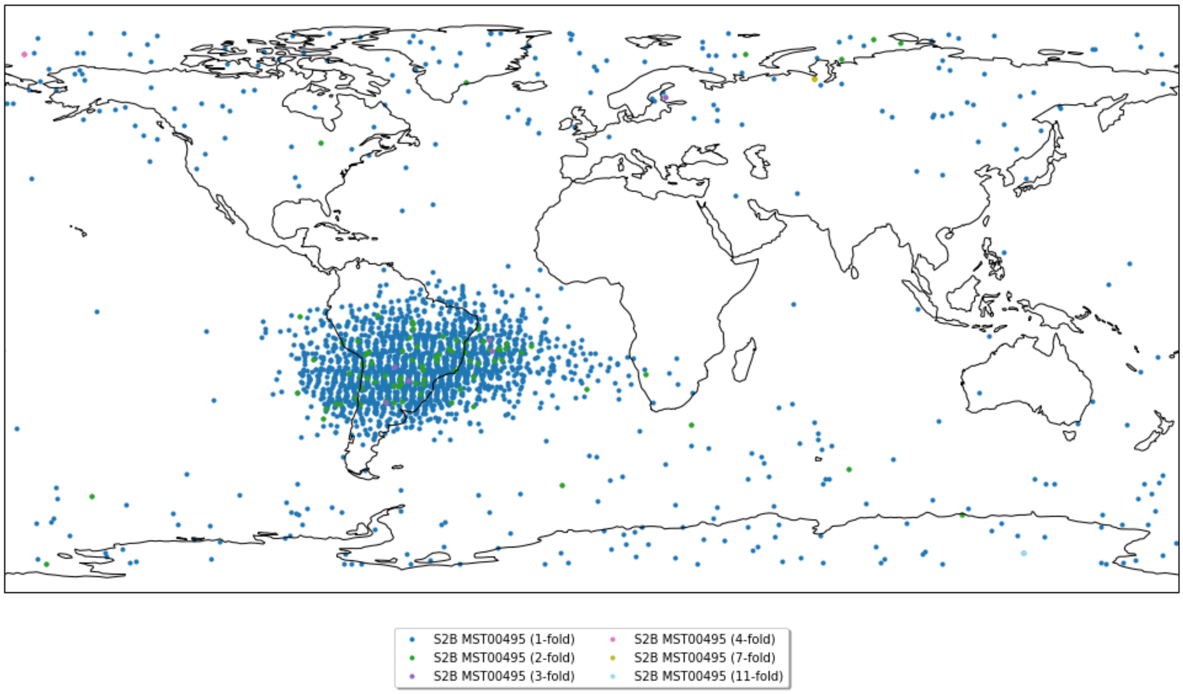


Figure 4.6: Map of Sentinel 2B MST00495 events

Counter Name	Coverage	Count (total)	Rate (total)	Count (individual)	Rate (individual)
Sentinel2A CST01830	1122 days 22:06:00	44	14.31	44	14.31
Sentinel2B CST01830	533 days 00:29:00	18	12.33	18	12.33
Sentinel2A CST01835	1122 days 22:06:00	39	12.69	39	12.69
Sentinel2B CST01835	533 days 00:29:00	27	18.5	27	18.5
Sentinel2A CST02259	1122 days 22:06:00	423	137.59	420	136.61
Sentinel2B CST02259	533 days 00:29:00	194	132.94	194	132.94
Sentinel2A CST02261	525 days 07:54:00	34	23.64	33	22.94
Sentinel2B CST02261	524 days 03:52:00	44	30.66	44	30.66
Sentinel2A CSTMA009	1122 days 22:14:00	2909	946.2	1012	329.17
Sentinel2B CSTMA009	535 days 12:05:00	3578	2440.44	23	15.69
Sentinel2A CSTMA010	1122 days 22:14:00	5682	1848.16	579	188.33
Sentinel2B CSTMA010	535 days 12:05:00	2588	1765.19	21	14.32
Sentinel2A CSTMA011	1122 days 22:14:00	3374	1097.45	30	9.76
Sentinel2B CSTMA011	535 days 12:05:00	1099	749.59	83	56.61
Sentinel2A CSTMA012	1122 days 22:14:00	3108	1010.93	26	8.46
Sentinel2B CSTMA012	535 days 12:05:00	2338	1594.68	24	16.37
Sentinel2A MST00495	789 days 08:06:00	4157	1923.57	4029	1864.34
Sentinel2B MST00495	524 days 03:42:00	2776	1934.42	2660	1853.59
Sentinel2A MST00599	658 days 18:44:33	60	33.27	20	11.09
Sentinel2B MST00599	434 days 05:12:00	1017	855.47	339	285.16
Sentinel2A MST00601	658 days 18:44:33	2196	1217.54	732	405.85
Sentinel2B MST00601	434 days 04:23:34	888	747.02	296	249.01
Sentinel2A SST00036	270 days 04:05:00	31	41.91	31	41.91
Sentinel2A SST03036	269 days 09:13:00	262	355.24	20	27.12

Table 4.1: Sentinel 2 event counter summary

5
Time Domain Analysis

Purpose of the time domain analysis is:

- to determine long term development in event rate, event latitude and event longitude.
- to find correlations between event rate and space weather proxies.
- to separate trend and seasonal effects in single event rate and single event positions.

The methodological preparation of these analyses is given in the next section, after which their implementations are explained, results are presented and finally discussed.

5.1. Methodology

Three signals from S2A+B are part of the time domain analysis:

- Event rate signal. This signal gives the number of events that have occurred in one sampling step. If in one time bin no event has occurred, the signal is zero. If no information is available for a certain time frame, the signal's value is left out.

  To draw links between space weather and event rate, it is necessary to consider each event as a single event only. The rationale for this is, that a multiple event is still caused by only one particle, and the defining link to space weather are particle fluxes.

  Another restriction is made on the basis of Figure 4.5 and Figure 4.6. The figures show that events inside and outside of the SAA differ strongly: while those inside the SAA are abundant and have little multiplicity, those outside occur sparsely and have a higher fraction of multiple events. The reason

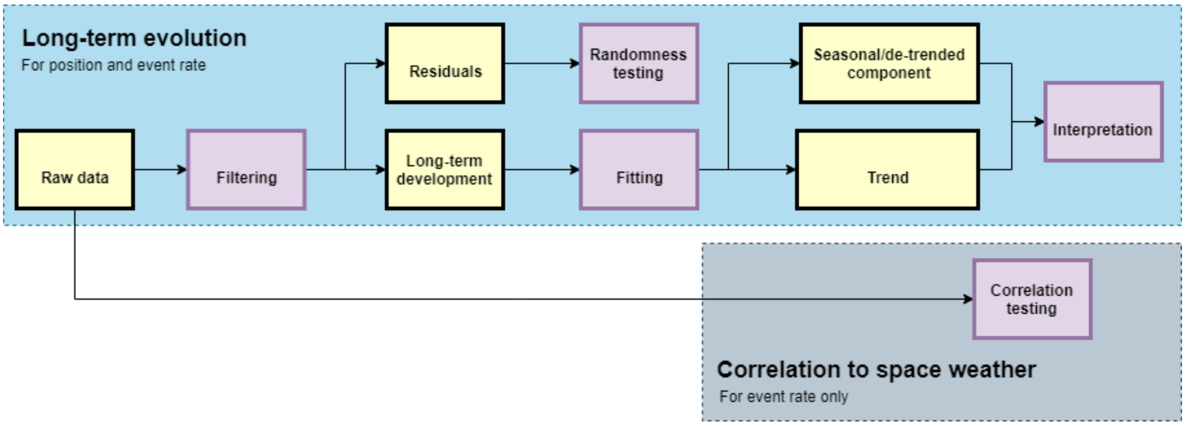


Figure 5.1: Processing of event data

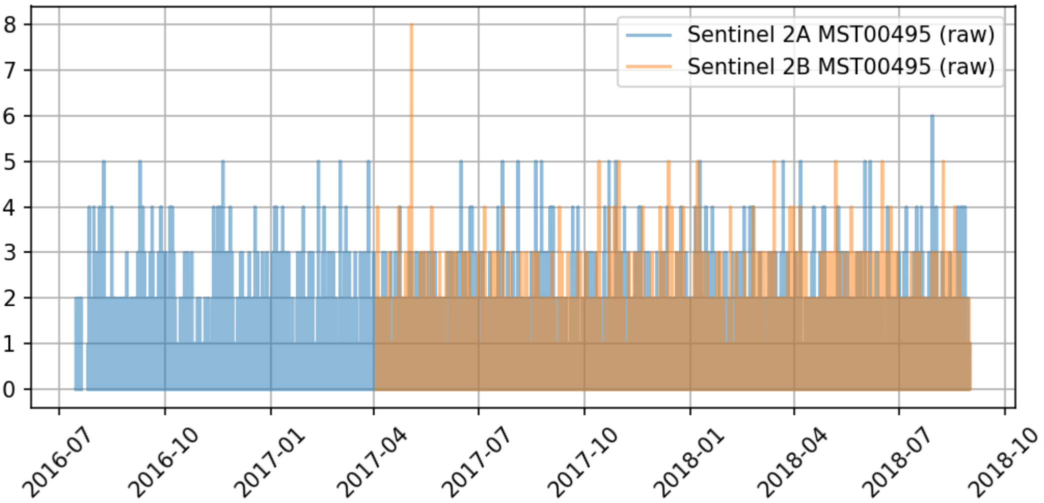


Figure 5.2: Sentinel 2 MST00495 raw

for this are their different causes: trapped particles and solar particles. For one, due to the sparsity of out-of-anomaly events and their irregular behaviour, these events will distort the subsequent study. But also, the study of theory has established, that studying links to space weather can be made solely on the basis of the SAA. This is why in the event rate, longitude, and latitude signal only those events that occurred in the SAA window given in subsection 2.2.5 are be considered. In doing so, less noise is created in the signal to begin with.

The raw signals are shown in Figure 5.2. S2A has a mean event rate of 0.324 per orbit and S2B has a mean event rate of 0.320 per orbit.

- Event longitude and latitude signal. These signals give the position of all events that have occurred in one time step. If no event has occurred within a time step, there is no value for the signal.

  With the event position signals, the same type of event selection as with the rate signals applies: each event is treated as a single one only, and only SAA events are considered.

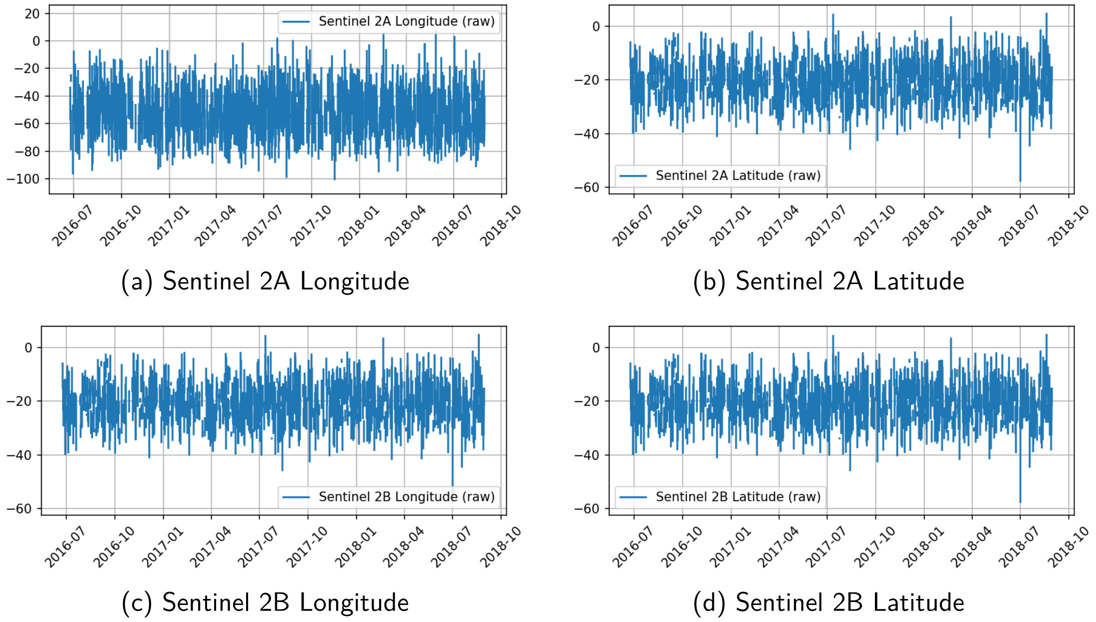
  The raw signals are shown in Figure 5.3.

With the analysis, all signals are sampled to one orbital period (6026 seconds, as calculated in section 2.3), because this is the smallest sampling period that corresponds to a repetitive pattern of any of the single event rate drivers. This also makes consideration of the scrubbing period unnecessary, because it is much smaller than the sampling. As it will be explained later, the analysis is to a large degree robust the chosen sampling.

The methodology of the data analysis differs slightly between event rate signals and the position signals. Both build upon the same initial steps, but the analysis of rate signals has some additional components that are not performed for the position signals. The work-flow for both types of signals is shown in Figure 5.1 and is worked out in detail in the following.

5.1.1. Long-Term Evolution

The analysis of long-term behavior follows the rationale that event rate, event latitude and event longitude are subject to noise on the very short term, but on the long term show deterministic, trend-like, behaviour driven by space weather and device degradation. There are also influences that vary periodically, as listed in section 2.4, so these result in a seasonal short- to medium-term pattern in the signals. These hypotheses translate to the signal complying with the classical decomposition model of time series [17]:



(a) Sentinel 2A Longitude

(b) Sentinel 2A Latitude

(c) Sentinel 2B Longitude

(d) Sentinel 2B Latitude

Figure 5.3: Raw event position data

$$X_t = m_t + s_t + Y_t \tag{5.1}$$

with:

X_t	Time series	m_t	Trend component
s_t	Seasonal component	Y_t	Random noise component

So the lower frequencies of the signal spectrum resemble seasonal and long-term components, while the higher frequencies purely resemble noise. It follows, that there must be a method of filtering the signals to separate the noise component from trend and seasonal component. The combination of trend and seasonal components is from now on called long-term development component. Supposedly, the signals of S2A and S2B only differ in their noise component and have equal long-term development components, because of the satellites being built the same way and are following the same pattern of operation.

When transforming time series data into the components of the described model, MA filters or differencing can be applied [17, 41]. The differences between the methods are:

- MA filter: Trend and seasonal component are estimated to remove the signal's noise component. This is done by using a MA filter, that transforms the signal into its trend and seasonal component. The residuals are noise.

- Differencing: Trend and seasonal component are removed to estimate the signal's noise. This is done by differencing the signal.

With the problem here, which is to find trend and seasonal component, moving-average filtering is the preferred approach, because differencing removes the signal components of interest. Further, it is easier to subjectively judge trend and season as determined by the moving average filter, in contrast to the noise component estimated by differencing.

A MA calculates the value at of the filtered time series at time t on the basis of prior values, or prior and future values, by weighing and summing them. One implementation of such filters is the flat MA filter, which weighs all past values equally to calculate the signal at present. Another implementation of the MA filter used with time series is the exponential moving average filter, which build on exponentially declining weighs [17, 79]. They are qualitatively shown in Figure 5.4 and traded-off in the following:

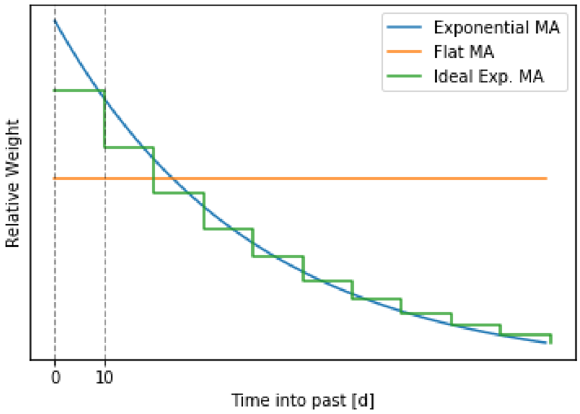


Figure 5.4: Moving-average windows

- **Flat MA filter**

  To calculate a value for an output signal Y at time n, a flat MA filter of window size k takes the average of all values in the input signal X between time $n-k$ to time n [17]:

  $$Y_n = \frac{\sum_{t=n-k}^{n} X_t}{k} \tag{5.2}$$

  The filter has the advantage that its output can be easily understood and interpreted. It is also implemented in the Pandas library already [79]. The flat weighted moving average window has the disadvantage that it does not place particular emphasis on recent values and therefore it takes its output some time to catch up on a trend change.

  To accurately represent the behaviour of the single counters, whose behaviour is driven by particle distributions bound to Earth, the moving average window size k has to be a multiple of the representative ground track of 10 days. Else, a particularly strong or weak phase of the orbit, in terms of single event rate or position, would appear non-proportionally in the filter output.

- **Exponential moving-average filter:**

  The exponentially weighted moving-average filter uses all past values available and weights them using an exponential function with a smoothing factor called α. Therefore, more recent values have a stronger influence on the output Y_n than past values do [79]:

  $$Y_t = \frac{X_t + (1-\alpha)\cdot X_{t-1} + (1-\alpha)^2 \cdot X_{t-2} + ... + (1-\alpha)^k \cdot X_{t-k}}{1 + (1-\alpha) + (1-\alpha)^2 + ... + (1-\alpha)^k} \tag{5.3}$$

  Pandas offers an implementation of the exponential moving-average window [79] filter. However, such a filter per-se does not respect the nature of the data: all values within a 10 day segment have to be given the same weight to achieve an output that is meaningful with the data. It is therefore not an option with single event data of S2A+B.

- **Ideal moving-average filter:**

  The ideal filter for single event data uses exponential weights, but not for individual values, but for 10 day segments, as shown in Figure 5.4. This accounts for the 10 day repetitive ground track and gives more emphasis to recent values.

Because the exponential moving average filter does not suit the nature of the data, and because the ideal filter is not implemented in the Pandas library [79], the flat moving-average filter shall be used in the following analysis. For the filter, a separation frequency, or period, has to be found to allow for filtering of the two signal components. This shall be selected through visual inspection of different filter outputs while considering the following criteria:

- The filtered signals for S2A and S2B must be made as similar as possible, because there is the assumption that they are indeed the same.
- There must be an equilibrium between detail and interpretability of features in the graph. This shall allow for getting all available insight in the data, making the MA window as small as possible, to preserve as much detail as possible.

Applying the filter results in a long-term development component and, by subtracting the long-term component from the raw input signal, its residuals.

To confirm the hypothesis that what has been removed from the signal is indeed noise, the residuals have to be analyzed. The residuals are noise, only if they are uncorrelated. This property can be tested by calculating their auto-correlations. If these are smaller than the confidence bands, indicating the minimum correlation necessary for statistical significance, the residual time series is noise. [51]

With the analysis of single event counters and position data, it is assumed that the trend component both in rate and position is due to the solar cycle and changes in device properties. Because the solar cycle is much longer than the time frame of data considered here, it is reasonable to model its influence on event rate by a linear function. As worked out in subsection 2.1.2, it is unclear what property changes in device single event sensitivity have. It is guessed that they are also approximately linear given the short time covered by single event data. Therefore, the general trend component is modelled by only a linear function. After finding an appropriate trend function through fitting the long-term development, the trend function is subtracted, resulting in the seasonal, respectively de-trended signal.

Upon applying this procedure to the input signals, the three input signals have been turned into:

- A long-term event signal, that supposedly allows to draw conclusions to space weather.
- Two long-term position signals, for longitude and latitude, that supposedly are signals for latitude and longitude position of the SAA's center.
- Linear trends for position in latitude and longitude and event rate.

These outputs shall be interpreted in the light of space weather conditions during S2A+B operation and the differences between the two satellites. For this, a comparison to space weather indicators is made. If a meaningful explanation for the behaviour of the signals can be found, the analysis is deemed successful.

5.1.2. Signal Correlation

The analysis of correlations focuses on event rate signal only and has the goal of finding a function in the form of

$$r = f(\mathsf{SW}) \tag{5.4}$$

where r represents event rate. This analysis has to be based off the unfiltered event rate data, because filtering can remove possible links to space weather variables. If a correlation, or any other deterministic relationship, between space weather variables and single event rate can be shown, this analysis is successful. Then the a prediction of single event rate for the counter under analysis can be made purely on space weather variables. The signal correlation analysis is only performed for the event rate signal, because it is not seen as meaningful to test correlation of event latitude and event longitude against space weather proxies, since these signals are perpendicularly related.

Meaningful proxy variables have to be be selected against which correlation can be checked. It was shown in subsection 2.2.5, that SAA activity is influenced by F10.7 and geomagnetic activity is influenced by DST. Thus, these variables suffice in evaluating the correlation between single event rate and space weather. It would also be possible to use the number of sunspots instead of F10.7 as a metric for solar activity, but their information is to mostly redundant and F10.7 offers higher resolution, which is why it is preferred. Data for these time series will have to be obtained and brought into a format that allows handling them.

Research has shown that particle flux in the SAA linearly correlates to $F_{10.7}$ as well as DST, as summarized in subsection 2.2.5. Therefore, only linear correlation can be expected between $F_{10.7}$ or DST and event rate. To quantify linear correlation, Pearson's correlation coefficient is used [41]. This also means, that more advanced techniques of data analysis cannot give better results on the relationship between space weather drivers and single event rate, because there can only be linearity in theory.

There is no time lag between particle flux and DST, but there can be a very long time lag between $F_{10.7}$ and particle flux in the SAA of approximately 700 days. Therefore, the inspection of correlations between single event rate and DST can be done instantaneously, but the correlation between $F_{10.7}$ needs to be done for past time-lags of $F_{10.7}$. [42, 46]

5.2. Implementation

In the following, the implementation of the time domain analysis, using the libraries of chapter 3, is described. Several processing steps are implemented and applied to the event rate and position data:

1. The single event rate time series for S2A+B are recovered with a sampling of one value per orbit as described in section 4.1, resulting in two Pandas dataframes. The position data is recovered as described in subsection 5.1.1.

   In both cases, only those events within the SAA window given in subsection 2.2.5 are counted, to filter out events due to solar flares and heavy ions to some extent. In recovering the event rate signal, a gap in data coverage of 20 minute is accepted before a time bin is considered unavailable.

   Lastly, the raw data is concatenated so that the different raw time series are aligned time-wise.

2. Next, Pandas' rolling window operator, in conjunction with aggregating by the mean of the considered window, is used for filtering the time series. First, a proper windows size is searched by using different multiples of 10 days as windows and interpreting the output. When this size has been found, the time series are filtered using that window size and only used in this form in subsequent analyses.

3. The raw time series is sampled down to 1 sample per day by averaging, again through Pandas' resampling operator. For testing correlation with F10.7 and DST, a common sampling rate is needed. 1 sample per day is the finest common rate possible, because this is the minimum interval by which $F_{10.7}$ data is available.

As a second input to the correlation analysis, data for F10.7 and DST is needed. Their time series are put into the time series database through Arctic and pandas. Time series data for F10.7 has been obtained through ESA's space situational awareness program[1] and has a sampling rate of one measurement per day. The data had to be pre-processed to make it fit to CSV format. Time series data for DST has been obtained from University of Kyoto [2]. It has a sampling rate of 1 recording per 3 hours. The data had to be pre-processed as well, by first copying the website contents to a local Excel file, reading it using pandas and finally pivoting the table. This resulted in a typical pandas CSV time series.

The inspection of auto-correlations of a residual time series can be done through statsmodels' plot_acf function, which immediately returns a plot for a specified number of time lags. Also, the function provides confidence bands to separate statistically significant correlations from non-significant correlations for a specified significance level α, usually being 0.05.

Besides functions for preparing the event rate data, functions for correlation testing are needed. scipy's linregress function performs linear regression and returns slope and intercept, Pearson's correlation coefficient and the p-value for a test for statistically significant correlation. Is is therefore used to determine the

[1] `http://swe.ssa.esa.int`, accessed 21.02.219

[2] `http://wdc.kugi.kyoto-u.ac.jp/dst_realtime/`, accessed 21.02.2019

correlation between two time series as well as for finding the linear trends. Time-lagged correlations are tested by first shifting the time-series through pandas shift-operator. For this task, a convenience wrapper, described in section A.5, has been written to evaluate several time-lags at once.

Several functions have been written to bundle functionality from different Python libraries. These functions are documented in section A.5.

5.3. Results

5.3.1. Long-Term Evolution

Event Rate

The figures of section A.2 give single event rate plots for MA windows with a window size up to 100 days, in steps of 10 days. These have been prepared for the selection of the most suitable window size using the subjective criteria given in the previous section.

It is visible, that the MA filter fortifies differences between the two satellites, so that with increasing window size, they are worked out more clearly. With a 10 day window size, the event rate curves of the two satellites show only subtle differences, but are somewhat distinct once a window size of more than 20 days is used. This means that to fulfill the previous requirement of similarity, a small window size is preferable.

On the basis of visual inspection of section A.2 it is maintained, that filtered event rate changes negligibly, once window length becomes larger than 30 days, so that using a window larger than that is not considered meaningful. Further, a 10 day window offers too much detail for it to be interpretable and is therefore not a reasonable choice to begin the analysis either. Still, it may be invoked later during the analysis to clarify observations that have been made in the interpretation of a plot with larger filter window size. It is thus decided to use a 20 day MA window in beginning the interpretation of the data.

The top frame of Figure 5.5 shows the long-term event rate development as well as the linear trends for S2A and S2B and the detrended signal at the selected MA window of 20 days. The residuals of the long-term components against the raw data are given in the bottom frame of Figure 5.5.

Figure 5.5 shows that the two satellite's event rates behave similarly, with some fluctuations on a time scale of several months. It can be seen, that the filtered event rates for S2A and S2B vary between 0.43 events per orbital period and 0.18 events per orbital period. The overall maximum event rate is found with S2A event rate around September 2017 and the overall minimum is found for S2A as well in November 2016. The average filtered event rate is 0.314 with S2A and 0.309 with S2B. The mean absolute errors between the filtered signal and the raw signal is 0.49 for S2A and 0.48 for S2B and are therefore higher than the remaining average event rates. The event rate trends of Figure 5.5 can be approximated linearly by slopes of $0.0164\ events \cdot orbit^{-1} \cdot year^{-1}$ for S2A and $0.0165\ events \cdot orbit^{-1} \cdot year^{-1}$ for S2B. Thus, the two satellites' trends practically do not differ.

It was introduced in subsection 2.2.5, that the activity of the SAA undergoes a semi-annual cycle. The maxima of this cycle are found around Autumn Equinox (AEX) and shortly before Vernal Equinox (VEX) and minima of these cycles occur in December and July. The de-trended and filtered event rate rate signal of Figure 5.5 shows this cycle in grey. For quantitative comparison of event rate to solar activity proxies, F10.7 and DST are plotted as well. In addition, outliers in F10.7 and DST, which indicate sunspots and geomagnetic storms, are highlighted by dashed vertical lines. A discussion of the figure follows in the subsequent section.

Figure 5.6 shows the auto-correlations of the residuals. It can be seen, that they are significant (exceeding confidence intervals) and repeat every 7 lags. Therefore, the residuals do not consist of noise only.

Figure 5.5 shows a histogram over de-trended event rate, the seasonal component. It can be inferred, that the maximum deviation is approximately 0.06 events per orbit, which means that a variation by approximately 20 % due to seasonal effects is possible. The standard deviation is about 0.037 for S2A and 0.031 for S2B.

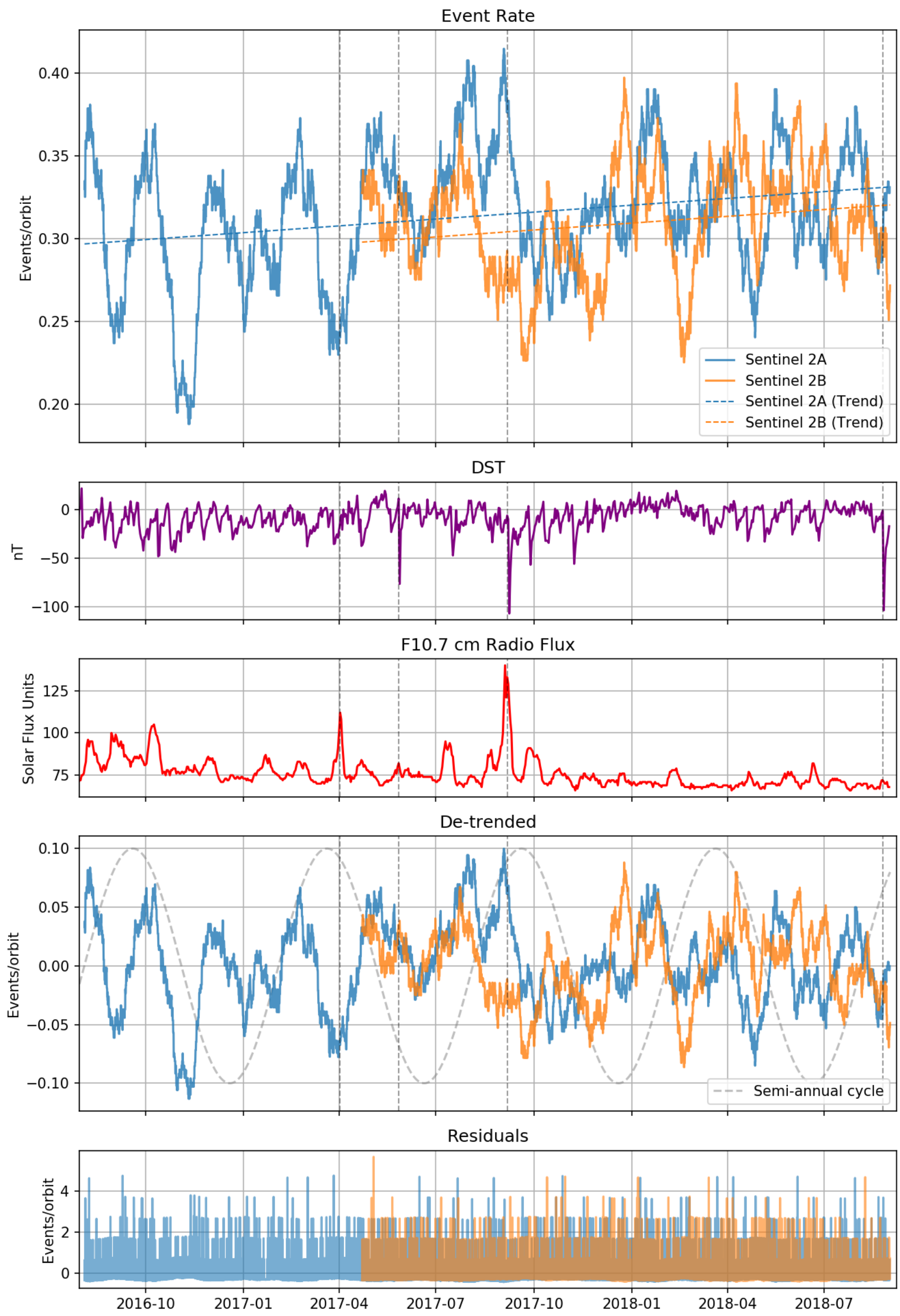
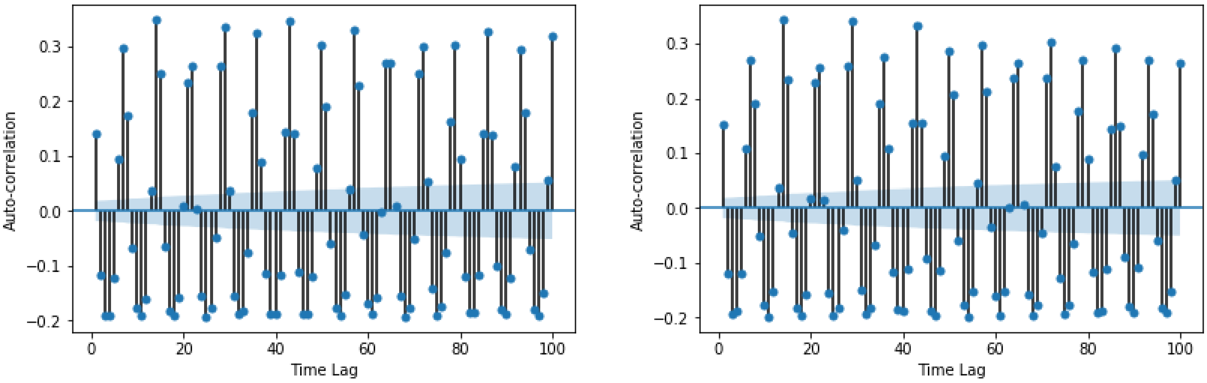


Figure 5.5: Single event rate long-term evolution



(a) Sentinel 2A

Sentinel 2B

Figure 5.6: Results for event rate residual auto-correlation

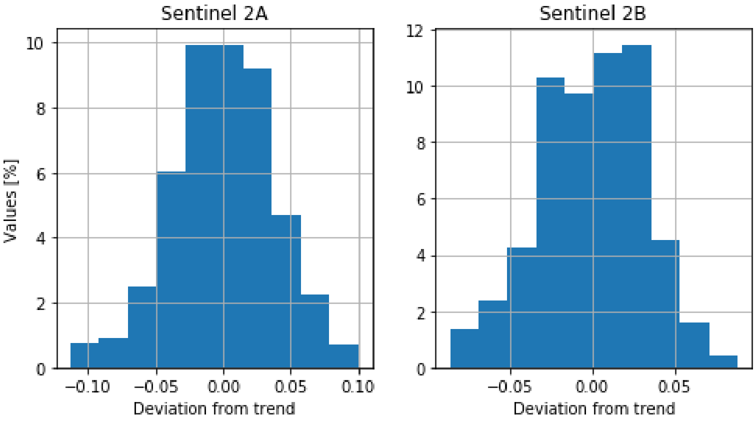


Figure 5.7: Deviation of event rate from linear trend in percent of values

Satellite	Longitudinal drift	Latitudinal drift
Sentinel 2A	-1.24°/y	-0.12°/y
Sentinel 2B	-2.08°/y	1.10°/y

Table 5.1: Linearized SAA drift

Event Position

To selected a suitable window for event position data filtering, the figures of section A.3 and section A.3 have been created. Following the reasoning of the previous section, the same MA window of 20 days has been selected.

