Autonomous on-board data processing and instrument calibration software for the Polarimetric and Helioseismic Imager on-board the Solar Orbiter mission

Kinga Albert[a,*] **Johann Hirzberger**[a] **Martin Kolleck**[a] **Nestor Albelo Jorge**[a] **Dennis Busse**[a] **Julian Blanco Rodríguez**[b] **Juan Pedro Cobos Carrascosa**[c] **Björn Fiethe**[d] **Achim Gandorfer**[a] **Dietmar Germerott**[a] **Yejun Guan**[d] **Lucas Guerrero**[a] **Pablo Gutierrez-Marques**[a] **David Hernández Expósito**[c] **Tobias Lange**[d] **Harald Michalik**[d] **David Orozco Suárez**[c] **Jesper Schou**[a] **Sami K. Solanki**[a,e] **José Carlos del Toro Iniesta**[c] **and Joachim Woch**[a]

[a]Max-Planck-Institut für Sonnensystemforschung, Göttingen, Germany
[b]Universidad de Valencia, Paterna, Valencia, Spain
[c]Instituto de Astrofísica de Andalucía—Consejo Superior de Investigaciones Científicas, Apartado, Granada, Spain
[d]Institute of Computer and Network Engineering at the Technical University Braunschweig, Braunschweig, Germany
[e]Kyung Hee University, School of Space Research, Yongin, Gyeonggi-Do, South Korea

Abstract. A frequent problem arising for deep space missions is the discrepancy between the amount of data desired to be transmitted to the ground and the available telemetry bandwidth. A part of these data consists of scientific observations, being complemented by calibration data to help remove instrumental effects. We present our solution for this discrepancy, implemented for the Polarimetric and Helioseismic Imager on-board the Solar Orbiter mission, the first solar spectropolarimeter in deep space. We implemented an on-board data reduction system that processes calibration data, applies them to the raw science observables, and derives science-ready physical parameters. This process reduces the raw data for a single measurement from 24 images to five, thus reducing the amount of downlinked data, and in addition, renders the transmission of the calibration data unnecessary. Both these on-board actions are completed autonomously.
 [DOI: 10.1117/1.JATIS.6.4.048004]

Keywords: spectropolarimetry; solar physics; space observatory; on-board data processing; data pipelines.

Paper 20063 received Jun. 5, 2020; accepted for publication Nov. 19, 2020; published online Dec. 18, 2020.

1 Introduction

Today's space missions are progressing in ambition and complexity, and the state-of-the-art instrumentation that they carry can produce vast amounts of data. This is especially true for remote sensing instruments, often producing multi-dimensional, high-resolution data products. In addition to that, the required precision and orbits with highly variable environments often require calibration data to be acquired on board as well. However, all raw observables and calibration data often cannot be transmitted to the ground due to low amounts of telemetry, especially in the case of deep space missions. These can, however, be processed on-board to result in calibrated, science-ready data that are more compact, hence increasing the science return of the mission. This processing often requires a high degree of autonomy, due to the limited telemetry and telecommand, which sometimes is also paired with long turnaround times.

*Address all correspondence to Kinga Albert, albert@mps.mpg.de

Solar Orbiter[1] (SO) is a mission for helioscience. It will follow unique, highly elliptical orbits around the Sun, with the closest approach at 0.28 astronomical units, and it will move out of the ecliptic plane during its lifetime to reach an inclination of 33° (including the extended mission phase). This provides a view of the polar regions of the Sun. To achieve the science goals, the spacecraft will carry a suite of four *in-situ* and six remote sensing instruments.

The Polarimetric and Helioseismic Imager[2] (SO/PHI) is part of the remote sensing package. It is a spectropolarimeter imaging the solar photosphere in the light of the Fe I 617.3-nm Zeeman sensitive absorption line. It takes images of the Sun in four polarization states of the light at six different wavelengths. Due to the Zeeman and Doppler effects, these observables carry information about the magnetic field vector and the flow velocities at the formation region of the spectral line in the solar atmosphere. Through the inversion of the radiative transfer equation (RTE), the magnetic field vector and the line of sight (LOS) velocity can be determined.[3]

SO/PHI is the first imaging spectropolarimeter to fly on a deep space mission, facing unprecedented challenges. It has an extremely low amount of guaranteed telemetry, it will see highly variable environment (especially large changes of temperature and radiation), and experience long command-response turnaround times. To cope with these restrictions, unprecedented for this type of instrument, SO/PHI implements autonomous on-board data reduction and autonomous on-board instrument calibration. The on-board data reduction consists of data pre-processing, i.e., the removal of instrumental effects and the inversion of the RTE[4–6]). The on-board instrument calibration includes both the characterization of the instrument (i.e., calculation of the flat and dark field) and the determination of the optimal operational parameters (e.g., integration time).

