

# **Determining the accuracy and completeness of Burst Mode data collection from the Solar Orbiter SWA electron sensor**

*Student 18019006*

A dissertation submitted in partial fulfillment  
of the requirements for the degree of

**Master of Science**

of

**University College London.**

Department of Space and Climate Physics  
University College London

July 22, 2024

**Word Count: 11,703**

I, Student 18019006, confirm that the work presented in this thesis is my own.  
Where information has been derived from other sources, I confirm that this has been  
indicated in the work.

# Abstract

The Solar Orbiter Solar Wind Analyser Electron Analyser System (EAS) produces pitch angle distributions (PADs) of solar wind electrons. These distributions are critical to the investigation of kinetic electron plasma dynamics, particularly at sub-second timescales. To achieve this temporal resolution, EAS must operate in “Burst Mode”, for which it relies on data from the Solar Orbiter Magnetometer (MAG) to be transmitted in real-time over the onboard S20 inter-instrument data link. However, by comparing onboard MAG data to calibrated MAG data from the Solar Orbiter Archive (SOAR), this project shows that the MAG data received by EAS is prone to inaccuracies. These inaccuracies are shown to be significant enough to result in the loss of data in Burst Mode PADs. This report details methodologies developed to visualise EAS and MAG data in novel ways, facilitating the development of algorithms that quantify MAG data inaccuracies and their effects on EAS data products, including a computationally inexpensive metric,  $C$ , for assessing EAS Burst Mode PAD data completeness and accelerating future solar wind electron research. This report also details an investigation of one possible cause of MAG data inaccuracy; time latency over the S20 data link itself. An approach was developed to quantify this latency error using cross-correlation of MAG time series received onboard with calibrated MAG time series available on the ground. A Monte Carlo method was applied to estimate time delay uncertainty for sample time series data, yielding promising, yet inconclusive results. Additionally, this report describes some of this project’s unexpected discoveries, including an error in the data processing pipeline onboard Solar Orbiter which led a significant amount of once-publically-available EAS data to be expunged from SOAR.

# **Impact Statement**

Within academia, this project primarily presents the concepts for tools and methodologies that could be used to augment existing repositories for space plasma physics science data, such as the online Solar Orbiter Archive, providing benefits in the form of useful metrics and data visualisation. Hopefully, this could accelerate future research in kinetic plasma physics and help to answer fundamental questions about magnetic turbulence and re-connection in the solar wind. The impact of this project's primary objectives is somewhat overshadowed by the accidental discovery of an error in Solar Orbiter's onboard data processing pipeline which led to the removal of  $\sim 6$  months of erroneous science data from the Solar Orbiter Archive.

Without academia, plasma physics research is important because of its unpredictable effects on countless industries, including the medical industry with novel approaches to hadron therapy, and the energy industry with exotic approaches to controlled fusion.

# Acknowledgements

Thank you to my supervisor, Chris, for his guidance and support.

Thank you, as well, to my other professors, Andrew, Alan, Bryan, Daniel, Georgios, Hamish, Ian, Kinwah, Marek, Mat, Matt, Paul, Saeed, Sarah, Steve, and Tom.

Thank you to Kwesi, Martha, and Iolanda for living with me. Thank you to Lilia and all my other friends and family for being in my life.

Thank you to Gang KBO for giving me time off from my spacecraft & subsystems responsibilities so I could finish this report.

# Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Solar Orbiter . . . . .	16
1.2	SWA-EAS . . . . .	17
1.3	MAG . . . . .	21
1.4	EAS Pitch Angle Distributions . . . . .	23
1.4.1	EAS Normal Mode . . . . .	25
1.4.2	EAS Burst Mode . . . . .	27
<b>2</b>	<b>Methods</b>	<b>31</b>
2.1	Data Access . . . . .	31
2.2	Time Axis Cropping . . . . .	33
2.3	Coordinate Systems . . . . .	35
2.4	Preliminary $B_{EAS}$ - $B_{MAG}$ Comparison: Method . . . . .	37
2.5	Simulating EAS Burst Mode Steering . . . . .	38
2.6	Visualisation for Bins and Vectors . . . . .	39
2.7	S20 Link Latency . . . . .	43
<b>3</b>	<b>Results and Discussion</b>	<b>49</b>
3.1	Preliminary $B_{EAS}$ - $B_{MAG}$ Comparison: Results . . . . .	49
3.2	SWA-DPU Binning Algorithm . . . . .	52
3.3	Completeness Algorithm . . . . .	58
3.4	S20 Link Latency . . . . .	69
<b>4</b>	<b>Conclusion</b>	<b>72</b>

<i>Contents</i>	7
-----------------	---

<b>Appendices</b>	<b>74</b>
<b>A EAS Angular Bin Tables</b>	<b>74</b>
<b>B Code Excerpts</b>	<b>77</b>
<b>C Colophon</b>	<b>79</b>
<b>Bibliography</b>	<b>81</b>

# List of Figures

- |   |    |
|---|----|
| 1.1 An image of Solar Orbiter with the -Y side panel removed to indicate the locations of onboard scientific instruments[1]. The boom is shown on the left-hand side of the image. Starting from the tip of the boom, in order of proximity to the spacecraft bus, the indicated instruments are SWA-EAS, MAG-OBS, RPW-SCM, and MAG-IBS[2][3]. At the bottom right of the image is SWA-PAS, and at the top right is SWA-HIS[4]. . . . .       | 16 |
| 1.2 An annotated schematic depicting SWA-EAS as it appears on the end of the Solar Orbiter instrument boom (see Figure 1.1), along with the axes of a few salient reference frames[5]. The spacecraft bus is towards the +X direction in the S/C (“spacecraft”) frame. The EAS1 sensor head (right) and the EAS2 sensor head (left) can be seen attached to a rectilinear electronics box. . . . .  | 18 |
| 1.3 A diagram showing a cross-section of an EAS sensor head with an example electron trajectory illustrated in blue, and the following annotated subsystems[4]: (1) “top-cap” anode for controlling the geometric factor; (2) (a) sensor shielding grid; (2) (b,c) deflector electrodes for elevation selection; (3) (a,b) hemispherical electrodes for energy selection; (4) MCP subsystem; (5) charge amplifier integrated circuit. . . . . | 20 |

1.4	An image of one of the Solar Orbiter fluxgate magnetometer sensors as seen in the lab with its protective radiation shield removed[2]. One ring core with its bright red outermost sense winding can be seen mounted on the underside of a white, ceramic bracket. The other core is unseen, mounted on the other side. . . . .	22
1.5	Pitch angle data recorded by Solar Orbiter SWA-EAS in Burst Mode over 5 minutes on 26th June 2020[5]. Electron counts are plotted as a PSD. Time is plotted in Universal Time. Electron pitch angle is plotted in degrees. . . . .	23
1.6	A plot of EAS1 (blue) and EAS2 (red) elevation and azimuth bins projected onto the sky in spherical spacecraft reference frame (SRF) coordinates. Note that “elevation” and “azimuth” as labelled in the axes refer to the coordinates in SRF; <i>not</i> the elevation/azimuth in the sensor head frames (e.g. Figure 1.2). Note also that between EAS1 and EAS2, it is possible to cover the entire sky with some overlap. . .	27
1.7	Another plot of projected EAS1 elevation and azimuth bins in SRF overlaid with a magnetic field vector from MAG. The point parallel to the vector is indicated by the green diamond, and the point antiparallel to the vector is indicated by the green square. The green contours represent pitch angles in 15°increments from 15°to 165°. MAG vector data are taken from the Solar Orbiter Archive, from a period of data collection on 12th June 2023. . . . .	28
1.8	A similar plot to Figure 1.7. Additionally, the bands of EAS1 elevation bins containing the parallel and antiparallel magnetic field vectors are highlighted in red, and the elevation and azimuth pixels containing the same vectors are highlighted in green. MAG vector data are taken from the Solar Orbiter Archive, from a period of data collection on 12th June 2023. . . . .	29

2.1	Example $B_{EAS}$ data from 31st August 2023. Top panel: Default cartesian $B_{EAS}$ in SRF frame. 2nd panel: Cartesian $B_{EAS}$ in EAS1 frame. 3rd panel: Elevation for spherical $B_{EAS}$ in EAS1 frame. bottom panel: Azimuth for spherical $B_{EAS}$ in EAS1 frame. The radius in spherical coordinates is not shown because all time series vectors are of length unity. . . . .	36
2.2	Three frames of an animation produced by plotting $B_{EAS}$ and $B_{MAG}$ data from 1st January 2023 along with pitch angle contours including loss contours, which are described in Section 3.3. Green represents $B_{EAS}$ and purple represents $B_{MAG}$ . Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Top panel: Data from 17:33:31 UTC. Middle panel: Data from 17:33:32 UTC. Bottom panel: Data from 17:33:33 UTC. . . .	42
2.3	Two figures illustrating CCF time delay determination. (a): Two arbitrary time series $x(t)$ and $z(t)$ , which are well-approximated as identical time series with a time delay $\tau_0$ between them. (b): A plot of the CCF applied to $x(t)$ and $z(t)$ , showing maximum correlation at the time delay $\tau_0$ , representing the latency. . . . .	44
2.4	Two figures illustrating the time delay algorithm for dummy data. Top panel: Normalised dummy time signals $f(t)$ and $g(t)$ plotted over 400s with a time delay $\tau_0$ between them. The peaks of $f(t)$ and $g(t)$ are labelled with their time values. Bottom panel: The CCF of $f(t)$ and $g(t)$ plotted over 800s of time delays $\tau$ . The peak of the CCF is labelled with its value for the time delay $\tau_r \approx \tau_0$ . . . . .	45
2.5	A histogram of the probability density function describing the $M = 1000$ simulations used to calculate the uncertainty in $\tau_r$ for the dummy data used in Figure 2.4. There are 10 equally-spaced time delay bins. . . . .	47

3.1 Example $B_{EAS}$ data from a 5 minute period of EAS Burst Mode on 31st August 2023. Top panel: Elevation for $B_{EAS}$ and $B_{MAG}$ in spherical EAS1. Middle panel: Azimuth for $B_{EAS}$ and $B_{MAG}$ in spherical EAS1. Bottom panel: Angular difference between $B_{EAS}$ and $B_{MAG}$ . . . . .	50
3.2 A detailed look at the last $\sim 5$ s of Figure 3.1. Top panel: Elevation for $B_{EAS}$ and $B_{MAG}$ in spherical EAS1. Middle panel: Azimuth for $B_{EAS}$ and $B_{MAG}$ in spherical EAS1. Bottom panel: Angular difference between $B_{EAS}$ and $B_{MAG}$ . . . . .	51
3.3 Example $B_{EAS}$ data from a 15-minute period of EAS Burst Mode on 5th November 2023 showing a simulation of the Burst Mode steering algorithm. The reference frame for the top two panels is labelled “EASX” to indicate that it switches between EAS1 and EAS2 over the 15 minute period depending on $B_{EAS}$ orientation and resulting head selection. Top panel: Elevation for $B_{EAS}$ in spherical EASX coordinates, along with the centers and upper/lower bounds of the selected elevation bins (“El_Bin_Center” and “El_Bin_Boundaries” respectively). Middle panel: Azimuth for $B_{EAS}$ in spherical EASX coordinates, along with the centers and upper/lower bounds of the selected azimuth bins (“Az_Bin_Center” and “Az_Bin_Boundaries” respectively). Bottom panel: The selected head over time. . . . .	52
3.4 A detailed view of the first $\sim 2.5$ minutes of Figure 3.3. Top panel: Elevation + binning for $B_{EAS}$ in spherical EASX coordinates. Middle panel: Azimuth + binning for $B_{EAS}$ in spherical EASX coordinates. Bottom panel: The selected head over time. . . . .	53

- 3.5 Similar data to Figure 3.3, but plotting the “Used” bin centers, bin bounds, and sensor heads. Top panel: Elevation for  $B_{EAS}$  in spherical EASX coordinates, along with centers and upper/lower bounds of the used elevation bins (“El\_Bin\_Center” and “El\_Bin\_Boundaries” respectively). Middle panel: Azimuth for  $B_{EAS}$  in spherical EASX coordinates, along with the centers and upper/lower bounds of selected azimuth bins (“Az\_Bin\_Center” and “Az\_Bin\_Boundaries” respectively). Bottom panel: The selected, or “Calculated” head and used head over time. . . . . 55
- 3.6 A detailed view of the first  $\sim 2.5$  minutes of Figure 3.3. Top panel: Elevation + used binning for  $B_{EAS}$  in spherical EASX. Middle panel: Azimuth + binning for  $B_{EAS}$  in spherical EASX. Bottom panel: The selected, or “Calculated” head and used head over time. . . . . 56
- 3.7 A cartoon to illustrate the process by which a discrepancy in the two elevation binning tables used by SWA DPU can lead to incorrect elevation binning. The vertical axis indicates the elevation angle corresponding to various electron detections, which are represented by yellow circles. The first four bins for the two binning tables are represented by rectangles labelled with their indices and the angles of their upper and lower elevation bounds, and vertically scaled according to their elevation widths. The short arrows indicate the correspondence between the indices of the mismatched binning tables. 57
- 3.8 Top panel: A plot of magnetic field vectors and calculated bin selection from 12th June 2023. Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Green represents  $B_{EAS}$  and purple represents  $B_{MAG}$ .  $\alpha_{min}$  and  $\alpha_{max}$  are plotted as contours in red. Bottom panel: A plot of min and max pitch angles  $\alpha_{min}$  and  $\alpha_{max}$  and PAD completeness  $C$  over the same period.  $\alpha_{min}$  and  $\alpha_{max}$  are plotted on the angular axis on the left, and  $C$  is plotted on the percentage axis on the right. . . . . 61

3.9	Top panel: A plot of magnetic field vectors and bin selection from 12th June 2023, where the binning issue is visible. Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Green represents $B_{EAS}$ and purple represents $B_{MAG}$ . $\alpha_{min}$ and $\alpha_{max}$ are plotted as contours in red. Bottom panel: A plot of min and max pitch angles $\alpha_{min}$ and $\alpha_{max}$ and PAD completeness $C$ over the same period. $\alpha_{min}$ and $\alpha_{max}$ are plotted on the angular axis on the left, and $C$ is plotted on the percentage axis on the right. . . . .	65
3.10	A plot of magnetic field vectors and calculated bin selection from 1st January 2023, where the parallel and anti-parallel elevation bins are adjacent to each other. Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Green represents $B_{EAS}$ and purple represents $B_{MAG}$ . $\alpha_{min}$ and $\alpha_{max}$ are plotted as (barely visible) contours in red. . . . .	67
3.11	Top panel: Two “identical” plots of $B_{EAS}$ elevation in EAS1 coordinates with an artificial, 50s time delay between them. Bottom panel: The result of the CCF algorithm applied to the top panel, where the uncertainty in $\tau_r$ is indicated by the small, horizontal, blue bar. . . . .	70
3.12	Top panel: Two “identical” plots of $B_{EAS}$ elevation in EAS1 coordinates with an artificial, 5s time delay between them. Bottom panel: The result of the CCF algorithm applied to the top panel. . . . .	71

# List of Tables

2.1	A table of transformation matrices between SRF and EAS1/2 coordinates in cartesian space. The matrices in the left column were found on SOAR (in reality, $\pm\sqrt{2}/2$ was approximated as $\pm 0.707$ on SOAR).	37
A.1	Bin Table: EAS1/2 Azimuth (as found in SOAR) - 17th July 2021 . . .	75
A.2	Bin Table: EAS1 Elevation (as found in SOAR) - 17th July 2021 . . .	76
A.3	Bin Table: EAS2 Elevation (as found in SOAR) - 17th July 2021 . . .	76

## **Chapter 1**

# **Introduction**

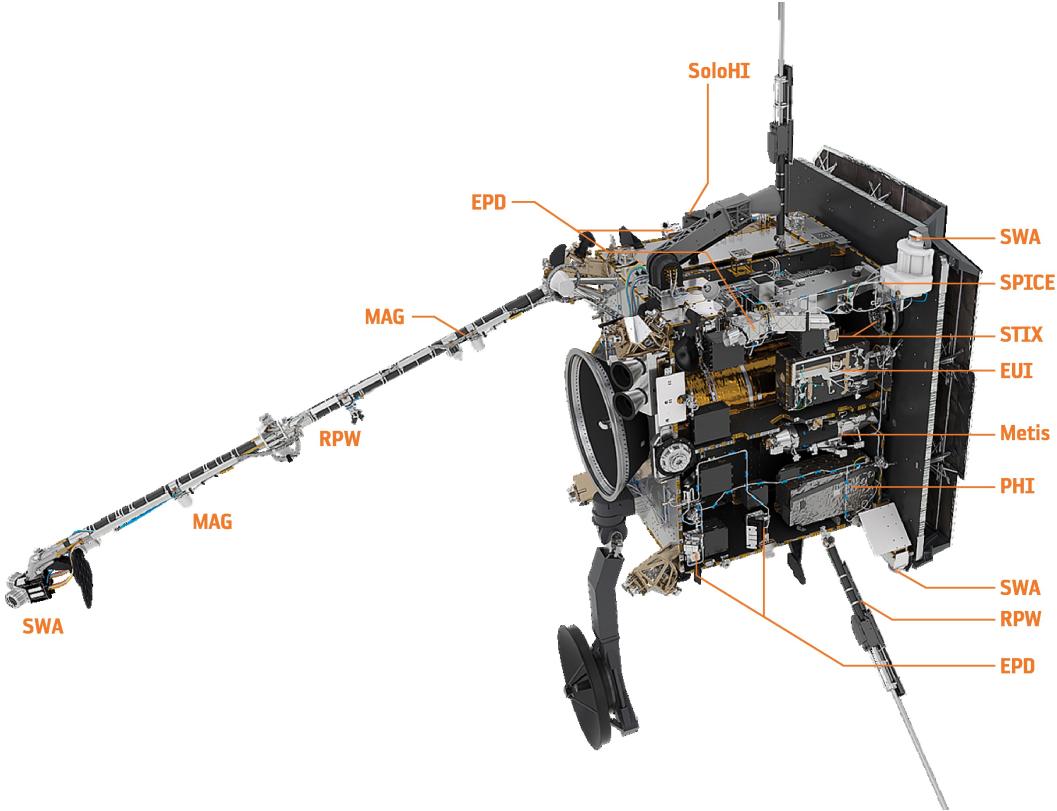
“although the tracking of the B-field vector received over the S20 link is excellent, the quality and relevance of the resulting PAD can only be as good as the quality and relevance of the input data”[5].

- Chris Owen, 2021

The Solar Orbiter Electron Analyser System instrument (EAS) is used to detect and characterise electrons in the solar wind. EAS can be commanded to follow various modes of operation depending on solar activity, spacecraft telemetry constraints, and other drivers. “Burst Mode” is a high-cadence operating mode that uses data from the Solar Orbiter Magnetometer instrument to make measurements at a timescale where electron plasma dynamics can be studied in particular detail. Unfortunately, the data transmitted from the magnetometer to EAS is of variable quality and relevance to the needs of EAS Burst Mode. The goal of the project described in this report is to investigate the quality and relevance of those transmitted magnetometer data, and the implications of their quality and relevance for EAS data products.

The following sections provide a background to the project and a description of the project’s goals in detail.

## 1.1 Solar Orbiter



**Figure 1.1:** An image of Solar Orbiter with the -Y side panel removed to indicate the locations of onboard scientific instruments[1]. The boom is shown on the left-hand side of the image. Starting from the tip of the boom, in order of proximity to the spacecraft bus, the indicated instruments are SWA-EAS, MAG-OBS, RPW-SCM, and MAG-IBS[2][3]. At the bottom right of the image is SWA-PAS, and at the top right is SWA-HIS[4].

Image Source: Müller et al[1]

The overarching science objective of the Solar Orbiter mission is “How does the Sun create and control the Heliosphere – and why does solar activity change with time?”[1]. In service of this objective, the Solar Orbiter spacecraft carries a payload of scientific instrumentation along an inclined, eccentric solar orbit taking it to a perihelion of 0.28AU ( $\sim 60$  solar radii), and up to  $18^\circ$  out of the ecliptic plane (extended mission phase may allow up to  $30^\circ$ ). While the results from other heliophysics missions, including SOHO[6], HINODE[7], STEREO[8], and the Parker Solar Probe[9] have allowed the scientific community to make great strides towards answering similar questions to Solar Orbiter’s overarching science objective,

some gaps remain in our understanding of the inner heliosphere and the Sun’s poles[1]. Solar Orbiter aims to fill those gaps.

To that end, Solar Orbiter’s instrumentation payload consists of a suite of remote sensing telescopes for purposes including solar wind and photospheric imaging and spectroscopy, as well as a suite of in-situ instrumentation for purposes including solar wind particle mass spectrometry and magnetic field sensing. These instruments are arranged around the spacecraft as illustrated in Figure 1.1. Of particular interest to this project are the interactions between two of these in-situ instruments: The first is the Solar Wind Analyser Electron Analyser System (SWA-EAS or simply EAS); a sensor in the Solar Orbiter Solar Wind Analyser (SWA) suite of particle detectors. The second is the Solar Orbiter Magnetometer (MAG).

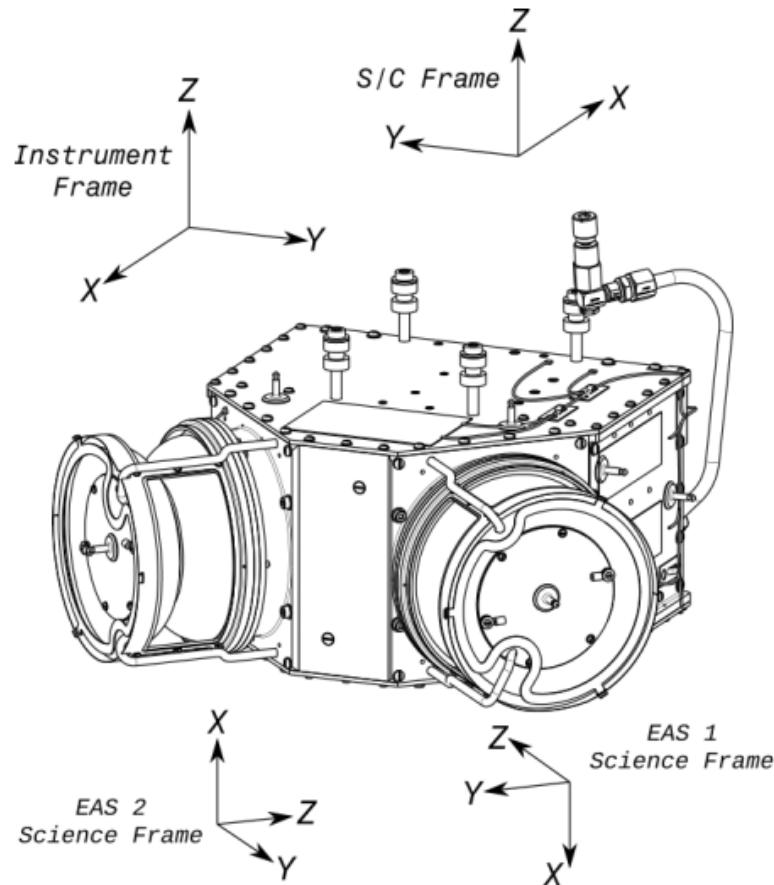
## 1.2 SWA-EAS

The Solar Wind Analyser suite comprises three in-situ particle sensors, each of which uses the well-established “top hat” electrostatic analyser design[10][4] to map incoming particles and their respective velocities/energies across each detector’s field of view. The resulting maps allow the determination of the 3D “velocity distribution functions” (VDFs) for the particles within a given energy range over the measurement period. The three SWA sensors are:

1. The Heavy Ion Sensor (SWA-HIS or HIS), which produces VDFs for alpha particles and heavier ions in the solar wind to determine their relative charge states and abundances, not only at thermal energies, but also at “suprothermal” energies above the solar wind bulk speed[11].
2. The Proton-Alpha Sensor (SWA-PAS or PAS), which produces VDFs for solar wind protons and alpha particles, mainly at thermal energies.
3. The Electron Analyser System, which produces VDFs for solar wind electrons at thermal and suprothermal energies.

Working in concert, these sensors provide important data about local solar

wind plasma to answer some of Solar Orbiter’s more specific science questions, such as “What are the source regions of the solar wind and heliospheric magnetic field?”, “How do coronal mass ejections (CMEs) evolve through the corona and inner heliosphere?” and “How and where are energetic particles accelerated at the Sun?”[4]. Of the three SWA sensors, this project is most concerned with EAS.



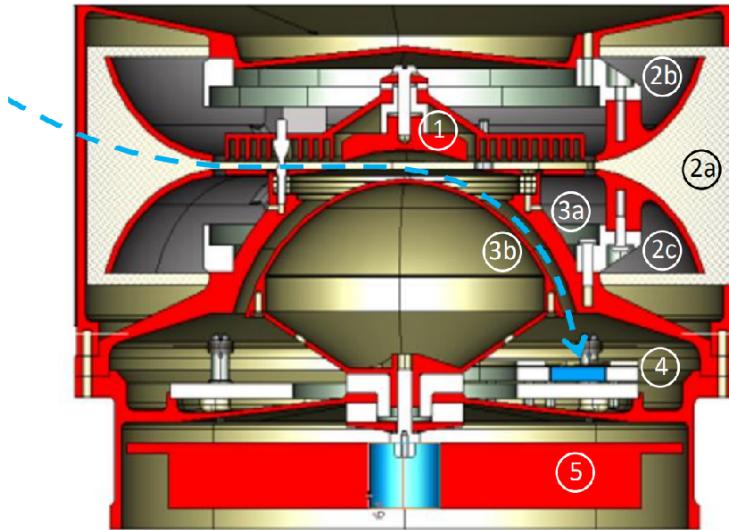
**Figure 1.2:** An annotated schematic depicting SWA-EAS as it appears on the end of the Solar Orbiter instrument boom (see Figure 1.1), along with the axes of a few salient reference frames[5]. The spacecraft bus is towards the +X direction in the S/C (“spacecraft”) frame. The EAS1 sensor head (right) and the EAS2 sensor head (left) can be seen attached to a rectilinear electronics box.