Figure 5.8 shows long-term average event positions in latitude and longitude, which is argued to represent the center of the SAA, and in addition, linear approximations to each of the signals. Table 5.1 gives linear fits to the latitude and longitude positions of S2A+B. The latitude and longitude positions for S2A and S2B comply well, except for some strong, timely-restricted differences. These differences and commonalities pointed out in more detail in the subsequent section.

Residuals of event positions are given in Figure 5.8 and tested for auto-correlation in Figure 5.9. It can be seen, that latitude residuals are practically uncorrelated, whereas the longitude residuals show an auto-correlation pattern similar to that of event rate.

Because the combination of longitude and latitude signal describes a position, it is presented in a two-dimensional coordinate system as well, shown in Figure 5.11. For better discernibility, this requires a different MA window than with the one-dimensional presentation of single event position in Figure 5.8, so that here, a MA window of 90 days has been applied. This offers the best compromise between detail and interpretability of the figure. Figure 5.11 also qualitatively gives F10.7, DST, SAA longitude and SAA latitude to allow for a visual inspection of how they are associated.

For clearer examination of the average event position paths, called single event trajectories, a 3D plot of the MA event position has been made using the 20 day window and time as third dimension. An extract of this plot is shown in Figure 5.10 for illustration. The 3D plot allows to evaluate the event trajectories at higher precision and was used in an thorough search for commonalities and differences among the two satellites. Figure 5.12 highlights these commonalities and differences, and highlights them for explanatory purposes. Each color marks the event trajectory of the same time frame for both satellites.

To begin with, commonalities are found:

- From approximately May 2017 to approximately September 2017, both satellites' average event position changes in a circular fashion, moving counter-clockwise. This is highlighted in Figure 5.12 in green.
- The average event position changes only little in both satellite's between April 2018 and July 2018 and describes a hook for both satellites (highlighted in orange).

Yet, the following features can only be seen with one of the satellites:

- Sentinel 2B takes a far northwards loop between 2017-12 and 2018-03 (blue). During that time, S2B's event position remains confined to the approximate center of event position history.
- Sentinel 2A moves northward, starting mid August 2018 (red). At the same time, Sentinel 2B continues to move about its initial August 2018 position.
- Sentinel 2A gets stuck in a eastwards position from mid September 2017 to end October 2017, while Sentinel 2B continues on a trajectory towards the event position center (purple). This deviation of Sentinel 2A happens just after completing the circular movement it undergoes together with Sentinel 2B.

In the next section, the features as well as reasons behind the movement of average event position and the differences between the two satellites shall be discussed.

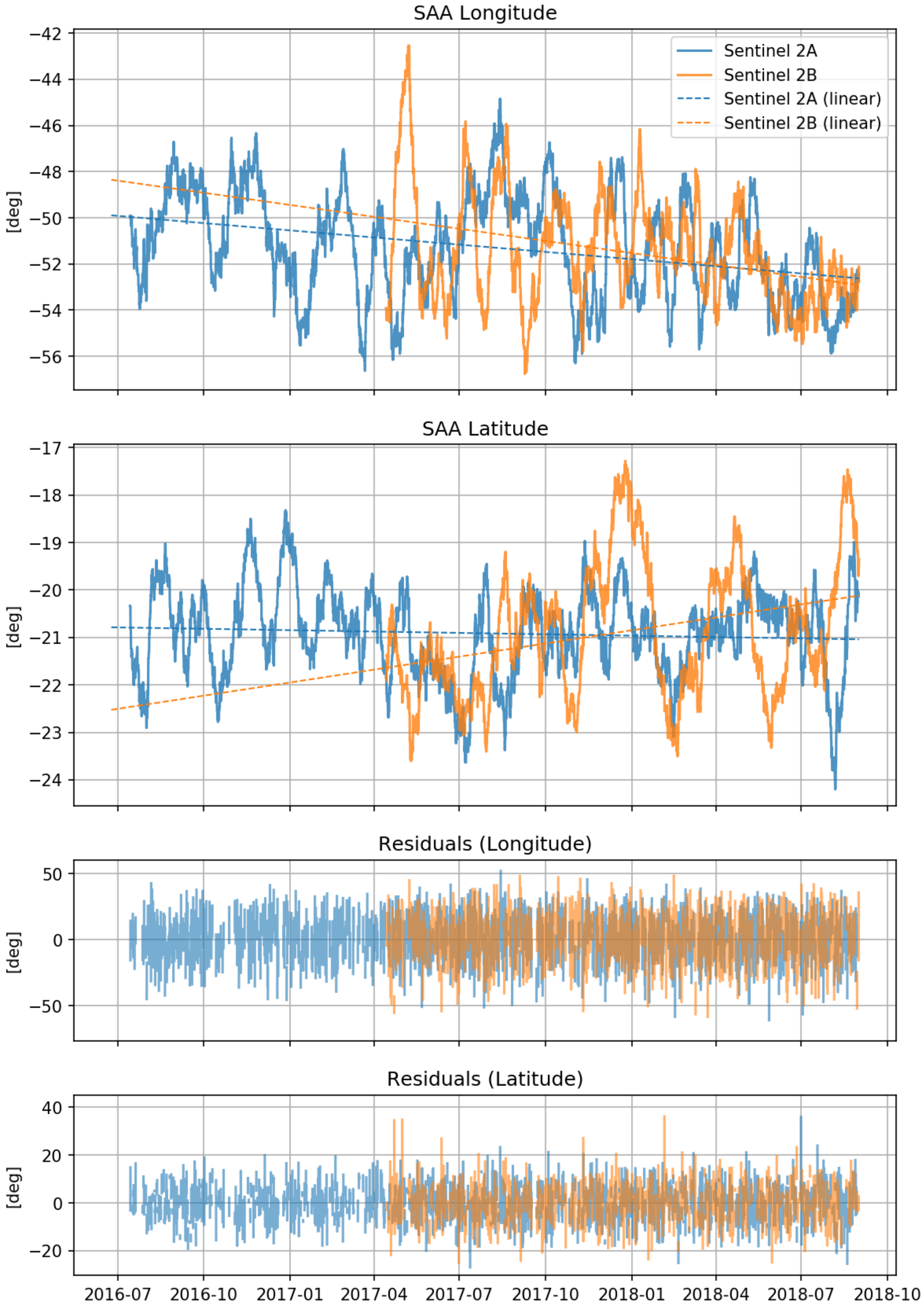
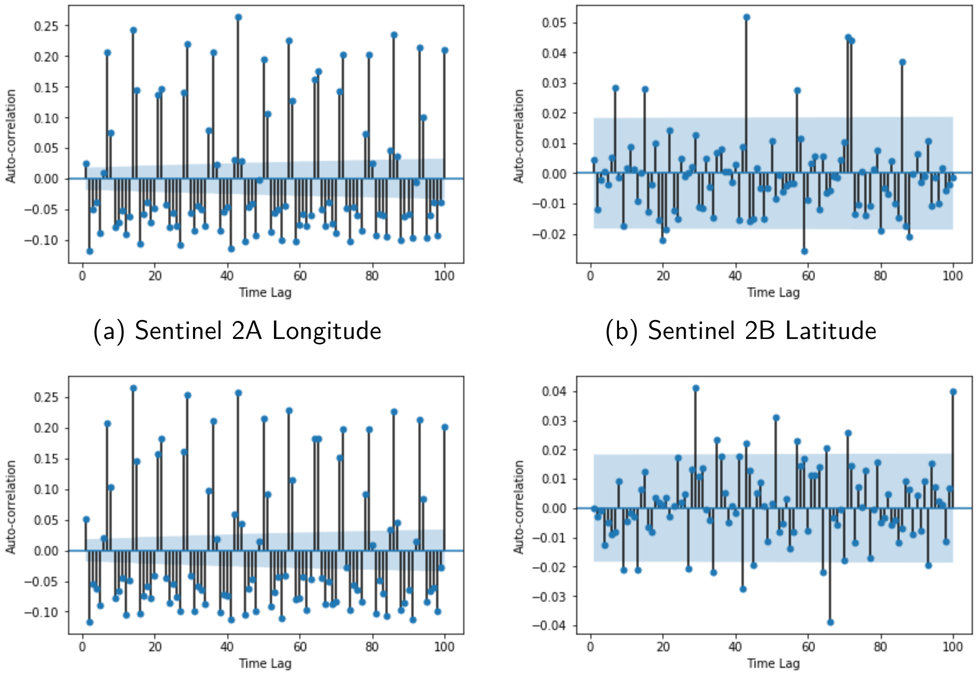


Figure 5.8: Single event position long-term evolution



(a) Sentinel 2A Longitude

(b) Sentinel 2B Latitude

(c) Sentinel 2B Longitude

(d) Sentinel 2B Latitude

Figure 5.9: Auto-correlation of position signal residuals

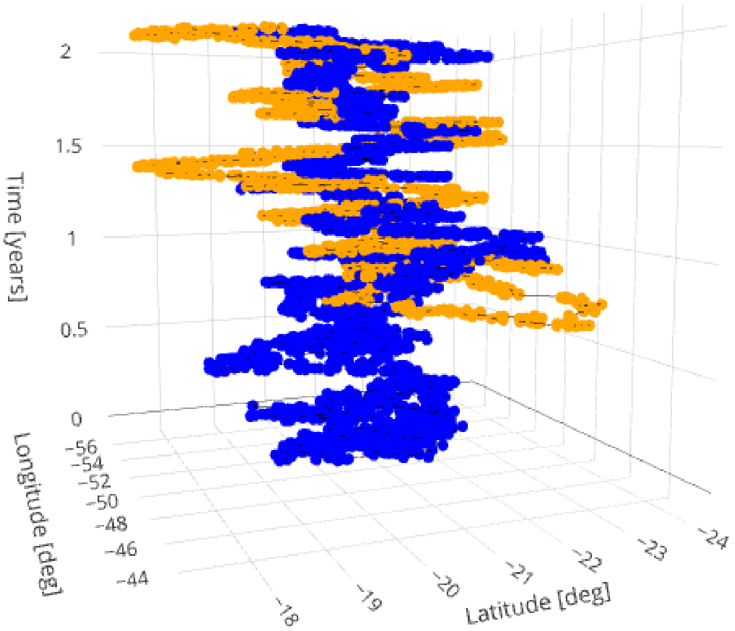


Figure 5.10: 20d MA 3D plot of event positions (blue: S2A, orange: S2B)

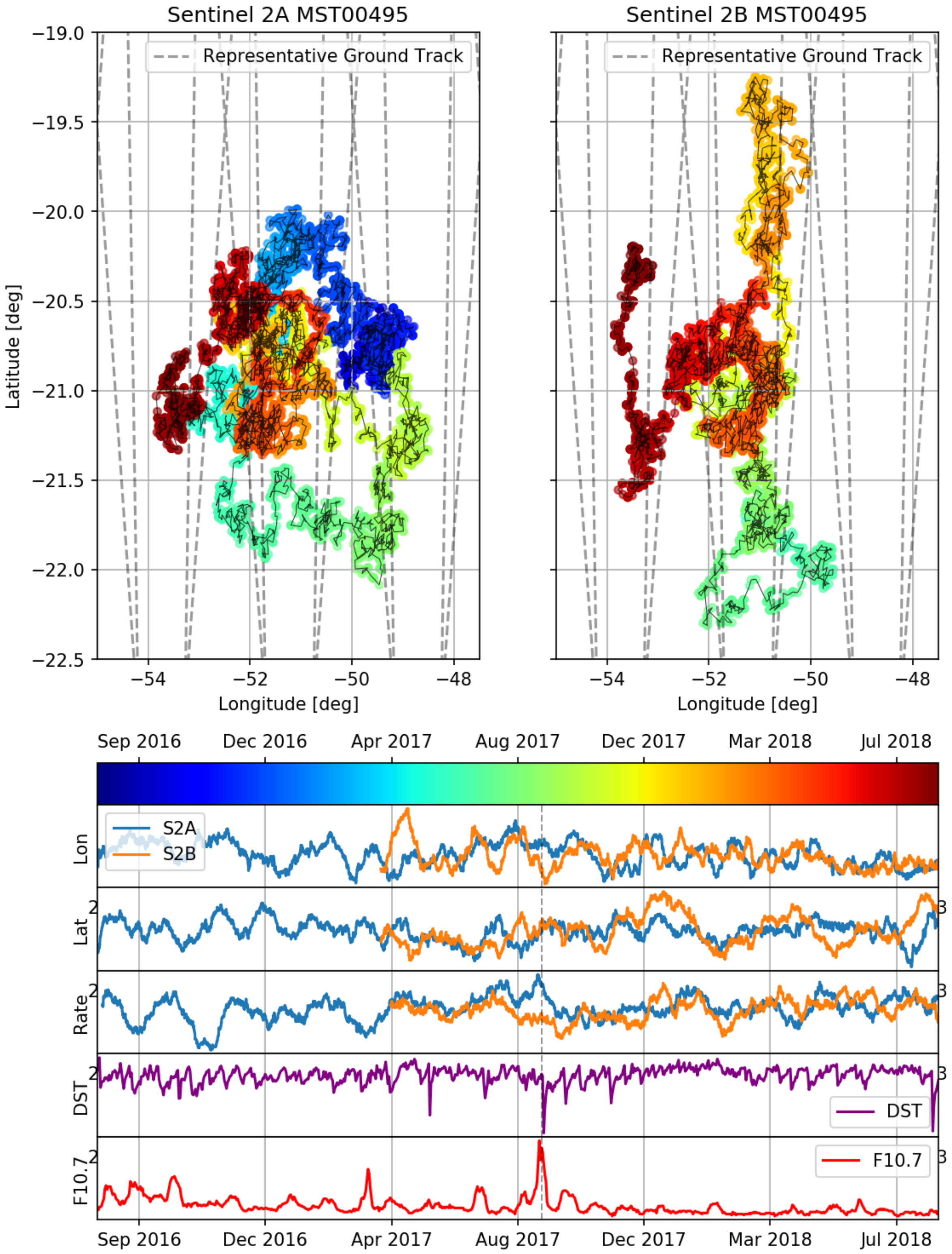


Figure 5.11: 90d SAA rolling average position

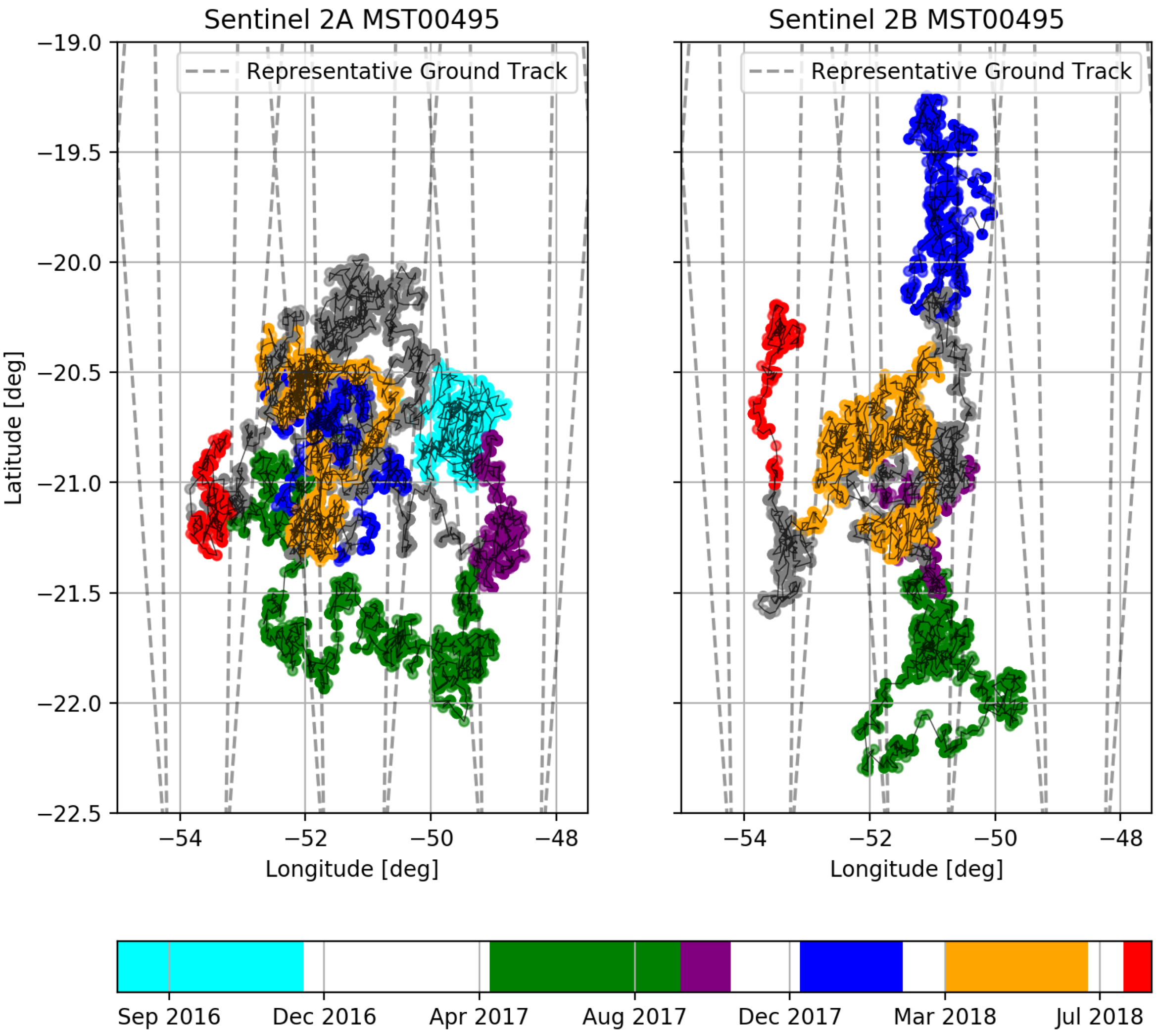
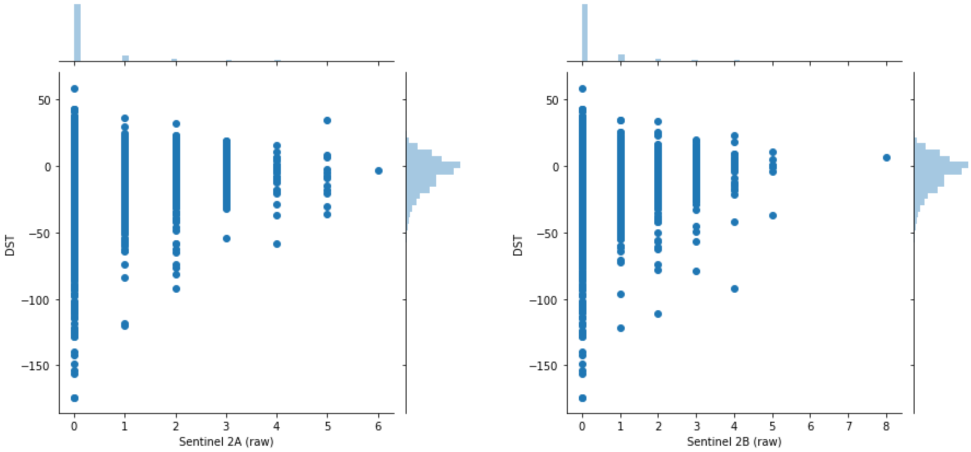
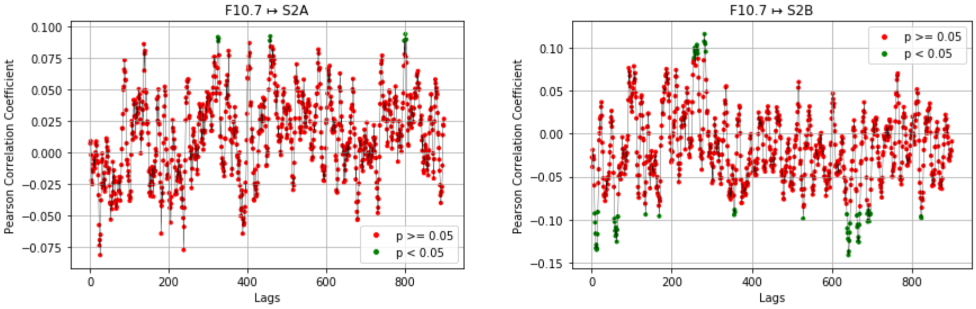


Figure 5.12: 90d SAA rolling average position w. highlights



(a) Sentinel 2A / DST scatter plot (b) Sentinel 2B / DST scatter plot

Figure 5.13: Results for instantaneous correlation between event rate and DST



(a) Sentinel 2A / F10.7 correlation (b) Sentinel 2B / F10.7 correlation

Figure 5.14: Results for time-lagged event rate/$F_{10.7}$ correlation

5.3.2. Correlation Results

The numeric results for correlation analysis are given in Table 5.2 for the instantaneous correlation between DST and event rate and and the corresponding scatter plots are shown in Figure 5.13. It can be seen that the scatter plots, putting event rate and DST into relation, show strong clustering and little reaction of event rate to DST. Next to that, outliers in DST, that occur as a result of CMEs, do not result in an outliers in single event rate.

Satellite	Correlation to DST
Sentinel 2A	0.01
Sentinel 2B	0.02

Table 5.2: Correlations between event rate and space weather variables

Results for the time-lagged correlation between $F_{10.7}$ and event rate are given in Figure 5.14 for a time-lag of up to 800 days. This maximum time-lag has been selected on the basis of the results of Qin et al. [42] and includes some margin. It can be seen that there are only few time lags at which statistically significant correlations exist. In addition, none of these correlations appears for both satellites at the same time.

5.4. Discussion

5.4.1. Long-Term Evolution

Event Rate

The mean filtered event rates of Figure 5.5 differ by less than 2% between the satellites and the linearized trends differ by less than 1 %. This shows that the components aboard the two satellites indeed behave very similarly over long times and is also seen as a confirmation for the approach of modelling degradation and solar cycle as linear function.

It fits to space weather theory in that both satellites event rate trends are positive. The data was recorded while approaching solar minimum, which is when the inner radiation belt is increasing in particle flux and therefore event rate must increase as well. A trend-change is expected to occur within the next years, when solar minimum has been reached and one life time of the event-causing protons has passed. The time delay between solar minimum and event rate maximum might allow for drawing conclusions on the energy of the protons causing events with S2A+B by linking their life-time to the energy they theoretically posses. To properly represent the trend in the analysis of future single event data, a more complex trend function must be fitted. Further, it has to be kept in mind that the future trend could be both positive and negative: on the one hand, the waxing solar cycle will decrease event rate, but on the other hand, device degradation could increase event rate. It has yet to be seen which effect is stronger. Such an observation has not been made with single event data yet, so monitoring of future data is strongly indicated.

The current trend results in a change of single event rate between 5 and 10% per year. One could therefore argue that it is practically irrelevant with respect to the overall event rate, but with mission time it gains in impact. Extrapolating the current event rate drift to S2A+B's approaching solar minimum, a maximum event rate in the range of 0.38 to 0.40 events per orbit can be expected, assuming a phase delay between solar minimum and SAA maximum fluxes of up to three years. This is a mild increase in contrast to the current event rate of 0.32 per orbital period.

Figure 5.6 shows that the residuals are auto-correlated, which means they are not purely noise. The auto-correlation pattern has a cycle of 7 orbital periods, after which it repeats, making its duration commensurate to 11.7 hours. In section 2.4, it was mentioned that a 12 hour periodicity in the single event data can be expected due to revisits of the SAA. This semi-diurnal periodicity is what causes the auto-correlation of the residuals. Because the pattern seems homogeneous and undisturbed by any other effect, the hypothesis is formed that the only relevant short-term influence on single event rate, not being considered in the long-term analysis here, are the repetitive passes of the SAA. This hypothesis will be verified in the subsequent spectral analysis, in case no other short-term influences event rate variation is found in that scope.

Now the variations seen in Figure 5.5 shall be interpreted. Three explaining influences exist: the SAA seasonal variation, solar particle events and memory hot spots. Not all features of Figure 5.5 can be interpreted on the basis of this and especially the disparity in the behaviour on shorter time scales between the two satellites shows that further data and study is necessary.

The plot of de-trended event rate in Figure 5.5 shows in grey, how event rate would have to develop if the only remaining influence was due to the semi-annual SAA cycle. One can see that there are indeed variations on a similar time-scale in the satellites' event rates, but their maxima and minima do not coincide with the semi-annual SAA cycle of Schaefer et al. in grey. This leaves three possible explanations for single event rate variations:

- The influence of solar particle events or memory hot spots covers up the semi-annual cycle.
- There is a time-shifted semi-annual cycle in Figure 5.5, arising because of the possible difference in particle energies between [54] and Sentinel 2 MSS SRAM.
- There are other influences, yet unknown.

When expanding the interpretation of Figure 5.5 to consider solar particle events as well, the following hypotheses can be made:

- A strong event rate decline starts for S2A in September 2017, while at the same time event rate for S2B stagnates and declines shortly thereafter. This coincides with two CMEs occurring shortly after each other[3], indicated by a dashed line in Figure 5.5. These CMEs can also be seen in the high amount of protons measured by GOES-15 in geostationary orbit (Figure 5.15), but also in spikes in DST and F10.7. Other CMEs during the time studied only affected DST, but did not show comparable proton fluxes with GOES-15, which speaks for the CMEs fierceness.

  Theoretically, CMEs can affect single event rate in three ways simultaneously:

  - Selesnick, Hudson, and Kress [31] showed, that CME can directly lead to particle loss in the inner radiation belt by freeing them from the magnetic field [31], so that a reduction in event rate is natural with strong CMEs. Given the small minimum DST peak of the September 2017 CMEs, compared to the minimum DST values of the CMEs studied by Selesnick, Hudson, and Kress, the effect however is weak or non-existent in September 2017.
  - The two CMEs cause a visible Forbush decrease. The particular decrease can be seen in measurements by the Moscow Neutron Monitor [4] (Figure 5.16, which indirectly monitors the flux of GCRs by measuring the flux of neutrons, the result of GCRs entering the Earth's atmosphere. Neutrons re-supply the radiation belts, and through a reduction in the re-supply of inner radiation belt protons, event rate is decreases in the wake of a CME.
  - The heating of outer atmosphere through the CMEs could result in an elevation of the atmosphere itself, that reduces proton fluxes in the inner radiation belt through particle scattering [58].

  Beginning of September, AEX had almost been reached, which means that a peak in the semi-annual variation of SAA particle count would have been reached anyways. However, it was previously shown how it is unclear in what extent or phase-delay the semi-annual SAA cycle affects S2A+B event rate.

  It is thus hypothesized, that the sharp decline of S2A, continuing until November 2017, can be attributed to the combination of previous phenomena. On the basis of the information available, it is not possible to clearly give the fraction by which each of the individual phenomena has contributed. In addition, it might also be hypothesized that the decline has led to particle levels reducing more than usual, because event rate increases even after November 2017, where it would normally be approaching minimum, in case the seasonal behaviour of SAA particle fluxes holds. The same increase despite seasonal minimum can be seen for S2B as well, but weaker.

  When the event rate decline of September 2017 begins for S2A, S2B already showed a decline in event rate going back to August 2017 and only declines until October 2017. Why S2B started before the three phenomena is not clear and must therefore have reasons external to the current explanatory framework.

- After the September 2017 decline, S2A+B event rates increase again. The increase is rather strong and stagnates at the supposed seasonal maximum at VEX 2018. The increase here could be exceptionally strong, because it is accompanied by quiet solar and geomagnetic activity (2017-12 to 2018-02), which would allow for a very stable particle population in the SAA.