In the domain of on-board data processing, we most often find cases for data characterization: data are sorted into different categories, or they identify features, such as ice or clouds.[7–9] In these cases, the necessary accuracy is determined by the algorithms used, without the goal of producing data for further scientific analysis, posing a different set of challenges. There are few precedents for scientific on-board data analysis. The Active Magnetospheric Particle Tracer Explorers Ion Release Module 3D Plasma Instrument and the Giotto RPA Experiment, three decades ago, computed moments of the distribution function and calculated pitch angle distribution on-board to minimize telemetry.[10] The reduction was done in real time for both these instruments, with severe resource limitations compared to today's state of the art, therefore the efforts were mainly concentrated on meeting the timing requirements. The solar wind analyzer on-board the SO calculates the moments of particle velocity distribution functions on-board through look-up-tables and implements an intelligent telemetry management system to meet the limitations set by the mission.[11] Another instrument, more comparable to SO/PHI, also implementing on-board data processing, is the Michelson Doppler Imager on-board the Solar and Heliospheric Observatory.[12] Due to telemetry limitations, it performs some of the initial steps of the data reduction prior to data downlink. It uses arithmetic operations and look-up-tables to calculate part of the observables, complemented by additional processing and calibration on ground. Instrumental effects are entirely determined on ground and uploaded to the instrument to be used in the data processing.

Typical calibration of an imaging spectropolarimeter[13–15] is based on extensive ground measurements. However, the precise calibration parameters (e.g., gain tables, dark current levels, and instrumental polarization parameters) are expected to change during the lifetime of most instruments, therefore calibration data are collected regularly. The calibration images are then analyzed and processed by the scientists operating the instrument on ground. The highly elliptical orbits of the SO mission introduce changes in several of the instrumental properties due to their temperature dependence (especially strong for the dark and the flat fields) and brings challenges due to changing Doppler shift and image scales. These dynamic changes render ground measurements insufficient and requires SO/PHI to collect calibration data from orbit as close to observational conditions as possible. To avoid the download of these data, SO/PHI processes them on-board, autonomously.

SO/PHI's data processing software is a first in multiple aspects. It implements complete autonomous on-board processing for spectropolarimetric solar data from the instrument characterization to the calculation of the final science data products using the inversion of the RTE. These steps distribute their calculations between hardware and software and are integrated seamlessly into a software framework, which runs on the SO/PHI data processing unit (DPU) with

limited computational resources. As the system's objective is to reduce data volume and increase science return, intermediate data products in nominal operations will not be available. Therefore, the robustness of the software is essential.

2 The SO/PHI Instrument

To provide an overview of the SO/PHI instrument, we describe its working principle, the available hardware for data processing and its operation concept.

2.1 *Instrument Principle*

SO/PHI images the Sun with two different optical paths.[16] One of the paths images the full solar disk at any point along the orbit, called the full disk telescope [with a 2° field of view (FOV)], while the other collects data with high spatial resolution, named the high-resolution telescope (HRT, with a 0.28° FOV).[17] The optical path of the HRT is additionally equipped with an image stabilization system, consisting of a correlation tracker camera and a tip-tilt mirror. Each optical path has its polarization modulation package, containing nematic liquid crystal variable retarders and a linear polarizer, to transform the polarization signals into intensity levels.[18] Both paths scan the photospheric line through a common narrow-band tunable filter system.[2,19] Finally, they lead to the common focal plane assembly (FPA), where the images are recorded on 2048×2048 pixels by a custom-built active pixel sensor (also referred to as CMOS sensor). Both apertures penetrate the heat shield of the spacecraft and are protected by a heat rejection entrance window, which filters out all spectral components of the solar light outside of a 30-nm passband around the observed absorption line.