Image Source: Owen et al[5]

EAS consists of two electrostatic analysers that inherit several aspects of their design from previous instruments such as the Cassini Plasma Spectrometer (CAPS) Electron Spectrometer (ELS)[12], the Mars Atmospheric and Volatile EvolutioN

(MAVEN) Solar Wind Electron Analyser (SWEA)[13], and the Cluster Plasma Electron And Current Experiment (PEACE)[14]. The two analysers, also called the “heads” of the instrument[5], are orthogonal to each other as shown in Figure 1.2. Unlike HIS and PAS, which can afford relatively narrow, Sun-pointing fields of view (FOV) of  $33^\circ$  to  $+66^\circ \times \pm 20^\circ$  and  $24^\circ$  to  $+42^\circ \times \pm 22.5^\circ$  respectively, the EAS sensor heads each observe  $360^\circ$  of azimuth and  $\pm 45^\circ$  of elevation in their respective reference frames (see “EAS 1/2 Science Frame” in Figure 1.2), theoretically covering more than the sky’s full  $4\pi$  steradians between them, where  $1.6\pi$  steradians are covered by the overlap between the two FOVs (see Figure 1.6)[4]. This FOV is required because, in comparison to protons and other ions, electron thermal velocities are expected to be much higher than the solar wind plasma bulk velocities. As a result, while protons and other ions are expected to enter their respective sensor primarily from the Sun-facing direction, electrons are expected to enter EAS more isotropically[4]. In reality, some of the EAS FOV is always obstructed by a set of support pillars and a shielding grid (to avoid spacecraft engine exhaust contamination) around each head, as well as the Solar Orbiter bus high-gain antenna and solar arrays[4][5][15]. To minimise this obstruction, as well as to avoid electrostatic interference from the rest of the spacecraft, EAS is positioned at the far end of the 4.4m Solar Orbiter instrument boom (see Figure 1.1)[4][16].

Ignoring obstructions, solar wind electrons can enter each sensor head from any elevation and azimuthal angle in its field of view. As they do so, the electrons fly through an electric field generated by a pair of deflector electrodes at the entrance of the head (subsystems 2b and 2c in Figure 1.3). The electrodes are charged with positive voltages such that only incoming electrons from a narrow, selected range of elevation angles can continue through the sensor head. Electrons with elevations outside of this range will collide with the sensor walls and go undetected. This range corresponds to a single elevation angle bin. Electrons within the selected elevation range subsequently fly through a pair of hemispherical electrodes (subsystems 3a and 3b in Figure 1.3). These electrodes create an electric field that only allows



**Figure 1.3:** A diagram showing a cross-section of an EAS sensor head with an example electron trajectory illustrated in blue, and the following annotated subsystems[4]: (1) “top-cap” anode for controlling the geometric factor; (2) (a) sensor shielding grid; (2) (b,c) deflector electrodes for elevation selection; (3) (a,b) hemispherical electrodes for energy selection; (4) MCP subsystem; (5) charge amplifier integrated circuit.

Image Source: Owen et al[4]

electrons within a narrow, selected range of velocities/energies to continue to the detector proper, filtering the electron population again. This range corresponds to a single energy bin. If an electron passes through these electron optics, then its trajectory ends when it impinges on a segment of an annular microchannel plate (MCP) corresponding to its azimuthal angle (subsystem 4 in Figure 1.3). As a result, the incoming electron induces the emission of  $2 \times 10^6 - 5 \times 10^6$  secondary electrons which collect at the MCP anode, resulting in a characteristic voltage pulse that triggers a detection[4].

During EAS operation, the deflector electrodes sweep through up to 16 elevation bins, sequentially, in the range of  $\pm 45^\circ$ . The elevation bins are unequally sized according to a<sup>1</sup> table of voltages aboard Solar Orbiter, which is designed to account

---

<sup>1</sup>SWA-EAS has used different elevation bin tables at different times over the course of the Solar Orbiter mission. Due to an error in the EAS data acquisition pipeline discovered during this project, this table has occasionally been incorrect (see Section 3.2).

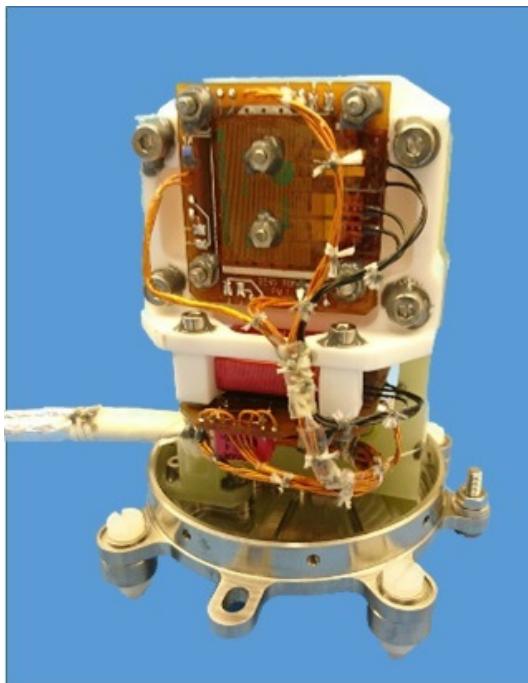
for increasing angular resolution at increasing deflection angles, with bin widths varying from  $\sim 2^\circ$  to  $\sim 10^\circ$ . For each elevation bin, the hemispherical electrodes sweep through up to 64 logarithmically-spaced energy bins, sequentially, in the range [ $\sim 1\text{eV}, \sim 5\text{keV}$ ]. The full range of  $0\text{-}360^\circ$  of azimuth is split into 32 equally-spaced azimuthal bins, all of which are sampled simultaneously for each combination of elevation and energy. The result is a distribution of solar wind electron energies per solid angle; an electron VDF.

## 1.3 MAG

The Solar Orbiter magnetometer consists of two three-axis fluxgate magnetometers located on the Solar Orbiter instrument boom. The inboard sensor (MAG-IBS) and outboard sensor (MAG-OBS) respectively (see Figure 1.1). The two sensors are at different positions along the boom to minimise the risk of magnetic contamination by the rest of Solar Orbiter both by distance attenuation and by subtracting the effect of Solar Orbiter through magnetic gradiometry. The design of MAG borrows heavily from that of previous magnetometers flown on the Cassini[17] and Double Star[18], missions, and forms the basis for future magnetometers on missions including IMAP[19][15]. The author has previously described the operation of a fluxgate magnetometer as follows[15]:

Each magnetometer consists of two orthogonal fluxgate ring-cores (one of which is partially visible in Figure 1.4). The fluxgate cores consist of three copper windings; One inner “drive” winding and two outer, orthogonal, “sense” windings. The inner “drive” winding is wound around a ring of magnetically-permeable material which it drives into successive states of oppositely-polarised magnetic saturation at the “drive frequency” through magnetic induction. In an environment with no external magnetic field, the magnetic field induced in one half of the ring is designed to perfectly counteract the antisymmetric field induced in the opposite half, resulting in zero magnetic field in the ring overall. However, if an external magnetic field is present and oriented along the plane of the ring, then the drive winding will drive

the ring core into magnetic saturation more quickly in the direction parallel to the external magnetic field and more slowly in the direction antiparallel to the external magnetic field. The effect of such an external magnetic field is to temporarily break the symmetry of the induced magnetic field in the ring core, causing a pulse of magnetic flux density which repeats at twice the drive frequency. These pulses induce voltages in the outer sense windings from which a 2D external magnetic field vector can be recovered. With two orthogonal ring-cores, each fluxgate magnetometer can recover an external magnetic field vector in full 3D[2][15].



**Figure 1.4:** An image of one of the Solar Orbiter fluxgate magnetometer sensors as seen in the lab with its protective radiation shield removed[2]. One ring core with its bright red outermost sense winding can be seen mounted on the underside of a white, ceramic bracket. The other core is unseen, mounted on the other side.

Image Source: Horbury et al[2]

The measured values of the axial components of the external magnetic field vector are known to occasionally “drift” due to imperfections in the magnetometer hardware, leading to offsets that affect magnetic field vector orientation. These offsets are corrected through data processing on the ground and through occasional updates to the software aboard Solar Orbiter[2]. The central issue in this project is

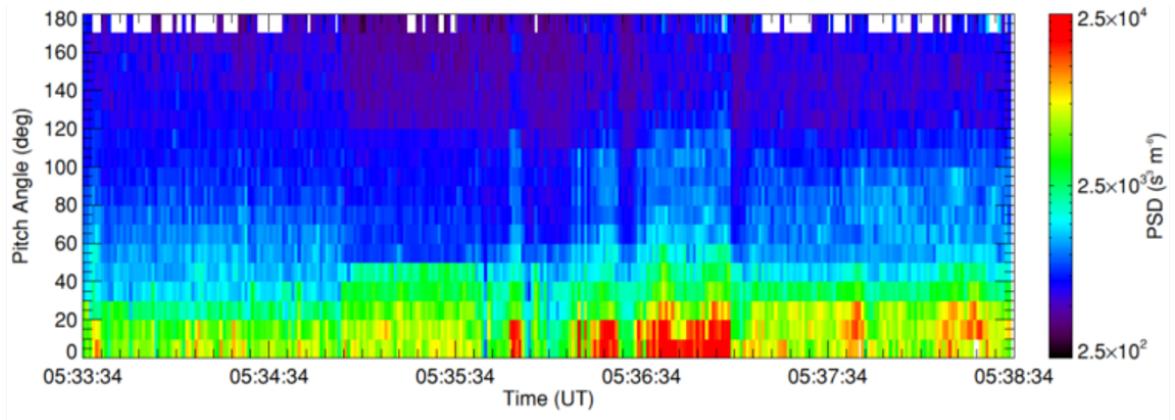
the effect of onboard MAG data inaccuracies, including offset, on EAS pitch angle data.

## 1.4 EAS Pitch Angle Distributions

A particle’s pitch angle, often denoted  $\alpha$ , is the angle between the particle’s velocity vector  $v$  and the local magnetic field vector  $B$ .  $\alpha$  is given by:

$$\alpha = \arctan \frac{v_{\perp}}{v_{\parallel}} \quad (1.1)$$

where  $v_{\parallel}$  and  $v_{\perp}$  are the particle’s velocity components parallel and perpendicular to  $B$  respectively[20]. Analysis of electron pitch angle distributions (PADs) has revealed distinct electron populations in the solar wind. Some populations have a tendency to collide amongst themselves, resulting in a “thermal” distribution with mainly isotropic pitch angles that can be modeled, to within a close approximation, by a Maxwell-Boltzmann distribution. These represent the “bulk” of the electron plasma. Other populations are more anisotropic, forming narrow electron beams with pitch angles that may be field-aligned, making up the so-called electron “strahl”. These anisotropic electrons tend to collide less frequently, so their trajectories are less uniformly distributed[20][21].



**Figure 1.5:** Pitch angle data recorded by Solar Orbiter SWA-EAS in Burst Mode over 5 minutes on 26th June 2020[5]. Electron counts are plotted as a PSD. Time is plotted in Universal Time. Electron pitch angle is plotted in degrees.

Image Source: Owen et al[5]

An example PAD generated using Solar Orbiter SWA-EAS data is shown in Figure 1.5 for a 5-minute period starting at 05:33:34 UT on 24th June 2020[5]. Shown here is the phase space density, or concentration, of strahl electrons with energies  $> 70\text{eV}$ . The concentration is presented in terms of power spectral density (PSD) in colours from black (fewest electrons) to red (most electrons)<sup>2</sup>. A different PAD is plotted at every 1/8th-second time step (corresponding to EAS Burst Mode; see below). Notice that, for each time step, only pitch angles are shown from  $0^\circ$  to  $180^\circ$ . It may seem that a full map of the sky would be required to fully capture electron concentrations at different phase angles about the local magnetic field at a given pitch angle, implying *two* angles in spherical coordinates for every time step. However, near this time resolution, it is expected that any instantaneous in-homogeneities in electron concentration at different gyrophase angles about the local magnetic field will disappear too quickly to discern, such that the concentration at all phase angles for a given pitch angle can be assumed to be uniform; in a state of so-called “gyrotropy”[5]. This is expected to be a safe assumption at this timescale[22][23][5]. This can be understood because the expected electron gyroperiod in the solar wind at  $\sim 1\text{AU}$  is much shorter than the 1/8th-second timescale. Assuming magnetic field strength  $B \approx 1\text{nT}$ , then the electron gyrofrequency  $\omega_e$  is given by:

$$\omega_e = \frac{eB}{m_e} = \frac{e \times 10^{-9}}{m_e} \approx 176 \text{ rad/s} \quad (1.2)$$

where  $m_e$  is the electron mass. The corresponding gyration frequency is therefore  $\omega/2\pi \approx 28.0/\text{s} \gg 8/\text{s}$ . Therefore, if this is a realistic estimate, then over a single 1/8th-second measurement, most electron agyrotropy is expected to be thoroughly averaged and rendered indiscernible. Lu et al 2020 has similarly supported the assumption of gyrotropy when using 1/4th-second time steps in 2D particle-in-cell simulations of magnetic reconnection at Earth, on the basis that the local electron gyroradius should be comparable in size to the local plasma ion skin depth - should this be analogous to the EAS environment with the solar wind’s

---

<sup>2</sup>White pitch angle bins on this plot represent missing data, relating to each PAD’s completeness score, which will be important in Section 3.3.

plasma parameters to within an order of magnitude, then the assumption of gyrotropy for EAS is expected to be safe[24][25].

The PAD in Figure 1.5 was taken while EAS was operating in “Burst Mode”, which differs from the more typical “Normal Mode” in that it generates PADs  $\sim 8 \times$  faster. A description of Normal Mode and Burst Mode PAD determination follows.

### 1.4.1 EAS Normal Mode

Most EAS data are collected in normal mode, where all 64 energy bins, 16 elevation bins, and 32 azimuth bins are sampled, yielding as close to a full-sky VDF as possible. The 32 azimuth bins are sampled simultaneously. Sampling a single energy bin for a single elevation bin takes  $\sim 0.96\text{ms}$ [5]. Therefore, cycling through all energy and elevation bins in sequence takes  $64 \times 16 \times 0.96\text{ms} = 0.983\text{s} \approx 1\text{s}$ , yielding Normal Mode data at a data rate of  $\sim 1\text{Hz}$ [5]. The pipeline involved in acquiring a VDF and a subsequent PAD in Normal Mode can be summarised as follows[4]:

1. Acquire a full-sky EAS VDF with 64 energy bins  $\times$  16 elevation bin  $\times$  32 azimuth bins at 1Hz,
2. Acquire a MAG vector at 1Hz,
3. Downlink EAS VDF and MAG vector to ground,
4. Post-process MAG vector on ground, applying any necessary corrections,
5. Combine EAS VDF and MAG vector to yield a PAD.

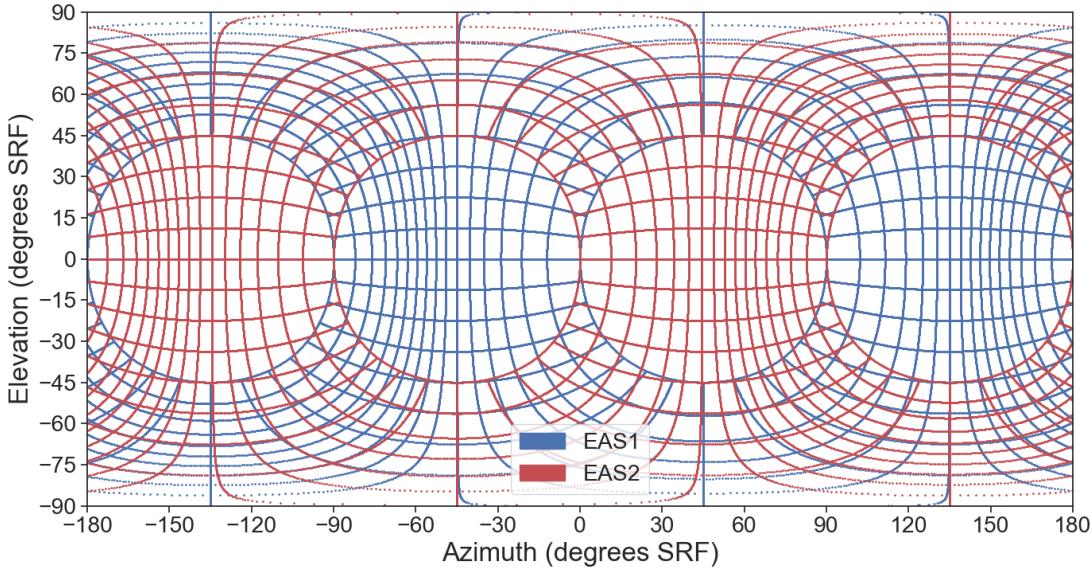
For a more detailed description, consider Figure 1.6. This figure shows the combined elevation and azimuth bin “pixels” available to each EAS sensor head (EAS1 in blue and EAS2 in red) projected on to a map of the full sky in spherical, spacecraft reference frame (SRF) coordinates where [0,0] points directly away from the Sun, and  $+90^\circ$  elevation points out of the ecliptic plane. This amounts to a total of  $16 = 512$  pixels per head. In normal mode, an EAS head first samples electrons from all of the pixels in its FOV. Meanwhile, MAG samples the local 3D

magnetic field vector in its own Burst Mode (128Hz) or Normal Mode (8Hz)[2]. Now, consider Figure 1.7, where a magnetic field vector from the same period as the 1s PAD is superimposed onto a projection of the EAS1 bins in SRF coordinates. Here, the green diamond represents the vector parallel to the magnetic field (i.e. the B-vector itself), and the green square represents the antiparallel vector. Surrounding the parallel vector are various contours of constant pitch angle at every  $15^\circ$  in the range  $[0^\circ, 180^\circ]$ . To aid explanation, let the vectors in Figure 1.7 represent the ground magnetic field post-processing<sup>3</sup>. In this case, wherever an EAS pixel is crossed by a pitch angle contour, that pixel has sampled electrons with the contour's pitch angle. Calculating the pitch angle for every pixel in the 3D VDF allows a 2D PAD to be constructed for every azimuthal bin, each of which corresponds to a range of gyrophase angles. By combining samples from all the azimuthal bins in post-processing (this is harmless under gyrotropy), the gyrophase dimension is collapsed, yielding a single, 2D PAD for the 3D VDF.

Even with compression, the volume of data implied by a full-sky, 3D VDF in Normal Mode has significant implications for the SWA telemetry budget, limiting full-cadence (1s/sample or 1Hz) data downlink to a 5-minute period per day. Instead, the SWA Data Processing Unit (SWA-DPU) typically downlinks moments of full VDFs at a cadence of 4s/sample, and full VDFs are limited to 10s/sample or 100s/sample cadences when downlinked. The limited telelmetry budget places even more constraints on what is already a relatively low cadence for measurements of solar wind electron dynamics (baseline of 1s/sample), motivating the design of a faster and less data-intensive method of operation; EAS Burst Mode.

---

<sup>3</sup>In reality, this vector was taken from data from the archive corresponding to the unprocessed magnetic field transmitted over the S20 data link in EAS Burst Mode (see Section 1.4.2). In this case, it is only treated as an analogy for the ground magnetic field.



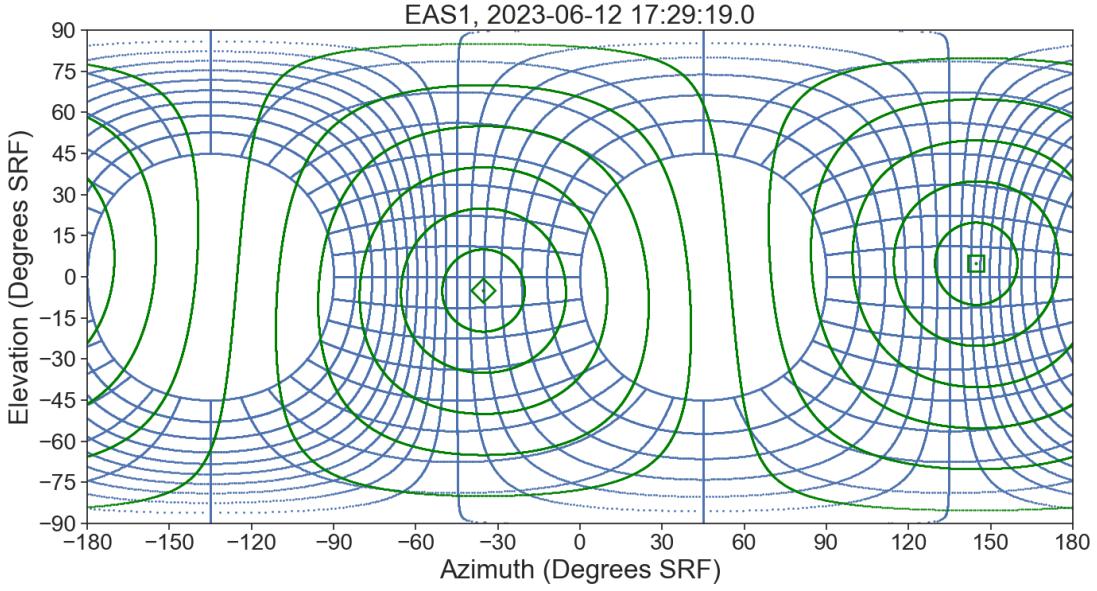
**Figure 1.6:** A plot of EAS1 (blue) and EAS2 (red) elevation and azimuth bins projected onto the sky in spherical spacecraft reference frame (SRF) coordinates. Note that “elevation” and “azimuth” as labelled in the axes refer to the coordinates in SRF; *not* the elevation/azimuth in the sensor head frames (e.g. Figure 1.2). Note also that between EAS1 and EAS2, it is possible to cover the entire sky with some overlap.

Image Source: Author’s own work

### 1.4.2 EAS Burst Mode

EAS Burst Mode data are typically collected over periods of  $\sim 5 - 15$  minutes, once per day. Compared to Normal Mode at full cadence, full Burst Mode PADs can be acquired after sampling only 2 of the 16 elevation bins in the sky, resulting in a theoretical  $8\times$  improvement in sample cadence (8Hz at  $\sim 0.125$ s/sample). Compared to other typical Normal Mode cadences of 4s/sample, 10s/sample, or 100s/sample, the improvement can be as good as  $32\times$ ,  $80\times$ , or  $800\times$  respectively. The Burst Mode PAD pipeline can be summarised as follows[5]:

1. Acquire a MAG vector at 8Hz,
2. Transmit the MAG vector directly to EAS via the onboard “S20” data link,
3. Select the EAS head that has the MAG vector closest to the center of its FOV,
4. For that head, select the **2** elevation bins that contain the parallel and antiparal-



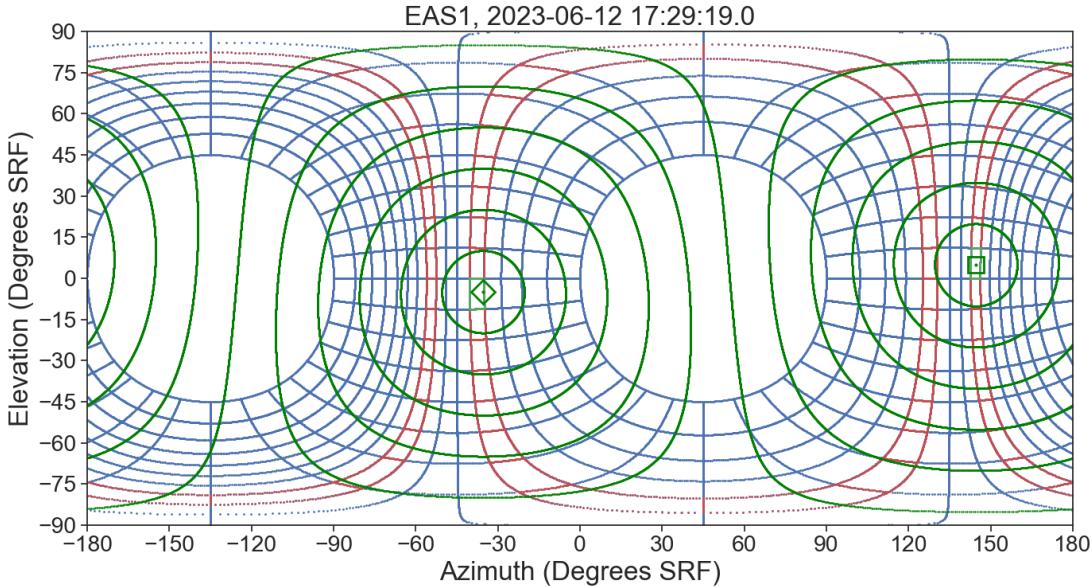
**Figure 1.7:** Another plot of projected EAS1 elevation and azimuth bins in SRF overlaid with a magnetic field vector from MAG. The point parallel to the vector is indicated by the green diamond, and the point anti-parallel to the vector is indicated by the green square. The green contours represent pitch angles in  $15^\circ$  increments from  $15^\circ$  to  $165^\circ$ . MAG vector data are taken from the Solar Orbiter Archive, from a period of data collection on 12th June 2023.