- Before the September 2017 peak, event rate for S2A rises (June 2017 to September 2017), starting at its possible seasonal minimum in June. The behaviour of S2B at the same time is different. S2B does not monotonously increase in event rate, but starts to decline already in August 2017. What causes this difference is unclear, but the spike in DST during June 2017 could be related to it, since it indicates a minor geomagnetic storm and at the same time marks the onset of the decrease for S2B. Because of the phase delay in the satellites' ground tracks, the effects of the geomagnetic storm could have been vanished already by the time S2A passed the same positions as S2B.

[3] http://www.srl.caltech.edu/ACE/ASC/DATA/level3/icmetable2.htm, accessed 10.03.2019

[4] http://cr0.izmiran.ru/mosc/, accessed 10.03.2019

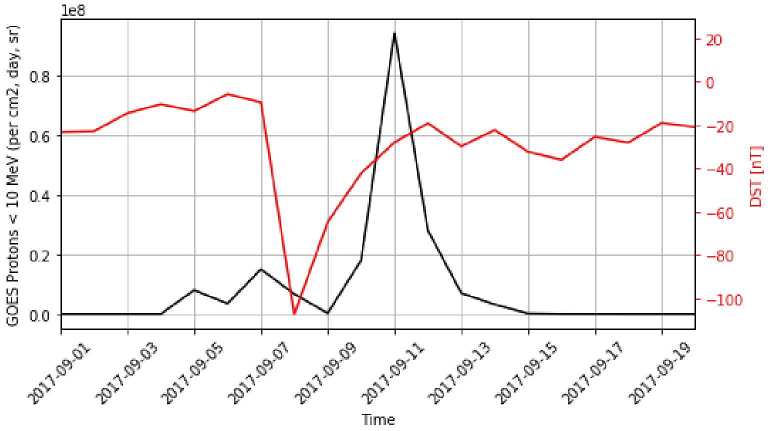


Figure 5.15: Proton flux in GEO and DST of 2019-09 CMEs

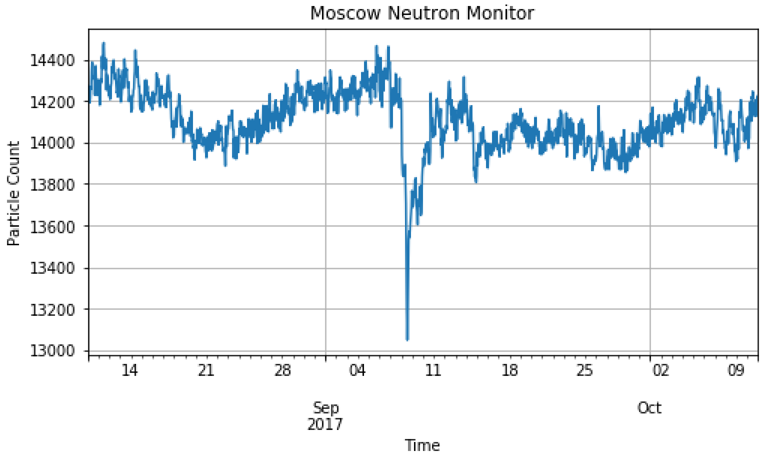


Figure 5.16: Moscow neutron monitor, showing a decrease in GCRs

- Some CMEs seem to not have an effect on event rate for both satellites. For instance, the spike in DST around June 2016 is not accompanied by a change in event rate for S2A. This could be due to the phase delay in the satellite's orbit. The spike in $F_{10.7}$ in April 2017, indicating a solar flare, should be accompanied by a decrease in event rate, because proton flux and F10.7 are anti-correlated, but instead event rate increases following it. Presumably, the solar flare's $F_{10.7}$ increase was to short to have a meaningful effect on particle populations.
- Another geomagnetic storm in May 2017, caused by a CME, is visible by a spike in DST. For both satellites, a decline in event rate follows shortly after, which would mean that this CME has had consequences on event rate. However, the seasonal minimum was to occur for June 2017 anyways, so this should normally also trigger an event rate decrease.
- There is a strong geomagnetic storm close to the end of the available data, following a CME in August 2018 (dashed line). Not enough data is available to describe the subsequent event rate development.
- No annual component due to Earth's orbit is visible. However, two reasons are thinkable: either, during the time scales considered it was overruled by other influences and thus is not clearly visible, or, it was overestimated to begin with and thus does not play a role in single event rate.

Thus, even when taking semi-annual behaviour and particle events into account, it remains intricate to explain event rate variations of the satellites and especially the differences between the two satellites. Therefore, either hot spots or another unknown phenomena must be the main driver behind single event rate seasonal effects, or there is an intricate combination of several effects. This shows that further measures are needed to gain clarity. It is proposed to:

- Gain clarity on when hot spots occurred with S2A+B. This can be achieved by studying the memory addresses at which single event upsets had occurred, which are recorded at event occurrence according to [61]. If the addresses cluster for some time, hot spots are likely to have contributed to them.
- Perform simulations of radiation belt particle distributions for the covered time frames and place them in context with the event rates of S2A+B. A possibility to do so is provided by the radbelt module of the SpacePy Python library. radbelt provides a model of the Earth's radiation belts and uses geomagnetic conditions to simulate resulting particle distributions. By performing such simulations for the times covered by S2A+B single event data, using DST and other actual measurements as input, a clearer understanding of the solar particle event influences will be gained. Possibly, there are time delays between the onset of a geomagnetic storm and the observation of consequences on S2A+B. It could become possible to discern radiation belt effects from other influences on event rate seen in Figure 5.5. Also, this allows for considering the impact of different particle energies. [30]
- Study future data from the satellites to increase the understanding of event rate long-term evolution. Up to now, because the data was recorded during quiet space weather conditions, only weak solar particle events have been seen by S2A+B. Stronger particle events will provide a stronger response in the event counters, so that their effect becomes visible more clearly.
- Re-iterate the identification of influences on event rate. It cannot be excluded yet that there are operational influences, currently not being considered, that confound the expected long-term variations. For example, it is thinkable that the attitude of the satellites differed which resulted in a different effective shielding for the components. A further iteration in identifying single event drivers with the satellites will clarify this.

Figure 5.7 shows, that the seasonal deviation is commensurate to approximately three years of trend increase. Further study, as described above, needs to be done, what fraction of this is due to CMEs and what fraction is due to semi-annual variation.

Position Data

The sole observation that the average event position undergoes non-random changes, shown in Figure 5.11 and Figure 5.12, is rather trivial to explain. Earth's magnetic field is permanently changing, because of internal processes and interaction with the Sun. Trapped particles, such as the SAA protons tracked by the single event counter under study, follow Earth's field lines, and with a change in the magnetic field the arrangement of field lines changes changes as well [16]. As such, the particle distribution in the SAA is permanently changing, which results in event position changes. Remarkably, the intra-year movements tracked by single events, shown in Figure 5.11, already go over differences in the same order of magnitude as the annual reported SAA movement in literature [54, 57]. This implies, that intra-year movement is actually relevant in defining the position of the SAA.

In light of the scrubbing period of 20 seconds, the recorded event position is not exactly the position at which the event occurred. However, the satellites pass the SAA on a north to south trajectory as many times as they pass it from south to north, meaning that errors in latitudinal direction will even out over the large MA window. Because of the inclination of the orbit, the passings also contain two longitudinal components, that are equal for passings from north to south and from south to north. Thus, there remains some error in east-west direction that is not compensated through averaging. This error can be quantified from its two components: the distance the satellites pass at longitudinal velocity within the scrubbing period and Earth's rotation during the scrubbing period. The longitudinally flow distance is:

$$\epsilon_{lon,V} = \frac{T_S}{T_O} \cdot \sin\left(\frac{\pi}{2} - i\right) = 0.026\ deg \tag{5.5}$$

The error due to Earth's rotation is calculated as follows:

$$\epsilon_{lon,E} = \frac{360\ deg}{24\ h} \cdot 20\ s = 0.083\ deg \tag{5.6}$$

So that the total error in east-west direction is:

$$\epsilon_{total} = \epsilon_{lon,E} - \epsilon_{lon,V} = 0.057\ deg \tag{5.7}$$

Given this small error, differentiation between event recorded position and event true position is not considered necessary for the SAA drift analysis.

Now, it shall again be assumed that the observed changes in event positions are due to space weather and that hot spots do not influence event positions. Only if the given observation cannot be explained on these premises, hot spots shall be used in the explanation. Drawing conclusions from single event average position to the state of Earth's magnetic field or space weather is rather involved, because it requires retracing complex processes from only one observation, the single event SAA position. Rules of thumb can be given on the basis of the assumption, that timely and spatially changing space weather influences lead to a change in the average position of event occurrences. This assumption requires to consider changes in magnetic field due to geo-internal processes as negligible, but given the timescales on which the internal processes of Earth's magnetic field evolve discussed in section 2.2, in contrast to the time-scale considered here, this assumption is valid.

The following guidelines are made:

- When average event positions cluster at a point for a longer time, it means that the magnetic field has been quiet for that time frame, and in turn, solar activity has been quiet for that period.
- If an effect is only visible with one satellite, it means that the effect vanishes, or significantly loses strength, on a time scale of less than 5 days, coming from the phase delay between the ground tracks. If an effect was to hold for more than 5 days, both satellites would pass the position at which the effect is working, which would result in the difference being compensated. In the depiction of event trajectories, Figure 5.11, any difference between the satellites is depicted for more than 5 days, but this is because they are part of the MA window that captures the short-term effect for a longer time frame.

  For the working of mentioned short-term effects, following hypothesis is formed: It was introduced in subsection 2.2.5, that size and peak particle fluxes of the SAA react to geomagnetic storms. Because of the phase delay between the satellites, this could result in one satellite seeing a storm-enhanced SAA, while the other, passing later, experiences a quiet SAA. The recovery of the SAA, indicated by area and peak flux, following geomagnetic storms is as fast as DST recovery, so a significant difference in SAA particle environment can be seen between the visits of S2A and S2B. [46]

  The difference in event position could also have its origin in timely differing operation or state of the SRAM memories, so that the devices react to different particles and thus experience events at different position. This has however been ruled out in the given theoretical framework, because both satellites and their components are assumed to behave exactly the same.

Coming back to Figure 5.11, Figure 5.12 and subsection 5.3.1, these rules of thumb allow to explain some of the observations. It can be seen that the event trajectory of S2A clusters before mid December 2016 (cyan), which indicates times of little SAA dynamic. The same applies to the time span of April 2018 to July 2018 (orange, Figure 5.12), during which both satellites show some tendency for event clustering. These observations are confirmed on the basis of DST not indicating any geomagnetic storms or irregular activity for these time frames. Thus, quiet geomagnetic conditions are a prerequisite for quiet event positions.

When there are differences, it is possible to differentiate the deviating satellite from the one which event positions evolve regularly by visual inspection of the 3D plot. The 3D plot shows, that event trajectories generally move slowly, but in times when there are differences between the two satellites, it can be seen that this difference is accompanied by fast movement of the events one satellite, whereas the other satellite's events continue to move a constant pace. Figure 5.17 gives an example of this, where S2A breaks out in longitudinal direction, moving much faster than S2B.

The purple, short-lived deviation of S2A in Figure 5.12 is likely explainable by SAA distortion, following the September 2017 CMEs. Despite the strong geomagnetic storm, spatially speaking this deviation is minor.

In contrast, the blue northwards deviation of S2B in Figure 5.12 is rather strong and not explainable within this theoretical framework, because it shows a strong drift in event position, and therefore SAA position, despite no geomagnetic storm being indicated by DST.

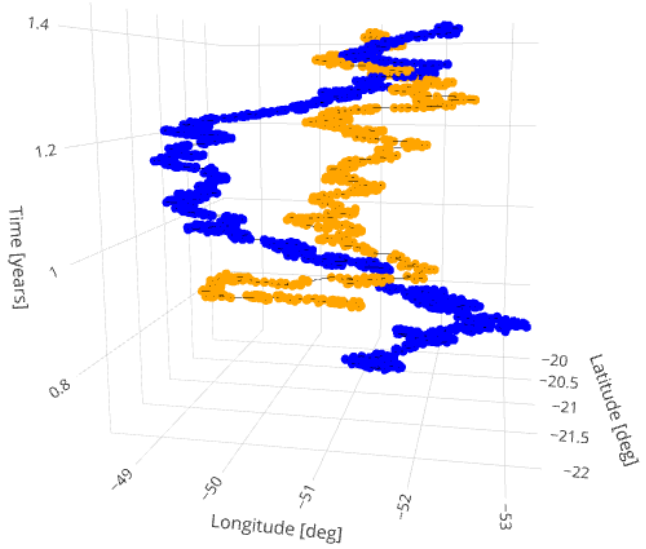


Figure 5.17: Example of event trajectory deviation in 3D (blue: S2A, orange: S2B)

Finally, S2B shows a position deviation at the end of its current trajectory (red). There is a CME in late August 2018, however the deviation already begins in mid August. Because a right-sided MA filter has been used in processing of the data, future values do not influence the present, and so this deviation can again not have its origin in solar activity. It may however not be excluded that the August 2018 geomagnetic storm has fortified the deviation.

The circular behaviour from May 2017 to to September 2017, but also the hook described from April 2018 to August 2018, show that there is a repetitive circular component to the position changes. From the time scale on which these occur, it is cannot be excluded that they are linked to the semi-annual component of SAA activity. However, this observation cannot be linked to any existing study.

Given the complexity of the system trapped particles/magnetic field, trying to explain such an obscuring element any further is not considered fruitful. Since the assumptions have resulted in accepting geomagnetic storm as the only possible explanation, they have to be revisited in light of the mismatch between geomagnetic storm and event position deviation of S2B (blue).

Either, there is a space weather related phenomenon not currently being considered, which is seen as unlikely, or there are platform effects that are not covered yet. It is thinkable, that:

- The scrubbing interval did not remain at constant 20 seconds. If scrubbing period changed during the mission, then averaging out errors is no longer possible consistently, which could cause a short-term deviation in the event trajectory that is corrected automatically as more events occur to compensate the error.
- The operation of the two satellites differed. Different attitude of the platform leads to a different level of effective shielding, because protons in the SAA are not omnidirectional. If one of the satellites changes its attitude, this will also change the SAA position where events occur normally.
- Hot spots within the memories that occurred during SAA passes changed the behaviour of the device, making some memory cells particularly susceptible to particle energies that are found in different regions of the SAA. This hypothesis can be evaluated by looking into the physical addresses at which these deviating events occurred. According to [61], the addresses are part of the single bit error handling process, and in case they are saved and down-linked, the physical proximity of single event affected memory cells can be studied to confirm or refute the role of hot spots. As already mentioned in subsection 4.2.2, repeated accesses to the Flash memory bad block table can be a cause of such a hot spot.

Lastly, the results of linearized longitude and latitude SAA drift are discussed. SAA drift rate is usually given linearly in literature, so this type of modelling is used here as well for comparability [54, 57].

In Table 5.1, drift rates in the same direction and within the same order of magnitude as summarized by Schaefer et al. [54] are found. In detail however, they differ. Some of the differences could be due to Schaefer et al. only providing results up to 2011, and current times showing different SAA drift rates. But also, there is a discrepancy between latitudinal drift as observed on S2A versus the observed drift on S2B. Inspection of Figure 5.11 yields, that longitude has not been affected by strong event position deviations as much as latitude, thus the better agreement of S2A and S2B. Finally, the latitudinal drift observed by S2A is expected to be more realistic, since S2A has been subjected to fewer sudden event position changes.

5.4.2. Correlations

The scatter plots of orbit event rate vs. DST (Figure 5.13 a and b) are similar among the satellites. No clear relationship is visible between either event rate/$F_{10.7}$ or event rate/DST.

Throughout the solar cycle, DST remains almost constant in its mean, but becomes more volatile when CMEs become more abundant. The responses of event rate to spikes in DST, indicating geomagnetic storms, observed so far do not allow for finding a clear relationship. However, this observation was made for for rather weak DST spikes. With more severe geomagnetic storms, deriving a clearer relationship between DST and event rate will become possible, because these also lead to a clearer response in particle fluxes [31]. It is therefore indicated to continue evaluating this relationship with future data.

$F_{10.7}$ takes on higher values in times of higher solar activity. Up to now, only little data was gathered at times of high solar activity, namely during solar flares. The short time in which solar flares cause an increase in $F_{10.7}$ are seemingly too short for the SAA particle fluxes respond visibly. This is underlined by the previous discussion of event rate responses to space weather. As the solar cycle progresses, the time span of data coverage with high $F_{10.7}$ will increase. Then, the SAA will be able to react by reducing its particle fluxes. By evaluating proton flux over a time span of 26 years, Anderson, Rich, and Borisov [64] showed that the proton flux changes in the order of 35% over one solar cycle. On a 2 year time scale, as was considered in the analysis here, a change will not be noticeable due to the sparsity of single event data, but when more data at increased $F_{10.7}$ becomes available, it will be possible to see better correlation. The additional data will allow for finding a relationship between event rate and $F_{10.7}$, that is supposed to be commensurate to the linear trend of Figure 5.5. Because of the volatility of DST as response to solar particle events, there are only only short-term deviations from the DST mean, which means that a strong correlation, as could be with $F_{10.7}$, will not be established in the future.

6 Frequency Domain Analysis

In the next analysis, it shall be determined what periodic variations there are in the unfiltered single event signal for S2A+B by spectral analysis. Because in the previous long-term analysis, all periodic variations up to a certain period have been removed from the data through the moving average filter, this dedicated step is necessary to establish a full understanding of the single event rate signal. The single event position signals are not part of this analysis, because no relevant finding is expected there.

As presented in section 2.4, periodic variations can be due to the operation of the satellite as well as space weather. As a result, it will be clarified, if other characteristics than the orbit are strong enough to be visible in the single event rate signal. This is a necessary step to determining if they are quantitatively important enough to be considered in single event rate prediction. To give a concrete example of what could be found: if the solar rotation has an influence on event rate, a peak in the spectrum of the single event signal must be visible around 25-31 days. In that case, an accurate single event prediction would have to consider this fluctuation in event rate.

6.1. Methodology

With spectral analysis, the particular nature of single event data has to be taken into account. In their most basic form, a single event can be characterized purely by a time stamp giving its occurrence. However, in the scope of this project, no method for performing immediate spectral analysis on time stamp data has been found.

Therefore, events have to be binned as described in section 4.1 to produce a time series giving the number of events per time bin. For small bins, up to orbital period, this means that sparse data is created: in most bins, no event happens, while those that capture an event will mostly only capture a single one. These bin sizes hold the finest spectral information, because, as Nyquist criterion states, the minimum period that can be resolved is twice the bin size.

When bins become larger, the number of unique event counts increases, up to bins holding 0 to 12 events when binning daily (see Figure 5.2). Large bins however hide part of spectral information, because the minimum discernible period increases with bin size. With very large bins, the number of unique event counts decreases again, because the number of bins to place in the covered time decreases. At this point however, analysis is not meaningful anymore anyways.

In addition to its sparsity, event data is unsteady as well. When event rate changes, it only changes in steps of 1 with raw signals and does not, like other physical quantities, increase steadily. In addition, a method to analyze the frequency spectrum of single events has to be capable of dealing with missing data and unevenly sampled data. Else, only those time frames of perfect data quality can be analyzed, which in turn limits the analysis to a certain maximum detectable periodicity, because it reduces the longest streak of data coverage. Spectral analysis with missing data and uneven sampling is possible by least-squares spectral analysis [49, 71]. This method relies on fitting periodic functions to the data and evaluating the goodness

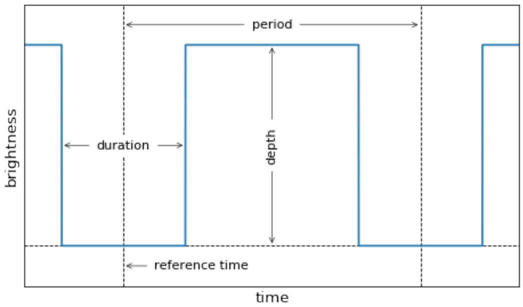


Figure 6.1: BLS basis function, taken from [65]

of fit, which gives a numerical description of the spectral power at the given periodicity.

Given the need for least-squares spectra analysis, and the sparse data with two states on the one hand and the less sparse data with several states on the other hand, two methods to finding the single event signal's periodogram need to be applied:

- Box Least Squares (BLS) periodogram: This type of spectral analysis measures the power of the signal when fitting a top hat function to the signal using least squares method. The top hat, as shown in inverse in Figure 6.1, is defined by a duration, a period and a depth. BLS periodograms are suitable where the signal's pulses are much shorter than the estimated period of the signal. Also, BLS periodograms give a sharp spectral response for such data, but are limited to signals where the on-phase is expected to last 5% of the period. The BLS method is therefore selected for determining the spectrum with small bin sizes. BLS periodograms are originally used in analyzing transit curves of exoplanets to find the orbital period of such planets. [12]
- Lomb-Scargle (LS) periodograms: To perform this type of spectral analysis, sine curves are fitted to the data using the least squares method. The goodness of fit again gives the spectral response of the signal. Unlike classic Fourier transform, it does not require evenly spaced data without gaps, making it appropriate for the given data. In contrast to BLS periodogram, it is not a good match for unsteady changes of state, but allows for growth and decline. It is therefore the better method for the event data sampled to larger bin sizes. [71]

In parallel, a manual analysis of event rate in eclipse versus events when sunlit shall be made. Because eclipses occur once per orbital period, it would not be possible to discern influences due to eclipse from other influences that occur at the same period, like changes in radiation environment. For this reason, the manual approach is chosen, calculating whether an event occurred in eclipse or sunlit from the position of its occurrence.

6.2. Implementation

Both BLS and LS fitting are available through the stats module of the Astropy library.

For the BLS spectrum, event data is sampled to bins of 5 minutes for BLS analysis. At an orbital period of more than 100 minutes, this allows for one event per orbit without crossing the 5% threshold required for proper BLS spectral analysis. In contrast to the previous long-term analysis, not only SAA events are considered, but also events occurring outside of SAA, because there is no need for such restriction. The Astropy BLS implementation needs top hat durations as an input as well and the one duration that results in the best fit. Since input durations are unknown, a wrapper around the original Astropy BLS function is written: For each probed period, linearly spaced top hat durations from zero seconds up to the duration of the tested period shall be used for fitting to the data. The top hat duration of best fit shall be selected as the actual result of the analysis. The number of durations in the linear space is left to the user and can be

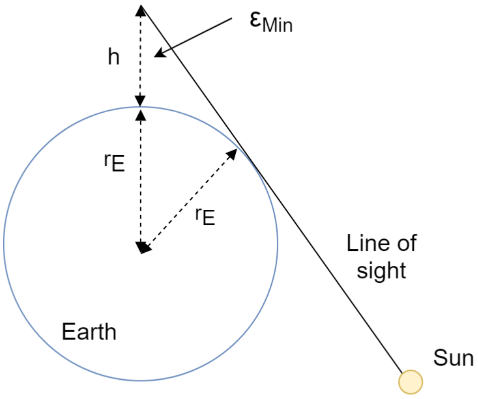


Figure 6.2: Minimum elevation angle for sunlit satellite

selected through a trade-off between computing time and accuracy.

Single events are placed in bins of 6026 seconds (orbital period) for LS spectral analysis. LS spectral analysis as implemented in Astropy can immediately be applied to the data and returns the power of the selected frequency.

In both analyses, the probed periods, are spaced linearly and are selected by iteratively calculating the spectrum until no more discernible change occurs through further refinement of the frequency grid. The quantitative importance of the found periodicities shall be evaluated on the basis of how many events they cover (BLS analysis), respectively how strong the periodic differences are (LS analysis). BLS is applied up to a sampling rate of 12 hours, up to 48 hours BLS and LS periodograms are used, and beyond that, only LS is used.

To determine whether an event had happened in sunlight or in eclipse, the elevation angle between sun and the satellite's position is calculated through the Python library PySolar. This indirect approach is chosen, because PySolar is built for terrestrial photovoltaics applications, and so no direct evaluation of eclipsing is available. As position of the satellite, the latitude and longitude of event occurrence and a height of 786 km above Earth's surface are used. If the calculated elevation angle is smaller than the minimum elevation at which the satellite can see the Sun, determined in Figure 6.2, the satellite was in eclipse.

$$\epsilon_{min} = -\frac{\pi}{2} - \sin \frac{r_e}{r_e + h} \tag{6.1}$$

In parallel, it was determined whether an event occurred within SAA or outside of it, to allow for determining the influence of eclipses for both cases individually. In doing so, the SAA window of subsection 2.2.5 had been applied.

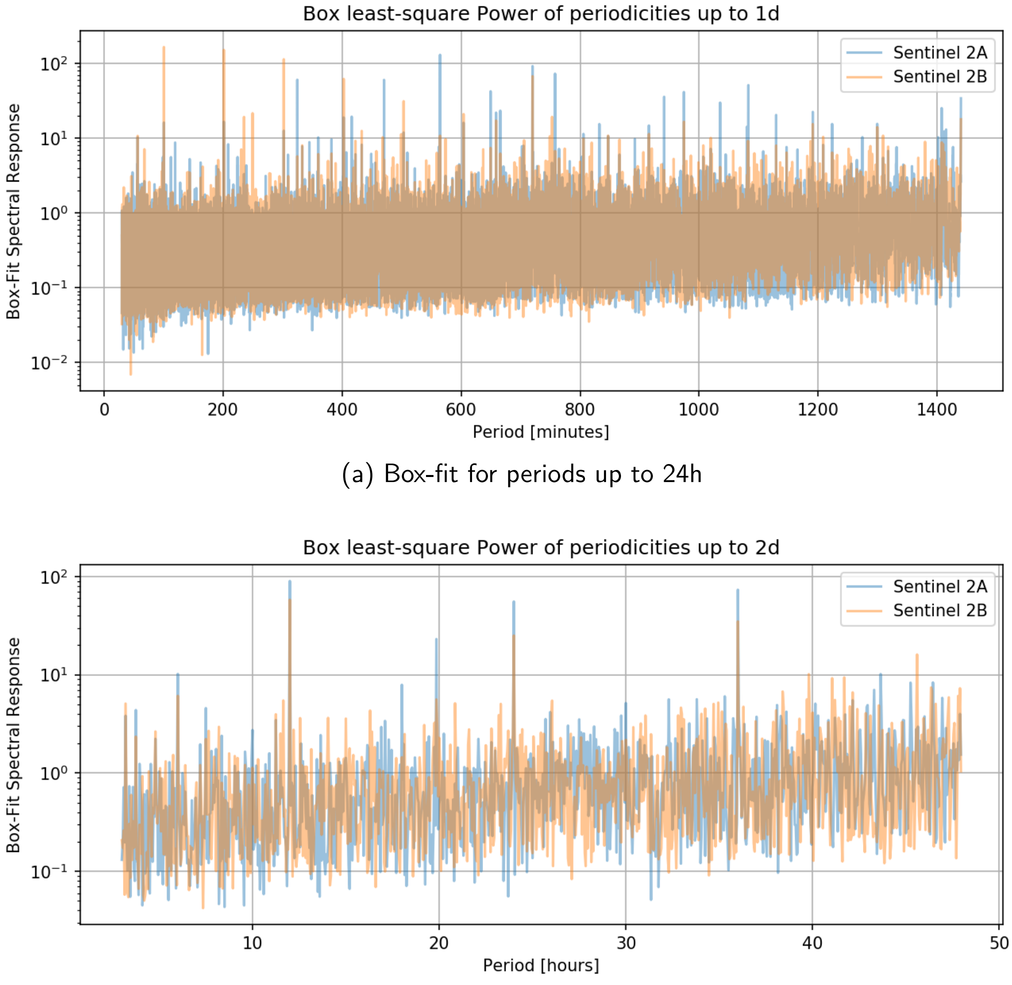
All of the steps above are implemented in listing A.4 using functions described in section A.5.

6.3. Results

Figure 6.3 shows the spectral results to BLS spectral analysis and Figure 6.4 shows spectral response for LS spectral analysis.

The BLS periodogram in Figure 6.3 (a) shows spikes at the orbital frequency of 6026, respectively 100.4 minutes, and its harmonics. Next to that, there is a spike at 12h and 24h. Spikes at 12h and 24h return in Figure 6.3 (b) and are followed by a spike at 36h, likely a harmonic of the 12h period.

The LS spectrum in Figure 6.4 (a) also shows spikes at 6h, 12h and 24h. Figure 6.4 (b) and (c) do not show any spikes, which means that there is no periodic response in the single event signal at high periods.