A science data set recorded by SO/PHI can be described by the following equation:

$$I_m^{obs\cdot}(\lambda,x,y) = \left[\left[c\sum_{p=1}^{4} M_{mp}(\lambda,x,y)S_p(\lambda,x,y)\right] * A_m(\lambda,x,y)\right] I_m^{\text{flat}}(\lambda,x,y) + I^{\text{dark}}(x,y), \quad (1)$$

where "*" stands for convolution. The indices m and p run over the four polarimetric modulation states, λ denotes the wavelength, and x and y are the spatial image coordinates in pixels. $\boldsymbol{I}^{obs\cdot}$ is the recorded data set, a four element vector of images at six different wavelengths, while $\boldsymbol{S}$ is a four element vector (Stokes vector) which describes the polarimetric state of the incoming sunlight[20] and which can be expressed as

$$\boldsymbol{S}(\lambda,x,y) = \begin{bmatrix} I(\lambda,x,y) \\ Q(\lambda,x,y) \\ U(\lambda,x,y) \\ V(\lambda,x,y) \end{bmatrix}.$$

$\boldsymbol{M}$ is the polarimetric modulation matrix, a 4×4 matrix of images, describing how the instrument transforms the polarization degrees of incoming sunlight into practically measurable light levels (intensities) for each pixel of its FOV, $\boldsymbol{A}$ are optical aberrations SO/PHI introduces, $\boldsymbol{I}^{\text{flat}}$ are the flat fields (gain tables) of the telescope, depending both on wavelength and polarization states, c is a constant gain factor that converts the pixels from number of photons accumulated on the detector to digital numbers (DN-s), and I^{dark} is the dark field of the sensor in DN-s, the same for all wavelengths and polarization states. For further details on solar spectropolarimetry see Ref. 3.

2.2 *Data Processing Hardware*

All data processing in SO/PHI is implemented in the DPU,[21–23] see Fig. 1. The DPU integrates a Leon-3FT microprocessor inside a GR712RC as central processing unit (CPU) which is a radiation-hardened processor by Gaisler, a Microsemi RTAX field-programmable gate array (FPGA) and two static random-access memory-based Xilinx Virtex-4 FPGAs, communicating through a

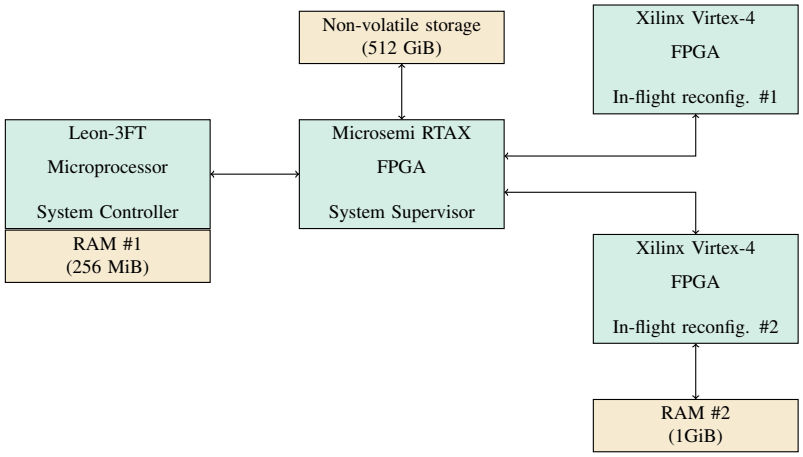

Fig. 1 The data processing of SO/PHI is performed on the DPU hardware. It runs distributed between a Leon-3FT microprocessor and two Xilinx Virtex-4 FPGAs. These processing units are aided by memories of different capacities. For long term data storage, a large capacity non-volatile memory is available.

SoCWire[24,25] network. The microprocessor is designated as the system controller, running the Real-Time Executive for Multiprocessor Systems[26] operating system, in version 4.10, implementing communications with the spacecraft, controlling image processing and memory transfers. The Microsemi FPGA is radiation hardened, and it implements the essential functions for the instrument (e.g., communication interfaces). It also acts as the system supervisor for the configuration of the reconfigurable Xilinx Virtex-4 FPGAs (RFPGAs) that are reconfigured dynamically during processing. The RFPGAs are used for data processing, compression, data accumulation and image stabilization in a time-sharing approach, saving volume, mass, and energy.

The memory budget of the DPU consists of memories aiding processing and non-volatile memory facilitating storage. A 256-MiB memory is attached to the system controller, of which 128 MiB are available to use for image processing. A 1-GiB fast synchronous dynamic random-access memory (SDRAM) supports the RFPGA designated for data pre-processing. A 512-GiB non-volatile NAND-Flash image data storage is available for storing the raw images awaiting processing, and the final products waiting for the last steps performed by the instrument: compression, packaging of telemetry packets, and transmission to the spacecraft platform.