Image Source: Author's own work

parallel MAG vector in their respective FOVs,

5. Acquire a partial-sky VDF with  $64$  energy bins  $\times$   $2$  selected elevation bins  $\times$   $32$  azimuth bins at  $8\text{Hz}$ ,
6. Downlink EAS VDF to ground,
7. Re-bin EAS VDF on ground, yielding a full PAD.

For a more detailed description, consider Figure 1.8. This figure shows the same bins, vectors, and pitch angle contours as Figure 1.7, but the bands of elevation bins containing the parallel and antiparallel magnetic field vectors are highlighted in red, and the pixels containing the parallel and antiparallel magnetic field vectors are highlighted in green. Note that each elevation band is separated into azimuthal bins. Within the elevation band containing the parallel vector, as angular distance increases from the azimuthal bin/pixel containing the vector, successive azimuthal



**Figure 1.8:** A similar plot to Figure 1.7. Additionally, the bands of EAS1 elevation bins containing the parallel and antiparallel magnetic field vectors are highlighted in red, and the elevation and azimuth pixels containing the same vectors are highlighted in green. MAG vector data are taken from the Solar Orbiter Archive, from a period of data collection on 12th June 2023.

Image Source: Author's own work

bins are crossed by higher and higher pitch angles, culminating in an angle  $\alpha \geq 90^\circ$ . Conversely, as angular distance increases from the azimuthal bin containing the antiparallel magnetic field vector, successive azimuthal bins are crossed by lower and lower pitch angles, culminating in an angle  $\alpha \leq 90^\circ$ . For the sensor head containing the magnetic field vector in its FOV, the geometry of the sensor heads guarantees that this is always true[5]; the two opposite sides of a hypothetical elevation bin at  $0^\circ$  elevation would reach a maximum of  $\sim 180^\circ$  apart, and the opposite sides of an elevation bin at  $\pm 45^\circ$  would reach a minimum of  $\sim 90^\circ$  apart. Therefore, by “steering” EAS to sample each azimuthal bin across only the two highlighted elevation bands, it is possible to sample electrons with a full range of  $0^\circ$ - $180^\circ$  pitch angles. With the assumption of gyrotropy, this represents a full PAD (e.g. Figure 1.5). Since the 32 azimuth bins at each elevation are sampled simultaneously, this costs no additional time, resulting in an eight-fold sample time reduction.

To sample only the elevation bins containing the parallel and antiparallel magnetic field vector, EAS must first be told the orientation of the magnetic field. On Solar Orbiter, this is communicated via the “S20 inter-instrument communications link” which is transmitted by the MAG Instrument Control Unit (MAG-ICU) and received by the SWA-DPU[5][4][2]. Once received, SWA-DPU must select the EAS head that has the magnetic field closest to the center of its FOV, which is equivalent to finding the smallest angle between the vector and the sensor head aperture center-plane[5].

While Burst Mode boasts far-superior time resolution, it is also vulnerable to errors that Normal Mode avoids. By virtue of the fact that Burst Mode uses MAG data transmitted directly over the S20 data link, it therefore also does not benefit from post-processing of MAG data on the ground. Errors that may result in a discrepancy between the vectors used to “steer” EAS and the “more-correct” ground vectors include:

- Un-calibrated offset drift in MAG measurement axes, which may be more significant during the period leading up to a routine calibration update[5][2].
- Data latency in the S20 inter-instrument link itself, which may be inherent and/or exaggerated by periods of heavy data traffic[5].

When errors like these cause EAS to select incorrect elevation and azimuth bins in Burst Mode, this can lead to missing and mislabelled PAD data. The next chapter describes the methods applied to analyse the impact of these and other errors on Burst Mode PAD accuracy and completeness, and to encapsulate some of that impact in the form of a “completeness score”.

## Chapter 2

# Methods

To determine the accuracy and completeness of Burst Mode data collection it is useful to compare 3D magnetic field vectors used to steer EAS in Burst Mode (which will be called “ $B_{EAS}$ ”) to corresponding examples of the ground-calibrated and “true” magnetic field vectors (which will be called “ $B_{MAG}$ ”).

### 2.1 Data Access

The online Solar Orbiter Archive (SOAR) regularly publishes science data at various processing levels from Solar Orbiter instruments, including MAG and EAS, where they can be accessed by the public[26]. Most science data, including MAG vector time series and EAS Burst Mode PADs, are published in CDF, which is a “Self-describing data format for the storage of scalar and multidimensional data in a platform- and discipline-independent way”[27]. Developed by NASA Goddard Space Flight Center, CDF is an industry-standard data format used frequently in NASA/ESA projects[2]. CDF files accommodate data fields for different variables including science data and associated metadata/support data, along with various tags and descriptions. The primary data and metadata in a CDF file can be previewed via the SOAR interface, and accessed/manipulated through freely available libraries built for Python and MATLAB[27].

EAS and MAG data products are published to SOAR at various levels of us-

ability; some are “raw” data directly from the instrument, and others are thoroughly processed and ready to be used in conventional, publishable research[2][4][28]. The author has previously described the data levels for EAS and MAG as follows[15]:

#### **EAS Data Levels[4]:**

- **Level 0:** Uncalibrated CCSDS[29] binary data packets.
- **Level 1:** Uncalibrated electron count data converted to CDF in Burst Mode and Normal Mode. At this level, the MAG vectors used to steer EAS during Burst Mode ( $B_{EAS}$ ) are also included.
- **Level 2:** Calibrated electron count data including Burst Mode PADs, but excluding  $B_{EAS}$ .
- **Level 3:** A subset of level 2 data with additional processing.

#### **MAG Data Levels[2]:**

- **Level 0:** Uncalibrated binary data.
- **Level 1:** Uncalibrated time series data in nT in each MAG sensor’s reference frame.
- **Level 2:** Calibrated time series data in radial-tangent-normal (RTN), and Spacecraft Reference Frame (SRF) coordinates, corrected for offset drift and other inaccuracies, published at various cadences including E8 normal mode (8Hz) and E128 burst mode (128Hz).
- **Level 3:** A subset of level 2 data with additional processing.

EAS level 1 (L1) Burst Mode CDF files are typically published at a rate between  $\sim$ 5min/day and  $\sim$ 15min/day. Some of these files are published to SOAR under the name “solo\_L1\_swa-eas-padc\_YYYYMMddThhmmss-YYYYMMddThhmmss\_V01.cdf”, and these particular files are packaged with  $B_{EAS}$  as metadata in the form of normalised vectors in the SRF frame under the variable “SWA\_EAS\_MagDataUsed”.

MAG level 2 (L2) SRF CDF files, published to SOAR under “solo\_L2\_mag-srf-normal\_YYYYMMdd\_V01”, contain several hour’s worth of ground-calibrated 8Hz MAG data per day (i.e.  $B_{MAG}$ ), all in the same reference frame as  $B_{EAS}$  at the time of publishing (SRF), under the variable “B\_SRF”.

The desired CDFs were downloaded and manipulated using Python scripts with the *pycdf* package in the Spacepy library. Spacepy is “a package for Python, targeted at the space sciences, that aims to make basic data analysis, modeling and visualization easier”[30]. Pycdf allows individual CDF data fields to easily be referenced by name and converted to *numpy* arrays as required[31].

## 2.2 Time Axis Cropping

$B_{EAS}$  and  $B_{MAG}$  can be plotted together at 8Hz to yield an initial estimate of the discrepancies between them, but their recent, respective time series are uploaded to SOAR with very different durations (typically 5-15 minutes vs several hours respectively). This makes it difficult to compare one time series to the other numerically using Python, unless the longer time series is first cropped. Several algorithms were implemented to crop magnetic field time series. Initial attempts involved simply cropping the start and end of the longer series ( $B_{MAG}$ ) to the shorter series ( $B_{EAS}$ ). While this worked for some EAS CDFs (e.g. solo\_L1\_swa-eas-padc\_20231105T172733-20231105T174232\_V01.cdf, lasting 15mins, and solo\_L1\_swa-eas-padc\_20230831T172734-20230831T173233\_V01.cdf, lasting 5mins), this could not be generalised to some older EAS CDFs which combined multiple Burst Mode periods totalling 1hr or more of duration (e.g. solo\_L1\_swa-eas-padc\_20200624T053335-20200624T063334\_V01.cdf, lasting 1hr, and solo\_L1\_swa-eas-padc\_20221201T052733-20221201T220001\_V01.cdf, lasting  $\sim$ 17hrs). A persistent issue with these uploaded time series was that the first few data points were occasionally in the 1700s. At the time of writing, this is not signposted on SOAR, and this cropping algorithm does not crop these data points specifically. In the interest of generality, the final cropping algorithm was adapted to read the

timestamps in the titles of the CDFs as the start and end times to crop to, and both time series were cropped to those times instead. At present, this leads to a small amount of missing data near the start/end of a time series because the CDF filenames do not include millisecond accuracy, but this is easily fixed in the future.

Once the desired start and end of an arbitrary time series (this can be  $B_{MAG}$ , for example) are defined, an algorithm was implemented to efficiently identify the data points in that time series that should be cropped. The algorithm is summarised here:

1. Determine if the time series to be cropped overlaps with the desired start and stop times by performing two comparisons.
2. If the two time series overlap, then take the time in the middle of the series and determine if it is before or after the start time.
3. Whichever region the start time lies in, divide that region in half again and repeat  $N - 2$  times where  $N$  is the number of search divisions (typically,  $N = 5$ . More divisions show diminishing returns in efficiency).
4. Once the number of divisions has been reached, loop through the time series until the time is greater than or equal to the start time. Crop when you find the start time.
5. Starting from the start time, loop through the time series again until the time is greater than or equal to the stop time, and then crop again.

Implementations of this algorithm without step 3 had to loop through comparisons with every data point in a  $B_{MAG}$  time array that was several hours long. Given that the time values are saved as relatively inefficient `datetime.datetime()` values, the result was that these earlier implementations took  $>20$ s to complete. The inclusion of step 3 reduced this time to  $<2$ s. The core algorithm is written in Python as per Appendix Listing B.1, where `time_array` is the array of time values being cropped, `time_reference` is the time in `time_array` to crop at, `time_index` is an index in `time_array` within one index of `time_reference`, and `length` is the length of

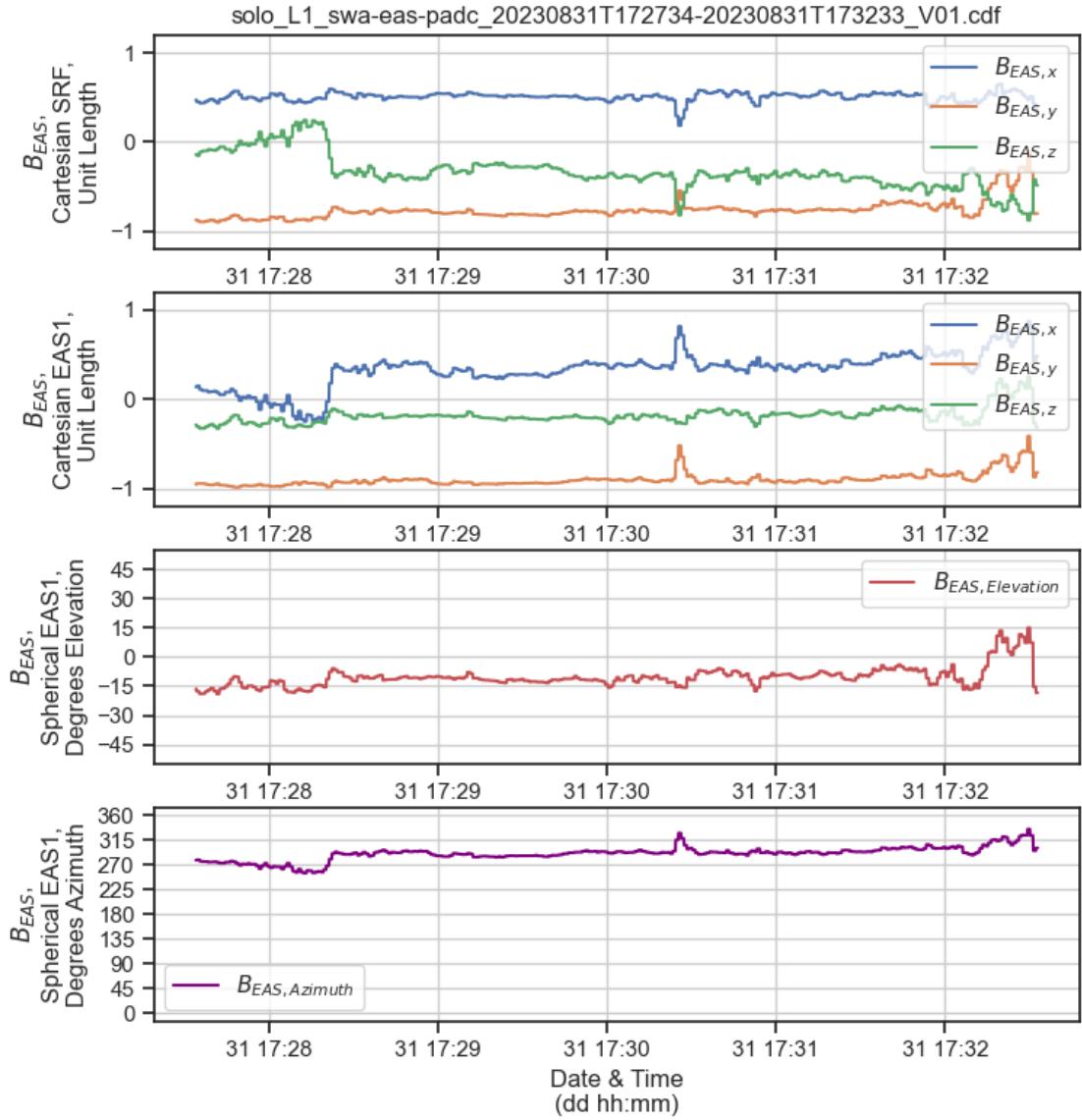
*time\_array*.

As-implemented, this algorithm sometimes leads to “off by one” errors in time series length. This may be the result of comparing time series that are not using the exact same time grid, because  $B_{EAS}$  is synchronised to EAS Burst Mode time and  $B_{MAG}$  is synchronised to MAG time. In response, if  $B_{EAS}$  and  $B_{MAG}$  are not the same length after cropping, then the longer of the two time series is cropped to the shorter starting with the last data point. This may lead to an artificial, “off-by-one” time delay of one data point (0.125s) between the two time series.

## 2.3 Coordinate Systems

$B_{EAS}$  and  $B_{MAG}$  are both acquired from SOAR in cartesian SRF coordinates. This is contrasted with data in cartesian radial-tangent-normal (RTN) coordinates (which are also provided for MAG[26]), where the “x-axis” is oriented radially away from the Sun, the “z-axis” is oriented towards the north celestial pole, and the “y-axis” is tangent to a prograde orbit in the plane of the ecliptic, completing the triad. SRF coordinates correspond to the “S/C Frame” shown in Figure 1.2, which is approximately equivalent to RTN coordinates with the “x and y” axes inverted. However, this equivalence depends on Solar Orbiter’s pointing accuracy, meaning that transformations between rigidly-coupled onboard reference frames (e.g. SRF and the EAS Instrument frame in Figure 1.2) are generally expected to be more consistent than transformations between onboard reference frames and celestial reference frames such as RTN[32].

To determine the effect of  $B_{EAS}$  and  $B_{MAG}$  on EAS elevation/azimuth binning, these time series had to be converted from cartesian SRF coordinates to spherical EAS1/EAS2 coordinates in terms of radius and elevation and azimuth angles, where the head should be chosen such that  $B_{EAS}$  and/or  $B_{MAG}$  lie within that head’s FOV. This was calculated in the following order: Cartesian SRF → Cartesian EAS1/2 → Spherical EAS1/2. This order was chosen because SOAR



**Figure 2.1:** Example  $B_{EAS}$  data from 31st August 2023. Top panel: Default cartesian  $B_{EAS}$  in SRF frame. 2nd panel: Cartesian  $B_{EAS}$  in EAS1 frame. 3rd panel: Elevation for spherical  $B_{EAS}$  in EAS1 frame. bottom panel: Azimuth for spherical  $B_{EAS}$  in EAS1 frame. The radius in spherical coordinates is not shown because all time series vectors are of length unity.

Image Source: Author's own work

provides “EASX\_TO\_SRF” transformation matrices packaged with level 2 EAS data CDF files published under “solo\_L2\_swa-easX-nm3d-dnf\_YYYYMMddThhhmmss-YYYYMMddThhhmmss\_V01.cdf” where  $X$  is the EAS head number (1 or 2). Despite being packaged with timestamped CDF files, these transformation matrices do not

appear to change over time (nor should they be expected to change over time). The matrices can be trivially inverted to enable a Cartesian SRF → Cartesian EAS1/2 transformation. The default and inverted forms are shown in Table 2.1, as would be expected geometrically based on Figure 1.2.

EAS1 to SRF	SRF to EAS1
$\begin{bmatrix} 0 & -\frac{\sqrt{2}}{2} & +\frac{\sqrt{2}}{2} \\ 0 & +\frac{\sqrt{2}}{2} & +\frac{\sqrt{2}}{2} \\ -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 \\ -\frac{\sqrt{2}}{2} & +\frac{\sqrt{2}}{2} & 0 \\ +\frac{\sqrt{2}}{2} & +\frac{\sqrt{2}}{2} & 0 \end{bmatrix}$
EAS2 to SRF	SRF to EAS2
$\begin{bmatrix} 0 & -\frac{\sqrt{2}}{2} & +\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ +\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \end{bmatrix}$

**Table 2.1:** A table of transformation matrices between SRF and EAS1/2 coordinates in cartesian space. The matrices in the left column were found on SOAR (in reality,  $\pm\sqrt{2}/2$  was approximated as  $\pm 0.707$  on SOAR).

Once in the EAS1/2 frame, the Cartesian EAS1/2 → Spherical EAS1/2 transformation was calculated using the *astropy* library for Python[33][34][35]. Figure 2.1 shows an example of normalised  $B_{EAS}$  data from 31st August 2023 at each stage of transformation from unit cartesian SRF to unit spherical coordinates in the EAS1 frame. Since the  $B_{EAS}$  vector time series is normalised from the start, a time series of the radius in spherical coordinates would be trivial and is not shown. In Figure 2.1, the x-axis, y-axis, and z-axis in the top panel (cartesian SRF) appear to align most closely with the z-axis (+ an offset), the y-axis (+ an offset), and the opposite of the x-axis in the 2nd panel (cartesian EAS1) respectively. This is also expected based on the “S/C” and “EAS 1 Science” frames in Figure 1.2, further validating the matrices in Table 2.1.

## 2.4 Preliminary $B_{EAS}$ - $B_{MAG}$ Comparison: Method

As described in the author’s earlier work, a preliminary comparison was performed between  $B_{EAS}$  and  $B_{MAG}$  from a small set of arbitrarily selected periods of Burst

Mode data to justify the creation of a completeness score[15].

$B_{EAS}$  time series are stored in EAS burst mode CDF files under “EPOCH” for the time axis and “SWA\_EAS\_MagDataUsed” for the series axis. Similarly,  $B_{MAG}$  time series are stored in MAG SRF normal mode CDF files under “EPOCH” and “B\_SRF” respectively. Every “B\_SRF” vector was normalised before comparison with  $B_{EAS}$ . The dot product between  $B_{EAS}$  and  $B_{MAG}$  vectors gives the angular offset between them. This angle was calculated for every vector in a few randomly chosen time series, yielding the results and discussion presented in Section 3.1, confirming the risk of PAD incompleteness for a few example datasets to justify the eventual development of a completeness metric.

## 2.5 Simulating EAS Burst Mode Steering

To better understand sources of error in Burst Mode data, the Burst Mode steering algorithm was simulated in Python using  $B_{EAS}$  vectors, Table 2.1, and the latest published binning tables shown in Appendix A. As per Section 1.4.2, the algorithm first requires a comparison between  $B_{EAS}$  and both EAS head aperture center planes in SRF coordinates, equivalent to their z-axes + 90° elevation in their respective EAS coordinates. This algorithm can be summarised as follows:

1. The angles between  $B_{EAS}$  and both EAS head aperture center planes are calculated.
2. The smallest angle selects the EAS frame to which  $B_{EAS}$  is transformed, in spherical coordinates.
3. The upper and lower bounds of the bin tables are compared to the elevation and azimuth angles of each vector in the spherical EAS frame until their correct bins are found.

The anti-parallel vector to  $B_{EAS}$  is also determined and binned, resulting in two selected elevation bins. To validate the simulated binning algorithm, the time series for  $B_{EAS}$  can be plotted against the selected bin centers and upper and lower bounds.

If head-selection has been performed correctly, then the  $B_{EAS}$  elevation component should never exceed  $\pm 45^\circ$ . The results of this approach to validation were positive and are presented in Section 3.2.

The most direct way to validate the algorithm is to compare the simulated bins and heads selected to the actual bins and heads selected for a given dataset. EAS level 1 Burst Mode CDF data files on SOAR provide this information. The variable “SWA\_EAS\_EasUsed” contains a time series indicating the head selected (0 for EAS1, 1 for EAS2). “SWA\_EAS\_ElevationUsed” contains a 2D time series with the bin indices (0-15) of the elevation bins selected for parallel and anti-parallel  $B_{EAS}$ . “SWA\_EAS\_ELEVATION”, “SWA\_EAS\_ELEVATION\_delta\_lower”, and “SWA\_EAS\_ELEVATION\_delta\_upper” contain 2D time series with the actual bin centers and boundaries used for the selected elevation bins (in degrees), for parallel and anti-parallel  $B_{EAS}$ . In contrast, while there are similarly-named variables for azimuth (“SWA\_EAS\_AZIMUTH”, “SWA\_EAS\_AZIMUTH\_delta\_lower”, and “SWA\_EAS\_AZIMUTH\_delta\_upper”), these do not contain time series. Instead, the azimuth variables contain the constant bin center and delta data found in Table A.1. Therefore, simulated azimuth binning cannot be verified this way. This is of no consequence to Burst Mode simulation because steering is only determined by elevation binning.

The results of this approach to validation led to the discovery of an error in the memory of the SWA Data Processing Unit (SWA-DPU). These results are also presented in Section 3.2.

## 2.6 Visualisation for Bins and Vectors

To facilitate the determination of Burst Mode data accuracy, a visualisation tool was developed for this project based on a plot produced by Owen et al (2021)[5]. This was the tool that was used to create Figures 1.6, 1.7, and 1.8 in Section 1.4. This visualisation is attractive because it concisely displays magnetic field vectors in

SRF, circumventing the need to change coordinates following a sensor head change. The choice of SRF over RTN has been justified in Section 2.3. This approach to visualisation also allows the difference in angular bin widths to be seen projected on to the sky, visualising each head and pixel's field of view along with  $B_{EAS}$  vectors in a form that may facilitate a more intuitive determination of the ensuing effects on Burst Mode steering compared to other approaches.

The visualisation was implemented in Python. The first step was to plot the grid of angular EAS bins in SRF space. Plotting the grid is easily (though inefficiently) accomplished in the EAS1/2 frame using spherical coordinates by simply plotting many closely-spaced points in straight lines along the upper and lower bounds of the elevation and azimuth bin tables (see Appendix A). This step is described in detail with added excerpts of code from its implementation in Appendix B. Plotting the parallel and anti-parallel magnetic field vectors seen in Figures 1.7 and 1.8 was also relatively straightforward. Since both  $B_{EAS}$  and  $B_{MAG}$  are acquired from SOAR in SRF coordinates, they only had to be transformed to spherical coordinates and plotted with distinct markers to differentiate them. The markers used throughout this report are a diamond for the parallel vector and a square for the anti-parallel vector.  $B_{EAS}$  and  $B_{MAG}$  vectors are also differentiated by colour (green and purple respectively).