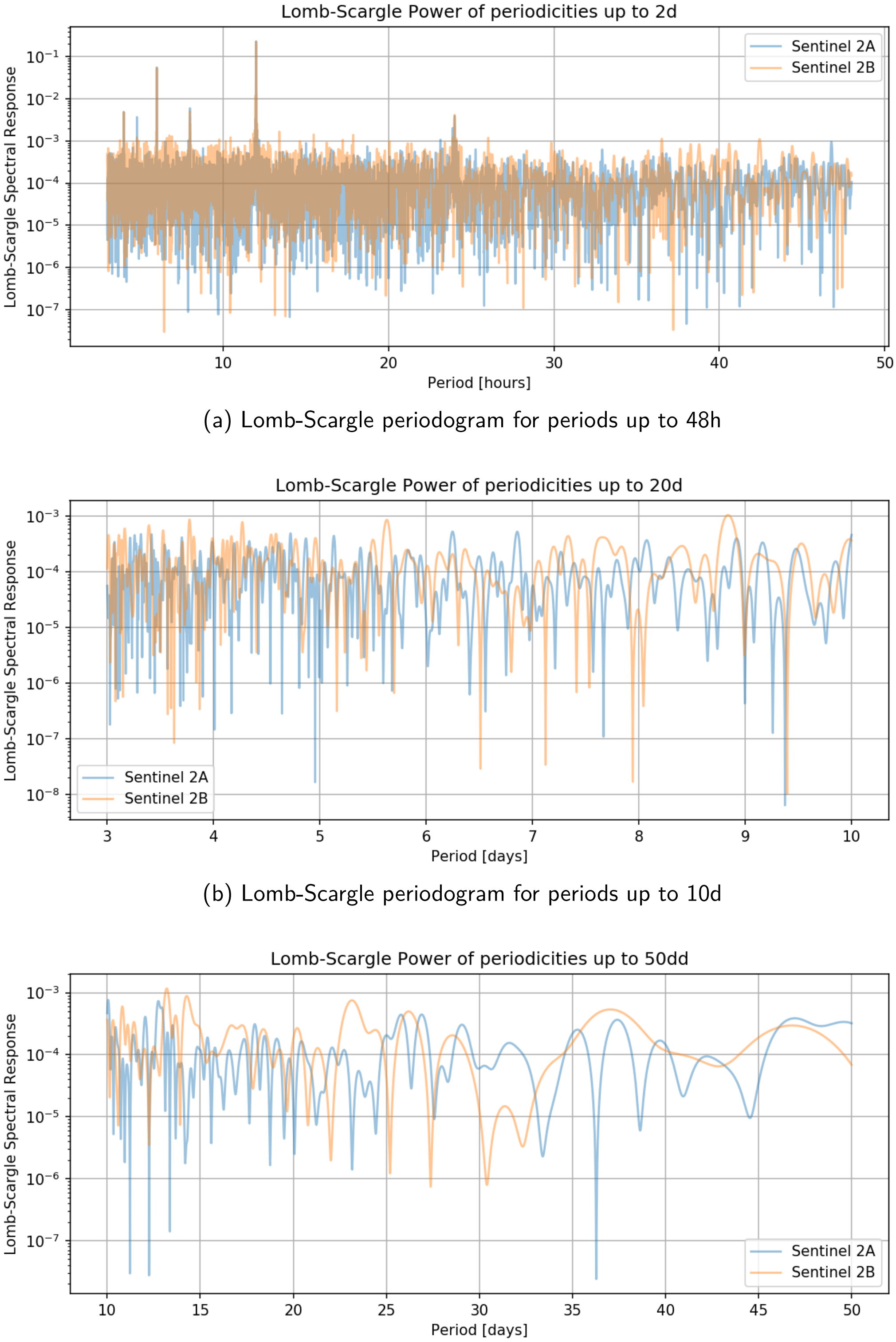
(a) Box-fit for periods up to 24h

(b) Box-fit for periods up to 48h

Figure 6.3: BLS periodograms

Satellite	OOA Eclipsed $[y^-1]$	OOA Sunlit $[y^-1]$	SAA Eclipsed $[y^-1]$	SAA Sunlit $[y^-1]$
Sentinel 2A	159.48	296.76	10509.48	10422.43
Sentinel 2B	232.31	314.16	9917.76	10780.87

Table 6.1: Event occurrence: SAA vs non-SAA, eclipse vs. sunlit



(a) Lomb-Scargle periodogram for periods up to 48h

(b) Lomb-Scargle periodogram for periods up to 10d

(c) Lomb-Scargle periodogram for periods up to 60d

Figure 6.4: LS periodograms

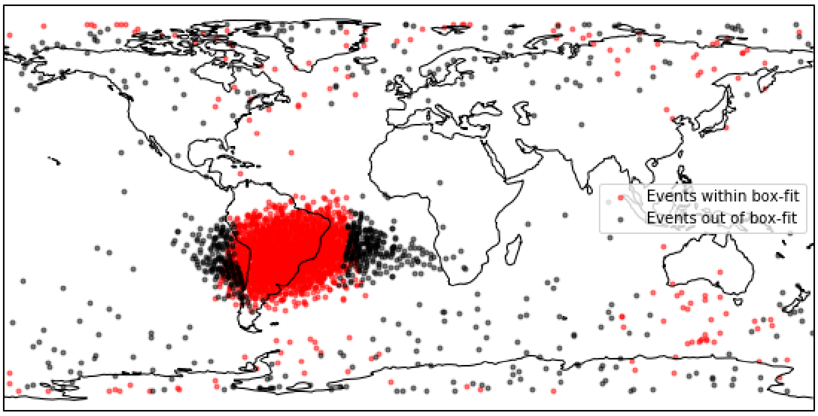


Figure 6.5: Events in/out of 12h Box-Fit

Table 6.1 shows the differences in event rate between eclipse and sunlit, and SAA and out of SAA

6.4. Discussion

Figure 6.3 and Figure 6.4 show, that with any of the spectral responses, the peaks of S2A and S2B are in good agreement, which confirms the assumption that the two satellites behave similarly.

Considering Figure 6.3, two types of common relevant peaks can be seen. In Figure 6.3 (a), the only peak is found at the orbital period or multiples of that. In Figure 6.3 (b), peaks are found at 12 hours and its harmonics as well as at harmonics of the orbital period. These peaks all go down to the same root cause, the SAA. The peak at orbital period arises, because the SAA is passed once per orbit by the satellite.

The 12 hour peak marks the time it takes for the satellites to pass the SAA in its full width, which happens every 12 hours. One can depict the SAA as an obstacle that moves through S2A orbit once in 12 hours, alternating between night-side and day-side. When using the box-fit for the 12h period and applying it to the event data of S2A, Figure 6.5 results. This figure shows the events that have been placed within the 12h fit in red. These are all the events occurring during the central passes of the SAA, which amounts to 57% of the total event count when considering S2A. Through the identification of this well-understandable single event rate variation, the methodology itself is verified.

Table 6.1 shows that there is a stronger difference between eclipsed and sunlit out of SAA events than there is between SAA events. This effect cannot be purely temperature related, because then there would have to be the same different in event rate eclipsed/sunlit for the SAA.

The differences in SAA events between eclipse and sunlight are more subtle and do not give a strong tendency. For S2A, event rate in eclipse is slightly higher than in sunlight, but for S2B the situation is flipped and stronger. Further data could help to clarify this, in case the event rates converge to the same values.

The analysis has revealed, that the only meaningful periodic influence is due to the orbital period and the therewith changing radiation environment of the satellite, be it on single orbit time scale or on 12h time scale. Other, more subtle, changes in the event rate drivers have not been recorded by single event rate of S2B. Also, there is no indication of periodic hot spots occurrences, causing deviations in event rate.

Performing the analysis again, once more data is available seems reasonable. Also, as the solar cycle progresses, the response on to solar rotation will be stronger because more surface-bound features will appear with the Sun.

7
Conclusion

7.1. Synthesis

7.1.1. Summary

On the basis of the idea, that single event counters are not only witnesses of malfunctions, but also sensors for space weather, satellite operations and device behavior, research questions have been formed to clarify what their actual potential is. The satellites Sentinel 2A and 2B served as platform for this study.

The background theory of single event effects and space weather have been reviewed. These were linked to the characteristics of the Sentinel 2 mission, which allowed to make hypotheses about what influences might be visible in the data and how to analyze them. It was established that both time domain analysis and a frequency domain analysis are necessary, because the noisiness of single event data will not allow for studying both long-term influences to single event rate and short-term influences at the same time.

In the time series analysis, noise was removed using a MA filter. It was shown through analysis of the residuals, that, along with the noise, the variation of event rate due to the 12h SAA revisit pattern was removed as well. A linear trend was identified that is similar for the two satellites and attributed to a combination of the 11 year solar cycle and component degradation. Future data can make it possible to distinguish between the two. The trends, as well as the mean filtered event rate, comply well among the satellites. Based on radiation belt theory, the variations in filtered event rate should have their cause in SAA seasonal effects and geomagnetic storms. Because these variations are different between the two satellites, hot spots in the memories whose event counters are studied likely cause these differences. An analysis of memory addresses can clarify this. Also, future data, especially data involving stronger geomagnetic storms, will provide further insight.

The position drifts of the SAA as seen by single events were extracted from the latitude and longitude information of the single events. In this data, the two satellites document the movement of the SAA center and comply during most of the time covered. However, strong deviations can be seen for short periods of time. In one instance, this is attributed to a geomagnetic storm, but not all deviations can be linked to storms. Further study on this issue is needed, and again memory hot spots are recommended as a starting point.

The spectral analysis is based on different methods, required to respect the nature of the data, and shows that the orbital period and the semi-diurnal period are the main short-term influences on event rate. In the future, further periodic phenomena might become visible, such as the solar rotation period, but currently there is too little data to reach a conclusion. An analysis of event rate in and out of eclipse shows that event rate with the SAA does not depend on eclipses, but out of SAA the event rate is twice as high in eclipse. Possibly, the increased temperature of the satellite can explain this.

Table 7.1 summarizes the results of the analyses for the single event rate influences hypothesized in subsection 2.2.5. Some of these results shall be considered as preliminary, because future data could reveal a different connection between these space weather effects and single event rate. Some of the preliminary results also imply the need of other measures to establish their true role.

Influence	Period	Analysis	Influence on Event Rate
Orbital Period	100.4 min	Spectral (Box-Fit)	Strong
Eclipses	100.4 min	Spectral (Manual)	Ambiguous
Semi-diurnal	12 h	Spectral (Box-Fit)	Strong
Solar rotation	25-31 days	Spectral (Lomb-Scargle)	No effect (TBC)
SAA Semi-annual	182.5 days	Time Series	Ambiguous
Annual	365 days	Time Series	No effect (TBC)
Solar cycle	11 years	Time Series	Weak (TBC)
Degradation & Aging	permanent	Time Series	Weak (TBC)
Geomagnetic storms	instantaneous	Time Series	Ambiguous
Hot spots	instantaneous	Time Series	Ambiguous

Table 7.1: Consolidation of event rate analysis results

Table 7.1 shows that the major influence is due to passes of the SAA that occur with orbital frequency and 12h frequency. Other influences are not notable or there is too little data to judge their existence conclusively. Operations seem not to influence single event rate at all. The SAA semi-annual cycle is different, because there is some variation of single event rate occurring on this time scale, and thinkable to be caused by the semi-annual SAA period, it cannot yet be confirmed what it is linked to.

7.1.2. Findings

The main findings of this thesis are:

- The long-term trend of event rate can be modelled linearly and contains degradation and solar cycle. Once the solar cycle will change, a more complex regression method has to be found, but will also reveal by which fractions the two components affect event rate individually.
- There is a variation in filtered single event rate, that is caused by the seasonal behaviour of the SAA, geomagnetic storms and hot spots in the memories. It was not possible to distinguish between these any further, but simulations and additional data can provide further insight.
- Event positions can be used to find the linear drift of the SAA. The drift rates differ between the two satellites due to short-term differences in their filtered event positions. Likely, these are caused by memory hot spots and geomagnetic storms.
- The spectrum of single event rate shows peaks at orbital period and at 12h. The 12h peak is due to the revisit of the SAA; the peak at orbital period is trivial. No other strong periodicities, such as an annual one or one due to solar rotation, were found.
- Eclipses do not affect event rate in SAA, but outside of SAA, event rate is almost twice as high in sunlit than in eclipse.

Lastly, the fulfillment of the three motivational goals, introduced in chapter 1, shall be evaluated:

1. For now, single event counters cannot be used as a space weather sensors. With the variations of single event rate and position it is not yet clear what component is due to space weather and what component is due to other causes, e.g. hot spots. Future study will allow for resolving this problem, so that in the usage of single event counters, this is just an obstacle but not a final indication for impossibility. Therefore, the fulfillment of this goal has been prepared.

2. Device effects on the long-term are likely to exist, but superimposed by the solar cycle. They will become visible more clearly once the solar cycle changes direction and thus, this goal has not been achieved finally, but a way forward is given.

3. It was confirmed that the only relevant effect of satellite operation on single event rate is the orbit, respectively its passing of the SAA. Therefore, the third motivational goal can be closed without having resulted in the discovery of unforeseen satellite operation single event influences.

7.2. Reflection

To some degree, the usefulness of single event counters for space weather research has been confirmed. The applied method allows to resolve the SAA position drift from single event counters with a resolution not achieved by previous single-event based methods, such as the cross-correlation method used by Lauriente, Vampola, and Gosier [8]. To the author's knowledge, there is only one study on the relationship between single event rates and the solar cycle up to now [7], that analyzes the link between single event rate and solar cycle. In contrast to that study, the research here is based on a device of high event rate and therefore allows for more resolved results and for a numeric link between single event rate and solar cycle. The added resolution however increases complexity of event rate interpretation, so that there are inconclusive results, such as the role of the semi-annual SAA cycle and the reasons for event position deviations, demanding for further research. If it turns out that these position deviations and the changes in event rate are due to hot spots in the memories, this thesis provides first witness of their occurrence in orbit.

In the following, it is reflected on the planning and execution of the given research project. Initially, one of the goal of this research was to find machine learning methods applicable to single event data and to evaluate their usefulness in the prediction process. The rationale was to follow the current surge of machine learning methods in many fields of research. Aside from the ambiguity of how to apply such methods with single event data, this thesis gives witness, that classical statistical approaches have not been applied in full with single event data yet. Results obtained from classical methods are also more easily to interpret and are therefore a better way of advancing single event data analysis.

Further, in the beginning of this project, some aspects of the analyses' implementations have been misjudged. It was assumed that large amounts of space weather data would be needed for relating the single event counters to their radiation environment, but later it became clear that there are only two meaningful variables. Also, alongside the research project here, there was the goal of building a Python platform for convenient and interactive study of time series data, focusing on single event data. It turned out that this step was premature, because at that point in time it was not clear yet what methods are actually needed in single event data analysis, because the potential of the data had not yet been judged completely. Also, the different data handling and visualization demands of time series data and position data, which are the two sides of single event data, make it difficult to develop a fully-integrated solution.

Next to that, this thesis has been of rather qualitative than quantitative nature and leaves space for improvement in this regard. There has not been a consideration of magnetic drift shells, but the insight gained in literature have been generalized for the entire LEO and SAA environment. This neglects, that there might be important differences in particle energies and magnetic drift shells. Also, this thesis ignores time delays: when a geomagnetic storm causes particle distributions to change, the effect might not instantaneously affect the S2A+B satellites. Particles diffuse radially inwards, lowering their magnetic drift shell and thus altitude over time [52]. Then a geomagnetic storm is visible in the orbit of S2A+B, but only later when particles that would have reached that orbital height are lacking. A similar issue is, that so far no comparison of single event rate data against radiation belt simulations has been made, which is why this is one of the main recommendations of this thesis. These could account for time-delays in particle distributions.

Finally, this thesis placed little emphasis on the technical workings of SRAM memories. This might mean that some influence factors have been overlooked or judged wrongly in their prominence. In particular, this affects the role of hot spots with SAA position drifts.

7.3. Way Forward

The following section gives recommendations for the immediate continuation of the research project at hand and provides a vision for the analysis and use of single event data, which shows its ultimate potential.

7.3.1. Recommendations

To continue the line of research opened in this thesis, the following recommendations are given:

- It is recommended to study future data of MST00495 from S2A+B, as has been laid out in section 5.4 already. This aims at clarifying the role of geomagnetic storms and separating degradation influences from solar cycle influences.
- It is recommended to analyze temperature data from the MMFU of S2A+B, available through the readings of three thermistors [61]. Thus, it will be possible to judge what temperature variations occurred and how these might have affected single event rate. Possibly, a correlation can be found.
- It is recommended to evaluate the memory addresses at which single events occurred, as this will clarify the role of memory hot spots. According to [61], when a single bit error occurs in the MSS SRAM, the address of the affected event is saved. By analyzing the physical proximity of affected memory cells, it can be clarified if the position deviations in filtered event position and the variations in filtered event positions were caused by hot spots.
- It is further recommended to go back to the counters that have been discarded in chapter 4. Some of these were discarded for showing implausible data, but it could be possible to clean this data and hence make use of this data. This will allow to study which particle types and energies the devices are sensitive to and will allow to show when solar particle events caused events in radiation hard devices. Nevertheless, none of them will provide the same amount of information as MST00495 does, because there are much less events with the other counters.

7.3.2. Potential

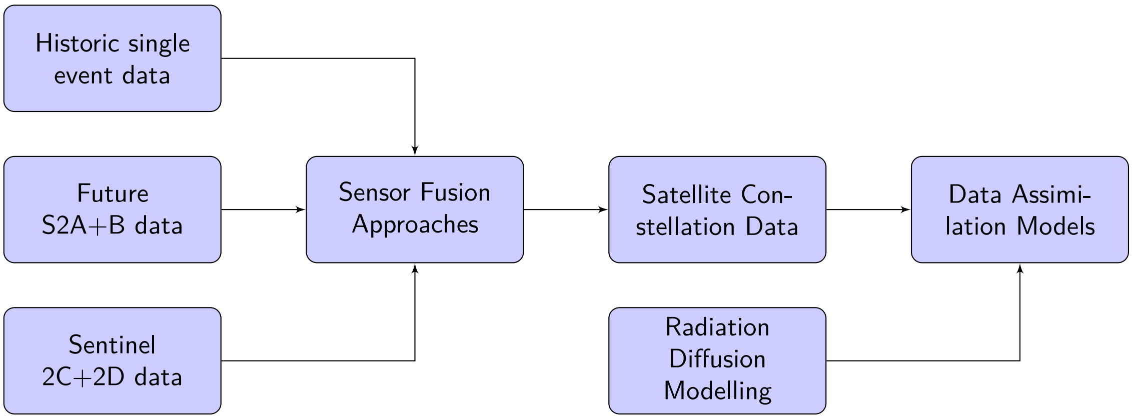


Figure 7.1: Single event data vision

The following ideas mark the conclusion of this research project. They integrate this thesis into a larger framework and provide a vision of single event data use, shown in Figure 7.1, reflecting different stages of it. Because single event data has received little attention yet, many ideas point at other types of single event data to be analyzed.

1. **Historic Data**: One limitation of this research was the data coverage of only about 2 years. This has limited the study of the relation between single event rate and degradation and solar cycle. Possibly, there is single event data of missions already finished that covers larger time frames. This data could shed some more light on this topic.

   Historic data also entails historic data from the Sentinel 2 missions on phenomena other than single events. Here, event addresses and MSS temperature shall be mentioned.

2. **Future Data**: The time series analysis has shown that future single event counter of MST00495 needs to be studied to gain a clearer understanding of its response to geomagnetic storms. Because some of this data is already available, and because with this research an adaptive analysis in the form of a Jupyter notebook is provided, it is possible do perform with little effort.

3. **Sentinel 2C and 2D**: The launch of Sentinel 2C and 2D will contribute to a preciser understanding of the relationship of degradation effects, because they bring new components to space, which allows for studying degradation differences.

4. **Sensor Fusion**: In theory, it is possible to combine multiple single event counters in a sensor fusion approach, creating a single, more accurate sensor for the respective radiation environment. The higher the event rate of the counters to be combined, the better quality the resulting sensor. What needs to be considered when combining single event data of different source is:

   - The counters have to observe radiation from behind a comparable shielding. In that respect, it is not so much the overall shielding thickness that is problematic, but the distribution of shielding around the device. In LEO, the east-west effect causes particles to impinge in inhomogeneously. Next to that, the attitude of the satellite has to be taken into account as well, as it can lead to a timely changing shielding strength.
   - The devices which counters are combined have to be comparable in terms of upset threshold and cross section. Also, when combining memories of different capacity, they have to be scaled for their number of cells.
   - Sensor fusion has to include a hot spot detection, so that their irregular influence can be erased. This requires to include device addresses in the process of sensor fusion.

   The first two points show that single event sensor fusion is only practically possible with identical satellites. Therefore, it is proposed to evaluate the possibility of such an approach with the single event data of S2A+B to create a more accurate radiation environment sensor. One must however keep in mind, that the differences between event counters also provide information on time scales in the order of magnitude of revisit time, that might be lost.

5. **Satellite Constellations**: In the past, constellations of spacecraft were exceptions. Through recent developments in space industry, these will become more common in the future, especially in LEO. An example of a future LEO constellation is OneWeb[1], which shall comprise 648 identical satellites in an orbit of 1200 km height in its final configuration[2]. If there is a device aboard of similar event rate as the one studied in this thesis, approximately 7000 events will be registered per day in different locations. This will pose a significant improvement in data quality with respect to S2A+B and will allow for a meaningful improvement in data quality. It is hence recommended to search for access to such data and to study its responses to radiation environment changes and operational changes.

6. **Data Assimilation**: Data assimilation is the approach of combining direct and indirect observations with theoretical models to forecast the state of a system. It is widely used in Earth sciences and relies mostly on probabilistic methods and Monte Carlo methods. Single event data, as well as other unrevealed space weather data, are observers of the space radiation environment system and as such provide information about that system. The usage of single event data in a data assimilation model of the radiation belts marks the ultimate step in single event data usage. It has the potential of contributing to real time models of particle distributions. Because data assimilation is limited by

[1] www.oneweb.world, accessed 18.04.2019

[2] https://directory.eoportal.org/web/eoportal/satellite-missions/o/oneweb, accessed 18.04.2019

spatial coverage, the usage of data from only a single orbit will not be sufficient, and so the data assimilation step builds upon the usage of satellite constellation data. [28]

Some efforts of building radiation belt data assimilation models are already ongoing and focus on past particle count data and radiation belt diffusion models [33]. Single event data is not currently part of such models and is of much lower information content than particle counters, but they have the advantage of being ubiquitous and so their applicability should be evaluated.

Lastly, it shall be mentioned that there are other unintended space weather sensors aboard satellites than single event counters. Ecoffet [38] provides the following ideas:

- Dark current increases: Payloads with imaging electronics can detect fine variations in pixel dark current when the imager protected from outside illumination. These dark current variations are associated to the radiation environment and can therefore provide space weather information. However, their potential is rather low, because this data is only generated when the imager is not in operation.
- Frequency shifts in oscillators. Some payloads require very accurate oscillators, that can shift in frequency when subjected to tiny amounts of radiation dose. These frequency shifts are also associated with the radiation environment and can identify the contour of the SAA. Thus, they can provide additional radiation environment information and do so regardless of operation.

To these types of data, the same research questions as the ones of this thesis can be applied.

List of References

[1] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. Online. 2001–. URL: `http://www.scipy.org/`.

[2] Met Office. *Cartopy: a cartographic python library with a matplotlib interface*. Exeter, Devon, 2010 - 2015. URL: `http://scitools.org.uk/cartopy`.

[3] John A. Lockwood. "Forbush decreases in the cosmic radiation". In: *Space Science Reviews* 12.5 (1971), pp. 658–715.

[4] A. B. Campbell and W. J. Stapor. "The Total Dose Dependence of the Single Event Upset Sensitivity of IDT Static RAMs". In: *IEEE Transactions on Nuclear Science* 31.6 (1984), pp. 1175–1177.

[5] Thomas F. Tascione. *Introduction to the space environment*. Malabar, Florida: Orbit Book Co., 1988.

[6] E. L. Petersen et al. "Rate prediction for single event effects-a critique". In: *IEEE Transactions on Nuclear Science* 39.6 (1992), pp. 1577–1599.

[7] R. Ecoffet et al. "Influence of solar cycle on SPOT-1,-2,-3 upset rates". In: *IEEE Transactions on Nuclear Science* 42.6 (1995), pp. 1983–1987.

[8] M. Lauriente, A. L. Vampola, and K. Gosier. "Experimental Validation of South Atlantic Anomaly Motion Using a Two-Dimensional Cross-Correlation Technique". In: *Radiation Belts: Models and Standards*. Geophysical Monograph Series. 1996, pp. 109–117.

[9] G. D. Badhwar. "Drift rate of the South Atlantic Anomaly". In: *Journal of Geophysical Research: Space Physics* 102.A2 (1997), pp. 2343–2349.

[10] George C. Messenger and Milton S. Ash. *Single event phenomena*. New York: Chapman & Hall : International Thomson Publishing, 1997.

[11] J. R. Heirtzler. "The future of the South Atlantic anomaly and implications for radiation damage in space". In: *Journal of Atmospheric and Solar-Terrestrial Physics* 64.16 (2002), pp. 1701–1708.

[12] G. Kovács, S. Zucker, and T. Mazeh. "A box-fitting algorithm in the search for periodic transits". In: *Astronomy and Astrophysics* 391 (2002), pp. 369–377.

[13] Peter Foukal. *Solar astrophysics*. 2004.

[14] John M. Goodman. *Space weather & telecommunications*. 2005.

[15] M. D. Looper, J. B. Blake, and R. A. Mewaldt. "Response of the inner radiation belt to the violent Sun-Earth connection events of October–November 2003". In: *Geophysical Research Letters* 32.3 (2005).

[16] Martin Walt. *Introduction to geomagnetically trapped radiation*. Cambridge; New York: Cambridge University Press, 2005.

[17] P.J. Brockwell and R.A. Davis. *Introduction to Time Series and Forecasting*. Springer New York, 2006.

[18] P. Olson and H. Amit. "Changes in earth's dipole". In: *Naturwissenschaften* 93.11 (2006), pp. 519–542.

[19] Walter G. Sannita, Livio Narici, and Piergiorgio Picozza. "Positive visual phenomena in space: A scientific case and a safety issue in space travel". In: *Vision Research* 46.14 (2006), pp. 2159–2165.

[20] J. R. Schwank et al. "Effects of Total Dose Irradiation on Single-Event Upset Hardness". In: *IEEE Transactions on Nuclear Science* 53.4 (2006), pp. 1772–1778.

[21] Volker Bothmer and Ioannis A. Daglis. *Space weather : physics and effects*. 2007.

[22] Jean-Claude Boudenot. "Radiation Space Environment". In: *Radiation Effects on Embedded Systems*. Ed. by Raoul Velazco, Pascal Fouillat, and Ricardo Reis. Dordrecht: Springer Netherlands, 2007, pp. 1–9.

[23] Christine E. Hellweg and Christa Baumstark-Khan. "Getting ready for the manned mission to Mars: the astronauts' risk from space radiation". In: *Naturwissenschaften* 94.7 (2007), pp. 517–526.

[24] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[25] Fernanda Lima Kastensmidt and Ricardo Reis. “Fault Tolerance in Programmable Circuits”. In: *Radiation Effects on Embedded Systems*. Ed. by Raoul Velazco, Pascal Fouillat, and Ricardo Reis. Dordrecht: Springer Netherlands, 2007, pp. 161–181.

[26] ECSS Technical Authority. *Methods for the calculation of radiation received and its effects, and a policy for design margins*. 2008.

[27] ECSS Technical Authority. *Calculation of radiation and its effects and margin policy handbook*. 2010.

[28] W. Lahoz, B. Khattatov, and R. Menard. *Data Assimilation: Making Sense of Observations*. Springer Berlin Heidelberg, 2010.

[29] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 51–56.

[30] Steven Morley et al. *SpacePy - a Python-based library of tools for the space sciences*. 2010.

[31] R. S. Selesnick, M. K. Hudson, and B. T. Kress. “Injection and loss of inner radiation belt protons during solar proton events and magnetic storms”. In: *Journal of Geophysical Research: Space Physics* 115.A8 (2010).

[32] N. Yu Ganushkina et al. “Locations of boundaries of outer and inner radiation belts as observed by Cluster and Double Star”. In: *Journal of Geophysical Research: Space Physics* 116.A9 (2011).

[33] Humberto Godinez Josef Koller. *Radiation Belt Data Assimilation: Overview and Challenges*. 2011.

[34] Wes McKinney. *pandas: a Foundational Python Library for Data Analysis and Statistics*. 2011.

[35] Edward Petersen. *Single event effects in aerospace*. 2011.

[36] Josef Aschbacher and Maria Pilar Milagro-Pérez. “The European Earth monitoring (GMES) programme: Status and perspectives”. In: *Remote Sensing of Environment* 120 (2012), pp. 3–8.

[37] M. Drusch et al. “Sentinel-2: ESA’s Optical High-Resolution Mission for GMES Operational Services”. In: *Remote Sensing of Environment* 120 (2012), pp. 25–36.

[38] R. Ecoffet. “Overview of In-Orbit Radiation Induced Spacecraft Anomalies”. In: *Ieee Transactions on Nuclear Science* 60.3 (2013), pp. 1791–1815.

[39] G. P. Ginet et al. “AE9, AP9 and SPM: New Models for Specifying the Trapped Energetic Particle and Space Plasma Environment”. In: *Space Science Reviews* 179.1-4 (2013), pp. 579–615.

[40] Jack J. Lissauer and Imke de Pater. *Fundamental planetary science : physics, chemistry and habitability*. New York, NY: Cambridge University Press, 2013.

[41] A.K. Chattopadhyay and T. Chattopadhyay. *Statistical Methods for Astronomical Data Analysis*. Springer New York, 2014.

[42] Murong Qin et al. “Solar cycle variations of trapped proton flux in the inner radiation belt”. In: *Journal of Geophysical Research: Space Physics* 119.12 (2014), pp. 9658–9669.

[43] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[44] Travis E. Oliphant. *Guide to NumPy*. CreateSpace Independent Publishing Platform, 2015.

[45] Karel Wakker. *Fundamentals of Astrodynamics*. 2015.

[46] Hong Zou et al. “Short-term variations of the inner radiation belt in the South Atlantic anomaly”. In: *Journal of Geophysical Research: Space Physics* 120.6 (2015), pp. 4475–4486.

[47] Li Cai et al. *Experimental study of temperature dependence of single-event upset in SRAMs*. Vol. 27. 2016.

[48] European Space Agency (ESA). “Brochure: Sentinel 2 - Colour Vision for Copernicus”. In: (2016).

[49] J. D. Hartman and G. Á Bakos. “Vartools: A program for analyzing astronomical time-series data”. In: *Astronomy and Computing* 17 (2016), pp. 1–72.

[50] G.V. Khazanov. *Space Weather Fundamentals*. CRC Press, 2016.

[51] Wilfredo Palma. *Time series analysis*. 2016.

[52] Juan G. Roederer, Hui Zhang, and H. Springer-Verlag Gmb. *Dynamics of Magnetically Trapped Particles Foundations of the Physics of Radiation Belts and Space Plasmas*. 2016.

[53] L. Salvy et al. "Total ionizing dose influence on the single event effect sensitivity of active EEE components". In: *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*. 2016, pp. 1–8.

[54] R. K. Schaefer et al. "Observation and modeling of the South Atlantic Anomaly in low Earth orbit using photometric instrument data". In: *Space Weather* 14.5 (2016), pp. 330–342.

[55] G.R. Blackwell. *The Electronic Packaging Handbook*. CRC Press, 2017.

[56] J.D. Cressler and H.A. Mantooth. *Extreme Environment Electronics*. CRC Press, 2017.

[57] A. D. Jones et al. "SAMPEX observations of the South Atlantic anomaly secular drift during solar cycles 22-24". In: *Space Weather-the International Journal of Research and Applications* 15.1 (2017), pp. 44–52.

[58] D. M. Oliveira et al. "Thermosphere Global Time Response to Geomagnetic Storms Caused by Coronal Mass Ejections". In: *Journal of Geophysical Research: Space Physics* 122.10 (2017), pp. 10, 762–10, 782.

[59] Salvatore Pezzino Peter Ehlig. *Error Detection in SRAM*. Tech. rep. Texas Instruments, 2017.

[60] T. Rousselin et al. "Impact of aging on the soft error rate of 6T SRAM for planar and bulk technologies". In: *Microelectronics Reliability* 76-77 (2017), pp. 159–163.

[61] A. Schingale et al. *Mass Memory & Formatting Unit (MMFU) For Sentinel-2 (Unit User Manual)*. Tech. rep. Airbus DS GmbH, 2017.

[62] Michael Waskom et al. *mwaskom/seaborn: v0.8.1 (September 2017)*. Sept. 2017. DOI: `10.5281/zenodo.883859`. URL: `https://doi.org/10.5281/zenodo.883859`.

[63] Man AHL. *Arctic TimeSeries and Tick store*. Oct. 2018. URL: `https://github.com/manahl/arctic`.

[64] P. C. Anderson, F. J. Rich, and Stanislav Borisov. "Mapping the South Atlantic Anomaly continuously over 27 years". In: *Journal of Atmospheric and Solar-Terrestrial Physics* 177 (2018), pp. 237–246.

[65] Astropy Collaboration et al. "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". In: 156, 123 (Sept. 2018), p. 123. DOI: `10.3847/1538-3881/aabc4f`. arXiv: `1801.02634 [astro-ph.IM]`.