2.3 *Instrument Operations*

SO implements an off-line commanding strategy. This strategy places the outline of commands and the establishment of final command parameters ahead of operations. They are defined in an iterative process, in a time frame of seven months to one week prior to their execution. Nominally, SO/PHI acquires data during three observation windows along one orbit, each of these windows lasting for 10 days. These windows are placed at special points of interests: closest approach and maximum and minimum solar latitude. There is a calibration campaign associated with each observation window, and a dedicated data processing campaign, taking place after the observations, and may last for the rest of the orbit. The operational constraints of the spacecraft require SO/PHI to anticipate its power, produced telemetry, electromagnetic compatibility, and time budget for all its operations. Consequently, all operations are planned and commanded from ground without autonomous decisions regarding the processing steps.

3 Requirements of the Data Processing System

The data processing system of SO/PHI is required to perform three different functionalities: process the raw science observables, calculate the calibration data from dedicated observations, and determine the optimal operational parameters for observations.

A typical science data processing pipeline for a spectropolarimeter is composed of pre-processing, the inversion of the RTE, and compression (see Fig. 2). The main aim of the pre-processing is to remove instrumental effects that appear in the recorded images [see Eq. (1)]. To achieve this in the simplest case, the following equations are applied:

$$I_m^{corr.}(\lambda, x, y) = [I_m^{obs.}(\lambda, x, y) - I^{\mathrm{dark}}(x, y)]/I^{\mathrm{flat}}(x, y), \tag{2}$$

$$S_p(\lambda, x, y) = \sum_{m=1}^{4} D_{pm}(x, y) I_m^{corr.}(\lambda, x, y), \tag{3}$$

where $\boldsymbol{D}$ is the demodulation matrix, expressed as

$$\boldsymbol{D}(\lambda, x, y) = \boldsymbol{M}(\lambda, x, y)^{-1}. \tag{4}$$

During pre-processing, we consider the term $\boldsymbol{A}$ (optical aberrations) negligible, and the dependence of $\boldsymbol{D}$ on λ, and the dependence of $\boldsymbol{I}^{\mathrm{flat}}$ on λ and p [see Eq. (1)]. The pipeline, however, must be able to correct the images without the simplifications as well, if found necessary in the future. This implies adding a number of optional steps. In addition, we know, based on experience that $\boldsymbol{D}$ usually deviates slightly from the real polarimetric behavior of the instrument. Therefore, further corrections are applied to remove the so-called cross-talks[3] (linear dependencies between the Stokes images). Furthermore, we may bin or crop the data sets to discard unused FOV (e.g., a full disk image not filling the entire detector) and to balance the downlink capabilities with the requirements of different science cases.

The implemented RTE inversion method is based on the Milne–Eddington approximation.[27] It is an iterative process, operating on all the 24 images at once, pixel by pixel. We shorten its run-time by estimating the starting conditions for the inversion through numerical calculations,