To plot the contours of constant pitch angle at arbitrary pitch angles around these vectors involved adapting an algorithm written by Dr. Chris Owen in Interactive Data Language (IDL). The algorithm starts by constructing the transformation matrix  $M$  from a cartesian reference frame whose z-axis is aligned with the magnetic field vector to cartesian SRF coordinates.  $M$  is constructed using the elevation and azimuth for a magnetic field vector  $B$  in spherical SRF coordinates as per Equation 2.1, where  $\theta = (\text{spherical SRF magnetic field elevation}) * \pi / 180^\circ$  and  $\phi = (\text{spherical SRF magnetic field azimuth}) * \pi / 180^\circ$ .

$$M = \begin{bmatrix} \cos \phi \cos \theta & -\sin \phi & -\cos \phi \sin \theta \\ \sin \phi \cos \theta & \cos \phi & -\sin \phi \sin \theta \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.1)$$

From there, the proceeding algorithm can be summarised as follows:

1. For a given contour with pitch angle  $\alpha$ , calculate the angular length (conceptually similar to “circumference”) of the contour in spherical SRF coordinates. Multiply this by the angular point density to yield a number of phase angles  $\varphi$  around  $B$  with which to plot points along the contour.  $\varphi$  is analogous to azimuth in the  $B$ -aligned frame.
2. Working in the  $B$ -aligned frame, for every phase angle  $\varphi$ , construct the cartesian vector  $\vec{p}_B$  according to:

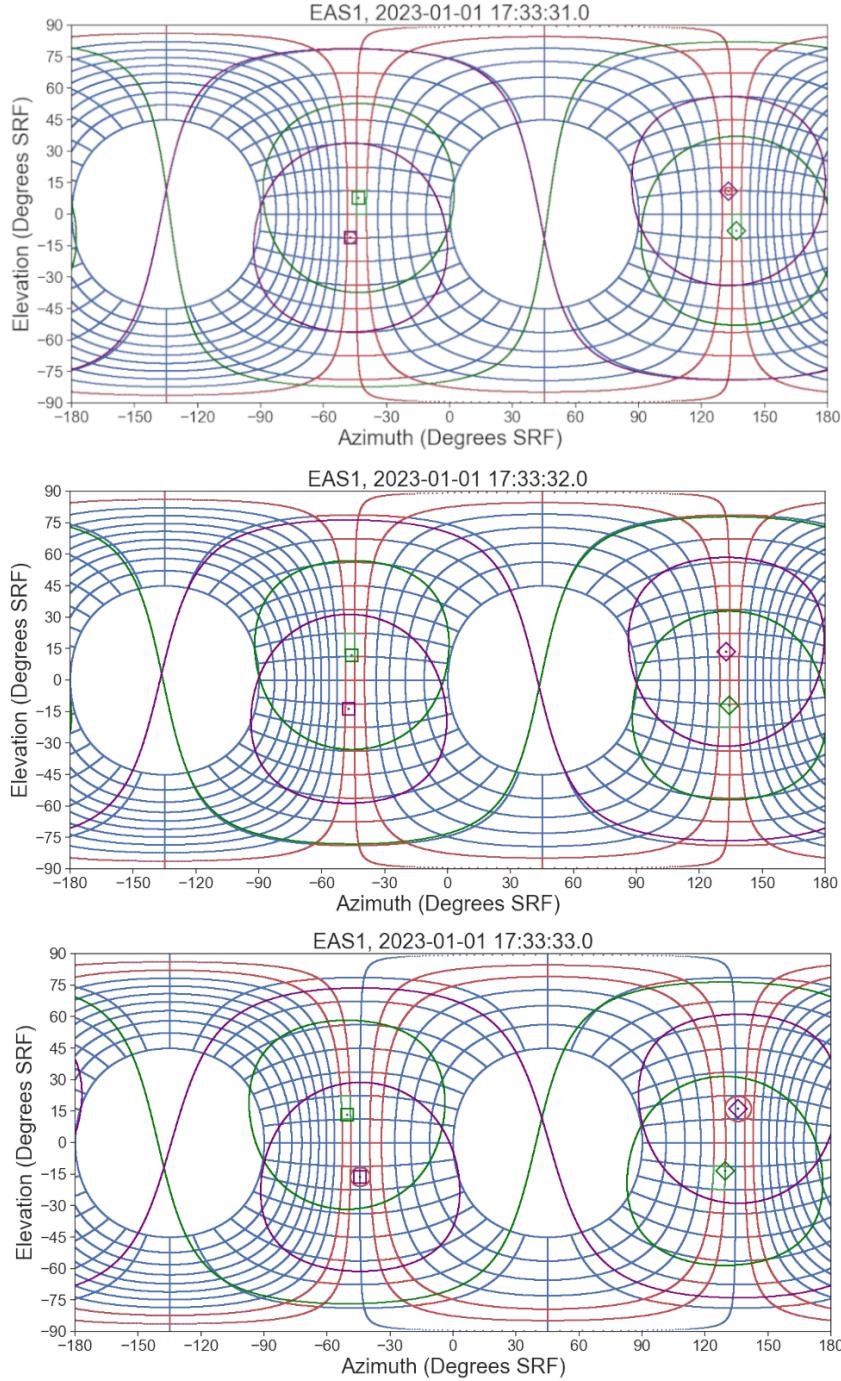
$$\vec{p}_B = \begin{bmatrix} x_{\alpha,\varphi} \\ y_{\alpha,\varphi} \\ z_{\alpha,\varphi} \end{bmatrix} = \begin{bmatrix} \cos \alpha \\ \sin \alpha \cos \varphi \\ \sin \alpha \sin \varphi \end{bmatrix} \quad (2.2)$$

3. Transform  $\vec{p}_B$  into  $\vec{p}_{SRF}$  in the cartesian SRF frame using  $M$  according to:

$$\vec{p}_{SRF} = M \vec{p}_B \quad (2.3)$$

4. Transform  $\vec{p}_{SRF}$  to spherical SRF coordinates, and its elevation and azimuth angles can be plotted directly.

Following this algorithm to plot all vectors for all phase angles in the range  $[0^\circ, 360^\circ]$  with appropriate increments and all pitch angles in the range  $[0^\circ, 180^\circ]$  in  $15^\circ$  increments yields Figure 1.7 when plotted alongside EAS bins and the  $B_{EAS}$  vector itself for the appropriate date and time.



**Figure 2.2:** Three frames of an animation produced by plotting  $B_{EAS}$  and  $B_{MAG}$  data from 1st January 2023 along with pitch angle contours including loss contours, which are described in Section 3.3. Green represents  $B_{EAS}$  and purple represents  $B_{MAG}$ . Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Top panel: Data from 17:33:31 UTC. Middle panel: Data from 17:33:32 UTC. Bottom panel: Data from 17:33:33 UTC.

Image Source: Author's own work

As SOAR’s magnetic field vector data comes in time series, this tool can also be used to visualise magnetic field variation over time in the form of an animation. Individual frames of an animation can be generated for every data point in a magnetic field time series, and then combined into a high-definition *.gif* using the *imageio* library for Python[36]<sup>1</sup>. To play the animation at a realistic frame rate, the duration of each frame must be specified. For Burst Mode data, the duration is set to 0.125s, matching the time delay between successive time series vectors. Using data from 1st January 2023, three (non-consecutive) frames of such an animation, 1s apart, are included in Figure 2.2 for example.

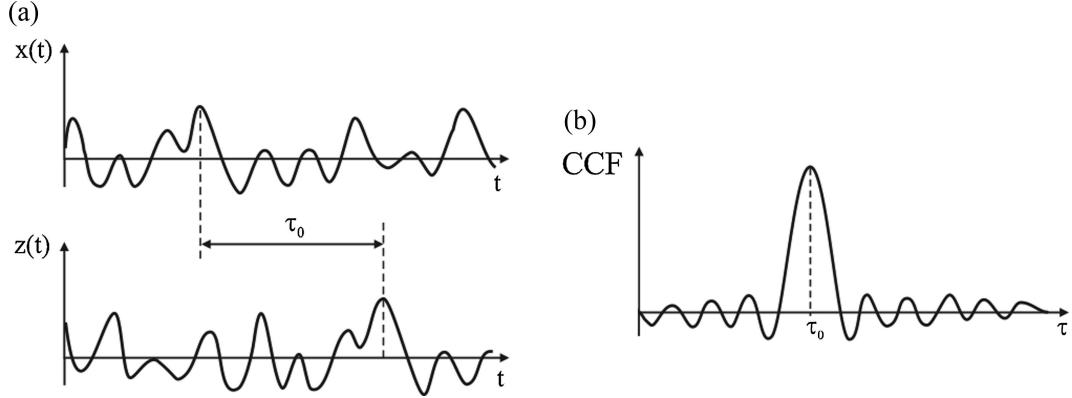
## 2.7 S20 Link Latency

It is possible that some of the  $B_{EAS}$ - $B_{MAG}$  discrepancy can be attributed to latency over the S20 inter-instrument data link[5]. This latency may be particularly noticeable on 1/8th second timeframe during periods of heavy data traffic[5]. If noticeable, then the latency is expected to cause a time delay between  $B_{EAS}$  and  $B_{MAG}$  where  $B_{MAG}$  leads by at least  $\sim 0.125$ s. One approach to measuring this time delay is to perform a cross-correlation analysis between  $B_{EAS}$  and  $B_{MAG}$ [37]); in principle, a cross-correlation function (CCF) takes a domain of time delays as input and the delay that gives the strongest correlation between the two time series can be selected to represent the latency. The principle behind this approach is well-illustrated by Figure 2.3 by Hanus et al[37] for two 1D time series  $x(t)$  and  $z(t)$  and latency  $\tau_0$ .

A version of the cross-correlation approach was implemented in Python to calculate the S20 latency  $\tau_{S20}$ . The *scipy* library for Python provides a wide range of signal processing methods and functions, including CCFs[38]. Given two 1D arrays of data, the discrete CCF between them can be determined using the *scipy.signal.correlate()* method[38]. The discrete CCF generated by this method is an approximation of a

---

<sup>1</sup>Other methods for composing *.gif* files are available and may be more straightforward. One of the most straightforward methods is to use free online “image-to-gif” converters, but the resolution of *.gif* files acquired by this method was found to be too low for this project’s purposes.



**Figure 2.3:** Two figures illustrating CCF time delay determination. (a): Two arbitrary time series  $x(t)$  and  $z(t)$ , which are well-approximated as identical time series with a time delay  $\tau_0$  between them. (b): A plot of the CCF applied to  $x(t)$  and  $z(t)$ , showing maximum correlation at the time delay  $\tau_0$ , representing the latency.

Image Source: Hanus et al[37]

theoretical CCF between two complex, 1D functions  $f(t)$  and  $g(t)$  over time  $t$ . The CCF, sometimes denoted “ $f \star g$ ”, is generated by convolving  $g(t)$  with the complex conjugate of  $f(t)$ ,  $\bar{f}(t)$  according to:

$$f \star g = \bar{f}(-t) * g(t) \quad (2.4)$$

where convolution is denoted by “ $*$ ”[39][40]. Convolution is defined by:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.5)$$

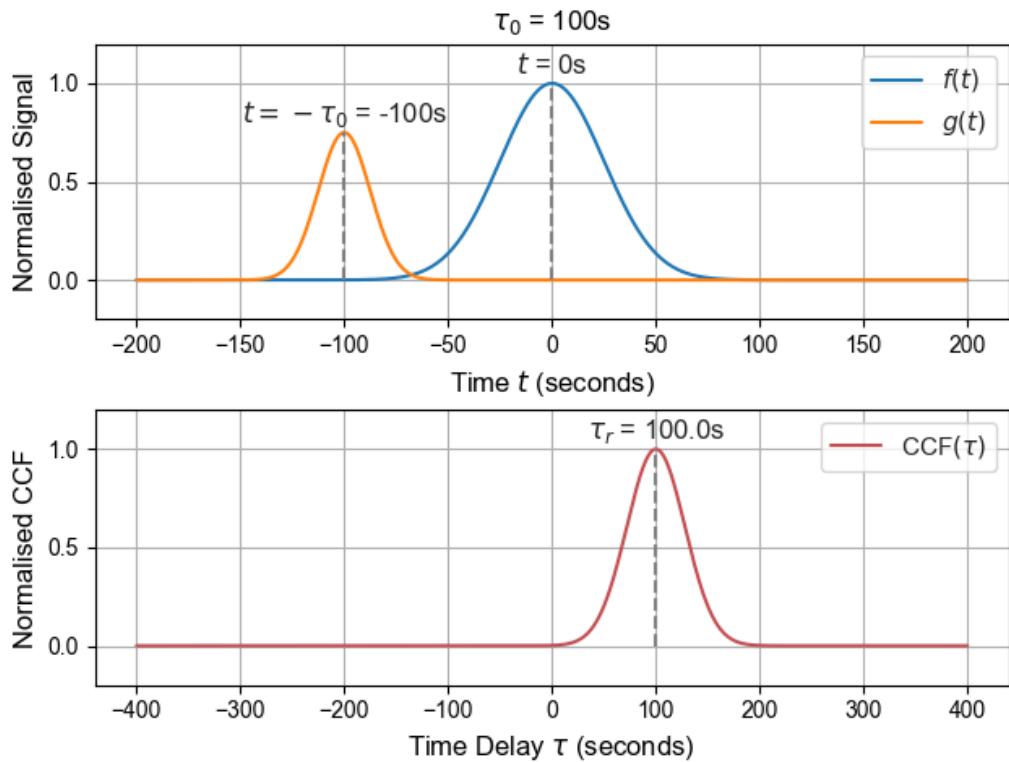
where  $\tau$  denotes the continuously-variable displacement in time between  $f(t)$  and  $g(t)$ [40], also known as the *lag*. Combining Equations 2.4 and 2.5 and changing integration variables from  $t$  to  $\tau$  yields Equation 2.6 for the cross-correlation in terms of  $\tau$ :

$$(f \star g)(\tau) = \int_{-\infty}^{\infty} \bar{f}(t)g(t + \tau)dt \quad (2.6)$$

Equation 2.6 can be expressed in its discrete form in terms of discrete functions  $f(t)$  and  $g(t)$ , yielding:

$$\text{CCF} \equiv (f \star g)(\tau) = \sum_{-\infty}^{\infty} \bar{f}(t)g(t + \tau) \quad (2.7)$$

This is the definition of “CCF” that will be used for the remainder of this section[38][41]. The translation of Equation 2.7 into Python is given by `scipy.signal.correlate()`. The array of lags  $\tau$  is generated by `scipy.signal.correlation_lags()`, which depends on the sample rate of  $f(t)$  and  $g(t)$  to define the timescale of  $\tau$ . For this project, that is equivalent to the EAS Burst Mode sample rate of 8Hz. With the CCF and the array of lags, the latency  $\tau_0$  can be estimated by calculating the maximum of the CCF and finding its corresponding lag.



**Figure 2.4:** Two figures illustrating the time delay algorithm for dummy data. Top panel: Normalised dummy time signals  $f(t)$  and  $g(t)$  plotted over 400s with a time delay  $\tau_0$  between them. The peaks of  $f(t)$  and  $g(t)$  are labelled with their time values. Bottom panel: The CCF of  $f(t)$  and  $g(t)$  plotted over 800s of time delays  $\tau$ . The peak of the CCF is labelled with its value for the time delay  $\tau_r \approx \tau_0$ .

Image Source: Author's own work

This time delay algorithm was implemented and validated using dummy time

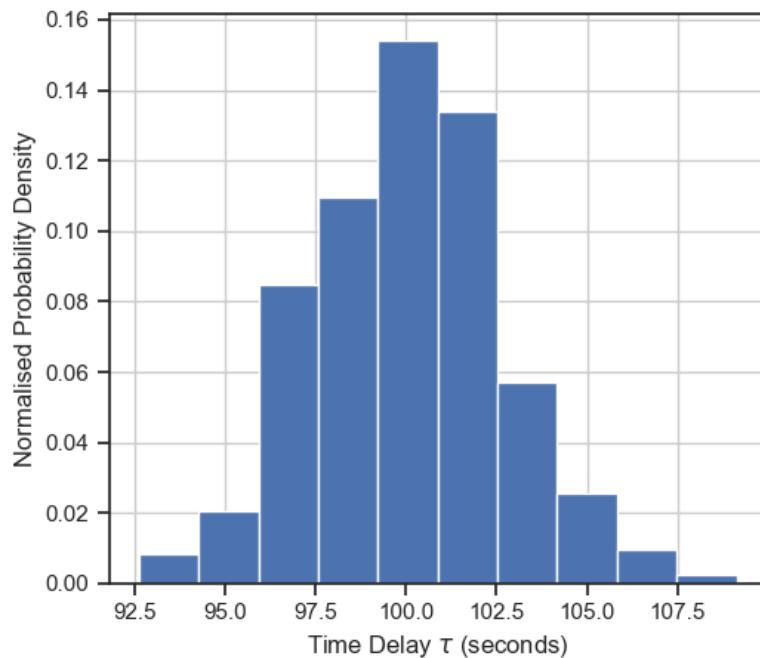
series data consisting of two Gaussian signals,  $f(t)$  and  $g(t)$  as shown in the upper half of Figure 2.4.  $f(t)$  has amplitude  $a_f = 1$  and standard deviation  $\sigma_f = 25\text{s}$ , and  $g(t)$  has  $a_g = 0.75$  and  $\sigma_g = 12.5\text{s}$ .  $f(t)$  and  $g(t)$  are distributed over 400s with  $N = 3200$  data points to simulate the Burst Mode sample rate of 8Hz. There is a delay  $\tau_0 = 100\text{s}$  between  $f(t)$  and  $g(t)$ , with  $f(t)$  in the lead. In the lower half of Figure 2.4, the calculated and normalised CCF is plotted against the array of lags over twice the duration of  $f(t)$  and  $g(t)$ , showing a clear Gaussian peak at the value of  $\tau_0$  that it recovered from the comparison of  $f(t)$  and  $g(t)$ , which is denoted  $\tau_r$ . Figure 2.4 shows reassuring results, correctly recovering a value of  $\tau_r = 100.000\text{s} = \tau_0$ .

The issue now is to determine the uncertainty in the calculated value. There is not much consensus on the “correct” approach to measuring the uncertainty CCF lags[42][43]. Peterson et al (1998) suggest Monte Carlo simulations of many small variations on the same cross-correlation to evaluate the sensitivity of the lag solution to individual data points. Specifically, they suggest a statistical test where the time series subject to the CCF is reduced to a large number of randomly selected subsets of the original dataset, each of which is shrunken by a factor  $\sim 1/e \approx 0.37$ . These subsets are each subjected to the CCF in the same way as before, allowing the calculation of a mean and standard deviation which provide a conservative estimate of the uncertainty. The principle behind this method inspired the implementation of a Python algorithm for calculating the uncertainty on the lag. The algorithm can be summarised as follows for arrays to be correlated,  $f$  and  $g$ :

1. Define two empty arrays  $f_i$  and  $g_i$ . These will be the subset arrays.
2. Starting from 0, take a random integer “step” through the indices of the arrays  $f$  and  $g$  and select their corresponding values to append to  $f_i$  and  $g_i$  respectively. The average length of the step should be  $\approx e$  to get the  $\sim 1/e$  shrink factor (e.g. the first values might be randomly distributed around  $f(3)$  and  $g(3)$ ).
3. Repeat step 2 until the end of  $f$  and  $g$  is reached, and then cross-correlate  $f_i$

and  $g_i$  to calculate  $\tau_r$ , dividing the sample rate by the average step length to reflect the subset's new data rate.

4. Repeat steps 1 through 3 over  $M$  iterations, and calculate the mean time delay  $\tau_r$  and standard deviation  $\sigma$  of the resulting collection of  $\tau_r$  values.  $\sigma$  is said to represent the uncertainty in  $\tau_r$ .



**Figure 2.5:** A histogram of the probability density function describing the  $M = 1000$  simulations used to calculate the uncertainty in  $\tau_r$  for the dummy data used in Figure 2.4. There are 10 equally-spaced time delay bins.

Image Source: Author's own work

The algorithm was validated using the same dummy data shown in Figure 2.4. In this case, the average step length was set to 3, and the random steps were calculated using a uniform random variable in the inclusive range (1, 5). After  $M = 1000$  iterations, a value of  $\tau_r = 100.0\text{s} \pm 2.6\text{s}$  was acquired. A probability density histogram resulting from the Monte Carlo simulation is shown in Figure 2.5. While an uncertainty of  $\pm 2.6\text{s}$  is too small to be displayed on a plot of the same scale as Figure 2.4, it is significantly larger than the time resolution of 8Hz magnetic field data. Given that the

dummy data used in this exercise are, by design, much “cleaner” than real magnetic field time series, this suggests a restrictive lower bound on the resolution of S20 data link latency determination using this method. The results of further experiments using real magnetic field data are presented in Section 3.4.

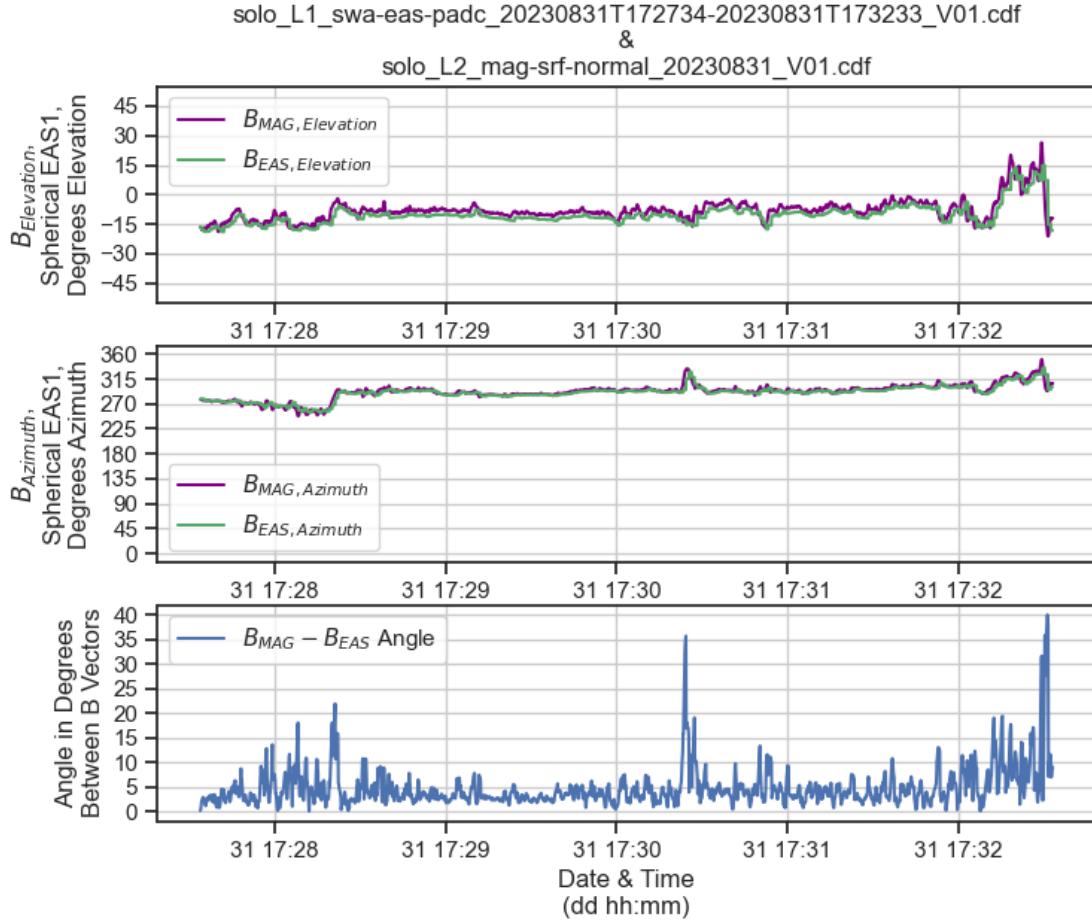
## Chapter 3

# Results and Discussion

### 3.1 Preliminary $B_{EAS}$ - $B_{MAG}$ Comparison: Results

The angle between  $B_{EAS}$  and  $B_{MAG}$  vectors, when plotted alongside those vectors for randomly chosen sample data from 31st August 2023, yields Figure 3.1.

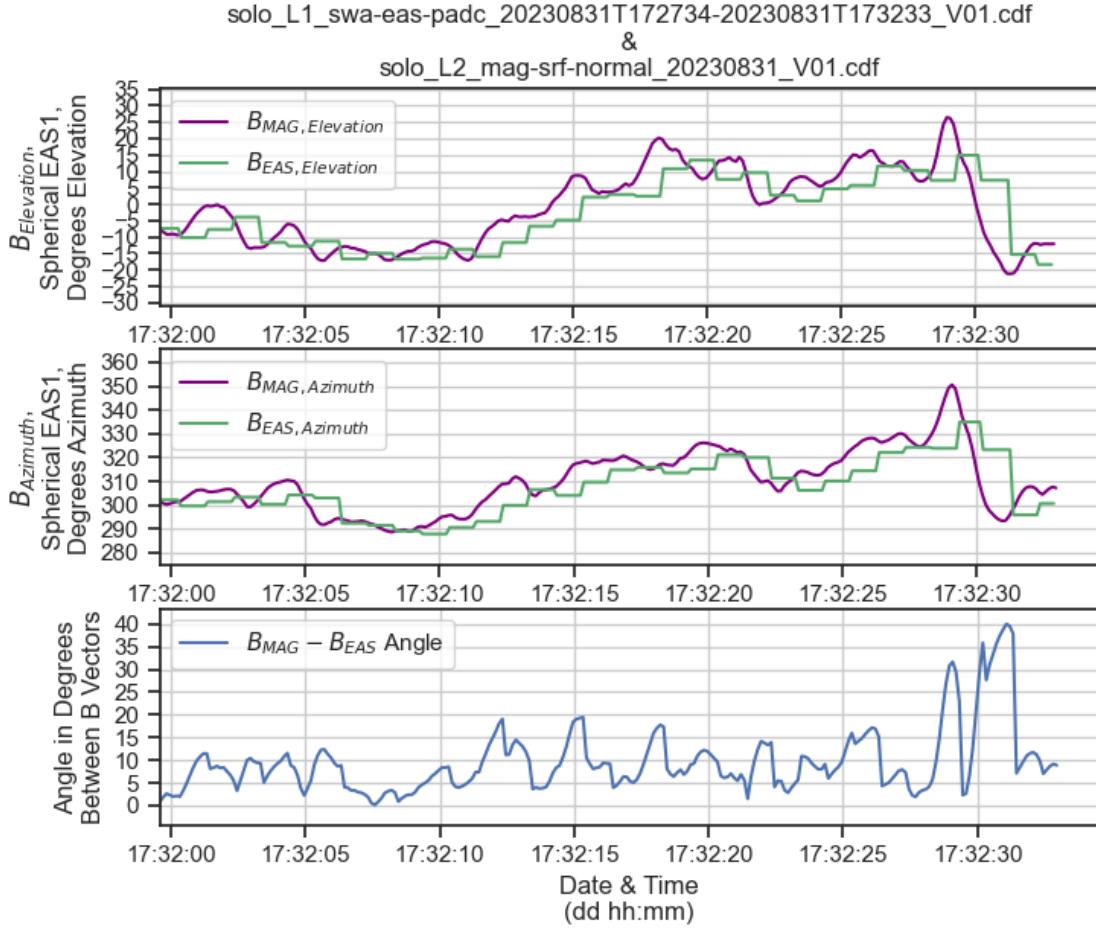
By inspection, while relatively good agreement can be seen between  $B_{EAS}$  and  $B_{MAG}$  over long timescales, there are clear discrepancies over short timescales as shown in Figure 3.2. An unexpected observation is that  $B_{EAS}$  has less time resolution than  $B_{MAG}$ , appearing only to update at 1Hz. Inspection of the raw “SWA\_EAS\_MagDataUsed” data time series data reveals that while this data array *is* published to SOAR with 8 vectors for every second of “EPOCH” as advertised, these vectors are only updated once per second, resulting in 8 identical vectors per second such that the effective cadence is 1Hz. Since these data are packaged in L1 .cdf files, they are expected to represent the true MAG data used to steer EAS, which implies that the alleged 1/8th second resolution of EAS Burst Mode is compromised by a magnetic field vector only updates at 1Hz. This will be a particular source of error during periods of fast-changing magnetic field. This effective 1Hz effect has been observed in other periods of Burst Mode from the sample including 5th November 2023 and 1st January 2023. It is absent from the relatively quiet period on 30th May 2023. Further research is required to determine the cause for this unexpectedly low update cadence, and how widespread of an issue this is in the SOAR dataset.