[66] Patrick Hornung. *Telemetry Data Mining - A Literature Study on the Evaluation of Single Event Telemetry using Data Mining Methods*. 2018.

[67] *IBM - Why you should use Python for scientific research*. Apr. 2018. URL: `https://developer.ibm.com/dwblog/2018/use-python-for-scientific-research/`.

[68] Wes McKinney. *Python for data analysis : data wrangling with pandas, NumPy, and IPython*. 2018.

[69] *missingno: Missing data visualization module for Python*. July 2018. URL: `https://github.com/ResidentMario/missingno`.

[70] H. Schmidt et al. *Assessment of In-Orbit SEE Observations on OBC to be used in Sentinel-2C/D and Sentinel-6 Missions*. 2018.

[71] Jacob T. VanderPlas. "Understanding the Lomb–Scargle Periodogram". In: *The Astrophysical Journal Supplement Series* 236.1 (2018), p. 16.

[72] *Why isn't there more open source in research?* Apr. 2018. URL: `https://www.software.ac.uk/blog/2016-10-06-why-isnt-there-more-open-source-research`.

[73] The Astropy Community. *Astropy Documentation — Astropy v3.1.1*. Feb. 2019. URL: `http://docs.astropy.org/en/stable/index.html`.

[74] The NumPy Community. *NumPy Reference, Release 1.16.1*. Feb. 2019. URL: `https://docs.scipy.org/doc/numpy-1.16.1/numpy-ref-1.16.1.pdf`.

[75] The Scipy Community. *SciPy Reference Guide, Release 1.2.1*. Feb. 2019. URL: `https://docs.scipy.org/doc/scipy-1.2.1/scipy-ref-1.2.1.pdf`.

[76] Wes McKinney, Josef Perktold, and Skipper Seabold. *Time Series Analysis in Python with statsmodels*. 2019.

[77] Wes McKinney and PyData Development Team. *pandas: powerful Python data analysistoolkit, Release 0.24.1*. Feb. 2019. URL: http://pandas.pydata.org/pandas-docs/stable/pandas.pdf.

[78] "Pysolar: Python libraries for simulating the irradiation of any point on earth by the sun". In: (2019).

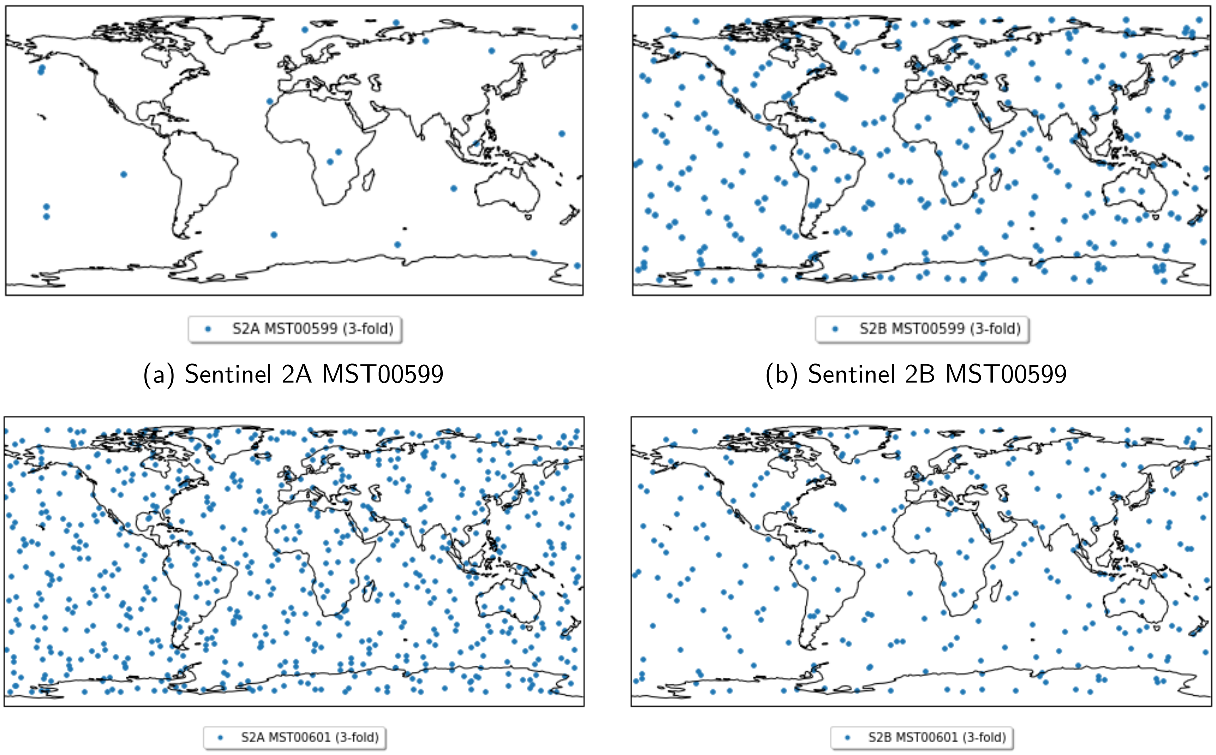
[79] *Computational tools in Pandas*. URL: http://pandas.pydata.org/pandas-docs/stable/user_guide/computation.html.

[80] *Using IPython for parallel computing*. URL: https://ipyparallel.readthedocs.io/en/latest/.

A

Appendix

A.1. Single Event Maps



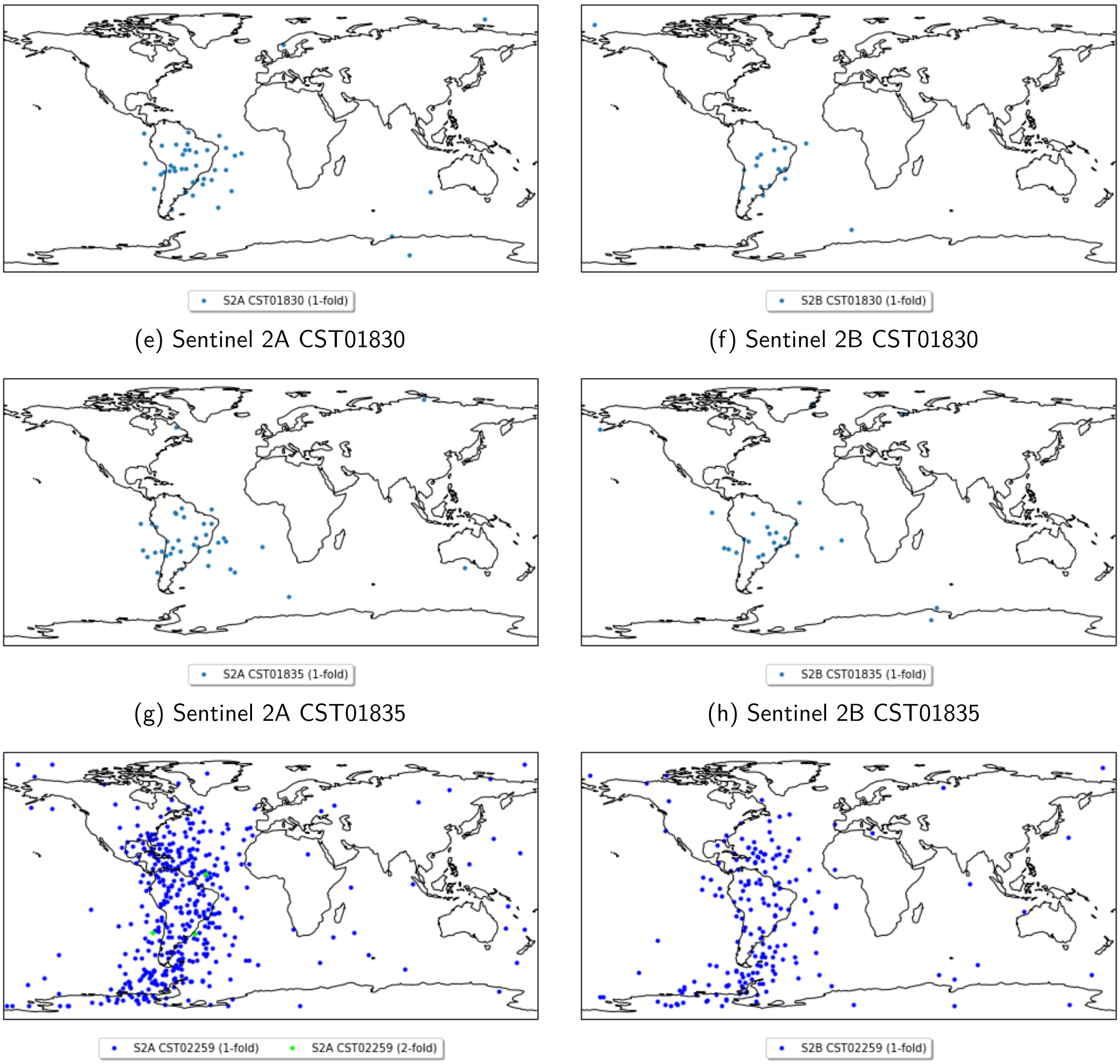
(a) Sentinel 2A MST00599

(b) Sentinel 2B MST00599

(c) Sentinel 2A MST00601

(d) Sentinel 2B MST00601

Figure A.1: Event positions for several counters (1)



(e) Sentinel 2A CST01830

(f) Sentinel 2B CST01830

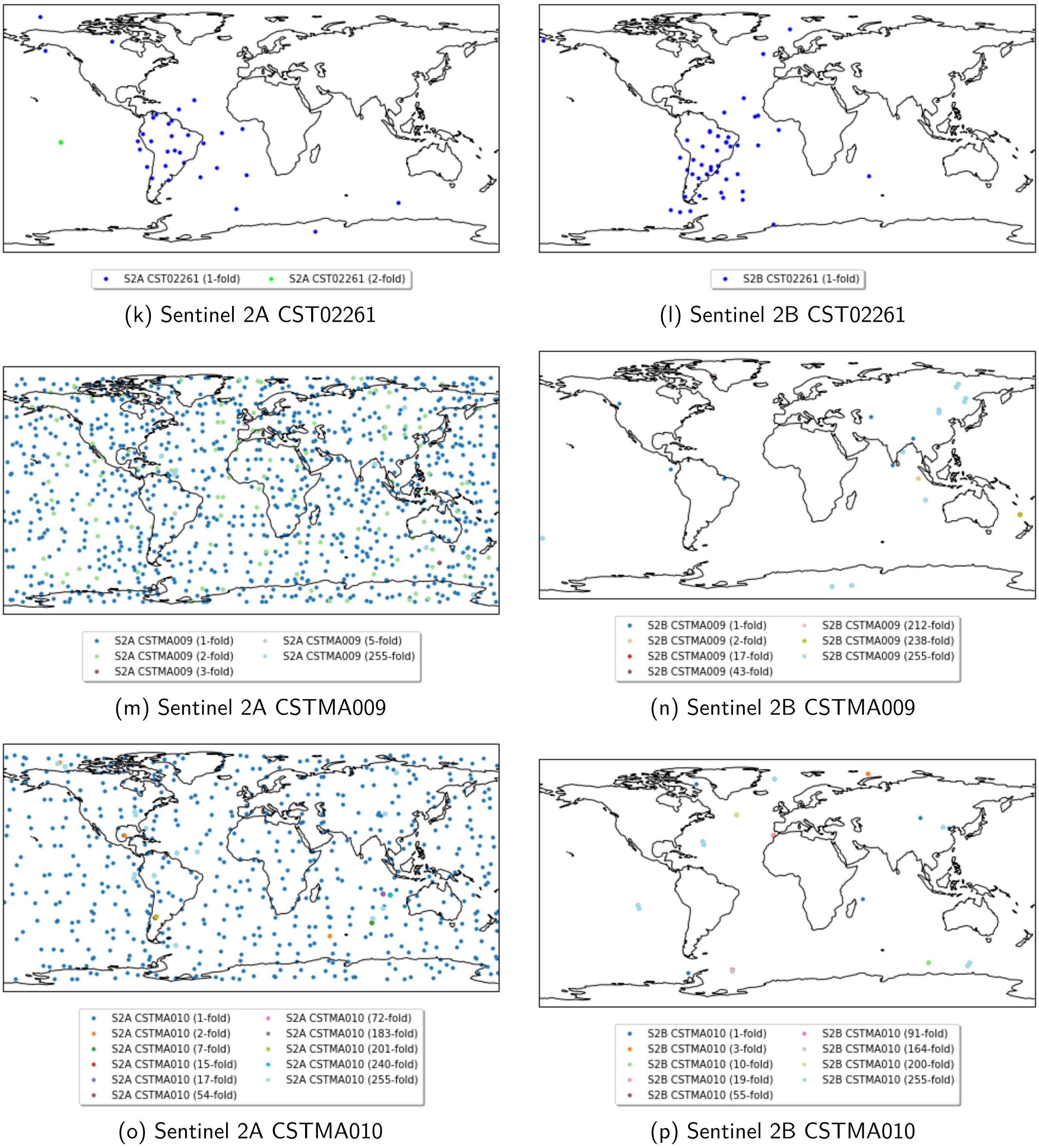
(g) Sentinel 2A CST01835

(h) Sentinel 2B CST01835

(i) Sentinel 2A CST2259

(j) Sentinel 2B CST02259

Figure A.2: Event positions for several counters (2)



(k) Sentinel 2A CST02261

(l) Sentinel 2B CST02261

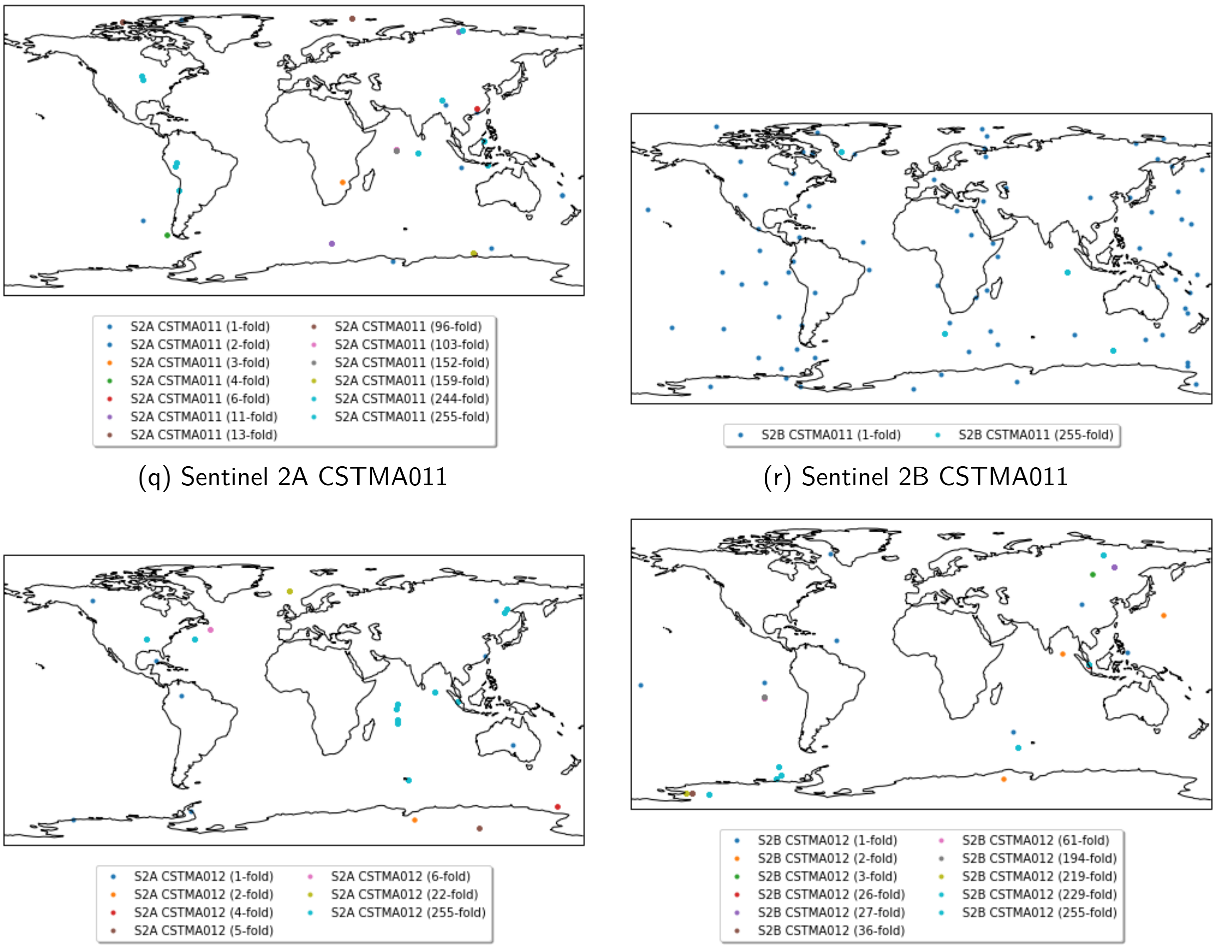
(m) Sentinel 2A CSTMA009

(n) Sentinel 2B CSTMA009

(o) Sentinel 2A CSTMA010

(p) Sentinel 2B CSTMA010

Figure A.3: Event positions for several counters (3)



(q) Sentinel 2A CSTMA011

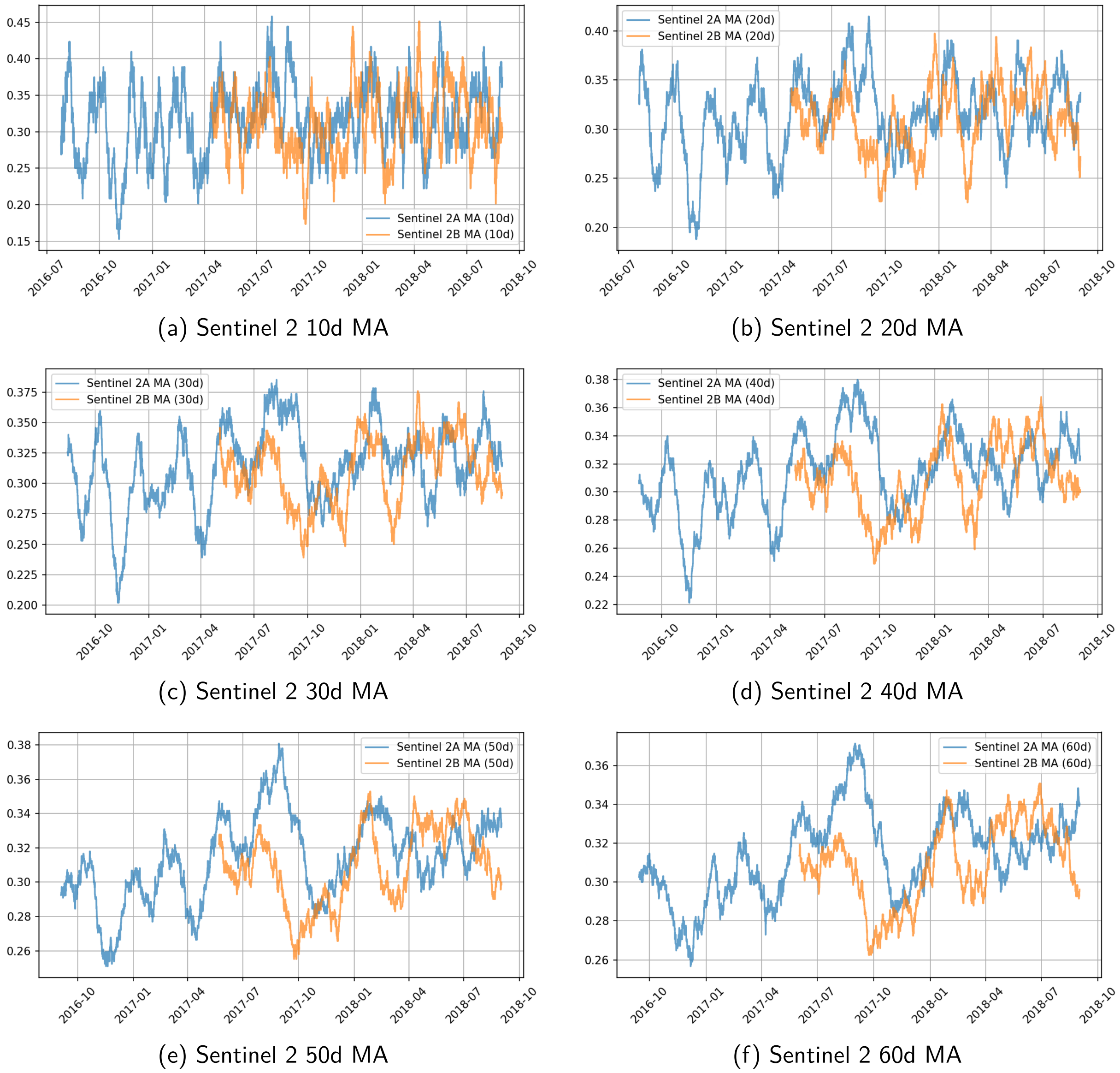
(r) Sentinel 2B CSTMA011

(s) Sentinel 2A CSTMA012

(t) Sentinel 2B CSTMA012

Figure A.4: Event positions for several counters (4)

A.2. Single Event Rate Moving-Average Windows



(a) Sentinel 2 10d MA

(b) Sentinel 2 20d MA

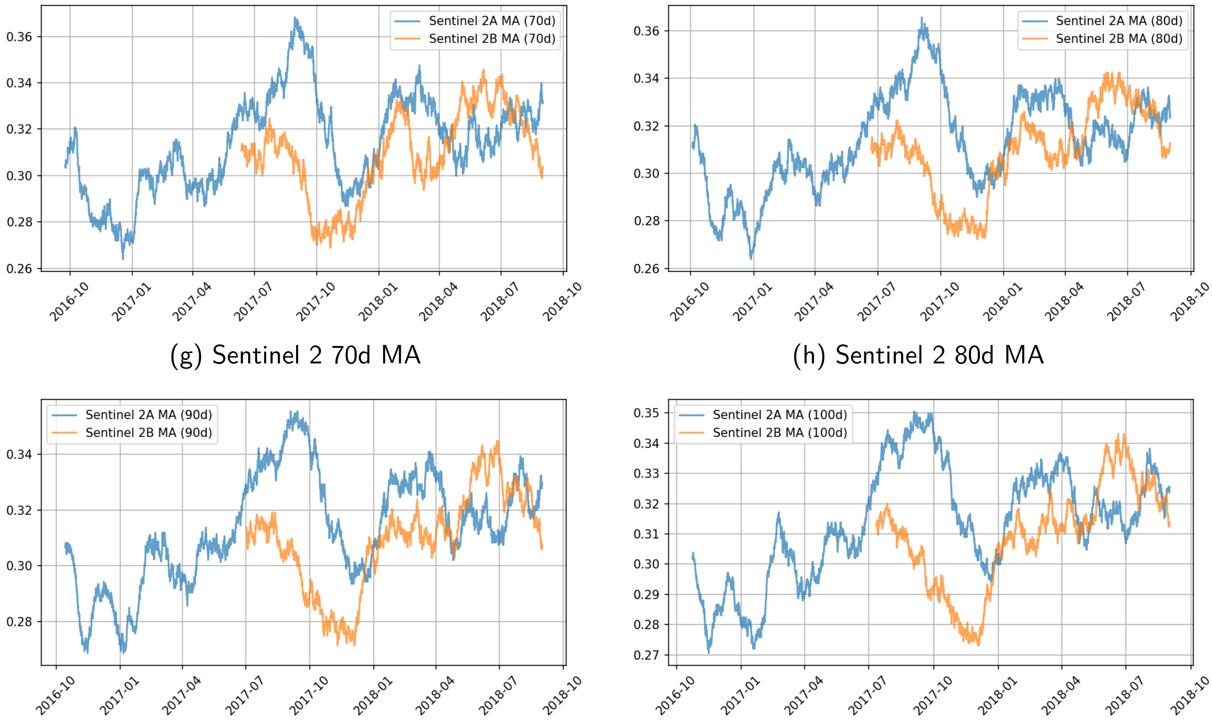
(c) Sentinel 2 30d MA

(d) Sentinel 2 40d MA

(e) Sentinel 2 50d MA

(f) Sentinel 2 60d MA

Figure A.5: Sentinel 2A and 2B rate signal for different moving-average Windows (1)



(g) Sentinel 2 70d MA

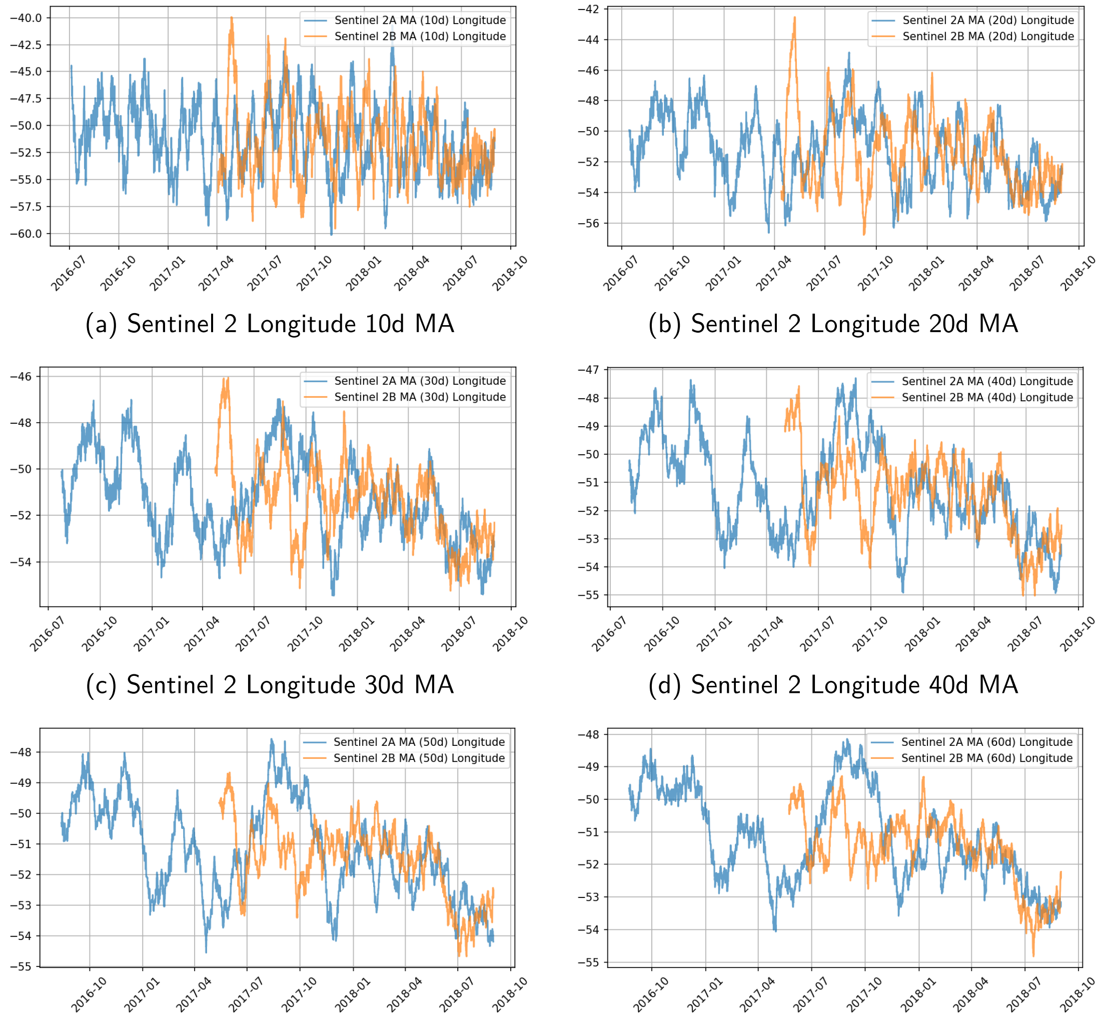
(h) Sentinel 2 80d MA

(i) Sentinel 2 90d MA

(j) Sentinel 2 100d MA

Figure A.6: Sentinel 2A and 2B rate signal for different moving-average Windows (2)

A.3. Single Event Longitude Moving-Average Windows



(a) Sentinel 2 Longitude 10d MA

(b) Sentinel 2 Longitude 20d MA

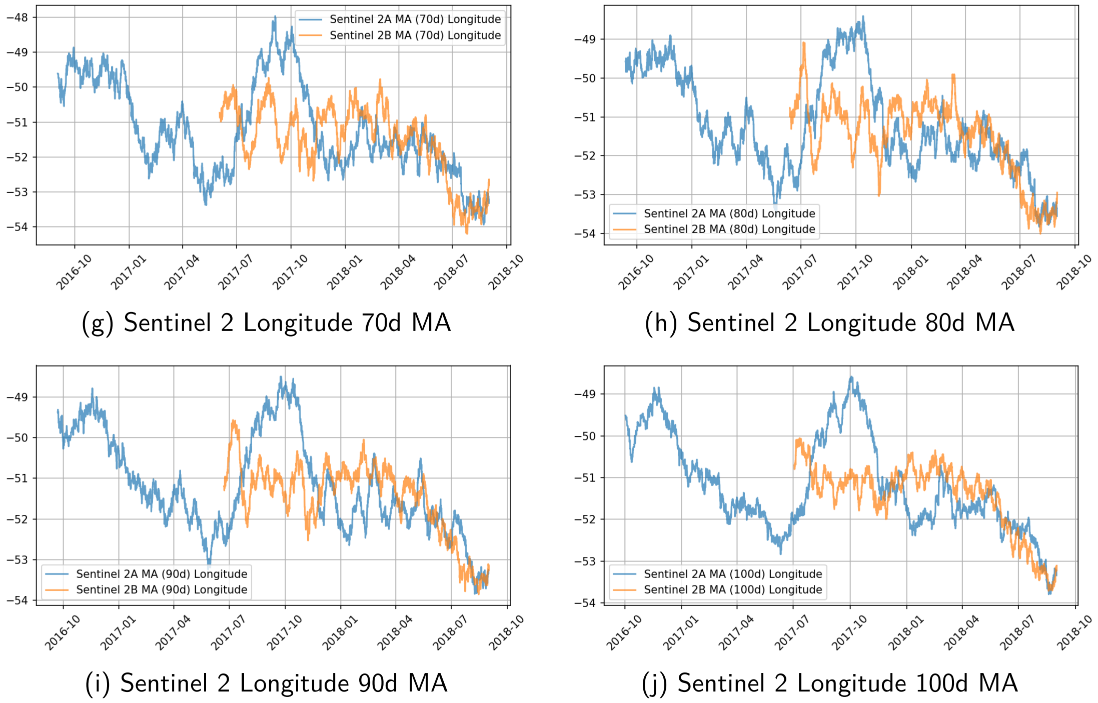
(c) Sentinel 2 Longitude 30d MA

(d) Sentinel 2 Longitude 40d MA

(e) Sentinel 2 Longitude 50d MA

(f) Sentinel 2 Longitude 60d MA

Figure A.7: Sentinel 2A and 2B longitude signal for different moving-average Windows (1)