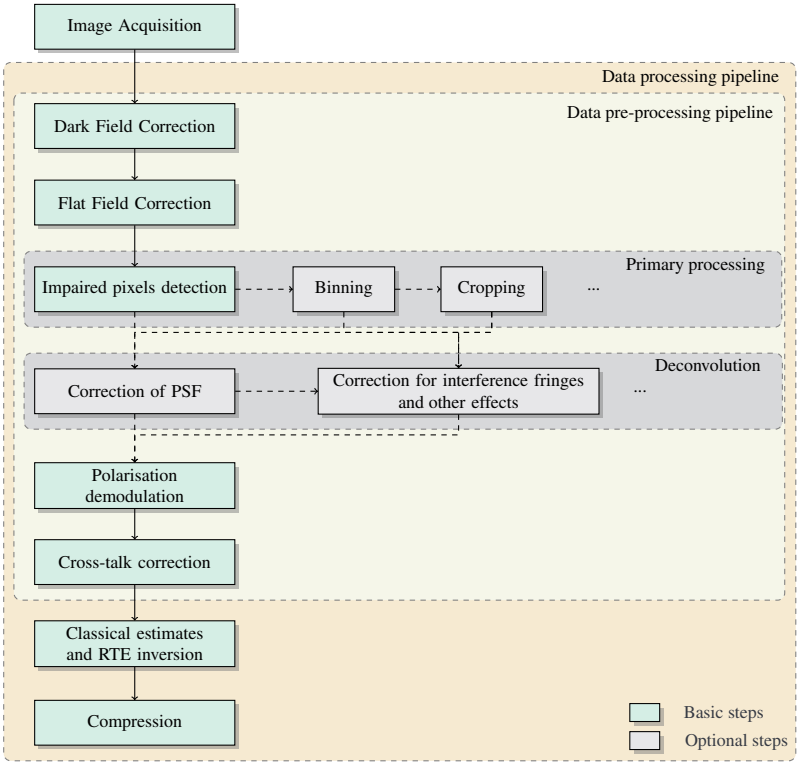

Fig. 2 The on-board science data processing on SO/PHI follows the typical ground processing used for spectropolarimeters. It consists of mandatory (basic, shaded green in the figure) and optional steps (shaded light gray). Some of the optional steps will be used in specific science cases, for others we decide during instrument commissioning whether they are necessary.

called classical estimates.[20,28,29] The final step is the compression of the resulting images with the Consultative Committee for Space Data Systems (CCSDS), CCSDS 122.0-B-1 algorithm, which only takes place before download.[30] This compression is only applied on the images, compression of metadata is possible with Zlib's[31] deflate algorithm on the already prepared telemetry packets. The saving of intermediate data products must be possible at certain steps of the processing. This is foreseen to be used extensively in the instrument commissioning phase to aid error search, and it adds opportunities for the extension and modification of processing for certain science cases (e.g., time averaging of pre-processed data sets, before the inversion of the RTE).

On-board instrument characterization involves dedicated observations, from which we derive the instrument characteristics. Some of these characteristics (e.g., the dark field) are computed regularly and determined on-board, while others are expected to be calculated more sporadically on ground and uploaded to the instrument (e.g., the optical point spread function). The on-board characterization process does not allow the interaction of scientists with the collected data, therefore it has to be autonomous.

The calculation of operational parameters requires immediate processing of the data after the observations in near real time. This is to calculate the starting point for a second iteration refining the results. They all require calculation of image qualifying parameters (e.g., contrast) for a series of images, and the methods have widely varying complexity.

The users of the system can be assigned to two categories: the scientists that describe the functionalities to be performed (e.g., defining what must be done to the science data before inversion) and the scientists operating the instrument (e.g., defining which pipeline to be executed on which data set and its parameters). A well-structured, simple definition of the processing functions is required to avoid user mistakes and to optimize on-board software updates. The main goals are describing the functions in an easily modifiable way, reducing code duplication, and creating standard error checks.

The computational demand of the performed functionalities is high due to the many image processing functions they contain. We shorten run-time by running the image processing functions in the RFPGAs. However, a back-up solution for the pre-processing portion of the pipeline, implemented in software is also required, excluding the RTE inversion and the CCSDS 122.0-B-1 compression (obtaining the physical parameters in the back-up solution would be possible with the classical estimates). The back-up solution with its significantly reduced implementation time aids the software development and testing and increases to the fault tolerance of the instrument by being able to take over these functionalities if necessary.

The number representation of the data has been chosen to be a fixed-point format during the processing, wherever possible. While the fixed-point format saves RFPGA resources, it comes with a significant overhead: all data must be scaled during the processing operations to avoid precision loss in decimals.

A metadata log associated with each data set must be created, recording the steps performed on the data set in full integrity. This is necessary both for scientific purposes and to offer a possibility for error search and improvement of the algorithms.

4 Implementation of the Data Processing System

To present the implemented system, we show its high-level structure and how a pipeline is constructed in the defined frame (Secs. 4.1 and 4.2). Later we describe the data scaling necessary to achieve the required scientific accuracy with the fixed-point representation in Sec. 4.3. Next, we show the implemented error handling (Sec. 4.4) and finally we describe the metadata logging system (Sec. 4.5).