**Figure 3.1:** Example  $B_{EAS}$  data from a 5 minute period of EAS Burst Mode on 31st August 2023. Top panel: Elevation for  $B_{EAS}$  and  $B_{MAG}$  in spherical EAS1. Middle panel: Azimuth for  $B_{EAS}$  and  $B_{MAG}$  in spherical EAS1. Bottom panel: Angular difference between  $B_{EAS}$  and  $B_{MAG}$ .

Image Source: Author's own work

The highest angular offset in Figure 3.1 is observed to exceed  $40^\circ$ , and it exceeds  $70^\circ$  in a different example from 5th November 2023. Otherwise, the offset is between  $0^\circ$  and  $10^\circ$  for  $\sim 82\%$  of the period in Figure 3.1. In comparison, the offset varies between  $3^\circ$  and  $5^\circ$  for  $\sim 80\%$  of the period in a different example from 30th May 2023, and between  $6^\circ$  and  $8^\circ$  for the remaining period. The bin centers and bin widths for elevation and azimuth are tabulated with level 2 EAS data files. While azimuth bins are identical for both heads, elevation bins are different due to minor differences in the assembly of each EAS head's electron optics. These tables have been updated

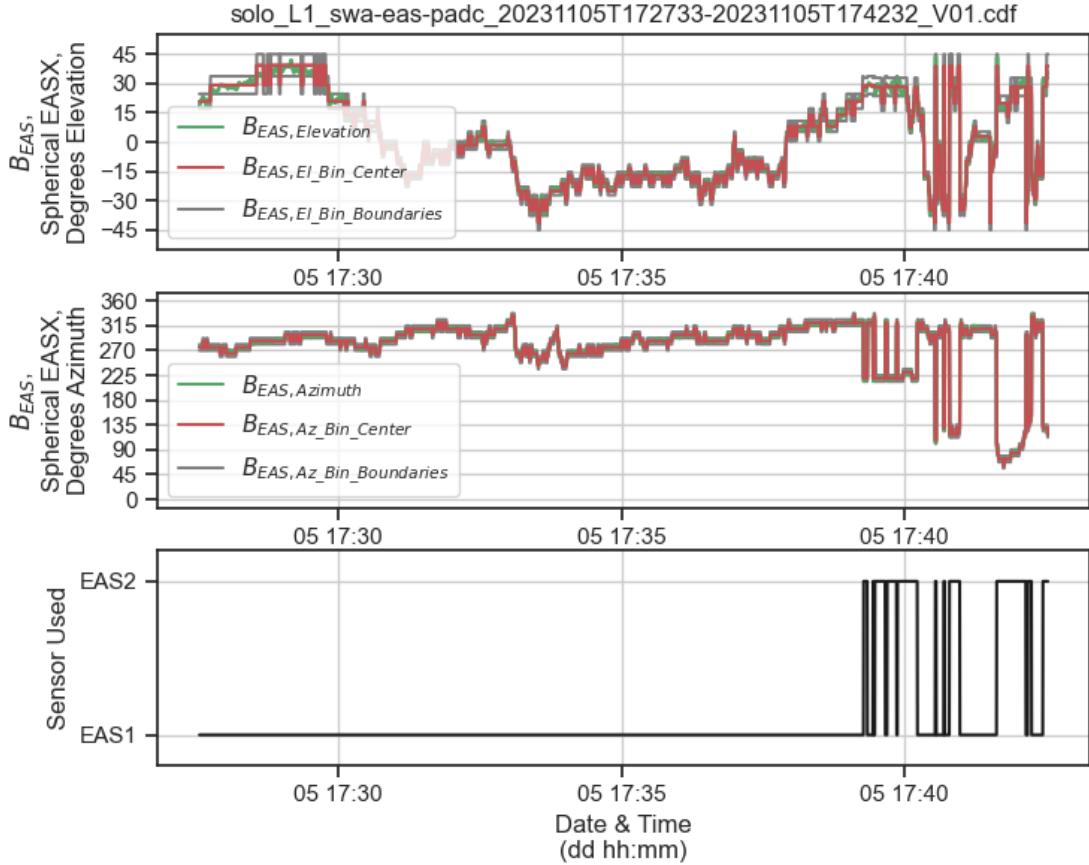


**Figure 3.2:** A detailed look at the last  $\sim 5$ s of Figure 3.1. Top panel: Elevation for  $B_{EAS}$  and  $B_{MAG}$  in spherical EAS1. Middle panel: Azimuth for  $B_{EAS}$  and  $B_{MAG}$  in spherical EAS1. Bottom panel: Angular difference between  $B_{EAS}$  and  $B_{MAG}$ .

Image Source: Author's own work

over time but remained constant from 17th July 2021 until January 2024, when they were reverted to a previous version due to the error described in Section 3.2. The tables from that 2021-2024 period are discussed here and are included in Appendix A. The widest angular bin in EAS1 has width  $\pm 6.06^\circ$ . This is well below much of the angular offset observed in the sample, therefore, it is established that  $B_{EAS}$  could be sufficiently offset from  $B_{MAG}$  to cause erroneous Burst Mode steering, justifying the development of a metric for PAD quality/completeness.

## 3.2 SWA-DPU Binning Algorithm

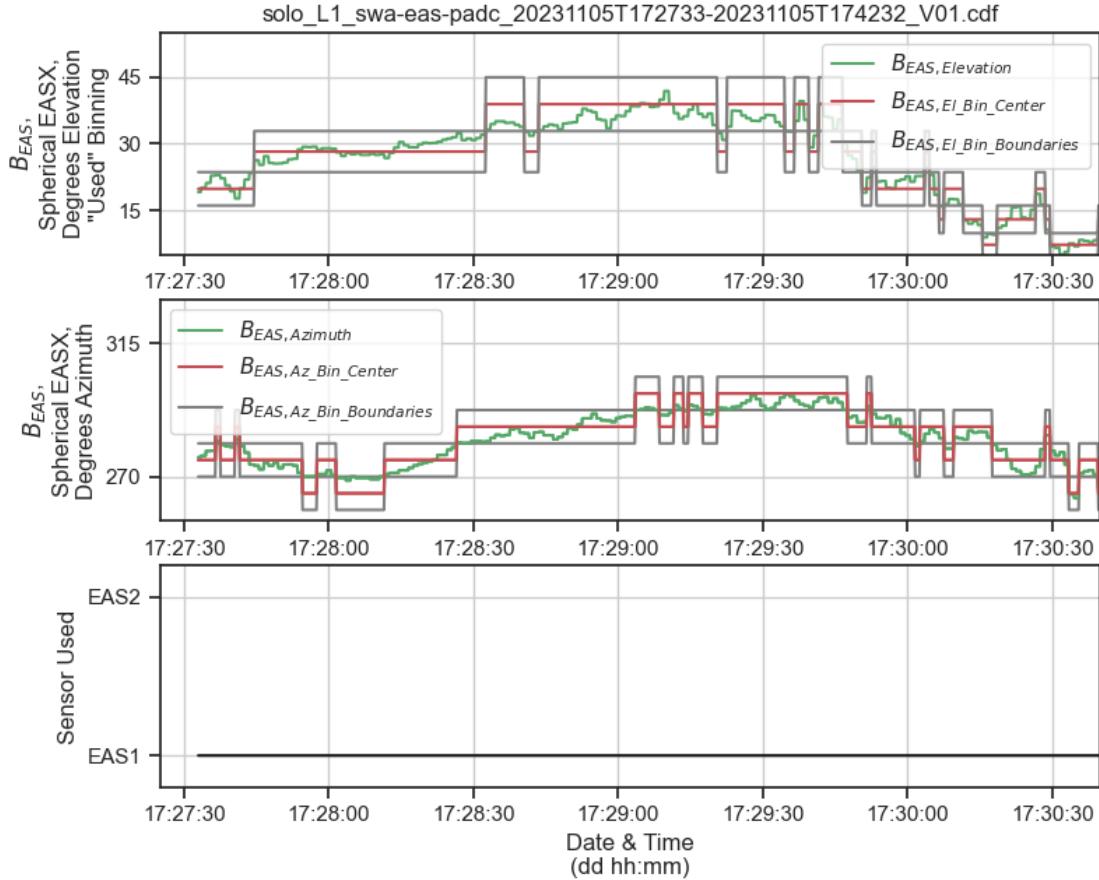


**Figure 3.3:** Example  $B_{EAS}$  data from a 15-minute period of EAS Burst Mode on 5th November 2023 showing a simulation of the Burst Mode steering algorithm. The reference frame for the top two panels is labelled “EASX” to indicate that it switches between EAS1 and EAS2 over the 15 minute period depending on  $B_{EAS}$  orientation and resulting head selection. Top panel: Elevation for  $B_{EAS}$  in spherical EASX coordinates, along with the centers and upper/lower bounds of the selected elevation bins (“El\_Bin\_Center” and “El\_Bin\_Boundaries” respectively). Middle panel: Azimuth for  $B_{EAS}$  in spherical EASX coordinates, along with the centers and upper/lower bounds of the selected azimuth bins (“Az\_Bin\_Center” and “Az\_Bin\_Boundaries” respectively). Bottom panel: The selected head over time.

Image Source: Author’s own work

In the interest of validating the simulated binning algorithm described in Section 2.5, the simulated binning for parallel  $B_{EAS}$  vectors is plotted in Figures 3.3 and 3.4 along with a plot of the chosen head over time, for a 15-minute period on 5th November 2023. Figure 3.3 shows that the elevation and azimuth values “jump” as

the reference frame switches from EAS1 to EAS2 towards the end of the time series. As expected, this switch can be triggered by the  $B_{EAS}$  elevation exceeding  $\pm 45^\circ$  in its previous head's reference frame. Figure 3.4 provides a detailed view of the algorithm's selection of appropriate binning such that the  $B_{EAS}$  trace stays between bin boundaries.



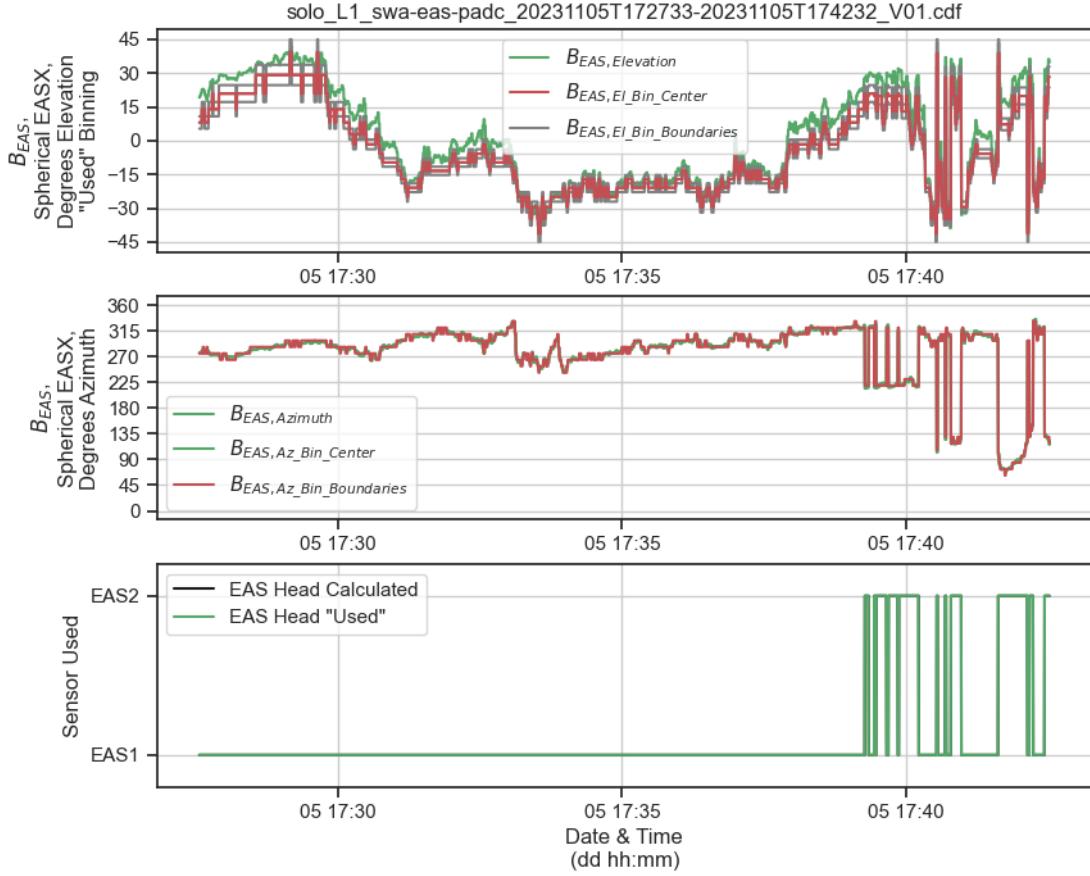
**Figure 3.4:** A detailed view of the first  $\sim 2.5$  minutes of Figure 3.3. Top panel: Elevation + binning for  $B_{EAS}$  in spherical EASX coordinates. Middle panel: Azimuth + binning for  $B_{EAS}$  in spherical EASX coordinates. Bottom panel: The selected head over time.

Image Source: Author's own work

In the interest of validating the binning algorithm even further, a more direct comparison was performed using the “Used” bin data directly from SOAR .cdf files. Plotting the “Used” heads and elevation bins in an attempt to recreate Figure 3.3 yields Figure 3.5. While the “SWA\_EAS\_EasUsed” head is perfectly correlated

with the calculated head in the bottom panel of Figure 3.5, the top panel shows that not only does the “SWA\_EAS\_ELEVATION” binning appear to disagree with the simulated binning in the top panel of Figure 3.5, but the “Used” binning also appears to capture the  $B_{EAS}$  elevation inaccurately over long periods of the time series. This is shown more clearly in Figure 3.6. In an attempt to fix this discrepancy, the data binning code was reviewed and multiple variations of the elevation bin tables were applied to the simulated algorithm. A possible bug was considered that might have caused the bin tables for the two heads to be swapped, but swapping them manually did not fix the discrepancy. The possibility that the uploaded data on SOAR had been reverted to a previous version was also considered, and older bin tables available on SOAR from 19th August 2020 to 17th July 2021, and from Solar Orbiter’s commissioning to 19th August 2020 were also applied, but these did not fix the discrepancy either. The discrepancy was brought to the attention of the Principal Investigator of Solar Orbiter SWA (Dr. Chris Owen) and the SWA operations team, and Solar Orbiter was commanded to downlink the binning tables in its memory along with routine housekeeping data. The data downlink revealed that Solar Orbiter had, indeed, been using outdated bin tables in the SWA-DPU onboard, explaining the failure of “SWA\_EAS\_ELEVATION\_delta\_lower” and “SWA\_EAS\_ELEVATION\_delta\_upper” to encapsulate  $B_{EAS}$ .

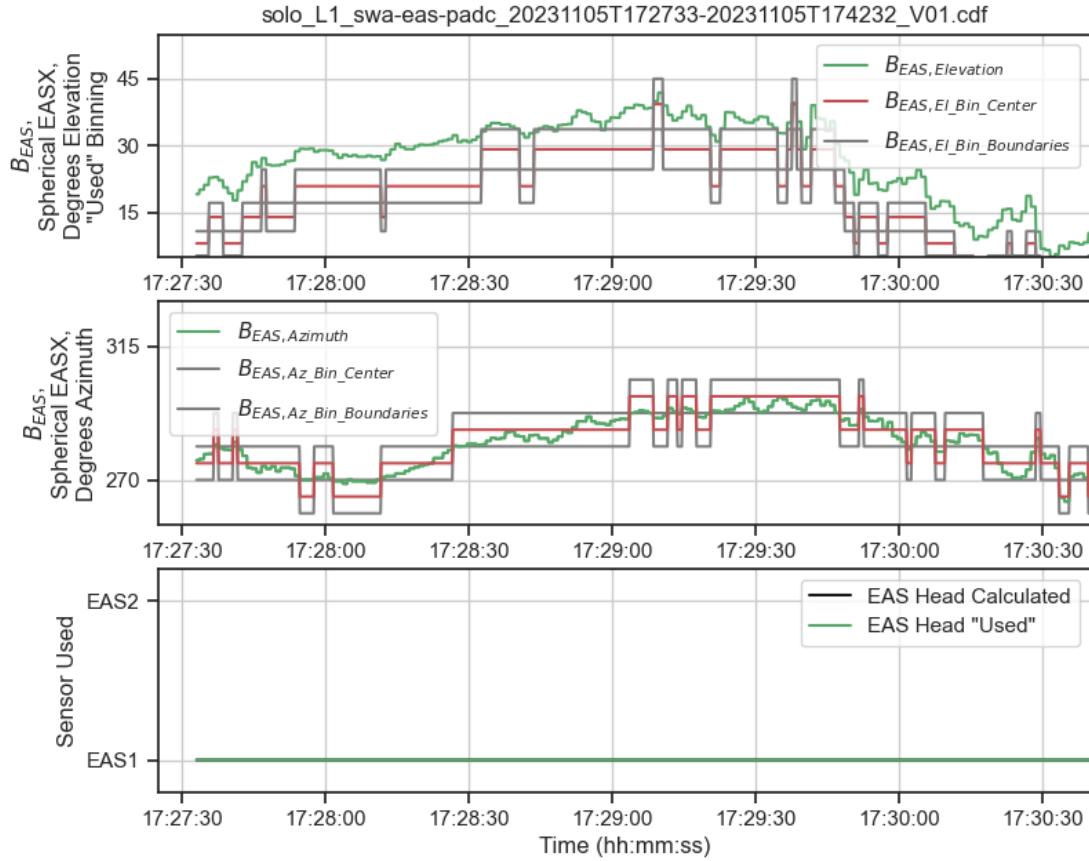
A partial explanation for this issue is that the SWA-DPU uses one bin table to steer the EAS heads to a particular elevation, and a different bin table to separate electron counts into elevation bins for further processing. While the second table may have been correctly updated, the first (“steering”) table had either not been updated or had been unintentionally reverted to an earlier state. Only the binning for the second table is uploaded to SOAR under “SWA\_EAS\_ELEVATION”. The effect that this mismatch can have on elevation binning is illustrated by Figure 3.7, where the vertical position of each yellow circle represents the elevation value associated with an individual electron. In Burst Mode, the elevation of the received  $B_{MAG}$  vector is first binned according to the steering table (Figure 3.7, left). When EAS



**Figure 3.5:** Similar data to Figure 3.3, but plotting the “Used” bin centers, bin bounds, and sensor heads. Top panel: Elevation for  $B_{EAS}$  in spherical EASX coordinates, along with centers and upper/lower bounds of the used elevation bins (“El\_Bin\_Center” and “El\_Bin\_Boundaries” respectively). Middle panel: Azimuth for  $B_{EAS}$  in spherical EASX coordinates, along with the centers and upper/lower bounds of selected azimuth bins (“Az\_Bin\_Center” and “Az\_Bin\_Boundaries” respectively). Bottom panel: The selected, or “Calculated” head and used head over time.

Image Source: Author’s own work

is then steered to the appropriate elevation bin (e.g. “bin 0”), that bin is “filled” with electron detections from various elevations within its elevation boundaries, represented by the bins filled with yellow circles in Figure 3.7, left. In order to label these raw electron detection data with their appropriate elevation angles, SWA-DPU then uses the indices of the “filled” bins (0-15) in the *other* binning table in Figure 3.7, right. While there is some overlap between bins with the same indices in both tables, the tables are mostly mismatched. Therefore, while some electron detections



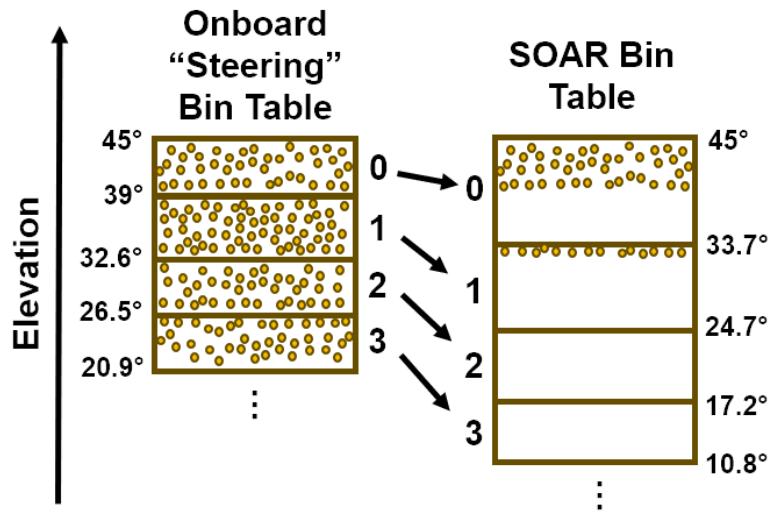
**Figure 3.6:** A detailed view of the first  $\sim 2.5$  minutes of Figure 3.3. Top panel: Elevation + used binning for  $B_{EAS}$  in spherical EASX. Middle panel: Azimuth + binning for  $B_{EAS}$  in spherical EASX. Bottom panel: The selected, or “Calculated” head and used head over time.

Image Source: Author’s own work

are correctly represented by both tables, this can lead the labelled elevation data to incorrectly report that certain elevation ranges are less “filled” with electron detections than reality. This is represented by the bins that are partially filled with electron detections in Figure 3.7, right.

This is the explanation for bin table mismatch which was found only to affect EAS Burst Mode data (which involves steering and therefore a steering table) prior to 2024. However, because of this project, it was later discovered that both the steering table *and* the table uploaded to SOAR had been incorrect since 1st January 2024, compromising both Burst Mode and Normal Mode data. As a result, as of 22nd July

2024, all EAS data since 1st January 2024 have been expunged from SOAR[26].



**Figure 3.7:** A cartoon to illustrate the process by which a discrepancy in the two elevation binning tables used by SWA DPU can lead to incorrect elevation binning. The vertical axis indicates the elevation angle corresponding to various electron detections, which are represented by yellow circles. The first four bins for the two binning tables are represented by rectangles labelled with their indices and the angles of their upper and lower elevation bounds, and vertically scaled according to their elevation widths. The short arrows indicate the correspondence between the indices of the mismatched binning tables.

Image Source: Author's own work

### 3.3 Completeness Algorithm

Research into the steering algorithm and data visualisation completed earlier in this project made it clear that there is a relatively fast approach to generating a “completeness” or “quality” score for Burst Mode PADs, circumventing the relatively expensive process of re-binning EAS Burst Mode VDF data to generate the PADs in the first place, and then assessing PAD completeness “by hand” (e.g. by counting the white bins in Figure 1.5).