(g) Sentinel 2 Longitude 70d MA

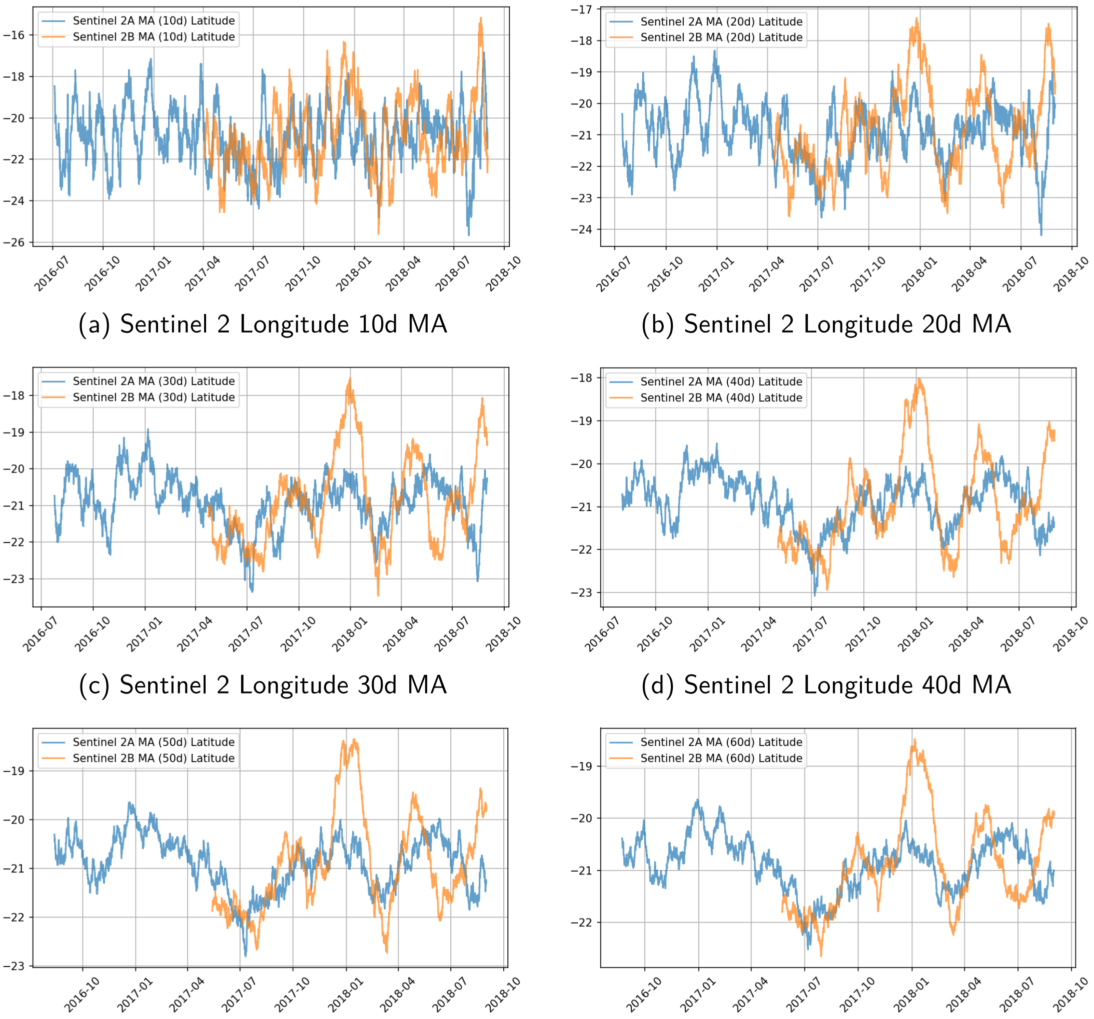
(h) Sentinel 2 Longitude 80d MA

(i) Sentinel 2 Longitude 90d MA

(j) Sentinel 2 Longitude 100d MA

Figure A.8: Sentinel 2A and 2B longitude signal for different moving-average Windows (2)

A.4. Single Event Latitude Moving-Average Windows



(a) Sentinel 2 Longitude 10d MA

(b) Sentinel 2 Longitude 20d MA

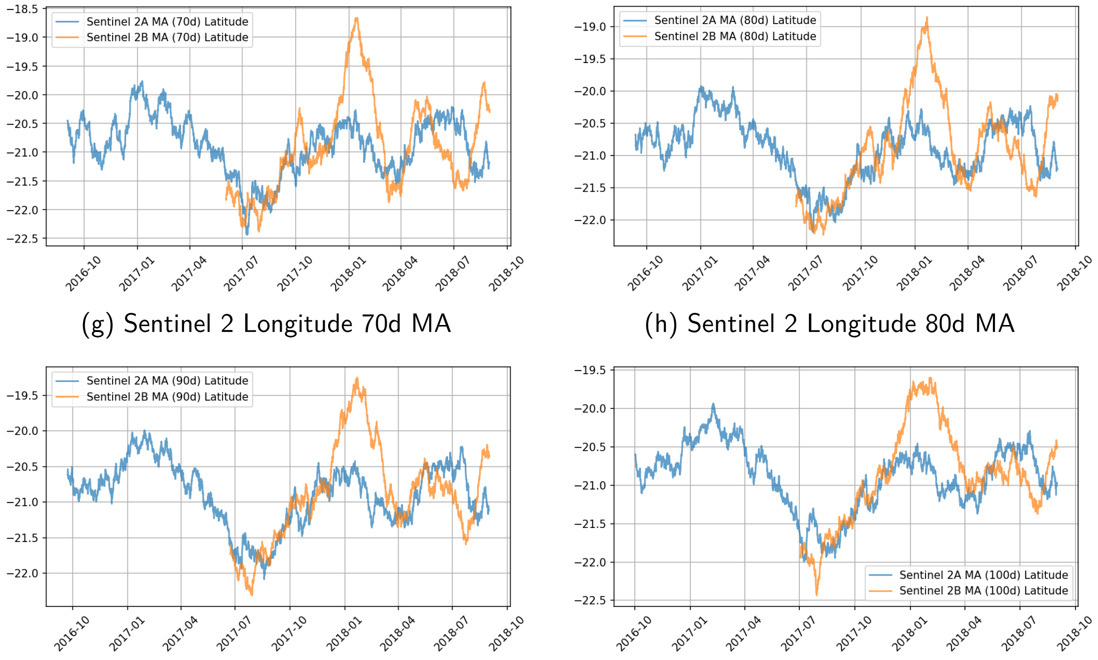
(c) Sentinel 2 Longitude 30d MA

(d) Sentinel 2 Longitude 40d MA

(e) Sentinel 2 Longitude 50d MA

(f) Sentinel 2 Longitude 60d MA

Figure A.9: Sentinel 2A and 2B latitude signal for different moving-average Windows (1)



(g) Sentinel 2 Longitude 70d MA

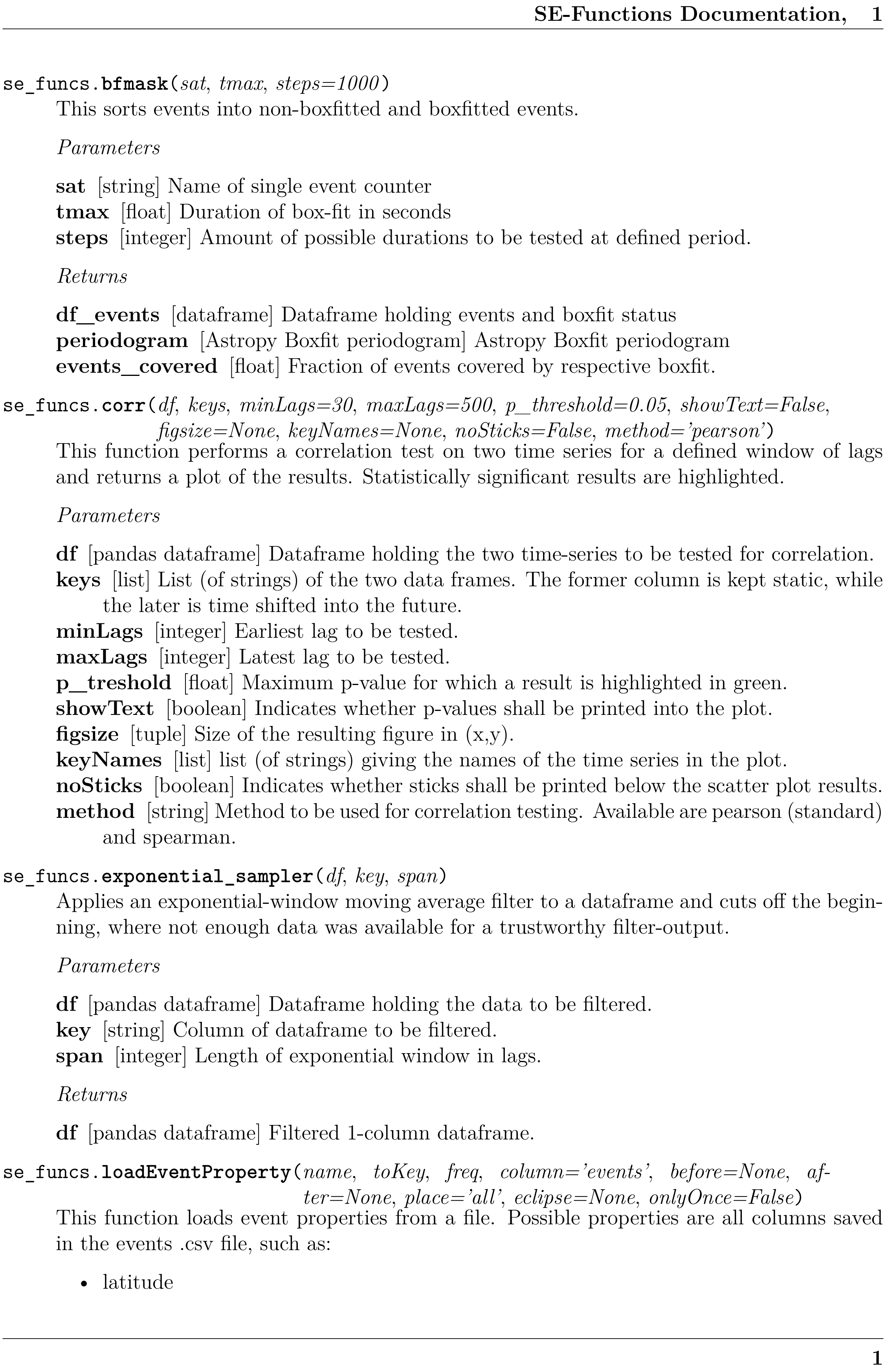
(h) Sentinel 2 Longitude 80d MA

(i) Sentinel 2 Longitude 90d MA

(j) Sentinel 2 Longitude 100d MA

Figure A.10: Sentinel 2A and 2B latitude signal for different moving-average Windows (2)

A.5. SE Functions API Documentation



se_funcs.**bfmask**(*sat*, *tmax*, *steps=1000*)

This sorts events into non-boxfitted and boxfitted events.

Parameters

sat [string] Name of single event counter
tmax [float] Duration of box-fit in seconds
steps [integer] Amount of possible durations to be tested at defined period.

Returns

df_events [dataframe] Dataframe holding events and boxfit status
periodogram [Astropy Boxfit periodogram] Astropy Boxfit periodogram
events_covered [float] Fraction of events covered by respective boxfit.

se_funcs.**corr**(*df*, *keys*, *minLags=30*, *maxLags=500*, *p_threshold=0.05*, *showText=False*, *figsize=None*, *keyNames=None*, *noSticks=False*, *method='pearson'*)

This function performs a correlation test on two time series for a defined window of lags and returns a plot of the results. Statistically significant results are highlighted.

Parameters

df [pandas dataframe] Dataframe holding the two time-series to be tested for correlation.
keys [list] List (of strings) of the two data frames. The former column is kept static, while the later is time shifted into the future.
minLags [integer] Earliest lag to be tested.
maxLags [integer] Latest lag to be tested.
p_treshold [float] Maximum p-value for which a result is highlighted in green.
showText [boolean] Indicates whether p-values shall be printed into the plot.
figsize [tuple] Size of the resulting figure in (x,y).
keyNames [list] list (of strings) giving the names of the time series in the plot.
noSticks [boolean] Indicates whether sticks shall be printed below the scatter plot results.
method [string] Method to be used for correlation testing. Available are pearson (standard) and spearman.

se_funcs.**exponential_sampler**(*df*, *key*, *span*)

Applies an exponential-window moving average filter to a dataframe and cuts off the beginning, where not enough data was available for a trustworthy filter-output.

Parameters

df [pandas dataframe] Dataframe holding the data to be filtered.
key [string] Column of dataframe to be filtered.
span [integer] Length of exponential window in lags.

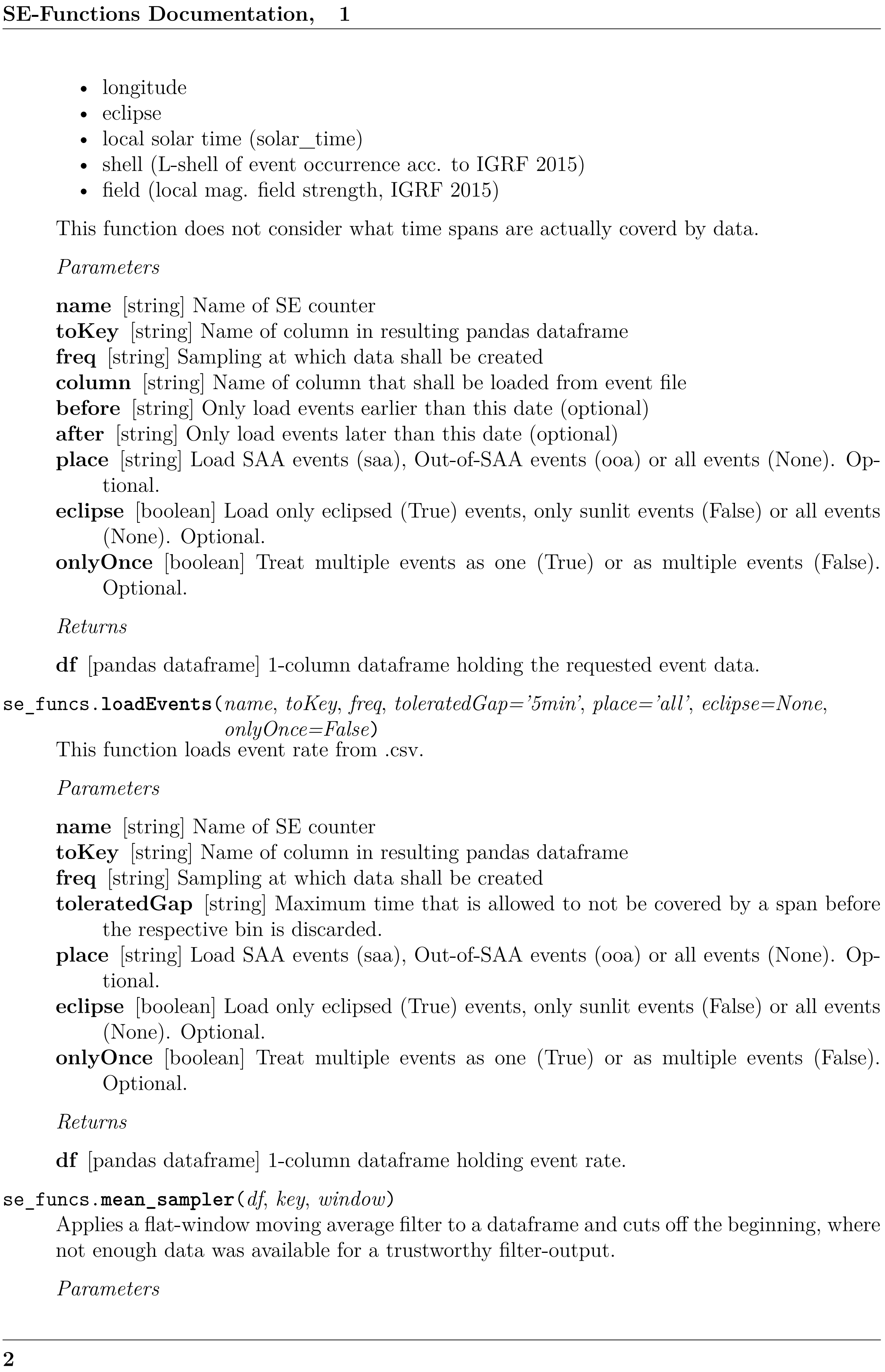
Returns

df [pandas dataframe] Filtered 1-column dataframe.

se_funcs.**loadEventProperty**(*name*, *toKey*, *freq*, *column='events'*, *before=None*, *after=None*, *place='all'*, *eclipse=None*, *onlyOnce=False*)

This function loads event properties from a file. Possible properties are all columns saved in the events .csv file, such as:

- latitude



- longitude
- eclipse
- local solar time (solar_time)
- shell (L-shell of event occurrence acc. to IGRF 2015)
- field (local mag. field strength, IGRF 2015)

This function does not consider what time spans are actually coverd by data.

Parameters

name [string] Name of SE counter
toKey [string] Name of column in resulting pandas dataframe
freq [string] Sampling at which data shall be created
column [string] Name of column that shall be loaded from event file
before [string] Only load events earlier than this date (optional)
after [string] Only load events later than this date (optional)
place [string] Load SAA events (saa), Out-of-SAA events (ooa) or all events (None). Optional.
eclipse [boolean] Load only eclipsed (True) events, only sunlit events (False) or all events (None). Optional.
onlyOnce [boolean] Treat multiple events as one (True) or as multiple events (False). Optional.

Returns

df [pandas dataframe] 1-column dataframe holding the requested event data.

`se_funcs.`**`loadEvents`**`(`*name*, *toKey*, *freq*, *toleratedGap='5min'*, *place='all'*, *eclipse=None*, *onlyOnce=False*`)`

This function loads event rate from .csv.

Parameters

name [string] Name of SE counter
toKey [string] Name of column in resulting pandas dataframe
freq [string] Sampling at which data shall be created
toleratedGap [string] Maximum time that is allowed to not be covered by a span before the respective bin is discarded.
place [string] Load SAA events (saa), Out-of-SAA events (ooa) or all events (None). Optional.
eclipse [boolean] Load only eclipsed (True) events, only sunlit events (False) or all events (None). Optional.
onlyOnce [boolean] Treat multiple events as one (True) or as multiple events (False). Optional.

Returns

df [pandas dataframe] 1-column dataframe holding event rate.

`se_funcs.`**`mean_sampler`**`(`*df*, *key*, *window*`)`

Applies a flat-window moving average filter to a dataframe and cuts off the beginning, where not enough data was available for a trustworthy filter-output.

Parameters

df [pandas dataframe] Dataframe holding the data to be filtered.
key [string] Column of dataframe to be filtered.
window [string or int] Length of window as pandas time-tag (string) or number of lags (int).

Returns

df [pandas dataframe] Filtered 1-column dataframe.

`se_funcs.`**`mongodb`**`(`*db='sentinel'*`)`

This method provides a Arctic library instance to freely work in MongoDB. Also, the configuration of MongoDB access (host, database) are stored in this function.

Returns

library : Arctic Library

`se_funcs.`**`plotGPS`**`(`*key*, *label*, *projection='pc'*, *before=None*, *after=None*, *filename=None*, *kde=False*, *cmap_kde='jet'*, *hue=None*, *cmap_hue='summer'*, *figsize=(15, 8)*`)`

This function plots Single event data on a map.

Parameters

key [string] Name of SE Counter.
label [string] Name of SE Counter in the plot.
before [string] Only use data before this date (optional).
after [string] Only use data prior to this date (optional).
filename [string] Save plot to this file (optinal)
kde [boolean] Add Kernel Density Estimate of events (optional).
cmap_kde [string] Colormap to be used with KDE (optional).
hue [boolean] Use different colors for different multi-event quantitites (optional).
cmap_hue [string] Matplotlib colormap to be used with multi-events (optional).
figsize [tuple] Size of figure in (x,y), optional.

`se_funcs.`**`sat2box`**`(`*name*, *periods*, *durations*, *verbose=False*, *cutDown=True*`)`

This function provides a wrapper around AstroPys Box-Fitting that pre-processes the output of loadEvent to use it as an input.

Parameters

name [string] Name of single event counter
periods [list] List of periods that shall be analysed in Astropy quantities
durations [list] List of periods that shall be analysed in Astropy quantities
verbose [boolean] Turns on execution time printing (default off, optional)
cutDown [boolean] Whether or not to treat multiple events as one (default true, optional).

Returns

periodogram [dict] power and corresponding periods (in seconds)

`se_funcs.`**`sat2boxTriangle`**`(`*name*, *periods*, *duration_acc*, *cutDown=True*`)`

This function provides a wrapper around AstroPys Box-Fitting that automatically

Parameters

name [string] Name of single event counter.
periods [list] List of periods that shall be analysed in Astropy quantities.
durations_acc [integer] Number of durations to be tested.
cutDown [boolean] Whether or not to treat multiple events as one (default true, optional).

Returns

periodogram [dict] power and corresponding periods (in seconds).

`se_funcs.`**`sat2ls`**`(`*name*, *pmin*, *pmax*, *step*, *sampling='6040s'*, *verbose=False*, *cutDown=True*, *full_output=False*`)`

This function provides a wrapper around AstroPys Lomb-Scargle fitting that pre-processes the output of loadEvent to use it as an input.

Parameters

name [string] Name of single event counter
pmin [string] Minimum period to analyse
pmax [string] Maximum period to analyse
step [integer] Number of seconds between two period steps
sampling [string] Sampling at which data shall be loaded
verbose [boolean] Turns on execution time printing (default off, optional)
cutDown [boolean] Whether or not to treat multiple events as one (default true, optional).

Returns

periodogram [dict] power and corresponding periods (in seconds)

`se_funcs.`**`savefig`**`(`*name*, *latex=False*`)`

Saves the current figure to a file (.png)

Parameters

name [string] File to which the plot is saved.

`se_funcs.`**`seasonal_adjustment`**`(`*df*, *column*, *lags*, *full_output=False*`)`

This function seasonally adjusts the column of a pandas dataframe. Mathematically seen, this means that for each element in the season length, an average is calculated that is then subtracted from the input data. The function returns the residuals, which are commensurate to a seasonally adjusted input.

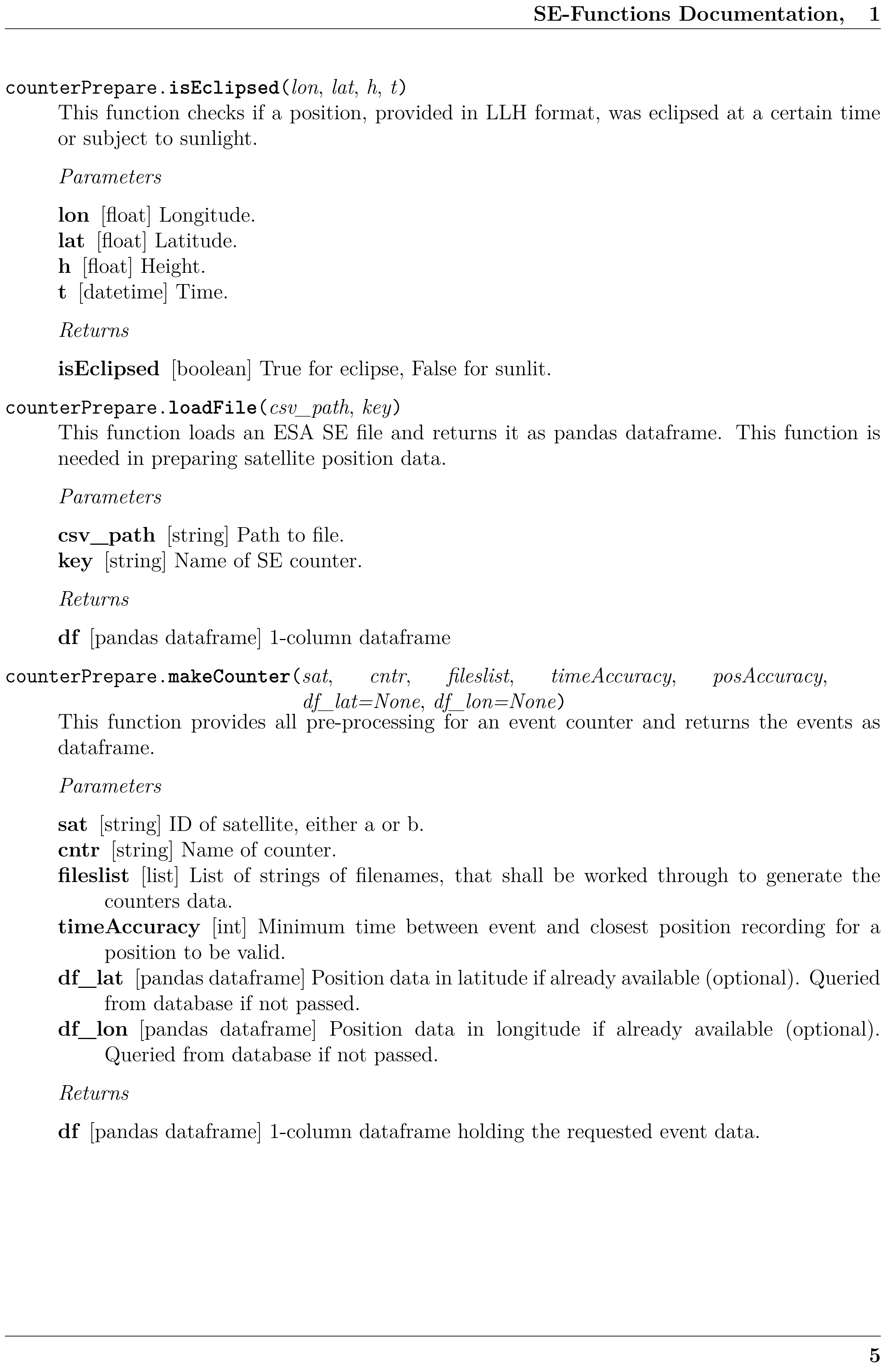
Parameters

df [pandas dataframe] Dataframe holding the input to be adjusted.
column [string] Name of dataframes column that shall be adjusted.
lags [int] Length of one season in rows of the data frame.
full_output [boolean] If set to true, the average season is returned as well.

Returns

df [pandas dataframe] Dataframe holding seasonally adjusted data.
season [pandas dataframe] Average season (optional).

A.6. Counter Preparation API Documentation



counterPrepare.**isEclipsed**(*lon*, *lat*, *h*, *t*)

This function checks if a position, provided in LLH format, was eclipsed at a certain time or subject to sunlight.

Parameters

lon [float] Longitude.
lat [float] Latitude.
h [float] Height.
t [datetime] Time.

Returns

isEclipsed [boolean] True for eclipse, False for sunlit.

counterPrepare.**loadFile**(*csv_path*, *key*)

This function loads an ESA SE file and returns it as pandas dataframe. This function is needed in preparing satellite position data.

Parameters

csv_path [string] Path to file.
key [string] Name of SE counter.

Returns

df [pandas dataframe] 1-column dataframe

counterPrepare.**makeCounter**(*sat*, *cntr*, *fileslist*, *timeAccuracy*, *posAccuracy*, *df_lat=None*, *df_lon=None*)

This function provides all pre-processing for an event counter and returns the events as dataframe.

Parameters

sat [string] ID of satellite, either a or b.

cntr [string] Name of counter.

fileslist [list] List of strings of filenames, that shall be worked through to generate the counters data.

timeAccuracy [int] Minimum time between event and closest position recording for a position to be valid.

df_lat [pandas dataframe] Position data in latitude if already available (optional). Queried from database if not passed.

df_lon [pandas dataframe] Position data in longitude if already available (optional). Queried from database if not passed.

Returns

df [pandas dataframe] 1-column dataframe holding the requested event data.

A.7. Source Codes

A.7.1. Data Preparation

Listing A.1: Source code: Data pre-processing

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  1 09:26:46 2018

@author: patri
"""

#import pyximport; pyximport.install(pyimport=True)

import pandas as pd
from arctic import Arctic
import numpy as np
import os
from tqdm import tqdm
import pytz
from pysolar.solar import  get_altitude_fast
import math


#sat = input("Enter Satellite (a/b): ")
#cntr = input("Enter counter name: ")


def loadFile(csv_path, key):

    # parsing file once to get essential information
    fp              = open(csv_path,"r")
    # lineNumber    = int(sum(1 for line in fp))
    # print ("File length: " + str(lineNumber) + " lines")
    headerLine = 0

    # Reset file to beginning if already manipulated
    fp.seek(0)

    # Skip through first 2 lines since these are useless
    for x in range(2):
        line = fp.readline()
        headerLine = headerLine + 1

    # Now find what timesets this file offers in line 3 and order them by satellite
    columnNames = []

    line = fp.readline()
    headerLine = headerLine + 1
    #print(line)
    entries = line.split(",")
    for x in range(len(entries)):
        entries[x] = entries[x].replace('\r', '')
```

```python
        entries[x] = entries[x].replace('\n', '')
        entries[x] = entries[x].replace(']','')
        entries[x] = entries[x].replace('[','')
        params = entries[x].split("␣")

        #print(params[1] + "_" + params[0])
        columnNames.append(params[1] + "_" + params[0])

    # skip ahead until descriptions
    for x in range(9):
        line = fp.readline()
        headerLine = headerLine + 1


    # Go through all field descriptions and count number of fields in the process
    # Then, assemble table creation query
    columnTypes = []
    line = fp.readline()
    headerLine = headerLine + 1
    hashFinder = line[0:1]

    while hashFinder != "#":
        entries = line.split(",")
        columnTypes.append(entries[3])
        line = fp.readline()
        headerLine = headerLine + 1
        hashFinder = line[0:1]

      # Print all found fields and their type
    #print("Fields found in file: ")
    for x in range(len(columnNames)):
        columnNames[x] = columnNames[x].lower()
        #print(columnNames[x] + " " + columnTypes[x])
    fp.close()
    fp = None
    namesArray = ["date"] + columnNames

    if key in columnNames:

        print(csv_path)
        df = pd.read_csv(csv_path, header=headerLine-1,  sep=',', names=namesArray)
        df = df[["date", key]]
        df = df[~df.iloc[:,0].str.startswith('Warn')]
        df = df[~(df.iloc[:,1].astype(str) == '␣')]

        if len(df) > 0:

            if len(str(df.iloc[0,0]).split(".")) == 5:
                df["date"] = pd.to_datetime(df.iloc[:,0], format = "%Y.%j.%H.%M.%S")
            if len(str(df.iloc[0,0]).split(".")) == 6:
                df["date"] = pd.to_datetime(df.iloc[:,0], format = "%Y.%j.%H.%M.%S.%f")

            df = df.set_index("date")

            return df

        else:

            return 0

    else:

        return 0


def isEclipsed(lon, lat, h, t):

    r_e = 6371*1000
    alt = get_altitude_fast(latitude_deg=lat, longitude_deg=lon, when=t)
    minElevation = - ((np.pi/2)  - math.asin(r_e/(r_e+h)))*(180/np.pi)
    return alt < minElevation

def makeCounter(sat, cntr, fileslist, timeAccuracy, posAccuracy, debug=False,
                df_lat=None, df_lon=None):

    key = "sentinel2" + sat + "_" + cntr
    #fileslist = fileopenbox(msg="Please select input files", multiple=True)

    # now opening each file and concatenating the results of each
    dfs = []
    for file in fileslist:
        df = loadFile(file, key)
        if type(df) == pd.DataFrame:
            dfs.append(df[key].to_frame(key))
    print("collected␣all␣dfs")

    df = pd.concat(dfs, axis=0)
    df = df.sort_index()
    print("concatenated␣all␣dfs")

    # next, make sure all data is numeric and remove nan data
    df = df.dropna()
    df[key] = pd.to_numeric(df[key])

    # satellite b mst00495 has some weird recording start-stop going on for times before 2017-03-24, therefore
    # all data (it is only little) in the time before is discarded
    if key=="sentinel2b_mst00495":
        df = df["2017-03-24":]

    print("cleaned␣all␣dfs")

    # differentiate to move from counter value to event occurrence
    df[key] = df[key].diff()
    mask = (df[key] >= 0)  # & (df[key] < 10)
    df = df[mask]

    if debug:
        return df

    # next, it is determined which are the usable and non-usable time frames
    # so that non-usable times can later be restored as nans when read from file
    # determine number of samples per time
    df_group = df[key].to_frame(key).groupby( \
                pd.Grouper(freq=str(timeAccuracy)+"s")).count()

    # keep only those time frames where the number of samples is sufficient, in this case
    #    1 sample per timeAccuracy
    usable = df_group[df_group[key] >= 1]

    # calculate time delta in seconds among the remaining samples
    df_unix = np.diff(usable.index.astype(np.int64))/10**9

    # where the time delta is higher than the allowed time delta, an interruption in data coverage
    # is present. this means that between these samples, nans have to be placed.
    transitions = np.where(df_unix > timeAccuracy)[0]

    t_coll = []

    # make sure there are actually usable time frames (len>0)
    if len(usable) > 0:

        # if there are no transitions in the usable data, that means that the entire data is
        # one big time span of good quality
        if len(transitions) == 0:
            t_coll.append([usable.index[0], usable.index[len(usable)-1]])

        # if there is one transition, then there are two spans of good data
        elif  len(transitions) == 1:
            t_coll.append([usable.index[0], usable.index[transitions[0]]])
            t_coll.append([usable.index[transitions[0]+1], usable.index[len(usable)-1]])

        # if there are more than 1 transition, the spans have to be written away by a loop
        else:
            # starting timespan
            t_coll.append([usable.index[0], usable.index[transitions[0]]])

            # all timespans in between
            for x in range(1, len(transitions) ):
                t_coll.append([usable.index[transitions[x-1]+1], usable.index[transitions[x]]])

            # final timespan
            t_coll.append([usable.index[transitions[\
                        len(transitions)-1]+1], usable.index[len(usable)-1]])


    # make sure that a folder exists for the analysis
    if not os.path.isdir("analysis/"+key):
        os.mkdir("analysis/"+key)

    # save the found spans of good data coverage to csv file
    df_spans = pd.DataFrame(t_coll)
    df_spans.to_csv("analysis/"+key+"/spans.csv", index=False)

    # kick out counter increases that exceed the allowable time threshold
    df["dt"] = df.index.astype(np.int64)/10**9
    df["dt"] = df["dt"].diff()
    df = df[1:] # remove first nan row
    df = df[df["dt"] <= timeAccuracy]
    df = df.drop(columns=["dt"])

    # now really only keep events
    df = df[ df[key] != 0 ]

    # establish connection to database so that position data can be fetched
    store = Arctic("localhost")
    satDB = store.get_library("satellites")


    # prepare columns to be filled
    df["lon"] = np.nan
    df["lat"] = np.nan
    df["eclipse"] = np.nan
    df["dt_lon"] = np.nan
    df["dt_lat"] = np.nan
    df["solar_time"] = np.nan
    df["orbit"] = np.nan

    ############# PREPARE POSITION DATA

    if df_lon is None:
        df_lon = satDB.read("sentinel2"+sat+"_gst0165f")
        print("lon␣in␣memory")
        df_lon = df_lon.sort_index()

    if df_lat is None:
        df_lat = satDB.read("sentinel2"+sat+"_gst01660")
        print("lat␣in␣memory")
        df_lat = df_lat.sort_index()


    ############ get orbit numbering

    df_orbits = pd.read_csv("analysis/orbits_"+sat+".csv")
    df_orbits["start"] = pd.to_datetime(df_orbits["start"])
    df_orbits["stop"] = pd.to_datetime(df_orbits["stop"])


    ############ DETERMINE ALL VALUES
    pbar = tqdm(total=len(df))
    for x in range(len(df)):

        time = df.index[x]
        #print(time)

        ###### longitude

        idx_before = df_lon.index.searchsorted(time)
        idx_after  = idx_before + 1

        tdist_before = (df_lon.index[idx_before] - time).total_seconds()
        tdist_after  = (df_lon.index[idx_after ] - time).total_seconds()

        posID = 0
        dt = 0

        if tdist_before <= tdist_after:
            posID = idx_before
            dt = tdist_before
        else:
            posID = idx_after
            dt = tdist_after

        if dt <= posAccuracy:
            df["lon"][x] = df_lon.iloc[posID, 0]
            df["dt_lon"][x] = dt


        ###### Latitude
        idx_before = df_lat.index.searchsorted(time)
        idx_after  = idx_before + 1

        tdist_before = (df_lat.index[idx_before] - time).total_seconds()
        tdist_after  = (df_lat.index[idx_after ] - time).total_seconds()

        posID = 0
        dt = 0

        if tdist_before <= tdist_after:
            posID = idx_before
            dt = tdist_before
        else:
            posID = idx_after
            dt = tdist_after


        if dt <= posAccuracy:
            df["lat"][x] = df_lat.iloc[posID, 0]
            df["dt_lat"][x] = dt

        ##### Number of orbit

        mask_orbits = (df_orbits["start"] > time) & (df_orbits["stop"] <= time)
        df_orbits_local = df_orbits[mask_orbits]
        if len(df_orbits_local) == 1:
            df["orbit"][x] = df_orbits_local["orbit"][0]



        if not np.isnan([df["lat"][x], df["lon"][x]]).any():

            ###### eclipses

            t = df.index[x].to_pydatetime()
            t = t.replace(tzinfo=pytz.UTC)
            df["eclipse"][x] = isEclipsed(df["lon"][x], df["lat"][x], 786*1000, t)

        ###### Progress Bar

        pbar.update()

    df_lat = None
    df_lon = None

    print(df["dt_lat"].max())
    print(df["dt_lon"].max())

    df["time"] = df.index
    df["events"] = df[key]

    df = df.drop(columns=[key])

    multiEvents = []

    for x in range(len(df)):
        if df["events"][x] > 1:
            for y in range(int(df["events"][x])-1):
                multiEvents.append(df[x:(x+1)])

    if len(multiEvents) > 0:
        df_multi = pd.concat(multiEvents, axis=0)
        df = df.append(df_multi)

    df["events"] = 1
    df = df.sort_index()
    df = df.drop(columns=["dt_lon", "dt_lat"])

    key = "sentinel2" + sat + "_" + cntr
    df.to_csv("analysis/"+key+"/events.csv")

    df = None
```

A.7.2. Event Rate Analysis

Listing A.2: Source code: Event rate analysis

```python
#!/usr/bin/env python
# coding: utf-8

# # Time Series Analysis

# This analysis combines:
#
# 1. Plotting for general inspection/illustration of raw data
# 2. Finding the long-term single event rate development
# 3. Testing auto-correlations of the single event rate signal
# 4. Testing correlations between long-term SE rate development and F10.7 / DST
# 5. First tries in prediction

# In[1]:


from se_funcs import *
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.tsa.api as smapi
import seaborn as sns
from statsmodels.tsa.ar_model import AR
from datetime import datetime
from scipy.stats import linregress
from pykalman import KalmanFilter
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.seasonal import seasonal_decompose
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
from pandas.plotting import autocorrelation_plot

from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()


# ## 1. Plotting for Illustration

# In[2]:


df_s2a_rate = loadEvents("sentinel2a_mst00495", "s2a", "6026s", toleratedGap="20min", onlyOnce=False, place="saa")
df_s2b_rate = loadEvents("sentinel2b_mst00495", "s2b", "6026s", toleratedGap="20min", onlyOnce=False, place="saa")


# In[3]:
```

```


plt.figure(figsize=(7,3.5), dpi=150)

plt.plot(df_s2a_rate, label="Sentinel␣2A␣MST00495␣(raw)", alpha=0.5)
plt.plot(df_s2b_rate, label="Sentinel␣2B␣MST00495␣(raw)", alpha=0.5)

plt.grid(True)
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()

savefig("s2a_s2b_daily.png", latex=True)


# In[4]:


df_s2a_lon = loadEventProperty("sentinel2a_mst00495", "s2a_lon", "6026s", column="lon", place="saa")
df_s2a_lat = loadEventProperty("sentinel2a_mst00495", "s2a_lat", "6026s", column="lat", place="saa")
df_s2b_lon = loadEventProperty("sentinel2b_mst00495", "s2b_lon", "6026s", column="lon", place="saa")
df_s2b_lat = loadEventProperty("sentinel2b_mst00495", "s2b_lat", "6026s", column="lat", place="saa")


# In[5]:


plt.figure(figsize=(6,3), dpi=150)
plt.plot(df_s2a_lon, label="Sentinel␣2A␣Longitude␣(raw)", alpha=1)
plt.grid(True)
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
savefig("s2a_lon_raw.png", latex=True)


# In[6]:


plt.figure(figsize=(6,3), dpi=150)
plt.plot(df_s2a_lat, label="Sentinel␣2A␣Latitude␣(raw)", alpha=1)
plt.grid(True)
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
savefig("s2a_lat_raw.png", latex=True)


# In[7]:


plt.figure(figsize=(6,3), dpi=150)
plt.plot(df_s2a_lat, label="Sentinel␣2B␣Longitude␣(raw)", alpha=1)
plt.grid(True)
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
savefig("s2b_lon_raw.png", latex=True)


# In[8]:


plt.figure(figsize=(6,3), dpi=150)
plt.plot(df_s2a_lat, label="Sentinel␣2B␣Latitude␣(raw)", alpha=1)
plt.grid(True)
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
savefig("s2b_lat_raw.png", latex=True)


# ## 2. Single event rate long-term development

# Rebuilding the event data from event list and usable time spans (see data pre-processing):

# In[9]:


ma_window = "50d"
sampling="6026s"

####### Sentinel 2A

# load events from file
df_s2a_rate_orb = loadEvents("sentinel2a_mst00495", "s2a_raw", sampling, toleratedGap="20min", onlyOnce=True, place="saa")

#measurements = df_s2a_rate_orb.values.flatten()
#measurements[np.isnan(measurements)] = np.ma.masked
#kf = KalmanFilter(initial_state_mean=0, n_dim_obs=1)
#kf.em(measurements, n_iter=10)
#df_s2a_rate_orb["s2a_ma"] = kf.smooth(measurements)[0]

# filter w. moving average
df_s2a_rate_orb["s2a_ma"] = mean_sampler(df_s2a_rate_orb, "s2a_raw", ma_window) * ( 6026 / pd.to_timedelta(sampling).total_seconds() )
#df_s2a_rate_orb["s2a_ma"] = exponential_sampler(df_s2a_rate_orb, "s2a_raw", 144*5)

####### Sentinel 2B

# load events from file
df_s2b_rate_orb = loadEvents("sentinel2b_mst00495", "s2b_raw", sampling, toleratedGap="20min", onlyOnce=True, place="saa")


```

```
# filter w. moving average
df_s2b_rate_orb["s2b_ma"] = mean_sampler(df_s2b_rate_orb, "s2b_raw", ma_window) * ( 6026 / pd.to_timedelta(sampling).total_seconds() )
#df_s2b_rate_orb["s2b_ma"] = exponential_sampler(df_s2b_rate_orb, "s2b_raw", 144*5)

####### BUNDLING
df_rate_orb = pd.concat([df_s2a_rate_orb, df_s2b_rate_orb], axis=1)

###### Difference
df_rate_orb["diff_raw"] = df_rate_orb["s2a_raw"] - df_rate_orb["s2b_raw"]
df_rate_orb["diff_ma"] = mean_sampler(df_rate_orb, "diff_raw", ma_window)


# Quantifying error between filtered and raw signal at orbital frequency

# In[10]:


print("S2A")
print("Absolute␣Mean␣Error:␣")
print( np.abs(df_s2a_rate_orb["s2a_raw"] - df_s2a_rate_orb["s2a_ma"]).mean() )

print("\nS2B")
print("Absolute␣Mean␣Error:␣")
print( np.abs(df_s2b_rate_orb["s2b_raw"] - df_s2b_rate_orb["s2b_ma"]).mean() )


# ### 2.1 Determining the linearized trend

# In[11]:


index2year = (10**9 * 60 * 60 * 24 * 365.25)
poly_degree = 1

xfit = df_rate_orb.index.astype(np.int64).values / index2year

x = df_rate_orb["s2a_ma"].dropna().index.astype(np.int64).values / index2year
y = df_rate_orb["s2a_ma"].dropna().values.flatten()
fit_a=np.polyfit(x,y,poly_degree)
print("\nSentinel␣2A␣rate␣drift")
print(fit_a)

x = df_rate_orb["s2b_ma"].dropna().index.astype(np.int64).values / index2year
y = df_rate_orb["s2b_ma"].dropna().values.flatten()
fit_b=np.polyfit(x,y,poly_degree)
print("\nSentinel␣2B␣rate␣drift")
print(fit_b)

df_rate_orb["s2a_fit"] = np.poly1d(fit_a)(df_rate_orb.index.astype(np.int64).values / index2year)
```

```
df_rate_orb["s2b_fit"] = np.poly1d(fit_b)(df_rate_orb.index.astype(np.int64).values / index2year)

df_rate_orb["s2a_detrend"] = df_rate_orb["s2a_ma"] - df_rate_orb["s2a_fit"]
df_rate_orb["s2b_detrend"] = df_rate_orb["s2b_ma"] - df_rate_orb["s2b_fit"]

df_rate_orb["s2a_residual"] = df_rate_orb["s2a_raw"] - df_rate_orb["s2a_ma"]
df_rate_orb["s2b_residual"] = df_rate_orb["s2b_raw"] - df_rate_orb["s2b_ma"]


print("Mean␣Rates")
print("S2a")

print(df_rate_orb["s2a_ma"].mean())

print("S2b")
print(df_rate_orb["s2b_ma"].mean())


# In[12]:


valid_idx = df_rate_orb["s2a_ma"].dropna().index
plt.plot(df_rate_orb["s2a_fit"][valid_idx])
plt.plot(df_rate_orb["s2a_ma"][valid_idx])


# In[13]:


valid_idx = df_rate_orb["s2b_ma"].dropna().index
plt.plot(df_rate_orb["s2b_fit"][valid_idx])
plt.plot(df_rate_orb["s2b_ma"][valid_idx])


# Are the residuals random (not auto-correlated)?

# In[14]:


autocorrelation_plot(df_rate_orb["s2a_residual"].fillna(0))


# In[15]:


autocorrelation_plot(df_rate_orb["s2b_residual"].fillna(0))


# Are the detrended functions stationary?

# In[16]:
```

```


adfuller(df_rate_orb["s2a_detrend"].fillna(0).values)


# In[17]:


adfuller(df_rate_orb["s2b_detrend"].fillna(0).values)


# In[18]:


plt.plot(df_rate_orb["s2a_detrend"])
plt.plot(df_rate_orb["s2b_detrend"])


# ### 2.2 Creating and analyzing daily sampled data

# In[19]:


df_rate_1d = df_rate_orb.resample("1d").mean()


# In[20]:


print("S2A")
print("Absolute␣Mean␣Error:␣")
print( np.abs(df_rate_1d["s2a_raw"] - df_rate_1d["s2a_ma"]).mean() )

print("\nS2B")
print("Absolute␣Mean␣Error:␣")
print( np.abs(df_rate_1d["s2b_raw"] - df_rate_1d["s2b_ma"]).mean() )


# DST and F10.7 are loaded for plotting.

# In[21]:


df_rate_1d["f107"] = internalize("esa_f107", freq="1d")["esa_f107"]
df_rate_1d["dst"]  = internalize("dst", freq="1d")["dst"]


# ### 2.3 Plotting: Event rate (filtered), DST, F10.7

# In[22]:


fig, axes = plt.subplots(nrows=5, ncols=1, figsize=(9,13),
                         gridspec_kw = {'height_ratios':[4, 1, 1, 2, 1]},
                         sharex=True, dpi=150)

```

```
date_min = 1469923200
date_max = 1535673600+3600*24*7

date_min = datetime.utcfromtimestamp(date_min)
date_max = datetime.utcfromtimestamp(date_max)

date_lims = [date_min, date_max]

axes[0].set_xlim(date_lims)
axes[1].set_xlim(date_lims)
axes[2].set_xlim(date_lims)

axes[0].set_title("Event␣Rate")


# plot real
idx_a = df_rate_orb["s2a_ma"].dropna().index
idx_b = df_rate_orb["s2b_ma"].dropna().index

axes[0].plot(df_rate_orb["s2a_ma"][idx_a], label="Sentinel␣2A", color="#1f77b4")
axes[0].plot(df_rate_orb["s2b_ma"][idx_b], label="Sentinel␣2B", color="#ff7f0e")

# plot fitted
axes[0].plot(df_rate_orb["s2a_fit"][idx_a], label="Sentinel␣2A␣(Trend)", linestyle='--', lw=1, color="#1f77b4")
axes[0].plot(df_rate_orb["s2b_fit"][idx_b], label="Sentinel␣2B␣(Trend)", linestyle='--', lw=1, color="#ff7f0e")

axes[0].legend()
axes[0].set_ylabel("Events/orbit")
axes[0].grid(True)

axes[1].set_title("DST")
axes[1].plot(df_rate_1d["dst"].dropna(), color="purple")
axes[1].set_ylabel("nT")
axes[1].grid(True)

axes[2].set_title("F10.7␣cm␣Radio␣Flux")
axes[2].plot(df_rate_1d["f107"].dropna(), color="red")
axes[2].set_ylabel("Solar␣Flux␣Units")
axes[2].grid(True)

axes[3].set_title("Detrended")
axes[3].plot(df_rate_orb["s2a_detrend"], color="#1f77b4", alpha=1, label=False)
axes[3].plot(df_rate_orb["s2b_detrend"], color="#ff7f0e", alpha=1, label=None)
#axes[3].plot(df_rate_orb["s2a_detrend"]-df_rate_orb["s2b_detrend"], color="black", alpha=0.6)
axes[3].set_ylabel("Events/orbit")
axes[3].grid(True)

axes[4].set_title("Residuals")
axes[4].plot(df_rate_orb["s2a_residual"], color="#1f77b4", alpha=0.6)
```

```
axes[4].plot(df_rate_orb["s2b_residual"], color="#ff7f0e", alpha=0.6)
axes[4].set_ylabel("Events/orbit")
axes[4].grid(True)

for axid in [0, 3]:

    # aex peak
    axes[axid].axvline("2016-09-23", color="green")
    axes[axid].axvline("2017-09-23", color="green")
    axes[axid].axvline("2018-09-23", color="green")

    # vex peak
    axes[axid].axvline("2016-02-01", color="green")
    axes[axid].axvline("2017-02-01", color="green")
    axes[axid].axvline("2018-02-01", color="green")

    # july minimum
    axes[axid].axvline("2016-06-15", color="red")
    axes[axid].axvline("2017-06-15", color="red")
    axes[axid].axvline("2018-06-15", color="red")

    # december minimum
    axes[axid].axvline("2016-12-01", color="red")
    axes[axid].axvline("2017-12-01", color="red")
    axes[axid].axvline("2018-12-01", color="red")


for axid in [0,1,2,3]:

    axes[axid].axvspan("2017-12-01", "2018-02-01", alpha=0.4, color="gold")
    axes[axid].axvspan("2017-01-15", "2017-02-20", alpha=0.4, color="gold")
    axes[axid].axvspan("2017-06-15", "2017-09-03", alpha=0.4, color="gold")

    axes[axid].axvline("2017-09-06", alpha=0.4, color="black", lw=1, linestyle="--")
    axes[axid].axvline("2017-04-02", alpha=0.4, color="black", lw=1, linestyle="--")
    axes[axid].axvline("2017-05-27", alpha=0.4, color="black", lw=1, linestyle="--")
    axes[axid].axvline("2018-08-25", alpha=0.4, color="black", lw=1, linestyle="--")

plt.tight_layout()

plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis␣Latex\\figures\\event_rate_orbit.png")


# ### 2.5 Deviation Trendline / Values

# In[23]:


```

```
fig, axes = plt.subplots(figsize=(8,4), ncols=2, nrows=1)

df_rate_orb["s2a_detrend"].hist(density=True, ax=axes[0])
df_rate_orb["s2b_detrend"].hist(density=True, ax=axes[1])

axes[0].set_title("Sentinel␣2A")
axes[1].set_title("Sentinel␣2B")

axes[0].set_xlabel("Deviation␣from␣trend")
axes[1].set_xlabel("Deviation␣from␣trend")

axes[0].set_ylabel("Values␣[%]")
#axes[1].set_ylabel("Values [%]")

savefig("event_rate_histogram.png", latex=True)


# ### 2.6 Autocorrelation (white-noise test) of residuals

# In[32]:


from statsmodels.stats.diagnostic import acorr_ljungbox

acorr_ljungbox(df_rate_orb["s2b_residual"].fillna(0).values)


# In[33]:


_=plot_acf(df_rate_orb["s2a_residual"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Auto-correlation")
plt.title("")
savefig("s2a_resid_acorr.png", latex=True)


# In[34]:


_=plot_acf(df_rate_orb["s2b_residual"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Auto-correlation")
plt.title("")
savefig("s2b_resid_acorr.png", latex=True)


# In[35]:


_=plot_pacf(df_rate_orb["s2a_residual"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Partial␣Autocorrelation")
plt.title("")


# In[36]:


_=plot_pacf(df_rate_orb["s2b_residual"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Partial␣Autocorrelation")
plt.title("")


# ### 2.7 Creating explanatory plots

# In[ ]:


df_goes = internalize("esa_p10", key="goes", after="2017-08-01", before="2018-01-01")

fig, ax1 = plt.subplots(figsize=(8,4))

ax1.set_xlabel('Time', color="black")
ax1.plot(df_goes["goes"].dropna(), label="GOES␣Protons", color="black")
ax1.set_xlim(["2017-09-01", "2017-09-20"])
ax1.set_ylabel("GOES␣Protons␣<␣10␣MeV␣(per␣cm2,␣day,␣sr)")

plt.grid()
plt.xticks(rotation=45)

ax2 = ax1.twinx()
ax2.tick_params('y', colors='r')
ax2.plot(df_rate_1d["dst"].dropna(), color="red", label="DST")
ax2.set_xlim(["2017-09-01", "2017-09-20"])
ax2.set_ylabel("DST␣[nT]", color='red')

savefig("goes_cmes.png", latex=True)


# In[ ]:


df_moscow = pd.read_csv("2019_sept_gcr.csv", delim_whitespace=True, header=None, names=["date", "time", "gcrs"])
df_moscow["idx"] = df_moscow["date"] + "␣" + df_moscow["time"]
df_moscow["idx"] = pd.to_datetime(df_moscow["idx"], format="%Y.%m.%d␣%H:%M")
df_moscow = df_moscow.set_index("idx")
df_moscow.index.name = "Time"
df_moscow = df_moscow.drop(columns=["time", "date"])
df_moscow["2017-08-10":"2017-10-10"]["gcrs"].plot(grid=True, figsize=(8,4))

plt.ylabel("Particle␣Count")
plt.title("Moscow␣Neutron␣Monitor")
```

```

savefig("moscow_neutron.png", latex=True)


# In[ ]:


t = np.arange(0,100,1)
y_exp     = 1*np.exp(-t*0.025)
y_ma      = np.ones((1,len(t))).flatten() * 0.55

t_ma_exp = np.arange(0,110,10)
y_ma_exp = [1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0]


plt.plot(t,y_exp, label="Exponential␣MA")
plt.plot(t,y_ma,  label="Flat␣MA")

plt.step(t_ma_exp,y_ma_exp, label="Ideal␣Exp.␣MA", where="post")

plt.legend()
plt.ylabel("Relative␣Weight")
plt.xlabel("Time␣into␣past␣[d]")

plt.yticks([])
plt.xticks([0, 10])

plt.axvline(0, alpha=0.4, color="black", lw=1, linestyle="--")
plt.axvline(10, alpha=0.4, color="black", lw=1, linestyle="--")

savefig("ma_windows.png", latex=True)


# In[ ]:


df_f107 = internalize("esa_f107", freq="1d")

plt.figure(figsize=(8,4), dpi=150)
plt.plot(df_f107)
plt.title("F10.7")

plt.ylabel("Solar␣flux␣units")
plt.xlabel("Time")
plt.grid()

savefig("f107.png", latex=True)


# In[ ]:


df_dst = internalize("dst", freq="1d")
df_dst["dst"].plot()


```

```
# ## 3. Studying SE (filtered) auto-correlations

# Sentinel 2A

# In[ ]:


fig ,ax = plt.subplots(figsize=(8,3))
_ =  plot_acf( df_rate_orb["s2a_ma"].fillna(df_rate_orb["s2a_ma"].mean()).values, lags=100, zero=False, ax=ax)
savefig("s2a_acorr.png", latex=True)


# Sentinel 2B

# In[ ]:


fig ,ax = plt.subplots(figsize=(8,3))
_ =  plot_acf( df_rate_orb["s2b_ma"].fillna(df_rate_orb["s2b_ma"].mean()).values, lags=100, zero=False, ax=ax)
savefig("s2b_acorr.png", latex=True)


# ## 4. Testing correlations between long-term SE rate development and F10.7 / DST

# In[ ]:


corr(df_rate_1d, ["s2a_ma", "f107"], 1, 200, noSticks=True, method="pearson")
plt.title("F10.7" + "␣\u21a6␣" + "S2A")
savefig("corr_s2a_f107.png", latex=True)


# In[ ]:


corr(df_rate_1d, ["s2b_ma", "f107"], 1, 200, noSticks=True, method="pearson")
plt.title("F10.7" + "␣\u21a6␣" + "S2B")
savefig("corr_s2b_f107.png", latex=True)


# In[ ]:


corr(df_rate_1d, ["s2a_ma", "dst"], 1, 200, noSticks=True, method="pearson")
plt.title("DST" + "␣\u21a6␣" + "S2A")
savefig("corr_s2a_dst.png", latex=True)


# In[ ]:

```

```

corr(df_rate_1d, ["s2b_ma", "dst"], 1, 200, noSticks=True, method="pearson")
plt.title("DST" + "␣\u21a6␣" + "S2B")
savefig("corr_s2b_dst.png", latex=True)


# ### Reduced complexity correlation

# In[ ]:


df_rate_1d["Sentinel␣2A␣MA"] = df_rate_1d["s2a_ma"]
df_rate_1d["Sentinel␣2B␣MA"] = df_rate_1d["s2b_ma"]

df_rate_1d["DST"] = df_rate_1d["dst"]
df_rate_1d["F10.7"] = df_rate_1d["f107"]


# In[ ]:


sns.jointplot("Sentinel␣2A␣MA", "F10.7", df_rate_1d)
savefig("corr_inst_s2a_f107.png", latex=True)

print(df_rate_1d[["s2a_ma", "f107"]].corr())


# In[ ]:


sns.jointplot("Sentinel␣2B␣MA", "F10.7", df_rate_1d),
savefig("corr_inst_s2b_f107.png", latex=True)

print(df_rate_1d[["s2b_ma", "f107"]].corr())


# In[ ]:


sns.jointplot("Sentinel␣2A␣MA", "DST", df_rate_1d)
savefig("corr_inst_s2a_dst.png", latex=True)
print(df_rate_1d[["s2a_ma", "dst"]].corr())


# In[ ]:


sns.jointplot("Sentinel␣2B␣MA", "DST", df_rate_1d)
savefig("corr_inst_s2b_dst.png", latex=True)
print(df_rate_1d[["s2b_ma", "dst"]].corr())
```

A.7.3. Event Position Analysis

Listing A.3: Source code: Event position analysis

```
#!/usr/bin/env python
# coding: utf-8

## Analysis of intra-year SAA (avg. event positition) shift

# The following analysis considers the average event position by using a rolling average on event latitude and longitude. Rolling average is applied to event chunks of 50 days (respectively 90d for plotting).