As previously described, while discrepancies between  $B_{EAS}$  and  $B_{MAG}$  are expected to cause some inaccuracies in Burst Mode PADs, the resulting PADs will only be left *incomplete* if these discrepancies disrupt Burst Mode steering. That is, under otherwise-ideal conditions, in order to lose part of a  $[0^\circ - 180^\circ]$  distribution, not only must  $B_{EAS}$  lie within a different angular bin than  $B_{MAG}$ , but, more specifically, it must lie within a different *elevation* bin than the actual bin(s) selected. Differences in azimuthal bin selection would lead the pitch angles recovered from the azimuthal bins in each selected elevation band (e.g. as shown in Figure 1.8) to be labelled incorrectly, but this can be corrected in post-processing as long as  $B_{MAG}$  is known. No amount of re-labelling can recover a pitch angle that would have fallen within an elevation bin that was not sampled. The exact amount of pitch angle “loss” therefore only depends on the elevation angles of  $B_{MAG}$  and the selected elevation bins. Consider the vector parallel to  $B_{EAS}$ , whose selected elevation bin is meant to capture a range of pitch angles from  $0^\circ$  (for the pixel containing the vector) to  $\geq 90^\circ$  (for the pixel in the same elevation bin with  $+180^\circ$  azimuth compared to the vector). If  $B_{MAG}$  lies outside of this elevation bin, then, after re-labelling pitch angle data, the smallest pitch angle that can be recovered is theoretically equal to the smallest difference in elevation between  $B_{MAG}$  and each of the two boundaries (upper and lower) of the selected parallel elevation bin. Any electron with a pitch angle smaller than this difference is travelling too close to parallel with  $B_{MAG}$  to be captured by the selected elevation bin and therefore goes undetected. More formally, let  $B_{EAS,\parallel\parallel}$ ,  $B_{EAS,\parallel\|}$ ,  $B_{MAG,\parallel\parallel}$ , and  $B_{MAG,\parallel\|}$  represent the unit vectors parallel and anti-

parallel to  $B_{EAS}$  and  $B_{MAG}$  respectively. Working in the selected EAS head frame, the pitch angle loss  $\Delta_{\uparrow\uparrow}$  associated with  $B_{EAS,\uparrow\uparrow}$  is given by:

$$\Delta_{\uparrow\uparrow} = \min\{|\theta_{MAG,\uparrow\uparrow} - \theta_{upper,\uparrow\uparrow}|, |\theta_{MAG,\uparrow\uparrow} - \theta_{lower,\uparrow\uparrow}|\} \quad (3.1)$$

where  $\theta_{MAG,\uparrow\uparrow}$  is the elevation of  $B_{MAG,\uparrow\uparrow}$  and  $\theta_{upper,\uparrow\uparrow}$  and  $\theta_{lower,\uparrow\uparrow}$  are the elevations of the upper and lower boundaries of the selected parallel elevation bin respectively.  $\Delta_{\uparrow\uparrow}$  is equivalent to the minimum theoretical pitch angle  $\alpha_{min}$  that can be recovered from a Burst Mode PAD, yielding the identity:

$$\alpha_{min} = \Delta_{\uparrow\uparrow} \quad (3.2)$$

Similarly, the pitch angle loss  $\Delta_{\downarrow\downarrow}$  associated with  $B_{EAS,\downarrow\downarrow}$  is given by:

$$\Delta_{\downarrow\downarrow} = \min\{|\theta_{MAG,\downarrow\downarrow} - \theta_{upper,\downarrow\downarrow}|, |\theta_{MAG,\downarrow\downarrow} - \theta_{lower,\downarrow\downarrow}|\} \quad (3.3)$$

where  $\theta_{MAG,\downarrow\downarrow}$  is the elevation of  $B_{MAG,\downarrow\downarrow}$  and  $\theta_{upper,\downarrow\downarrow}$  and  $\theta_{lower,\downarrow\downarrow}$  are the elevations of the upper and lower boundaries of the selected anti-parallel elevation bin respectively. As before,  $\Delta_{\downarrow\downarrow}$  implies that any electrons with a pitch angle larger than a maximum theoretical pitch angle  $\alpha_{max}$  cannot be detected. Therefore, if all angles are expressed in degrees, then the maximum theoretical pitch angle that can be recovered from a Burst Mode PAD is given by:

$$\alpha_{max} = 180^\circ - \Delta_{\downarrow\downarrow} \quad (3.4)$$

Notably, the fact that pitch angle loss occurs first at an  $\alpha_{min}$  and  $\alpha_{max}$  at either end of a full PAD before “spreading” towards the center of the distribution may imply that EAS Burst Mode is particularly ineffective at capturing narrow electron strahl populations in the solar wind, which are aligned with  $B_{MAG,\uparrow\uparrow}$  or  $B_{MAG,\downarrow\downarrow}$  by definition[44].

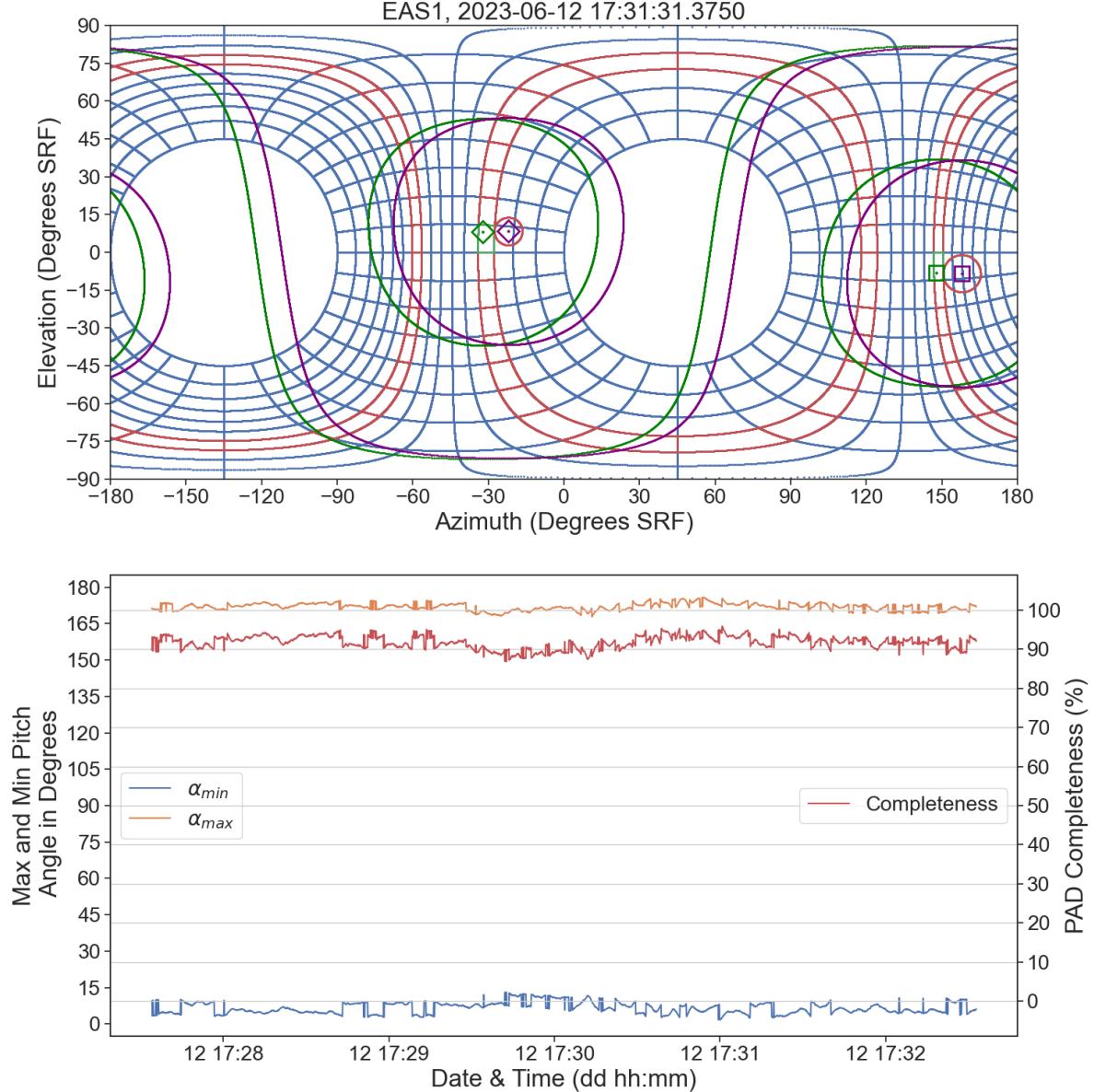
In general, it should *not* be assumed that  $\Delta_{\uparrow\uparrow} = \Delta_{\downarrow\downarrow}$  because the specific elevation bins associated with  $B_{EAS,\uparrow\uparrow}$  and  $B_{EAS,\downarrow\downarrow}$  are calculated independently of each

other and, in general, the elevation bins for both sensor heads are of non-uniform width[4][5]. In other words, a loss of  $\Delta_{\uparrow\uparrow}$  with respect to the parallel bin does *not* imply a symmetric and equal loss  $\Delta_{\uparrow\downarrow}$  with respect to the anti-parallel bin. This asymmetry can be captured using the visualisation tool described in Section 2.6. Using pre-existing code for plotting pitch angle contours,  $\alpha_{min}$  and  $\alpha_{max}$  can be treated as any other pitch angle and visualised in the same way as the pitch angles in Figures 1.7 and Figure 1.8. When  $\alpha_{min}$  and  $\alpha_{max}$  are visualised this way, the sizes (or “radii”) of their contours around  $B_{MAG,\uparrow\uparrow}$  and  $B_{MAG,\uparrow\downarrow}$  on the plot represent the sizes of the losses  $\Delta_{\uparrow\uparrow}$  and  $\Delta_{\uparrow\downarrow}$  respectively. Using the period of Burst Mode data from 12th June 2023 as an example,  $\alpha_{min}$  and  $\alpha_{max}$  are presented in Figure 3.8 as pitch angle contours plotted in red around  $B_{EAS,\uparrow\uparrow}$  (the green diamond) and  $B_{EAS,\uparrow\downarrow}$  (the green square) respectively. In this example,  $\alpha_{min}$  and  $\alpha_{max}$  (and, by extension,  $\Delta_{\uparrow\uparrow}$  and  $\Delta_{\uparrow\downarrow}$ ) are of visibly different sizes.

The total angular loss in a Burst Mode PAD depends on the combined contributions of the parallel and anti-parallel losses  $\Delta_{\uparrow\uparrow}$  and  $\Delta_{\uparrow\downarrow}$ . Subtracting the total loss from the full  $[0^\circ - 180^\circ]$  range of an ideal PAD with lossless binning and steering gives the widest theoretical range of pitch angles that can be captured by that PAD. Taking the size of that range as a fraction of a full  $180^\circ$  yields the following metric of the PAD’s completeness  $C$ , where all angles are expressed in degrees:

$$C = \frac{180^\circ - \Delta_{\uparrow\downarrow} - \Delta_{\uparrow\uparrow}}{180^\circ} = \frac{\alpha_{max} - \alpha_{min}}{180^\circ} \quad (3.5)$$

EAS data: solo\_L1\_swa-eas-padc\_20230612T172734-20230612T173233\_V01.cdf  
 MAG data: solo\_L2\_mag-srf-normal\_20230612\_V01.cdf



**Figure 3.8:** Top panel: A plot of magnetic field vectors and calculated bin selection from 12th June 2023. Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Green represents  $B_{EAS}$  and purple represents  $B_{MAG}$ .  $\alpha_{min}$  and  $\alpha_{max}$  are plotted as contours in red. Bottom panel: A plot of min and max pitch angles  $\alpha_{min}$  and  $\alpha_{max}$  and PAD completeness  $C$  over the same period.  $\alpha_{min}$  and  $\alpha_{max}$  are plotted on the angular axis on the left, and  $C$  is plotted on the percentage axis on the right.

Image Source: Author's own work

In summary, an algorithm for calculating the completeness of a Burst Mode PAD at a particular point in time  $t$  can be described as follows:

1. Acquire the upper and lower boundaries of the parallel and anti-parallel elevation bins selected by SWA-DPU at  $t$ . In terms of the variables in SOAR EAS L1 *.cdf* files, these boundaries can be calculated by summing the vectors from “SWA\_EAS\_ELEVATION” and “SWA\_EAS\_ELEVATION\_delta\_upper”, and from “SWA\_EAS\_ELEVATION” and “SWA\_EAS\_ELEVATION\_delta\_lower” at  $t$  respectively.
2. Acquire the EAS sensor head selected by SWA-DPU at  $t$ . In terms of the variables in SOAR EAS L1 *.cdf* files, these can be found under “SWA\_EAS\_EasUsed”.
3. Identify the 8Hz MAG dataset containing the contemporaneous  $B_{MAG}$  vector at  $t$ , and acquire the parallel and anti-parallel  $B_{MAG}$  vectors for that time. For simplicity, this dataset should be chosen such that the  $B_{MAG}$  vectors are already in cartesian SRF coordinates (see Section 2.1).
4. Using the transformation matrices in the right column of Table 2.1, transform the parallel and anti-parallel  $B_{MAG}$  vectors from cartesian SRF coordinates to the cartesian coordinates of the EAS sensor head selected at  $t$  (“EASX”).
5. Transform the parallel and anti-parallel  $B_{MAG}$  vectors from cartesian EASX coordinates to spherical EASX coordinates, and acquire their elevation angles in the range  $[-90^\circ, +90^\circ]$ <sup>1</sup>.
6. For each of the parallel and anti-parallel  $B_{MAG}$  vectors, check if their elevation angles are correctly binned by checking if they are between the upper and lower selected elevation bin boundaries. If either one is already correctly binned, then its associated loss is zero.

---

<sup>1</sup>Even though each EAS head can only sample the elevation range  $[-45^\circ, +45^\circ]$ ,  $B_{MAG}$  vectors are not prevented from having an elevation outside of that range.

7. For each vector that is not already correctly binned, calculate the difference between its elevation and the elevation of the upper and lower selected elevation bin boundaries. The smallest of those two differences represents  $\Delta_{\uparrow\downarrow}$  for the parallel  $B_{MAG}$  vector or  $\Delta_{\downarrow\uparrow}$  for the anti-parallel  $B_{MAG}$  vector.
8. Use  $\Delta_{\uparrow\downarrow}$  and  $\Delta_{\downarrow\uparrow}$  to calculate  $C$  at  $t$  according to Equation 3.5.

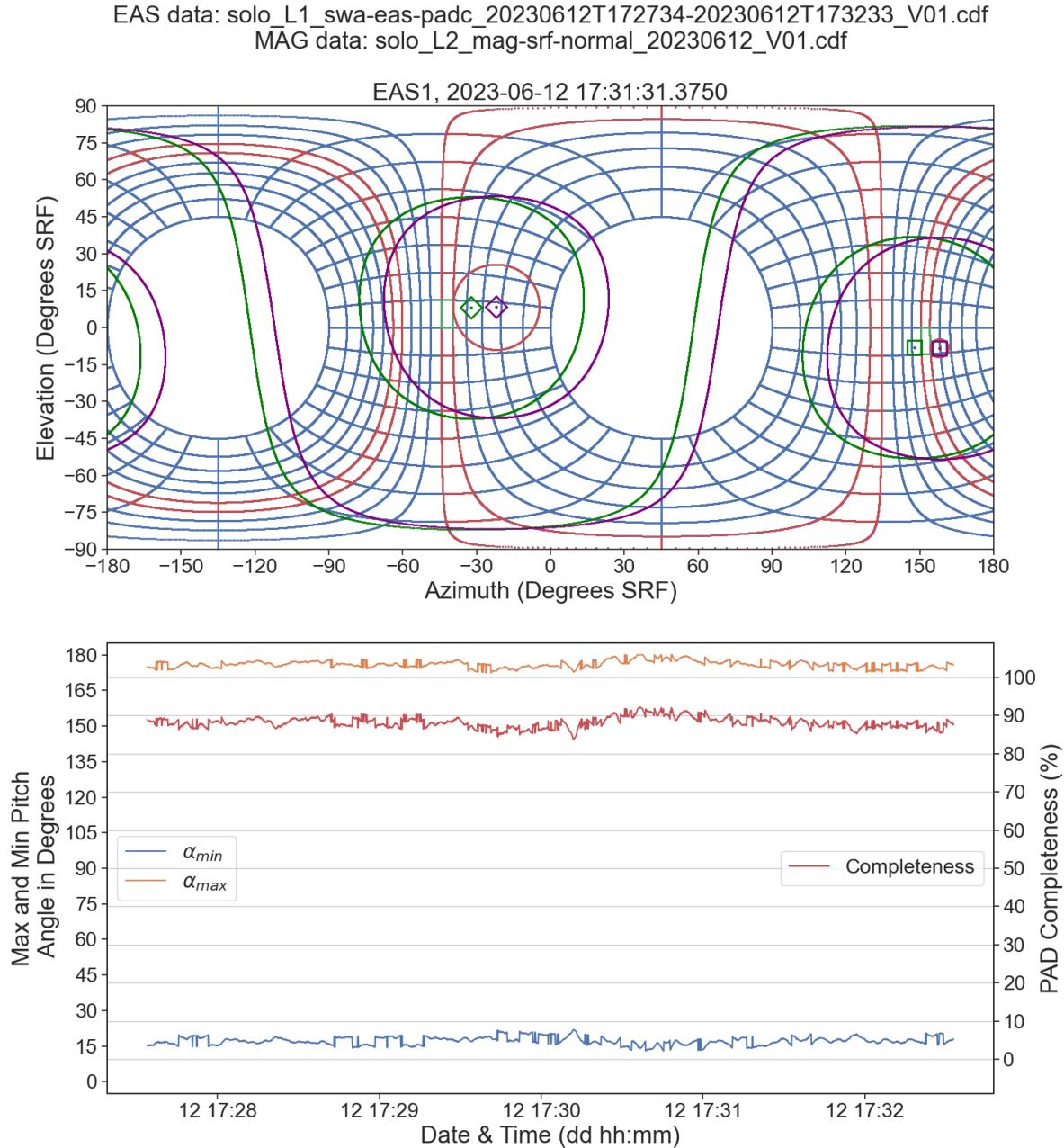
This algorithm is relatively inexpensive to compute when compared to an alternative approach where EAS L1 data are completely re-binned into an accurate PAD and the missing pitch angle bins are counted manually. This is an algorithm that could be computed shortly after the receipt of EAS L1 data on the condition that contemporaneous MAG data are also downlinked and post-processed on the ground in a timely manner. Furthermore, the convenience of expressing PAD completeness as a single number,  $C$ , may allow for faster determination of useful/unusable PADs for specific research purposes. For example, when investigating narrow, field-aligned electron strahl populations, high- $C$  may be a necessity to avoid missing all or most of the beams being studied. On the other hand, PADs with very low  $C$  might be expected to have a limited range of pitch angle measurements, but multiple measurements of the same pitch few angles at varying gyrophase angles. Repeated measurements of the same pitch angle are redundant under the assumption of gyrotropy, but 8Hz measurements of gyrophase may provide a way to test that assumption using science data that are already available on SOAR, which  $C$  could be used to identify[5].

While  $C$  represents the theoretical maximum fraction of pitch angles available in a PAD, the effective range also depends on the choice of pitch angle binning. When EAS Burst Mode was introduced by Owen et al (2021), it was proposed that EAS VDFs could be resampled into eighteen 10°-wide pitch angle bins (this can be seen in the example in Figure 1.5)[5]. If this number of bins is kept constant, then pitch angle loss would cause the bins at either end of the distribution either not to be “filled” or to be completely empty; their own form of incompleteness. Incomplete bins, if not completely discarded, should be analysed with caution as they may lead to misleading conclusions about the true electron PAD. Another approach is

to vary the bin width, uniformly or non-uniformly, to maximise the actual range of pitch angles that can be analysed. In large datasets with varying completeness, the range of usable pitch angles may have to be sacrificed for the convenience of using standardised bin widths and bin numbers.

So far, only the discrepancy between  $B_{EAS}$  and  $B_{MAG}$  has been examined for its effect on PAD completeness. However, the binning issue described in Section 3.2 should be expected to have its own effects on PAD completeness. Luckily, these effects are expected to be completely captured by the completeness algorithm. The binning issue concerns the discrepancy between  $B_{EAS}$  and its selected elevation bins, while the completeness algorithm concerns the discrepancy between  $B_{MAG}$  and the selected elevation bins. Whether or not these elevation bins were selected correctly for  $B_{EAS}$  only affects the completeness score indirectly depending on the agreement between  $B_{EAS}$  and  $B_{MAG}$ . In fact, given sufficient disagreement between  $B_{EAS}$  and  $B_{MAG}$ , the binning issue may lead  $B_{MAG}$  to be binned *more accurately* than  $B_{EAS}$  (for example, if  $B_{EAS}$  lies outside the selected bin while  $B_{MAG}$ , by chance, lies inside the selected bin).

In Figure 3.8,  $\alpha_{min}$  and  $\alpha_{max}$  have been calculated using binning according to the 17th July 2021 bin tables from SOAR (see Appendix A). The binning issue described in Section 3.2 makes this calculation inaccurate. By extension, Figure 3.8 is also inaccurate. A more accurate figure can be generated by calculating  $\alpha_{min}$  and  $\alpha_{max}$  with the actual bins selected. The result of reproducing Figure 3.8 with this “corrected” approach is Figure 3.9. By comparing Figure 3.8 and Figure 3.9, it can be seen that  $\alpha_{min}$  is significantly higher in the latter plot ( $\sim 15^\circ$ ) than in the former plot ( $\sim 5^\circ$ ), meaning that  $\Delta_{\parallel\parallel}$  is larger as a consequence of the wrongly-selected elevation bin being farther from both  $B_{EAS,\parallel\parallel}$  and  $B_{MAG,\parallel\parallel}$ . In contrast, it can also be seen that  $\alpha_{max}$  is somewhat higher in the latter plot ( $\sim 175^\circ$ ) than in the former plot ( $\sim 170^\circ$ ), meaning that  $\Delta_{\parallel\parallel}$  is *smaller* as a consequence of the wrongly-selected elevation bin being farther from  $B_{EAS,\parallel\parallel}$  but closer to  $B_{MAG,\parallel\parallel}$  by coincidence.



**Figure 3.9:** Top panel: A plot of magnetic field vectors and bin selection from 12th June 2023, where the binning issue is visible. Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Green represents  $B_{EAS}$  and purple represents  $B_{MAG}$ .  $\alpha_{min}$  and  $\alpha_{max}$  are plotted as contours in red. Bottom panel: A plot of min and max pitch angles  $\alpha_{min}$  and  $\alpha_{max}$  and PAD completeness  $C$  over the same period.  $\alpha_{min}$  and  $\alpha_{max}$  are plotted on the angular axis on the left, and  $C$  is plotted on the percentage axis on the right.

Image Source: Author's own work

Equation 3.5 assumes the use of a basic VDF re-binning pipeline where pitch angles in the range  $[0^\circ, > 90^\circ]$  are re-binned from the parallel selected elevation bin and pitch angles in the range  $[< 90^\circ, 180^\circ]$  are re-binned from the anti-parallel selected elevation bin. However, with sufficient disagreement between  $B_{EAS}$  and  $B_{MAG}$ , it is theoretically possible for  $B_{MAG, \uparrow\downarrow}$  to be closer to either boundary of the anti-parallel selected elevation bin than it is to the parallel selected elevation bin. Therefore, in that scenario, there may be a theoretical reduction in loss if the pitch angles in the range  $[0^\circ, > 90^\circ]$  were re-binned from the anti-parallel selected elevation bin instead of the parallel selected elevation bin. This scenario is particularly likely if  $B_{EAS}$  is close to the selected EAS head's aperture center plane, because this should cause EAS to select parallel and anti-parallel elevation bins that are close to each other. An example of this phenomenon is found in Burst Mode data from 1st January 2023, and is shown in Figure 3.10 along with contours for  $\alpha_{min}$  and  $\alpha_{max}$  as calculated using Equations 3.2 and 3.4 respectively. In Figure 3.10, the two selected elevation bins are adjacent to each other, and  $B_{MAG, \uparrow\downarrow}$  lies inside a different elevation bin than  $B_{EAS, \uparrow\downarrow}$ . Although  $\alpha_{min}$  is still calculated and plotted under the assumption that it will be recovered from the parallel elevation bin, it is possible that  $\alpha_{min}$  (and therefore  $\Delta_{\uparrow\downarrow}$ ) could be eliminated entirely if the anti-parallel elevation bin, in which  $B_{MAG, \uparrow\downarrow}$  lies, was used instead.

This alternative approach would imply alternative calculations for the loss, yielding  $\Delta_{alt, \uparrow\downarrow}$  and  $\Delta_{alt, \uparrow\downarrow}$ , which are given by Equations 3.6 and 3.7. These equations differ from Equations 3.1 and 3.3 in that  $\theta_{MAG, \uparrow\downarrow}$  and  $\theta_{MAG, \uparrow\downarrow}$  are each compared to the elevations of *all four* boundaries across both selected elevation bins instead of considering only the two boundaries of a single bin. In this case, the loss is given by the minimum angular distance of all four comparisons. As shown in Equations 3.6 and 3.7, this can be expressed succinctly by substituting  $\Delta_{\uparrow\downarrow}$  and  $\Delta_{\uparrow\downarrow}$  for their equivalent expressions in Equations 3.1 and 3.3.