# In[1]:


from se_funcs import *

import matplotlib.pyplot as plt
import matplotlib.colors
import pandas as pd
import numpy as np
import datetime
from arctic import Arctic
from mpldatacursor import datacursor
from scipy.signal import find_peaks
from scipy.stats import linregress
import matplotlib.dates as mdates
from statsmodels.graphics.tsaplots import plot_acf
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()


# In[2]:


freq = "90d"
groundTrack = True

date_min = 1469923200
date_max = 1535673600

cmap = matplotlib.cm.get_cmap('jet')
normalize = matplotlib.colors.Normalize(vmin=date_min, vmax=date_max)

# colorbar
def cmapFormatter(x,pos):
    d = datetime.datetime(1970,1,1) + datetime.timedelta(seconds=int(x))
    return d.strftime("%b %Y")

formatter = matplotlib.ticker.FuncFormatter(cmapFormatter)


# In addition, a representative ground track (10 days revisit) is shown in the figures and prepared in the following.

# ## Loading the data + pre-processing

# In[42]:


db = Arctic("localhost")
arc = db.get_library("satellites")

t = pd.date_range("2018-01-01", "2018-01-11", freq=None)

df_lon = arc.read("sentinel2a_gst0165f", chunk_range=t)
df_lat = arc.read("sentinel2a_gst01660", chunk_range=t)

df_ground = df_lon
df_ground["lon"] = df_lon["sentinel2a_gst0165f"]
df_ground["lat"] = df_lat["sentinel2a_gst01660"]
df_ground = df_ground.drop(columns=["sentinel2a_gst0165f"])

df_ground = df_ground[df_ground["lon"] > -115]
df_ground = df_ground[df_ground["lon"] < 30]
df_ground = df_ground[df_ground["lat"] > -60]
df_ground = df_ground[df_ground["lat"] < 10]


# In[43]:


df_dst  = internalize("dst",       freq="1d", after="2016-07-31", before="2018-09-01")
df_f107 = internalize("esa_f107", freq="1d", after="2016-07-31", before="2018-09-01")

# event rate s2a
df_s2a_rate_orb = loadEvents("sentinel2a_mst00495", "s2a_raw", "6026s", toleratedGap="20min", onlyOnce=False, place="saa")
df_s2a_rate_orb["s2a_50d"] = mean_sampler(df_s2a_rate_orb, "s2a_raw", "50d")
#df_s2a_rate_orb["s2a_50d"] = exponential_sampler(df_s2a_rate_orb, "s2a_raw", 144*5)
df_s2a_rate_orb["s2a_90d"] = mean_sampler(df_s2a_rate_orb, "s2a_raw", "90d")
#df_s2a_rate_orb["s2a_90d"] = exponential_sampler(df_s2a_rate_orb, "s2a_raw", 144*9)

# event rate s2b
df_s2b_rate_orb = loadEvents("sentinel2b_mst00495", "s2b_raw", "6026s", toleratedGap="20min", onlyOnce=False, place="saa")
df_s2b_rate_orb["s2b_50d"] = mean_sampler(df_s2b_rate_orb, "s2b_raw", "50d")
#df_s2b_rate_orb["s2b_50d"] = exponential_sampler(df_s2b_rate_orb, "s2b_raw", 144*5)
df_s2b_rate_orb["s2b_90d"] = mean_sampler(df_s2b_rate_orb, "s2b_raw", "90d")
#df_s2b_rate_orb["s2b_90d"] = exponential_sampler(df_s2b_rate_orb, "s2b_raw", 144*9)

```

```
# longitude s2a
df_s2a_lon = loadEventProperty("sentinel2a_mst00495", "s2a_lon", "6026s",
    column="lon", place="saa")
df_s2a_lon["s2a_lon_50d"] = mean_sampler(df_s2a_lon, "s2a_lon", "50d")
#df_s2a_lon["s2a_lon_50d"] = exponential_sampler(df_s2a_lon, "s2a_lon",
    144*5)
df_s2a_lon["s2a_lon_90d"] = mean_sampler(df_s2a_lon, "s2a_lon", "90d")
#df_s2a_lon["s2a_lon_90d"] = exponential_sampler(df_s2a_lon, "s2a_lon",
    144*9)

# latitude s2a
df_s2a_lat = loadEventProperty("sentinel2a_mst00495", "s2a_lat", "6026s",
    column="lat", place="saa")
df_s2a_lat["s2a_lat_50d"] = mean_sampler(df_s2a_lat, "s2a_lat", "50d")
#df_s2a_lat["s2a_lat_50d"] = exponential_sampler(df_s2a_lat, "s2a_lat",
    144*5)
df_s2a_lat["s2a_lat_90d"] = mean_sampler(df_s2a_lat, "s2a_lat", "90d")
#df_s2a_lat["s2a_lat_90d"] = exponential_sampler(df_s2a_lat, "s2a_lat",
    144*9)

# longitude s2b
df_s2b_lon = loadEventProperty("sentinel2b_mst00495", "s2b_lon", "6026s",
    column="lon", place="saa")
df_s2b_lon["s2b_lon_50d"] = mean_sampler(df_s2b_lon, "s2b_lon", "50d")
#df_s2b_lon["s2b_lon_50d"] = exponential_sampler(df_s2b_lon, "s2b_lon",
    144*5)
df_s2b_lon["s2b_lon_90d"] = mean_sampler(df_s2b_lon, "s2b_lon", "90d")
#df_s2b_lon["s2b_lon_90d"] = exponential_sampler(df_s2b_lon, "s2b_lon",
    144*9)

# latitude s2b
df_s2b_lat = loadEventProperty("sentinel2b_mst00495", "s2b_lat", "6026s",
    column="lat", place="saa")
df_s2b_lat["s2b_lat_50d"] = mean_sampler(df_s2b_lat, "s2b_lat", "50d")
#df_s2b_lat["s2b_lat_50d"] = exponential_sampler(df_s2b_lat, "s2b_lat",
    144*5)
df_s2b_lat["s2b_lat_90d"] = mean_sampler(df_s2b_lat, "s2b_lat", "90d")
#df_s2b_lat["s2b_lat_90d"] = exponential_sampler(df_s2b_lat, "s2b_lat",
    144*9)

# bundle
df_orb = pd.concat([df_s2a_rate_orb, df_s2b_rate_orb, df_s2a_lon,
    df_s2a_lat, df_s2b_lon, df_s2b_lat], axis=1)


# In[68]:


year2seconds = 365.25*24*60*60


# In[95]:


df_orb["s2a_vel_lon"] = df_orb["s2a_lon_50d"].diff() / 6026 * year2seconds
df_orb["s2a_vel_lat"] = df_orb["s2a_lat_50d"].diff() / 6026 * year2seconds
```

```

df_orb["s2a_vel_lat"] = mean_sampler(df_orb, "s2a_vel_lat", "50d")
df_orb["s2a_vel_lon"] = mean_sampler(df_orb, "s2a_vel_lon", "50d")

df_orb["s2a_saa_vel"] = (df_orb["s2a_vel_lat"]**2 + df_orb["s2a_vel_lat"]**2)**0.5

#df_orb["s2a_vel"] = mean_sampler(df_orb, "s2a_vel", "10d")


# In[96]:


df_orb["s2b_vel_lon"] = df_orb["s2b_lon_50d"].diff() / 6026 * year2seconds
df_orb["s2b_vel_lat"] = df_orb["s2b_lat_50d"].diff() / 6026 * year2seconds

df_orb["s2b_vel_lat"] = mean_sampler(df_orb, "s2b_vel_lat", "50d")
df_orb["s2b_vel_lon"] = mean_sampler(df_orb, "s2b_vel_lon", "50d")

df_orb["s2b_saa_vel"] = (df_orb["s2b_vel_lat"]**2 + df_orb["s2b_vel_lat"]**2)**0.5

#df_orb["s2b_vel"] = mean_sampler(df_orb, "s2b_vel", "10d")


# In[97]:


plt.figure(figsize=(15,7))
plt.grid()
plt.plot(df_orb["s2a_saa_vel"], label="Sentinel␣2A")
plt.plot(df_orb["s2b_saa_vel"], label="Sentinel␣2B")

plt.legend()


# ## Large 2D plot

# In[98]:


fig, axes = plt.subplots(nrows=1, ncols=2, sharey=True, sharex=True, figsize=(8.5,6.5), dpi=150)

xlims = [-55.0, -47.5]
ylims = [-22.5, -19.0]

colors_a = [cmap(normalize(value)) for value in df_orb.index.astype(np.int64)/10**9]
sc1 = axes[0].scatter(df_orb["s2a_lon_90d"].values, df_orb["s2a_lat_90d"].values, c=colors_a, marker=".", alpha=0.6, s=75)
axes[0].plot(df_orb["s2a_lon_90d"].values, df_orb["s2a_lat_90d"].values, color="black", alpha=0.75, linewidth=0.5)
axes[0].set_xlim(xlims)
axes[0].set_ylim(ylims)
axes[0].set_xlabel("Longitude␣[deg]")
```

```
axes[0].set_ylabel("Latitude␣[deg]")
axes[0].grid(True)
axes[0].set_title("Sentinel␣2A␣MST00495")

axes[0].plot(df_ground["lon"], df_ground["lat"], alpha=0.4, color="black",
        label="Representative␣Ground␣Track", linestyle="--")
axes[0].legend()

colors_b = [cmap(normalize(value)) for value in df_orb.index.astype(np.int64)/10**9]
sc2 = axes[1].scatter(df_orb["s2b_lon_90d"].values, df_orb["s2b_lat_90d"].values, c=colors_b, marker=".", alpha=0.6, s=75)
axes[1].plot(df_orb["s2b_lon_90d"].values, df_orb["s2b_lat_90d"].values, color="black", alpha=0.75, linewidth=0.5)
axes[1].set_xlim(xlims)
axes[1].set_ylim(ylims)
axes[1].set_xlabel("Longitude␣[deg]")
axes[1].grid(True)
axes[1].set_title("Sentinel␣2B␣MST00495")

axes[1].plot(df_ground["lon"], df_ground["lat"], alpha=0.4, color="black",
        label="Representative␣Ground␣Track", linestyle="--")
axes[1].legend()

plt.draw()

p0 = axes[0].get_position().get_points().flatten()
p1 = axes[1].get_position().get_points().flatten()

dt = date_max - date_min

common_ticks = np.linspace(date_min + 0.05*dt, date_max-0.05*dt, 7)
common_ticks_datetime = [datetime.datetime.utcfromtimestamp(date) for date in common_ticks]

ax_cbar     = fig.add_axes([p0[0]+0.0, -0.05, p1[2]-p0[0], 0.05])
ax_plot_lon = fig.add_axes([p0[0]+0.0, -0.15, p1[2]-p0[0], 0.10])
ax_plot_lat = fig.add_axes([p0[0]+0.0, -0.25, p1[2]-p0[0], 0.10])
ax_vel      = fig.add_axes([p0[0]+0.0, -0.35, p1[2]-p0[0], 0.10])
ax_rate     = fig.add_axes([p0[0]+0.0, -0.45, p1[2]-p0[0], 0.10])
ax_dbar     = fig.add_axes([p0[0]+0.0, -0.55, p1[2]-p0[0], 0.10])
ax_fbar     = fig.add_axes([p0[0]+0.0, -0.65, p1[2]-p0[0], 0.10])

sm = plt.cm.ScalarMappable(cmap=cmap, norm=normalize)
sm.set_array([])
cbar = fig.colorbar(sm, cax=ax_cbar, orientation="horizontal", ticks=common_ticks, format=formatter, label="")
cbar.ax.tick_params(labeltop=True, labelbottom=False, top=True, bottom=False)

ax_plot_lon.plot(df_orb["s2a_lon_50d"], label="S2A", color="#1f77b4")
ax_plot_lon.plot(df_orb["s2b_lon_50d"], label="S2B", color="#ff7f0e")
ax_plot_lon.set_ylabel("Lon")
ax_plot_lon.legend()
ax_plot_lon.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])
ax_plot_lon.grid()
ax_plot_lon.set_yticks([])
ax_plot_lon.set_xticks(common_ticks_datetime)

ax_plot_lat.plot(df_orb["s2a_lat_50d"], label="S2A", color="#1f77b4")
ax_plot_lat.plot(df_orb["s2b_lat_50d"], label="S2B", color="#ff7f0e")
ax_plot_lat.set_ylabel("Lat")
ax_plot_lat.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])
ax_plot_lat.grid()
ax_plot_lat.set_yticks([])
ax_plot_lat.set_xticks(common_ticks_datetime)

ax_vel.plot(df_orb["s2a_saa_vel"], color="#1f77b4")
ax_vel.plot(df_orb["s2b_saa_vel"], color="#ff7f0e")
ax_vel.set_ylabel("Velocity")
ax_vel.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])
ax_vel.grid()
ax_vel.set_yticks([])
ax_vel.set_xticks(common_ticks_datetime)

ax_rate.plot(df_orb["s2a_50d"], label="S2A", color="#1f77b4")
ax_rate.plot(df_orb["s2b_50d"], label="S2B", color="#ff7f0e")
ax_rate.set_ylabel("Rate")
ax_rate.set_yticks([])
ax_rate.set_xticks(common_ticks_datetime)
ax_rate.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])
ax_rate.grid()

ax_dbar.plot(df_dst["dst"].dropna(), label="DST", color="purple")
ax_dbar.legend()
ax_dbar.grid()
ax_dbar.set_yticks([])
ax_dbar.set_xticks(common_ticks_datetime)
ax_dbar.set_ylabel("DST")
ax_dbar.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])

ax_fbar.plot(df_f107["esa_f107"].dropna(), label="F10.7", color="red")
ax_fbar.legend()
ax_fbar.grid()
ax_fbar.set_yticks([])
ax_fbar.set_xticks(common_ticks_datetime)
ax_fbar.set_ylabel("F10.7")
ax_fbar.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])

#set major ticks format
ax_fbar.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y'))


#plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis Latex\\figures\\saa_test.png")

```

```

# ## Linear regression for drift rate modelling

# In[99]:


usToYear = 365*24*60*60*10**9

print("S2A␣-␣Lon")
reg_alon = linregress( df_orb["s2a_lon_50d"].dropna().index.astype(np.int64).values / usToYear,
                       df_orb["s2a_lon_50d"].dropna().values)
print(reg_alon)

print("\nS2B␣-␣Lon")
reg_blon = linregress( df_orb["s2b_lon_50d"].dropna().index.astype(np.int64).values / usToYear,
                       df_orb["s2b_lon_50d"].dropna().values)
print(reg_blon)

print("\nS2A␣-␣Lat")
reg_alat = linregress( df_orb["s2a_lat_50d"].dropna().index.astype(np.int64).values / usToYear,
                       df_orb["s2a_lat_50d"].dropna().values)
print(reg_alat)

print("\nS2B␣-␣Lat")
reg_blat = linregress( df_orb["s2b_lat_50d"].dropna().index.astype(np.int64).values / usToYear,
                       df_orb["s2b_lat_50d"].dropna().values)
print(reg_blat)

print("\nS2A␣-␣Vel")
reg_avel = linregress( df_orb["s2a_saa_vel"].dropna().index.astype(np.int64).values / usToYear,
                       df_orb["s2a_saa_vel"].dropna().values)
print(reg_avel)

print("\nS2B␣-␣Vel")
reg_bvel = linregress( df_orb["s2b_saa_vel"].dropna().index.astype(np.int64).values / usToYear,
                       df_orb["s2b_saa_vel"].dropna().values)
print(reg_bvel)

df_orb["s2a_lon_50d_fitted"] = reg_alon.intercept + (df_orb.index.astype(np.int64).values / usToYear) * reg_alon.slope
df_orb["s2a_lat_50d_fitted"] = reg_alat.intercept + (df_orb.index.astype(np.int64).values / usToYear) * reg_alat.slope

df_orb["s2b_lon_50d_fitted"] = reg_blon.intercept + (df_orb.index.astype(np.int64).values / usToYear) * reg_blon.slope
df_orb["s2b_lat_50d_fitted"] = reg_blat.intercept + (df_orb.index.astype(np.int64).values / usToYear) * reg_blat.slope

df_orb["s2a_saa_vel_fitted"] = reg_avel.intercept + (df_orb.index.astype(np.int64).values / usToYear) * reg_avel.slope
```

```
df_orb["s2b_saa_vel_fitted"] = reg_bvel.intercept + (df_orb.index.astype(np.int64).values / usToYear) * reg_bvel.slope


# Generate residuals of MA filter

# In[100]:


df_orb["s2a_lon_50d_resid"] = df_orb["s2a_lon_50d"] - df_orb["s2a_lon"]
df_orb["s2a_lat_50d_resid"] = df_orb["s2a_lat_50d"] - df_orb["s2a_lat"]
df_orb["s2b_lon_50d_resid"] = df_orb["s2b_lon_50d"] - df_orb["s2b_lon"]
df_orb["s2b_lat_50d_resid"] = df_orb["s2b_lat_50d"] - df_orb["s2b_lat"]


# ## Explanatory plot

# In[101]:


df_orb["color"] = "grey"
df_orb["time"] = df_orb.index

starts = ["2017-05", "2017-09-18", "2018-04", "2017-12-15", "2018-08-10", "2016"]
stops  = ["2017-09-18", "2017-10-25", "2018-07-15", "2018-03", "2018-12", "2016-12-15"]
pcolors = ["green", "purple", "orange", "blue", "red", "cyan"]

for x in range(len(starts)):
    df_orb.loc[(df_orb["time"] > starts[x]) & (df_orb["time"] < stops[x]), "color"] = pcolors[x]

fig, axes = plt.subplots(nrows=1, ncols=2, sharey=True, sharex=True, figsize=(8.5,6.5), dpi=200)

xlims = [-55.0, -47.5]
ylims = [-22.5, -19.0]

colors_a = [cmap(normalize(value)) for value in df_orb.index.astype(np.int64)/10**9]
sc1 = axes[0].scatter(df_orb["s2a_lon_90d"].values, df_orb["s2a_lat_90d"].values, c=df_orb["color"], marker=".", alpha=0.6, s=75)
axes[0].plot(df_orb["s2a_lon_90d"].values, df_orb["s2a_lat_90d"].values, color="black", alpha=0.75, linewidth=0.5)
axes[0].set_xlim(xlims)
axes[0].set_ylim(ylims)
axes[0].set_xlabel("Longitude␣[deg]")
axes[0].set_ylabel("Latitude␣[deg]")
axes[0].grid(True)
axes[0].set_title("Sentinel␣2A␣MST00495")

axes[0].plot(df_ground["lon"], df_ground["lat"], alpha=0.4, color="black",
          label="Representative␣Ground␣Track", linestyle="--")
axes[0].legend()

```

```
colors_b = [cmap(normalize(value)) for value in df_orb.index.astype(np.int64)/10**9]
sc2 = axes[1].scatter(df_orb["s2b_lon_90d"].values, df_orb["s2b_lat_90d"].values, c=df_orb["color"], marker=".", alpha=0.6, s=75)
axes[1].plot(df_orb["s2b_lon_90d"].values, df_orb["s2b_lat_90d"].values, color="black", alpha=0.75, linewidth=0.5)
axes[1].set_xlim(xlims)
axes[1].set_ylim(ylims)
axes[1].set_xlabel("Longitude␣[deg]")
axes[1].grid(True)
axes[1].set_title("Sentinel␣2B␣MST00495")

axes[1].plot(df_ground["lon"], df_ground["lat"], alpha=0.4, color="black",
         label="Representative␣Ground␣Track", linestyle="--")
axes[1].legend()


ax_cbar     = fig.add_axes([p0[0]+0.0, -0.05, p1[2]-p0[0], 0.05])
ax_cbar.set_xlim([datetime.datetime.utcfromtimestamp(date_min), datetime.datetime.utcfromtimestamp(date_max)])
for x in range(len(starts)):
    ax_cbar.axvspan(starts[x], stops[x], facecolor=pcolors[x])
ax_cbar.set_yticks([])
ax_cbar.set_ylabel("")
ax_cbar.grid()
ax_cbar.set_xticks(common_ticks_datetime)
ax_cbar.xaxis.set_major_formatter(mdates.DateFormatter('%b␣%Y'))

plt.draw()

#plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis Latex\\figures\\saa_explain.png")


# ## 50d resolution plots

# In[102]:


fig, axes = plt.subplots(nrows=5, ncols=1, sharey=False, sharex=True,
                         gridspec_kw = {'height_ratios':[2, 2, 2, 1, 1]},
                         figsize=(9,14), dpi=150)

axes[0].plot(df_orb["s2a_lon_50d"], label="Sentinel␣2A", color="#1f77b4")
axes[0].plot(df_orb["s2b_lon_50d"], label="Sentinel␣2B", color="#ff7f0e")
axes[0].plot(df_orb["s2a_lon_50d_fitted"], label="Sentinel␣2A␣(linear)", lw=1, linestyle="--", color="#1f77b4")
axes[0].plot(df_orb["s2b_lon_50d_fitted"], label="Sentinel␣2B␣(linear)", lw=1, linestyle="--", color="#ff7f0e")
axes[0].grid()
axes[0].legend()
axes[0].set_ylabel("[deg]")
axes[0].set_title("SAA␣Longitude")

axes[1].plot(df_orb["s2a_lat_50d"], label="Sentinel␣2A", color="#1f77b4")
axes[1].plot(df_orb["s2b_lat_50d"], label="Sentinel␣2B", color="#ff7f0e")
```

```
axes[1].plot(df_orb["s2a_lat_50d_fitted"], label="Sentinel␣2A␣(linear)", lw=1, linestyle="--", color="#1f77b4")
axes[1].plot(df_orb["s2b_lat_50d_fitted"], label="Sentinel␣2B␣(linear)", lw=1, linestyle="--", color="#ff7f0e")
axes[1].grid()
#axes[1].legend()
axes[1].set_ylabel("[deg]")
axes[1].set_title("SAA␣Latitude")

axes[2].plot(df_orb["s2a_saa_vel"], label="Sentinel␣2A", color="#1f77b4")
axes[2].plot(df_orb["s2b_saa_vel"], label="Sentinel␣2B", color="#ff7f0e")
axes[2].plot(df_orb["s2a_saa_vel_fitted"], label="Sentinel␣2A␣(linear)", lw=1, linestyle="--", color="#1f77b4")
axes[2].plot(df_orb["s2b_saa_vel_fitted"], label="Sentinel␣2B␣(linear)", lw=1, linestyle="--", color="#ff7f0e")
axes[2].grid()
#axes[2].legend()
axes[2].set_ylabel("[deg/orbit]")
axes[2].set_title("SAA␣Velocity")

axes[3].plot(df_orb["s2a_lon_50d_resid"], label="Sentinel␣2A␣Longitude", alpha=0.6)
axes[3].plot(df_orb["s2b_lon_50d_resid"], label="Sentinel␣2B␣Longitude", alpha=0.6)
axes[3].grid()
#axes[3].legend()
axes[3].set_ylabel("[deg]")
axes[3].set_title("Residuals␣(Longitude)")

axes[4].plot(df_orb["s2a_lat_50d_resid"], label="Sentinel␣2A␣Latitude", alpha=0.6)
axes[4].plot(df_orb["s2b_lat_50d_resid"], label="Sentinel␣2B␣Latitude", alpha=0.6)
axes[4].grid()
#axes[4].legend()
axes[4].set_ylabel("[deg]")
axes[4].set_title("Residuals␣(Latitude)")


# ## Test of residual whiteness

# Generate residuals of MA filter

# In[103]:


_=plot_acf(df_orb["s2a_lon_50d_resid"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Auto-correlation")
plt.title("")
savefig("s2a_lon_resid_acorr.png", latex=True)


# In[104]:

```

```

_=plot_acf(df_orb["s2a_lat_50d_resid"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Auto-correlation")
plt.title("")
savefig("s2a_lat_resid_acorr.png", latex=True)


# In[105]:


_=plot_acf(df_orb["s2b_lon_50d_resid"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Auto-correlation")
plt.title("")
savefig("s2b_lon_resid_acorr.png", latex=True)


# In[106]:


_=plot_acf(df_orb["s2b_lat_50d_resid"].fillna(0).values, lags=100, zero=False)
plt.xlabel("Time␣Lag")
plt.ylabel("Auto-correlation")
plt.title("")
savefig("s2b_lat_resid_acorr.png", latex=True)


# In[ ]:
```

A.7.4. Frequency Domain Analysis

Listing A.4: Source code: Frequency Domain analysis

```
#!/usr/bin/env python
# coding: utf-8

# # Spectral Analysis of single event counters

# In[2]:


#%matplotlib notebook
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from astropy.stats import BoxLeastSquares, LombScargle
import astropy.units as u
import time
from se_funcs import *
from copy import copy
from tqdm import tqdm
from scipy.signal import find_peaks
import matplotlib.patheffects as PathEffects
import cartopy.crs as ccrs


# ## Create counter CSV files

# In[4]:


loadEvents("sentinel2a_mst00495", toKey="s2a", freq="5min", toleratedGap="2min", onlyOnce=True).to_csv("s2a_5min.csv")


# In[5]:


loadEvents("sentinel2b_mst00495", toKey="s2b", freq="5min", toleratedGap="2min", onlyOnce=True).to_csv("s2b_5min.csv")


# ## Spectral Analysis functions

# Next, three functions are prepared
# - loadEvents
# - Boxfit
# - LS fit

# ## Boxfit up to 1440 minutes

# In[17]:


test_periods   = np.arange(30, 1440, 0.1)
```

```

periodogram_a = sat2boxTriangle("s2a", test_periods, 50, cutDown=True)
periodogram_b = sat2boxTriangle("s2b", test_periods, 50, cutDown=True)

periodogram_a["period"] = periodogram_a["period"] / 60
periodogram_b["period"] = periodogram_b["period"] / 60


# In[18]:


plt.figure(figsize=(10, 4), dpi=150)

plt.semilogy(periodogram_a["period"], periodogram_a["power"], alpha=0.45, label="Sentinel␣2A")
plt.semilogy(periodogram_b["period"], periodogram_b["power"], alpha=0.45, label="Sentinel␣2B")

plt.xlabel("Period␣[minutes]")
plt.ylabel("Box-Fit␣Spectral␣Response")
plt.title("Box␣least-square␣Power␣of␣periodicities␣up␣to␣1d")
plt.grid()
plt.legend()

plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis␣Latex\\pyfig\\boxfit_1400mins.png")


# In[24]:


#hThreshold = 10

#pa = find_peaks(periodogram_a["power"], height=hThreshold, distance=10)
#pa = pd.DataFrame(np.stack([pa[0], pa[1]["peak_heights"]])).T.sort_values(by=1, ascending=False)[0].astype(int).values
#pb = find_peaks(periodogram_b["power"], height=hThreshold, distance=10)
#pb = pd.DataFrame(np.stack([pb[0], pb[1]["peak_heights"]])).T.sort_values(by=1, ascending=False)[0].astype(int).values
#df = pd.DataFrame([periodogram_a["period"][pa], periodogram_a["power"][pa],
#                    periodogram_b["period"][pb], periodogram_b["power"][pb]]).T

#df.columns = ["thesaperiod", "thesapower", "thesbperiod", "thesbpower"]
#df.index = df.index+1
#df.index.name = "therank"
#print(str(df.round(2).to_csv()))


# ## Boxfit up to 48h

# In[11]:


test_periods   = np.arange(3, 48, 0.05) * 60
```

```

periodogram_a = sat2boxTriangle("s2a", test_periods, 50, cutDown=True)
periodogram_b = sat2boxTriangle("s2b", test_periods, 50, cutDown=True)

periodogram_a["period"] = periodogram_a["period"] / 60 / 60
periodogram_b["period"] = periodogram_b["period"] / 60 / 60


# In[12]:


plt.figure(figsize=(10, 4), dpi=150)

plt.semilogy(periodogram_a["period"], periodogram_a["power"], alpha=0.45, label="Sentinel␣2A")
plt.semilogy(periodogram_b["period"], periodogram_b["power"], alpha=0.45, label="Sentinel␣2B")

plt.xlabel("Period␣[hours]")
plt.ylabel("Box-Fit␣Spectral␣Response")
plt.title("Box␣least-square␣Power␣of␣periodicities␣up␣to␣2d")
plt.grid()
plt.legend()
plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis␣Latex\\pyfig\\boxfit_48h.png")


# # Lomb-Scargle

# ## Lomb-Scargle up to 48 h

# In[13]:


pmin = "3h"
pmax = "48h"
step = 20 #5

periodogram_a = sat2ls("s2a", pmin=pmin, pmax=pmax, step=step, sampling="6040s", verbose=True, cutDown=False)
periodogram_b = sat2ls("s2b", pmin=pmin, pmax=pmax, step=step, sampling="6040s", verbose=True, cutDown=False)

periodogram_a["period"] = periodogram_a["period"] / 60 / 60
periodogram_b["period"] = periodogram_b["period"] / 60 / 60


# In[14]:


plt.figure(figsize=(10, 4), dpi=150)

plt.semilogy(periodogram_a["period"], periodogram_a["power"], alpha=0.45, label="Sentinel␣2A")
plt.semilogy(periodogram_b["period"], periodogram_b["power"], alpha=0.45, label="Sentinel␣2B")

plt.xlabel("Period␣[hours]")
plt.ylabel("Lomb-Scargle␣Spectral␣Response")
plt.title("Lomb-Scargle␣Power␣of␣periodicities␣up␣to␣2d")
plt.grid()
plt.legend()
plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis␣Latex\\pyfig\\ls_48h.png")


# ## Lomb-Scargle up to 10d

# In[15]:


pmin = "3d"
pmax = "10d"
step = 100 #5

periodogram_a = sat2ls("s2a", pmin=pmin, pmax=pmax, step=step, sampling="6040s", verbose=True, cutDown=False)
periodogram_b = sat2ls("s2b", pmin=pmin, pmax=pmax, step=step, sampling="6040s", verbose=True, cutDown=False)

periodogram_a["period"] = periodogram_a["period"] / 60 / 60 / 24
periodogram_b["period"] = periodogram_b["period"] / 60 / 60 / 24


# In[16]:


plt.figure(figsize=(10, 4), dpi=150)

plt.semilogy(periodogram_a["period"], periodogram_a["power"], alpha=0.45, label="Sentinel␣2A")
plt.semilogy(periodogram_b["period"], periodogram_b["power"], alpha=0.45, label="Sentinel␣2B")

plt.xlabel("Period␣[days]")
plt.ylabel("Lomb-Scargle␣Spectral␣Response")
plt.title("Lomb-Scargle␣Power␣of␣periodicities␣up␣to␣20d")
plt.grid()
plt.legend()
plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis␣Latex\\pyfig\\ls_10d.png")


# ## Lomb-Scargle up to 50d

# In[17]:


pmin = "10d"
pmax = "50d"
step = 250 #5

```

```
periodogram_a = sat2ls("s2a", pmin=pmin, pmax=pmax, step=step, sampling="6040s", verbose=True, cutDown=False)
periodogram_b = sat2ls("s2b", pmin=pmin, pmax=pmax, step=step, sampling="6040s", verbose=True, cutDown=False)

periodogram_a["period"] = periodogram_a["period"] / 60 / 60 / 24
periodogram_b["period"] = periodogram_b["period"] / 60 / 60 / 24


# In[18]:


plt.figure(figsize=(10, 4), dpi=150)

plt.semilogy(periodogram_a["period"], periodogram_a["power"], alpha=0.45, label="Sentinel␣2A")
plt.semilogy(periodogram_b["period"], periodogram_b["power"], alpha=0.45, label="Sentinel␣2B")

plt.xlabel("Period␣[days]")
plt.ylabel("Lomb-Scargle␣Spectral␣Response")
plt.title("Lomb-Scargle␣Power␣of␣periodicities␣up␣to␣50dd")
plt.grid()
plt.legend()
plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis␣Latex\\pyfig\\ls_50d.png")


# ## Testing discrete Periods w. Boxfit

# at orbital period

# In[21]:


df_events, periodogram, events_covered = bfmask("s2a", 12*60*60, 1000)

print(periodogram)
print("Events␣covered:␣" + str(events_covered))
fig = plt.gcf()
ax = plt.gca()
fig.set_size_inches(10,5)
ax = plt.axes(projection=ccrs.PlateCarree())
ax.set_global()
ax.coastlines()
#plt.title("Events in/out of 12h Box-Fit")
ax.scatter(df_events[df_events["masked"] == True]["lon"], df_events[df_events["masked"] == True]["lat"],
color="red", alpha=0.5, marker=".", label="Events␣within␣box-fit")
ax.scatter(df_events[df_events["masked"] == False]["lon"], df_events[df_events["masked"] == False]["lat"],
color="black", alpha=0.5, marker=".", label="Events␣out␣of␣box-fit")
ax.grid()
ax.legend()

```

```
plt.savefig("C:\\Users\\patri\\Dropbox\\Apps\\ShareLaTeX\\Thesis Latex\\
    pyfig\\boxfit_12h_map.png")


# In[ ]:
```