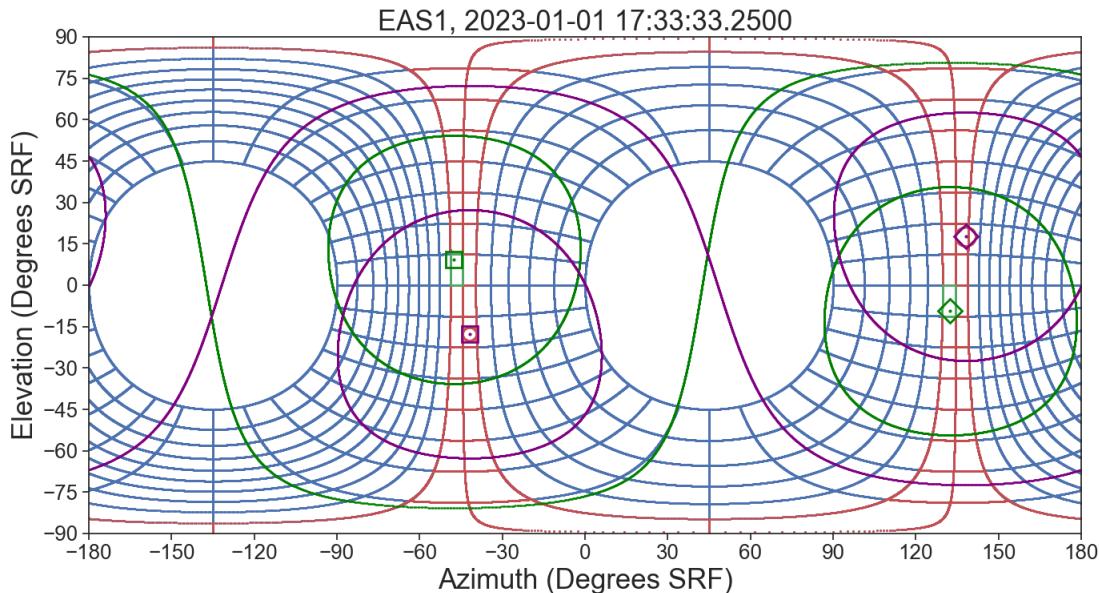
$$\Delta_{alt, \uparrow\downarrow} = \min\{\Delta_{\uparrow\downarrow}, |\theta_{MAG, \uparrow\downarrow} - \theta_{upper, \uparrow\downarrow}|, |\theta_{MAG, \uparrow\downarrow} - \theta_{lower, \uparrow\downarrow}|, |\theta_{MAG, \uparrow\downarrow} - \theta_{mid, \uparrow\downarrow}|, |\theta_{MAG, \uparrow\downarrow} - \theta_{mid+1, \uparrow\downarrow}|\}\quad (3.6)$$

$$\Delta_{alt,1\downarrow} = \min\{\Delta_{1\downarrow}, |\theta_{MAG,1\downarrow} - \theta_{upper,1\uparrow}|, |\theta_{MAG,1\downarrow} - \theta_{lower,1\uparrow}|\} \quad (3.7)$$

An equivalent completeness score  $C_{alt}$ , is given by:

$$C_{alt} = \frac{180^\circ - \Delta_{alt,1\downarrow} - \Delta_{alt,1\uparrow}}{180^\circ} \quad (3.8)$$

$C_{alt}$  can be incorporated into the algorithm for calculating  $C$  presented on page 62 by changing steps 6 and 7 to require both the parallel and anti-parallel selected bins to be checked and compared against the  $B_{MAG}$  vector, and by changing step 8 to use Equation 3.8.



**Figure 3.10:** A plot of magnetic field vectors and calculated bin selection from 1st January 2023, where the parallel and anti-parallel elevation bins are adjacent to each other. Diamond and square markers represent parallel and anti-parallel magnetic field vectors respectively. Green represents  $B_{EAS}$  and purple represents  $B_{MAG}$ .  $\alpha_{min}$  and  $\alpha_{max}$  are plotted as (barely visible) contours in red.

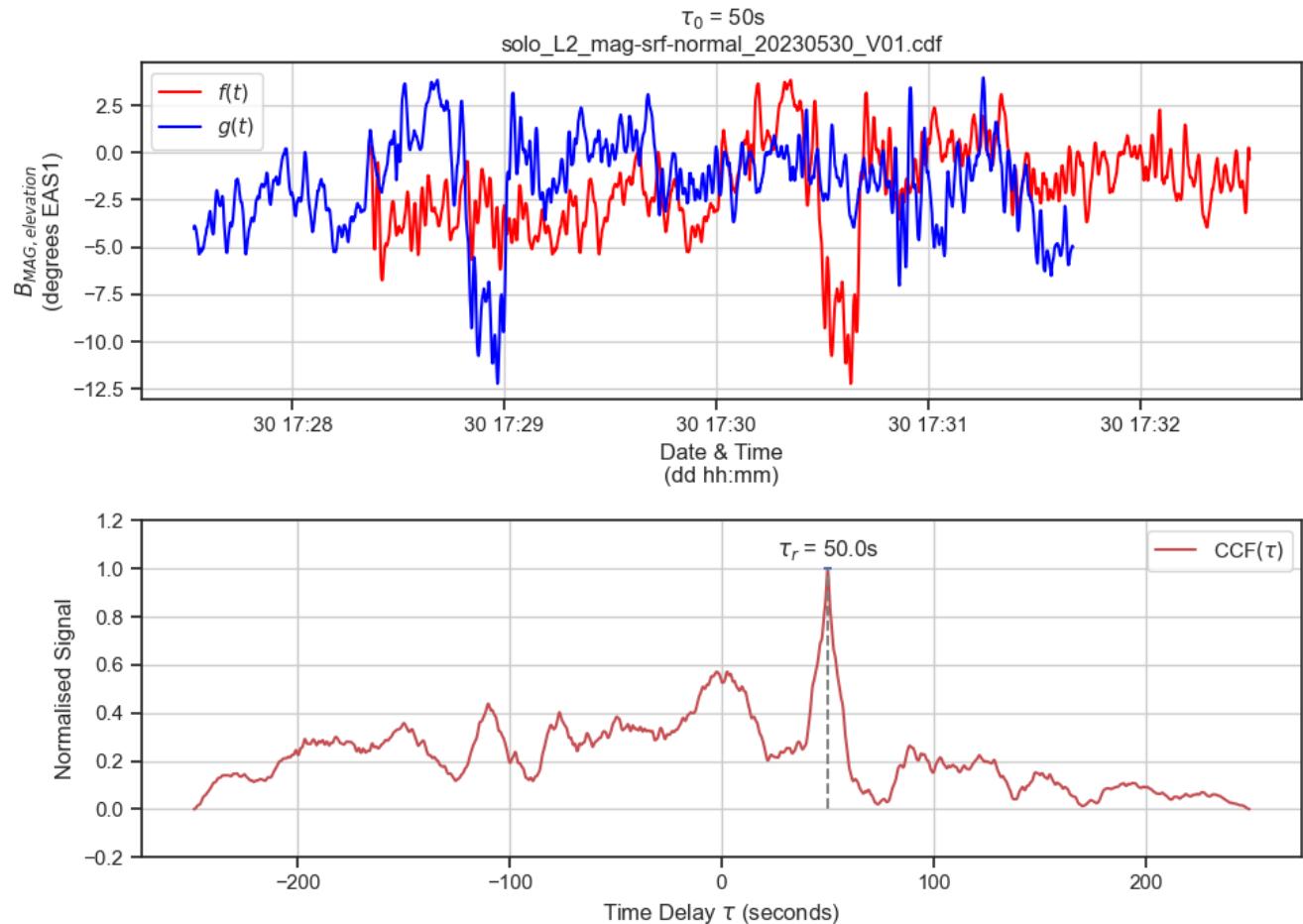
Image Source: Author's own work

$C$ ,  $C_{alt}$  and the algorithms used to calculate them represent two relatively inexpensive approaches to determining the completeness of any given Burst Mode PAD based only on data for the contemporaneous MAG vector used ( $B_{MAG}$ ) and the EAS

elevation bins selected. In particular,  $C$  and its associated losses and min/max pitch angles have been calculated and presented for various instances of randomly selected sample data in this report, but future work could investigate the long-term variation of  $C$ , collating months or years of EAS Burst Mode data available from SOAR. This may reveal a relationship between completeness and the time elapsed since the last routine calibration update to correct MAG offset drift, or a relationship between completeness and the rate of change of the magnetic field, potentially driven by magnetic turbulence in the solar wind[45].

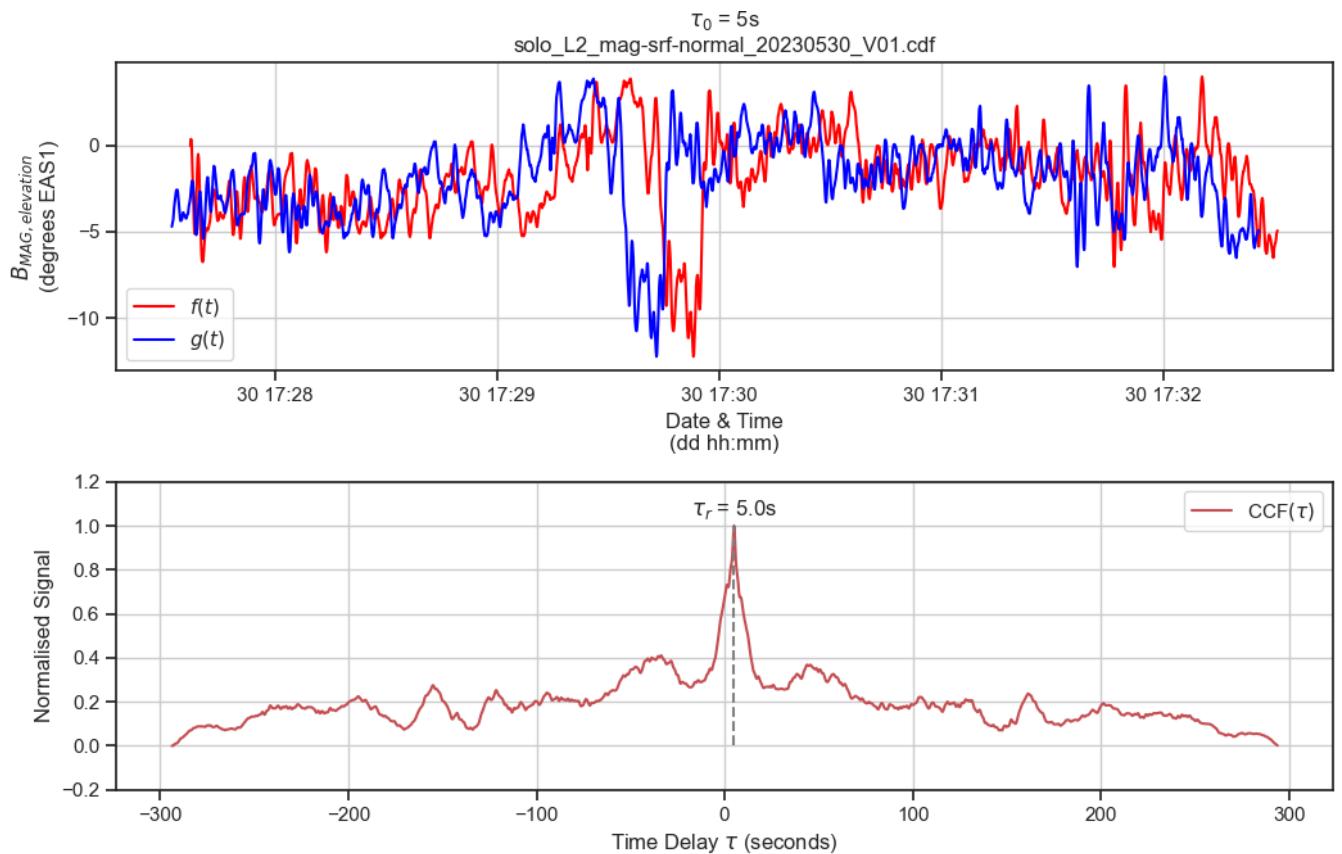
### 3.4 S20 Link Latency

To test the S20 latency algorithm described in Section 2.7 using real magnetic field data, a sample of  $B_{MAG}$  data from 30th May 2023 was given a large, artificial time offset of 50s and cross-correlated with itself, yielding two versions of it to represent an equivalent of  $f(t)$  and  $g(t)$  as previously described. The result for the  $B_{MAG}$  elevation in EAS1 coordinates is presented in Figure 3.11. Only a single axis is considered here to avoid having to combine different measurements of the time delay from different axes, as described in detail in the author’s previous work[15]. The precision here is surprisingly high, yielding  $\tau_r = 49.9s \pm 1.8s$ , and the error bar associated with this uncertainty is barely visible on the lower panel of Figure 3.11. The algorithm was then applied again, with a shorter time delay of 5s, yielding Figure 3.12. Surprisingly again, the precision in this case was also high, yielding  $\tau_r = 4.90s \pm 0.2s$ . If this trend in increasing precision continues, then the S20 latency algorithm may be able to accurately characterise real differences between  $B_{MAG}$  and  $B_{EAS}$  data more effectively than the author’s previous attempts[15]. This is an exciting subject that warrants future investigation.



**Figure 3.11:** Top panel: Two “identical” plots of  $B_{EAS}$  elevation in EAS1 coordinates with an artificial, 50s time delay between them. Bottom panel: The result of the CCF algorithm applied to the top panel, where the uncertainty in  $\tau_r$  is indicated by the small, horizontal, blue bar.

Image Source: Author’s own work



**Figure 3.12:** Top panel: Two “identical” plots of  $B_{EAS}$  elevation in EAS1 coordinates with an artificial, 5s time delay between them. Bottom panel: The result of the CCF algorithm applied to the top panel.

Image Source: Author’s own work

## Chapter 4

# Conclusion

The goal of this project was to determine the accuracy and completeness of Solar Orbiter SWA-EAS Burst Mode PADs as a result of inaccuracies in  $B_{EAS}$  the magnetic field vector data received by the SWA-DPU from MAG over the onboard S20 data link. These inaccuracies were expected to manifest as an offset between  $B_{EAS}$  and  $B_{MAG}$ ; the ground-processed magnetic field vector, and this was confirmed by preliminary comparisons between magnetic field time series performed using methodologies developed to crop and transform time series data acquired from the Solar Orbiter Archive. These offsets were found to be significant enough to lead to lost data in Burst Mode PADs, and this led to the development of an algorithm for  $C$ ; a concise metric for Burst Mode PAD completeness depending on pitch angle loss, represented by  $\Delta_{\parallel\parallel}$  and  $\Delta_{\parallel\|}$ . Pitch angle loss due to the inaccuracies described in this project manifests first near the extremities of a PAD at  $0^\circ$  and  $180^\circ$ , which may have implications for future research investigating field-aligned electron strahl. Particularly high pitch angle loss (i.e. low PAD completeness) may also be valuable in its own right if it can be used to study agyrotropy. The metric  $C$  can be a computationally inexpensive tool for classifying PAD data for any of these purposes.

The development of  $C$  was aided by the development of a tool for creating images and animations visualising EAS elevation-azimuth bins projected on to SRF coordinates along with time-varying magnetic field vectors, pitch angle contours, and PAD loss.

One source of  $B_{EAS}$  inaccuracy that this project aimed to investigate in particular was a possible latency over the S20 link. A method was developed to determine this latency by cross-correlation of  $B_{EAS}$  and  $B_{MAG}$  using Monte Carlo error estimation, yielding promising, though inconclusive results.

In addition to the project's primary goals, it also led to the discovery of some unexpected features in the EAS Burst Mode dataset. One such feature is a recurring issue with purported 8Hz  $B_{EAS}$  data that is only updated at a 1Hz cadence, which may have consequences for Burst Mode data accuracy. Much has been said about the error in SWA-DPU whose discovery led to the deletion of  $\sim 6$  months of EAS data from SOAR.

For future work, a clear direction would be to apply the methods developed in this project for calculating  $C$  and S20 latency to much larger sets of EAS and MAG data and investigate how those quantities vary over months or years. Similar surveys might lead to other discoveries, such as the prevalence and/or origin of the 8Hz-1Hz data issue. There is also plenty of room for improvement of the time series cropping and S20 latency determination algorithms that were hurriedly developed over the course of this project.

## **Appendix A**

# **EAS Angular Bin Tables**

Tables containing elevation and azimuth binning data for both EAS heads can be found in some *.cdf* files uploaded to SOAR. The uploaded tables have changed over the years since Solar Orbiter’s commissioning, and tables associated with data uploaded at any time between 17th July 2021 and the time of writing are included here.

“EAS1/2 Bin Index” refers to the index for each bin. “EAS1/2 Bin Index” is only used for the azimuthal bin table, because this table is identical for EAS1 and EAS2. The elevation bin tables are different for Eas1 and EAS2, hence the distinction between “EAS1 Bin Index” and “EAS2 Bin Index”. “Bin Center” refers to the angle of the center of each bin. “Bin Upper Delta” and “Bin Lower Delta” refer to the angular differences between “Bin Center” and the upper and lower boundaries of each bin respectively.

<b>EAS1/2 Bin Index</b>	<b>Bin Center (°)</b>	<b>Bin Upper Delta (°)</b>	<b>Bin Lower Delta (°)</b>
0	5.625	5.625	5.625
1	16.875	5.625	5.625
2	28.125	5.625	5.625
3	39.375	5.625	5.625
4	50.625	5.625	5.625
5	61.875	5.625	5.625
6	73.125	5.625	5.625
7	84.375	5.625	5.625
8	95.625	5.625	5.625
9	106.875	5.625	5.625
10	118.125	5.625	5.625
11	129.375	5.625	5.625
12	140.625	5.625	5.625
13	151.875	5.625	5.625
14	163.125	5.625	5.625
15	174.375	5.625	5.625
16	185.625	5.625	5.625
17	196.875	5.625	5.625
18	208.125	5.625	5.625
19	219.375	5.625	5.625
20	230.625	5.625	5.625
21	241.875	5.625	5.625
22	253.125	5.625	5.625
23	264.375	5.625	5.625
24	275.625	5.625	5.625
25	286.875	5.625	5.625
26	298.125	5.625	5.625
27	309.375	5.625	5.625
28	320.625	5.625	5.625
29	331.875	5.625	5.625
30	343.125	5.625	5.625
31	354.375	5.625	5.625

**Table A.1:** Bin Table: EAS1/2 Azimuth (as found in SOAR) - 17th July 2021

<b>EAS1 Bin Index</b>	<b>Bin Center (°)</b>	<b>Bin Upper Delta (°)</b>	<b>Bin Lower Delta (°)</b>
0	39.34	5.66	5.66
1	29.17	4.514	4.514
2	20.91	3.748	3.748
3	13.98	3.179	3.179
4	8.06	2.74	2.74
5	2.91	2.409	2.409
6	-1.66	2.161	2.161
7	-5.82	1.996	1.996
8	-9.7	1.886	1.886
9	-13.43	1.841	1.841
10	-17.13	1.856	1.856
11	-20.94	1.95	1.95
12	-25.0	2.115	2.115
13	-29.53	2.413	2.413
14	-34.82	2.88	2.88
15	-41.36	3.655	3.655

**Table A.2:** Bin Table: EAS1 Elevation (as found in SOAR) - 17th July 2021

<b>EAS2 Bin Index</b>	<b>Bin Center (°)</b>	<b>Bin Upper Delta (°)</b>	<b>Bin Lower Delta (°)</b>
0	38.94	6.06	6.06
1	28.25	4.633	4.633
2	19.86	3.761	3.761
3	12.99	3.111	3.111
4	7.25	2.624	2.624
5	2.35	2.272	2.272
6	-1.93	2.012	2.012
7	-5.78	1.838	1.838
8	-9.37	1.747	1.747
9	-12.84	1.722	1.722
10	-16.32	1.759	1.759
11	-19.97	1.887	1.887
12	-23.97	2.113	2.113
13	-28.57	2.485	2.485
14	-34.13	3.071	3.071
15	-41.1	3.897	3.897

**Table A.3:** Bin Table: EAS2 Elevation (as found in SOAR) - 17th July 2021

## Appendix B

# Code Excerpts

The full codebase for this project can be found in the author’s GitHub repository:

*<https://github.com/joratto/Solar-Orbiter-SWA-EAS-Completeness-MSc>*

```
time_index = 0
for i in range(1, searchdivisions+1):
    time_index += int(length/2**i)*(time_array[time_index+int(length/2**i)] < time_reference)
while (time_array[time_index] < time_reference) and (time_index < length-1):
    time_index += 1
```

**Listing B.1:** “Time Axis Cropping Algorithm”

For the bin visualisation algorithm described in Section 2.6, a single elevation bin can be defined by keeping the elevation angle of successive points constant (and equal to an upper/lower bound) while the azimuth phase angle is varied according to a defined angular point density, and vice versa for a single azimuth bin. Repeating this for the upper and lower edge of an elevation bin and an azimuth bin defines an elevation-azimuth pixel. Once defined, the points delineating each pixel can be projected to SRF space in spherical coordinates with the following sequence of transformations: Spherical EAS1/2 → Cartesian EAS1/2 → Cartesian SRF → Spherical SRF. This pixel point plotting algorithm was implemented in Python as described in Appendix Listing B.2, where k\_az and k\_el represent integer numbers of azimuth and elevation phase angles along a single bin edge respectively. This algorithm double-counts edges shared between adjacent pixels, trading a linear amount of computational efficiency for some convenience in plotting individual pixels. Once the points are transformed to spherical SRF coordinates, their azimuth

angles are converted from the range [-180,180] to the range [0,360] for uniformity with EAS azimuth conventions. Given sufficient point density ( $20^{\circ}$  works well), this results in the visually curved bins shown in Figure 1.6. Another result of high point density is that this algorithm's many matrix multiplications for each point transformation lead to a computation time of several minutes for a single image. To save time, the algorithm was adapted to save the point coordinates for each pixel to .csv files in a directory under “bin\_projections\{EASX\}\el{Y}\EASX\_el{Y}\_azZ.csv” where  $X$  is the EAS head number and  $Y$  and  $Z$  are the elevation and azimuth bin indices respectively. These files can then be read and plotted with an order of magnitude reduction in computation time. Part of this algorithm's Python implementation is as shown in Appendix Listing B.3.

```

angleStep = 1/pointDensity
azimuthPointCount = int(pointDensity*360/azimBinCount)
for i in range(elevBinCount):
    elevationBinWidth = 2*elevDeltaLowerArray[i]
    elevationPointCount = int(pointDensity*elevationBinWidth)
    binBoundaryProjectionArray.append([])
    for j in range(azimBinCount):
        pointArray = []
        for k_az in range(0,azimuthPointCount):
            lowerEdge_elev = np.array([1, elevLowerBoundArray[i], azimLowerBoundArray[j]+k_az*angleStep])
            lowerEdge_elev = cartToSphere(cart_proj_tuple[head].dot(sphereToCart(lowerEdge_elev)))
            pointArray.append(lowerEdge_elev)

            upperEdge_elev = np.array([1, elevUpperBoundArray[i], azimLowerBoundArray[j]+k_az*angleStep])
            upperEdge_elev = cartToSphere(cart_proj_tuple[head].dot(sphereToCart(upperEdge_elev)))
            pointArray.append(upperEdge_elev)

        for k_el in range(0,elevationPointCount):
            lowerEdge_azim = np.array([1, elevLowerBoundArray[i]+k_el*angleStep, azimLowerBoundArray[j]])
            lowerEdge_azim = cartToSphere(cart_proj_tuple[head].dot(sphereToCart(lowerEdge_azim)))
            pointArray.append(lowerEdge_azim)

            upperEdge_azim = np.array([1, elevLowerBoundArray[i]+k_el*angleStep, azimUpperBoundArray[j]])
            upperEdge_azim = cartToSphere(cart_proj_tuple[head].dot(sphereToCart(upperEdge_azim)))
            pointArray.append(upperEdge_azim)

```

#### **Listing B.2:** “Pixel Point Plotting Algorithm”

```

for i in range(16):
    for j in range(32):
        filename = 'bin_projections\{EAS\}\el{i}\EAS\{head\}\el{i}\az{j}.csv'
        pixelArray = np.genfromtxt(filename, delimiter=",")
        el = pixelArray[1]
        az = pixelArray[2]
        ax.scatter(az,el,color=bin_color,marker='.',s=s)

```

#### **Listing B.3:** “Pixel Point Saving Algorithm”

## Appendix C

# Colophon

This document was set in the Times New Roman typeface using L<sup>A</sup>T<sub>E</sub>X and BibT<sub>E</sub>X. It was composed using Overleaf. The document format is based on the “UCL Thesis L<sup>A</sup>T<sub>E</sub>X Template” by Ian Kirker, which is available on GitHub[46]. In addition, a few custom L<sup>A</sup>T<sub>E</sub>X functions were used to make the writing process a bit easier. For example:

Quotations are written using this function:

```
\newcommand{\q}[1]{‘#1’}
```

The symbols  $B_{EAS}$  and  $B_{MAG}$  are written using these functions:

```
\newcommand{\Beas}{\left(B_{EAS}\right)}  
\newcommand{\Bmag}{\left(B_{MAG}\right)}
```

The symbols “ $\upharpoonleft$ ” and “ $\downharpoonright$ ” are dependent on the *amssymb* package for L<sup>A</sup>T<sub>E</sub>X and are written using these functions:

```
\newcommand{\ppara}{\upharpoonleft \! \upharpoonright}  
\newcommand{\apara}{\upharpoonleft \! \downharpoonright}
```

Large figures are occasionally plotted with a width  $> 1$ , which causes them to be un-centered on the page by default. The following function by *tex.stackexchange.com* user Marc Baudoin keeps them centered[47]:

```
\makeatletter
\newcommand*\centerfloat}{%
\parindent \z@
\leftskip \z@ \plus 1fil \minus \textwidth
\rightskip\leftskip
\parfillskip \z@skip}
\makeatother
```

“I have made an end at last, and my weary hand can rest.”[48]

# Bibliography

- [1] Müller, D., St. Cyr, O. C., Zouganelis, I., Gilbert, H. R., Marsden, R., Nieves-Chinchilla, T., Antonucci, E., Auchère, F., Berghmans, D., Horbury, T. S., Howard, R. A., Krucker, S., Maksimovic, M., Owen, C. J., Rochus, P., Rodriguez-Pacheco, J., Romoli, M., Solanki, S. K., Bruno, R., Carlsson, M., Fludra, A., Harra, L., Hassler, D. M., Livi, S., Louarn, P., Peter, H., Schühle, U., Teriaca, L., del Toro Iniesta, J. C., Wimmer-Schweingruber, R. F., Marsch, E., Velli, M., De Groof, A., Walsh, A., and Williams, D. The solar orbiter mission - science overview. *AA*, 642:A1, 2020.
- [2] Horbury, T. S., O'Brien, H., Carrasco Blazquez, I., Bendyk, M., Brown, P., Hudson, R., Evans, V., Oddy, T. M., Carr, C. M., Beek, T. J., Cupido, E., Bhattacharya, S., Dominguez, J.-A., Matthews, L., Myklebust, V. R., Whiteside, B., Bale, S. D., Baumjohann, W., Burgess, D., Carbone, V., Cargill, P., Eastwood, J., Erdös, G., Fletcher, L., Forsyth, R., Giacalone, J., Glassmeier, K.-H., Goldstein, M. L., Hoeksema, T., Lockwood, M., Magnes, W., Maksimovic, M., Marsch, E., Matthaeus, W. H., Murphy, N., Nakariakov, V. M., Owen, C. J., Owens, M., Rodriguez-Pacheco, J., Richter, I., Riley, P., Russell, C. T., Schwartz, S., Vainio, R., Velli, M., Vennerstrom, S., Walsh, R., Wimmer-Schweingruber, R. F., Zank, G., Müller, D., Zouganelis, I., and Walsh, A. P. The solar orbiter magnetometer. *AA*, 642:A9, 2020.
- [3] M. Maksimovic, S. D. Bale, T. Chust, Y. Khotyaintsev, V. Krasnoselskikh, M. Kretzschmar, D. Plettemeier, H. O. Rucker, J. Souček, M. Steller, Š. Štverák, P. Trávníček, A. Vaivads, S. Chaintreuil, M. Dekkali, O. Alexandrova, P. A.

Astier, G. Barbary, D. Bérard, X. Bonnin, K. Bougedada, B. Cecconi, F. Chapron, M. Chariet, C. Collin, Y. de Conchy, D. Dias, L. Guéguen, L. Lamy, V. Leray, S. Lion, L. R. Malac-Allain, L. Matteini, Q. N. Nguyen, F. Pantellini, J. Parisot, P. Plasson, S. Thijs, A. Vecchio, I. Fratter, E. Bellouard, E. Lorfèvre, P. Danto, S. Julien, E. Guilhem, C. Fiachetti, J. Sanisidro, C. Laf-faye, F. Gonzalez, B. Pontet, N. Quéruel, G. Jannet, P. Fergeau, J. Y. Brochot, G. Cassam-Chenai, T. Dudok de Wit, M. Timofeeva, T. Vincent, C. Agrapart, G. T. Delory, P. Turin, A. Jeandet, P. Leroy, J. C. Pellion, V. Bouzid, B. Katra, R. Piberne, W. Recart, O. Santolík, I. Kolmašová, V. Krupař, O. Krupařová, D. Píša, L. Uhlíř, R. Lán, J. Baše, L. Ahlén, M. André, L. Bylander, V. Cripps, C. Cully, A. Eriksson, S. E. Jansson, E. P. G. Johansson, T. Karlsson, W. Puccio, J. Břínek, H. Öttacher, M. Panchenko, M. Berthomier, K. Goetz, P. Hellinger, T. S. Horbury, K. Issautier, E. Kontar, S. Krucker, O. Le Contel, P. Louarn, M. Martinović, C. J. Owen, A. Retino, J. Rodríguez-Pacheco, F. Sahraoui, R. F. Wimmer-Schweingruber, A. Zaslavsky, and I. Zouganelis. The Solar Orbiter Radio and Plasma Waves (RPW) instrument. , 642:A12, October 2020.

- [4] Owen, C. J., Bruno, R., Livi, S., Louarn, P., Al Janabi, K., Allegrini, F., Amoros, C., Baruah, R., Barthe, A., Berthomier, M., Bordon, S., Brockley-Blatt, C., Brysbaert, C., Capuano, G., Collier, M., DeMarco, R., Fedorov, A., Ford, J., Fortunato, V., Fratter, I., Galvin, A. B., Hancock, B., Heirtzler, D., Kataria, D., Kistler, L., Lepri, S. T., Lewis, G., Loeffler, C., Marty, W., Mathon, R., Mayall, A., Mele, G., Ogasawara, K., Orlandi, M., Pacros, A., Penou, E., Persyn, S., Petiot, M., Phillips, M., Přech, L., Raines, J. M., Reden, M., Rouillard, A. P., Rousseau, A., Rubiella, J., Seran, H., Spencer, A., Thomas, J. W., Trevino, J., Verscharen, D., Wurz, P., Alapide, A., Amoruso, L., André, N., Anekallu, C., Arciuli, V., Arnett, K. L., Ascolese, R., Bancroft, C., Bland, P., Brysch, M., Calvanese, R., Castronuovo, M., Čermák, I., Chornay, D., Clemens, S., Coker, J., Collinson, G., D'Amicis, R., Dandouras, I., Darnley, R., Davies, D., Davison, G., De Los Santos, A., Devoto, P., Dirks, G., Edlund, E., Fazakerley, A., Ferris, M., Frost, C., Fruit, G., Garat, C., Génot, V., Gibson, W., Gilbert,

- J. A., de Giosa, V., Gradone, S., Hailey, M., Horbury, T. S., Hunt, T., Jacquay, C., Johnson, M., Lavraud, B., Lawrenson, A., Leblanc, F., Lockhart, W., Maksimovic, M., Malpus, A., Marcucci, F., Mazelle, C., Monti, F., Myers, S., Nguyen, T., Rodriguez-Pacheco, J., Phillips, I., Popecki, M., Rees, K., Rogacki, S. A., Ruane, K., Rust, D., Salatti, M., Sauvaud, J. A., Stakhiv, M. O., Stange, J., Stubbs, T., Taylor, T., Techer, J.-D., Terrier, G., Thibodeaux, R., Urdiales, C., Varsani, A., Walsh, A. P., Watson, G., Wheeler, P., Willis, G., Wimmer-Schweingruber, R. F., Winter, B., Yardley, J., and Zouganelis, I. The solar orbiter solar wind analyser (swa) suite. *AA*, 642:A16, 2020.
- [5] Owen, C. J., Kataria, D. O., Berčič, L., Horbury, T. S., Berthomier, M., Verscharen, D., Bruno, R., Livi, S., Louarn, P., Anekallu, C., Kelly, C. W., Lewis, G. R., Watson, G., Fortunato, V., Mele, G., Nicolaou, G., Wicks, R. T., O'Brien, H., Evans, V., and Angelini, V. High-cadence measurements of electron pitch-angle distributions from solar orbiter swa-eas burst mode operations. *AA*, 656:L9, 2021.
- [6] V. Domingo, B. Fleck, and A. I. Poland. The SOHO Mission: an Overview. , 162(1-2):1–37, December 1995.
- [7] T. Kosugi, K. Matsuzaki, T. Sakao, T. Shimizu, Y. Sone, S. Tachikawa, T. Hashimoto, K. Minesugi, A. Ohnishi, T. Yamada, S. Tsuneta, H. Hara, K. Ichimoto, Y. Suematsu, M. Shimojo, T. Watanabe, S. Shimada, J. M. Davis, L. D. Hill, J. K. Owens, A. M. Title, J. L. Culhane, L. K. Harra, G. A. Doschek, and L. Golub. The Hinode (Solar-B) Mission: An Overview. , 243(1):3–17, June 2007.
- [8] M. L. Kaiser, T. A. Kucera, J. M. Davila, O. C. St. Cyr, M. Guhathakurta, and E. Christian. The STEREO Mission: An Introduction. , 136(1-4):5–16, April 2008.
- [9] N. J. Fox, M. C. Velli, S. D. Bale, R. Decker, A. Driesman, R. A. Howard, J. C. Kasper, J. Kinnison, M. Kusterer, D. Lario, M. K. Lockwood, D. J. McComas,

- N. E. Raouafi, and A. Szabo. The Solar Probe Plus Mission: Humanity’s First Visit to Our Star. , 204(1-4):7–48, December 2016.
- [10] Glyn Collinson and Dhiren Kataria. On variable geometric factor systems for top-hat electrostatic space plasma analyzers. *Measurement Science and Technology*, 21:105903, 08 2010.
- [11] Mason, G. M., Ho, G. C., Allen, R. C., Kouloumvakos, A., Wimmer-Schweingruber, R. F., and Rodriguez-Pacheco, J. Quiet-time suprathermal ions in the inner heliosphere during the rising phase of solar cycle 25. *AA*, 673:L12, 2023.
- [12] D. T. Young, J. J. Berthelier, M. Blanc, J. L. Burch, A. J. Coates, R. Goldstein, M. Grande, T. W. Hill, R. E. Johnson, V. Kelha, D. J. McComas, E. C. Sittler, K. R. Svenes, K. Szegö, P. Tanskanen, K. Ahola, D. Anderson, S. Bakshi, R. A. Baragiola, B. L. Barraclough, R. K. Black, S. Bolton, T. Booker, R. Bowman, P. Casey, F. J. Crary, D. Delapp, G. Dirks, N. Eaker, H. Funsten, J. D. Furman, J. T. Gosling, H. Hannula, C. Holmlund, H. Huomo, J. M. Illiano, P. Jensen, M. A. Johnson, D. R. Linder, T. Luntama, S. Maurice, K. P. McCabe, K. Mursula, B. T. Narheim, J. E. Nordholt, A. Preece, J. Rudzki, A. Ruitberg, K. Smith, S. Szalai, M. F. Thomsen, K. Viherkanto, J. Vilppola, T. Vollmer, T. E. Wahl, M. Wüest, T. Ylikorpi, and C. Zinsmeyer. Cassini Plasma Spectrometer Investigation. , 114(1-4):1–112, September 2004.
- [13] D. L. Mitchell, C. Mazelle, J. A. Sauvaud, J. J. Thocaven, J. Rouzaud, A. Fedorov, P. Rouger, D. Toublanc, E. Taylor, D. Gordon, M. Robinson, S. Heavner, P. Turin, M. Diaz-Aguado, D. W. Curtis, R. P. Lin, and B. M. Jakosky. The MAVEN Solar Wind Electron Analyzer. , 200(1-4):495–528, April 2016.
- [14] Johnstone, A. D. et al. PEACE: A plasma electron and current experiment. , 79(1572-9672):351–398, January 1997.

- [15] Pacheco Dickson, J. R. O. Determining the accuracy and completeness of burst mode data collection from the solar orbiter SWA electron sensor. Interim report, University College London, 2024.
- [16] X. Olaskoaga and Javier Andión. Solar orbiter instrument boom subsystem. 2017.
- [17] M. K. Dougherty, S. Kellock, D. J. Southwood, A. Balogh, E. J. Smith, B. T. Tsurutani, B. Gerlach, K. H. Glassmeier, F. Gleim, C. T. Russell, G. Erdos, F. M. Neubauer, and S. W. H. Cowley. The Cassini Magnetic Field Investigation. , 114(1-4):331–383, September 2004.
- [18] C. Carr, P. Brown, T. L. Zhang, J. Gloag, T. Horbury, E. Lucek, W. Magnes, H. O’Brien, T. Oddy, U. Auster, P. Austin, O. Aydogar, A. Balogh, W. Baumjohann, T. Beek, H. Eichelberger, K.-H. Fornacon, E. Georgescu, K.-H. Glassmeier, M. Ludlam, R. Nakamura, and I. Richter. The double star magnetic field investigation: instrument design, performance and highlights of the first year’s observations. *Annales Geophysicae*, 23(8):2713–2732, 2005.
- [19] David Mccomas, Eric Christian, Nathan Schwadron, Nicola Fox, J. Westlake, F. Allegrini, D. Baker, D. Biesecker, M. Bzowski, G. Clark, C. Cohen, Ian Cohen, Maher Dayeh, R. Decker, G. de Nolfo, M. Desai, Robert Ebert, H. Elliott, H. Fahr, and E. Zirnstein. Interstellar mapping and acceleration probe (imap): A new nasa mission. *Space Science Reviews*, 214, 10 2018.
- [20] W. G. Pilipp, H. Miggenrieder, M. D. Montgomery, K. H. Mühlhäuser, H. Rosenbauer, and R. Schwenn. Characteristics of electron velocity distribution functions in the solar wind derived from the helios plasma experiment. , 92(A2):1075–1092, February 1987.
- [21] Eckart Marsch. Kinetic Physics of the Solar Corona and Solar Wind. *Living Reviews in Solar Physics*, 3(1):1, July 2006.
- [22] Christian Vocks. *Electron kinetic processes in the solar corona and wind*. PhD thesis, Leibniz-Institut fur Astrophysik Potsdam, 2012.

- [23] Susanne F. Spinnangr, Michael Hesse, Paul Tenfjord, Cecilia Norgren, Håkon M. Kolstø, Norah K. Kwagal, Therese Moretto Jørgensen, and Tai Phan. Electron behavior around the onset of magnetic reconnection. *Geophysical Research Letters*, 49(23):e2022GL102209, 2022. e2022GL102209 2022GL102209.
- [24] San Lu, A. V. Artemyev, V. Angelopoulos, and P. L. Pritchett. Energetic electron acceleration by ion-scale magnetic islands in turbulent magnetic reconnection: Particle-in-cell simulations and artemis observations. *The Astrophysical Journal*, 896(2):105, jun 2020.
- [25] San Lu, V. Angelopoulos, A. V. Artemyev, P. L. Pritchett, J. Liu, A. Runov, A. Tenerani, C. Shi, and M. Velli. Turbulence and particle acceleration in collisionless magnetic reconnection: Effects of temperature inhomogeneity across pre-reconnection current sheet. *The Astrophysical Journal*, 878(2):109, jun 2019.
- [26] ESA. Solar Orbiter Archive — soar.esac.esa.int. <https://soar.esac.esa.int/soar/#home>, 2024. [Accessed 04-04-2024].
- [27] Michael Liu and Robert M. Candey. CDF. <https://cdf.gsfc.nasa.gov/>, 2024. [Accessed 08-04-2024].
- [28] Zouganelis, I., De Groof, A., Walsh, A. P., Williams, D. R., Müller, D., St Cyr, O. C., Auchère, F., Berghmans, D., Fludra, A., Horbury, T. S., Howard, R. A., Krucker, S., Maksimovic, M., Owen, C. J., Rodríguez-Pacheco, J., Romoli, M., Solanki, S. K., Watson, C., Sanchez, L., Lefort, J., Osuna, P., Gilbert, H. R., Nieves-Chinchilla, T., Abbo, L., Alexandrova, O., Anastasiadis, A., Andretta, V., Antonucci, E., Appourchaux, T., Aran, A., Arge, C. N., Aulanier, G., Baker, D., Bale, S. D., Battaglia, M., Bellot Rubio, L., Bemporad, A., Berthomier, M., Bocchialini, K., Bonnin, X., Brun, A. S., Bruno, R., Buchlin, E., Büchner, J., Bucik, R., Carcaboso, F., Carr, R., Carrasco-Blázquez, I., Cecconi, B., Cernuda Cangas, I., Chen, C. H. K., Chitta, L. P., Chust, T., Dalmasse, K.,

- D'Amicis, R., Da Deppo, V., De Marco, R., Dolei, S., Dolla, L., Dudok de Wit, T., van Driel-Gesztelyi, L., Eastwood, J. P., Espinosa Lara, F., Etesi, L., Fedorov, A., Félix-Redondo, F., Fineschi, S., Fleck, B., Fontaine, D., Fox, N. J., Gandorfer, A., Génot, V., Georgoulis, M. K., Gissot, S., Giunta, A., Gizon, L., Gómez-Herrero, R., Gontikakis, C., Graham, G., Green, L., Grundy, T., Haberreiter, M., Harra, L. K., Hassler, D. M., Hirzberger, J., Ho, G. C., Hurford, G., Innes, D., Issautier, K., James, A. W., Janitzek, N., Janvier, M., Jeffrey, N., Jenkins, J., Khotyaintsev, Y., Klein, K.-L., Kontar, E. P., Kontogiannis, I., Krafft, C., Krasnoselskikh, V., Kretzschmar, M., Labrosse, N., Lagg, A., Landini, F., Lavraud, B., Leon, I., Lepri, S. T., Lewis, G. R., Liewer, P., Linker, J., Livi, S., Long, D. M., Louarn, P., Malandraki, O., Maloney, S., Martinez-Pillet, V., Martinovic, M., Masson, A., Matthews, S., Matteini, L., Meyer-Vernet, N., Moraitis, K., Morton, R. J., Musset, S., Nicolaou, G., Nindos, A., O'Brien, H., Orozco Suarez, D., Owens, M., Pancrazzi, M., Papaioannou, A., Parenti, S., Pariat, E., Patsourakos, S., Perrone, D., Peter, H., Pinto, R. F., Plainaki, C., Plettemeier, D., Plunkett, S. P., Raines, J. M., Raouafi, N., Reid, H., Retino, A., Rezeau, L., Rochus, P., Rodriguez, L., Rodriguez-Garcia, L., Roth, M., Rouillard, A. P., Sahraoui, F., Sasso, C., Schou, J., Schühle, U., Sorriso-Valvo, L., Soucek, J., Spadaro, D., Stangalini, M., Stansby, D., Steller, M., Strugarek, A., Štverák, Š., Susino, R., Telloni, D., Terasa, C., Teriaca, L., Toledo-Redondo, S., del Toro Iniesta, J. C., Tsiropoulou, G., Tsounis, A., Tziotziou, K., Valentini, F., Vaivads, A., Vecchio, A., Velli, M., Verbeeck, C., Verdini, A., Verscharen, D., Vilmer, N., Vourlidas, A., Wicks, R., Wimmer-Schweingruber, R. F., Wiegmann, T., Young, P. R., and Zhukov, A. N. The solar orbiter science activity plan - translating solar and heliospheric physics questions into action. *AA*, 642:A3, 2020.
- [29] Consultative Committee for Space Data Systems. *CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. CCSDS Secretariat, 2022.
- [30] Jonathan T Niehof, Steven K Morley, Daniel T Welling, and Brian A Larsen. The spacepy space science package at 12 years. *Frontiers in Astronomy and*

*Space Sciences*, 9, 2022.

- [31] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [32] Dmitry Erofeev. Effect of large-scale inhomogeneity of solar wind velocity on distribution of directions of interplanetary magnetic field vector. *Solar-Terrestrial Physics*, 5(3):42–53, September 2019.
- [33] Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, W. E. Kerzendorf, A. Conley, N. Crighton, K. Barbary, D. Muna, H. Ferguson, F. Grollier, M. M. Parikh, P. H. Nair, H. M. Unther, C. Deil, J. Woillez, S. Conseil, R. Kramer, J. E. H. Turner, L. Singer, R. Fox, B. A. Weaver, V. Zabalza, Z. I. Edwards, K. Azalee Bostroem, D. J. Burke, A. R. Casey, S. M. Crawford, N. Dencheva, J. Ely, T. Jenness, K. Labrie, P. L. Lim, F. Pierfederici, A. Pontzen, A. Ptak, B. Refsdal, M. Servillat, and O. Streicher. Astropy: A community Python package for astronomy. , 558:A33, October 2013.
- [34] Astropy Collaboration, A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig, N. Dencheva, A. Ginsburg, J. T. VanderPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso,

S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D’Avella, C. Deil, É. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Ž. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P. Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska, T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terrón, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and Astropy Contributors. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. , 156(3):123, September 2018.

- [35] Astropy Collaboration, Adrian M. Price-Whelan, Pey Lian Lim, Nicholas Earl, Nathaniel Starkman, Larry Bradley, David L. Shupe, Aarya A. Patil, Lia Corrales, C. E. Brasseur, Maximilian N”othe, Axel Donath, Erik Tollerud, Brett M. Morris, Adam Ginsburg, Eero Vaher, Benjamin A. Weaver, James Tocknell, William Jamieson, Marten H. van Kerkwijk, Thomas P. Robitaille, Bruce Merry, Matteo Bachetti, H. Moritz G”unther, Thomas L. Aldcroft, Jaime A. Alvarado-Montes, Anne M. Archibald, Attila B’odi, Shreyas Bapat, Geert Barentsen, Juanjo Baz’an, Manish Biswas, M’ed’eric Boquien, D. J. Burke, Daria Cara, Mi-hai Cara, Kyle E. Conroy, Simon Conseil, Matthew W. Craig, Robert M. Cross, Kelle L. Cruz, Francesco D’Eugenio, Nadia Dencheva, Hadrien A. R. Devillepoix, J’org P. Dietrich, Arthur Davis Eigenbrot, Thomas Erben, Leonardo

Ferreira, Daniel Foreman-Mackey, Ryan Fox, Nabil Freij, Suyog Garg, Robel Geda, Lauren Glattly, Yash Gondhalekar, Karl D. Gordon, David Grant, Perry Greenfield, Austen M. Groener, Steve Guest, Sebastian Gurovich, Rasmus Handberg, Akeem Hart, Zac Hatfield-Dodds, Derek Homeier, Griffin Hosseinzadeh, Tim Jenness, Craig K. Jones, Prajwel Joseph, J. Bryce Kalmbach, Emir Karamehmetoglu, Mikolaj Kaluszy'nski, Michael S. P. Kelley, Nicholas Kern, Wolfgang E. Kerzendorf, Eric W. Koch, Shankar Kulumani, Antony Lee, Chun Ly, Zhiyuan Ma, Conor MacBride, Jakob M. Maljaars, Demitri Muna, N. A. Murphy, Henrik Norman, Richard O'Steen, Kyle A. Oman, Camilla Pacifici, Sergio Pascual, J. Pascual-Granado, Rohit R. Patil, Gabriel I. Perren, Timothy E. Pickering, Tanuj Rastogi, Benjamin R. Roulston, Daniel F. Ryan, Eli S. Rykoff, Jose Sabater, Parikshit Sakurikar, Jes'us Salgado, Aniket Sanghi, Nicholas Saunders, Volodymyr Savchenko, Ludwig Schwardt, Michael Seifert-Eckert, Albert Y. Shih, Anany Shrey Jain, Gyanendra Shukla, Jonathan Sick, Chris Simpson, Sudheesh Singanamalla, Leo P. Singer, Jaladh Singhal, Manodeep Sinha, Brigitta M. SipHocz, Lee R. Spitler, David Stansby, Ole Streicher, Jani Sumak, John D. Swinbank, Dan S. Taranu, Nikita Tewary, Grant R. Tremblay, Miguel de Val-Borro, Samuel J. Van Kooten, Zlatan Vasović, Shresth Verma, Jos'e Vin'icius de Miranda Cardoso, Peter K. G. Williams, Tom J. Wilson, Benjamin Winkel, W. M. Wood-Vasey, Rui Xue, Peter Yoachim, Chen Zhang, Andrea Zonca, and Astropy Project Contributors. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. , 935(2):167, August 2022.

- [36] Almar Klein, Sebastian Wallkötter, Steven Silvester, Anthony Rynes, actions user, Paul Müller, Juan Nunez-Iglesias, Mark Harfouche, Lukas Schrangl, Dennis, Antony Lee, Pandede, Matt McCormick, OrganicIrradiation, Arash Rai, Ariel Ladegaard, Hugo van Kemenade, Tim D. Smith, Ghislain Vaillant, jackwalker64, Joel Nises, Miloš Komarčević, rreilink, Chris Barnes, Zulko, Po-Chuan Hsieh, Niklas Rosenstein, Michał Górný, scivision, and Joe Singleton. imageio/imageio: v2.34.2, June 2024.

- [37] Robert Hanus. Time delay estimation of random signals using cross-correlation with hilbert transform. *Measurement*, 146:792–799, 2019.
- [38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [39] R.N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill series in electrical engineering. McGraw-Hill, 1986.
- [40] Eric W. Weisstein. "cross-correlation." from mathworld—a wolfram web resource. <https://mathworld.wolfram.com/Cross-Correlation.html>, 2024. [Accessed 04-04-2024].
- [41] L.R. Rabiner. *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Prentice-Hall, 1978.
- [42] Bradley M. Peterson, Ignaz Wanders, Keith Horne, Stefan Collier, Tal Alexander, Shai Kaspi, and Dan Maoz. On Uncertainties in Cross-Correlation Lags and the Reality of Wavelength-dependent Continuum Lags in Active Galactic Nuclei. , 110(748):660–670, June 1998.
- [43] Bradley Efron. *Bootstrap Methods: Another Look at the Jackknife*, pages 569–593. Springer New York, New York, NY, 1992.
- [44] P. Kajdič, O. Alexandrova, M. Maksimovic, C. Lacombe, and A. N. Fazakerley. Suprathermal Electron Strahl Widths in the Presence of Narrow-band Whistler Waves in the Solar Wind. , 833(2):172, December 2016.

- [45] Charles W. Smith and Bernard J. Vasquez. Driving and dissipation of solar-wind turbulence: What is the evidence? *Frontiers in Astronomy and Space Sciences*, 7, 2021.
- [46] Ian Kirker. UCL L<sup>A</sup>T<sub>E</sub>X thesis templates. <https://github.com/UCL/ucl-latex-thesis-templates>, 2014. [Accessed 20-07-2024].
- [47] Marc Baudoin. “Center figure that is wider than textwidth”. <https://tex.stackexchange.com/questions/16582/center-figure-that-is-wider-than-textwidth>, 2011. [Accessed 20-07-2024].
- [48] Ruth Johnston. *All Things Medieval: An Encyclopedia of the Medieval World*. 2019.