4.1 *Software Architecture*

The data processing software is organized on three layers, see Fig. 3. The lowermost layer (primitives) implements the image processing functions, e.g., addition of images or Fourier transform of an image. These are implemented both as RFPGA functionalities[32] and as software functions running on the system controller microprocessor. Due to the large number of functions one

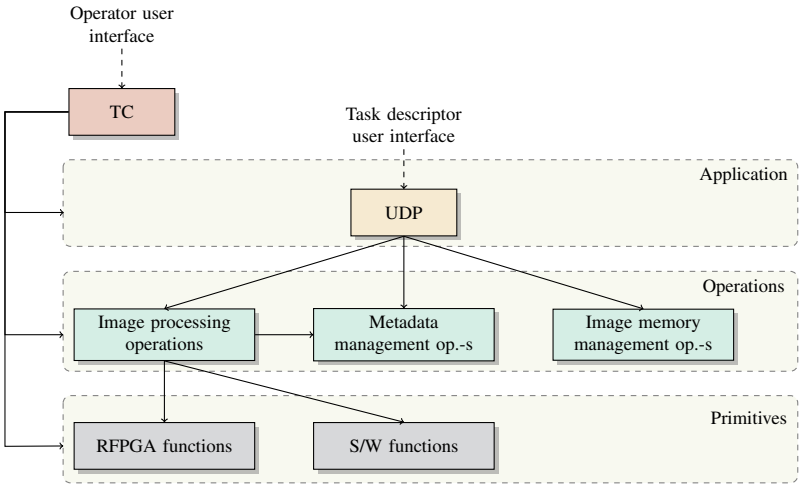

Fig. 3 The data processing software is organized on three layers. This organization facilitates information hiding, and the removal of responsibilities from the application developer (task descriptor).

RFPGA configuration is not sufficient, therefore the RFPGA is reconfigured on demand, during the processing. The necessity for reconfiguration is determined by the on-board software when there is a call to a function that is not available in the currently loaded configuration.

The second abstraction layer (operations) hides hardware details from the application and implements other lower-level functionalities. It integrates the primitives into image processing operations, hiding the hardware details. The image processing operations record metadata directly from this layer to improve metadata collection completeness by removing the full responsibility from the application developer. This layer also provides the functions for data transfer between different memories, and the functions for recording and storing metadata.

The application layer is where the processing functionalities are described, through user defined programs (UDPs). The UDPs are a special category of applications that are not compiled into the on-board software but handled by a UDP manager. Therefore, uploading a new UDP does not require a full on-board software exchange (on-board software is possible through UDP exchanges, full on-board software exchange, or FPGA configuration exchanges). The application layer further logs metadata, containing information that is only known on higher layers (e.g., the division of images performed is part of the flat fielding of the data set with a given ID).

We use three different languages in the architecture. On the primitives layer, the RFPGA firmware is described in Very High-Speed Integrated Circuit Hardware Description Language. The corresponding software functions in the same layer, and the operations layer is written in C. The application layer is written in On-board Command Language (OCL),[33] a high-level programming language based on C implemented for space instrumentation.

4.2 *Pipeline Construction*

To unify the approach to data processing, all of its three functionalities (science observables processing, calibration data calculation, and determination of operational parameters) are implemented in form of pipelines. We extend the definition of a pipeline to a series of changes done to a target (a data set, a number of data sets, or a subset of a data set), resulting either in parameters, or a new data set and their associated metadata. Observed data sets are available in the non-volatile data storage. The data sets are identified by a data set ID and each has a metadata file associated with them. They may contain multiple images, e.g., different wavelengths and polarization states in the case of a science observation, or different focal positions in case of a focusing observation. The pipelines are implemented in the application layer, and therefore are written in the form of UDPs. A block approach is adapted in its definition to achieve the required flexibility.

A pipeline block is defined as a unit that executes a number of functions on its input data, forming a logical unit and writes dedicated metadata. The abstraction level of a block is decided

from case to case. In respect to Fig. 2, some blocks are defined as much smaller functionalities that are performed more often, e.g., store data to non-volatile memory and can be added anywhere in the pipeline. In other cases, some of the separate blocks are combined, e.g., all Fourier space operations are performed in one block to avoid multiple Fourier transforms of the data set. To support their combination in multiple ways, a unified block interface is defined. Within a pipeline, blocks have a common target: a data set or part of a data set that is loaded into the processing memory. A block may load additional data for the processing step as needed (e.g., demodulation matrix), which is not written back to the non-volatile memory or passed to further blocks. Therefore all changes done to it are lost after the block. Due to the possibility of different combination of the blocks, the history of the data set inside each block must be determined, for which the interface is through the metadata file (e.g., if we cropped our data set in the first step, we also need to crop our dark field to the same area of the detector).

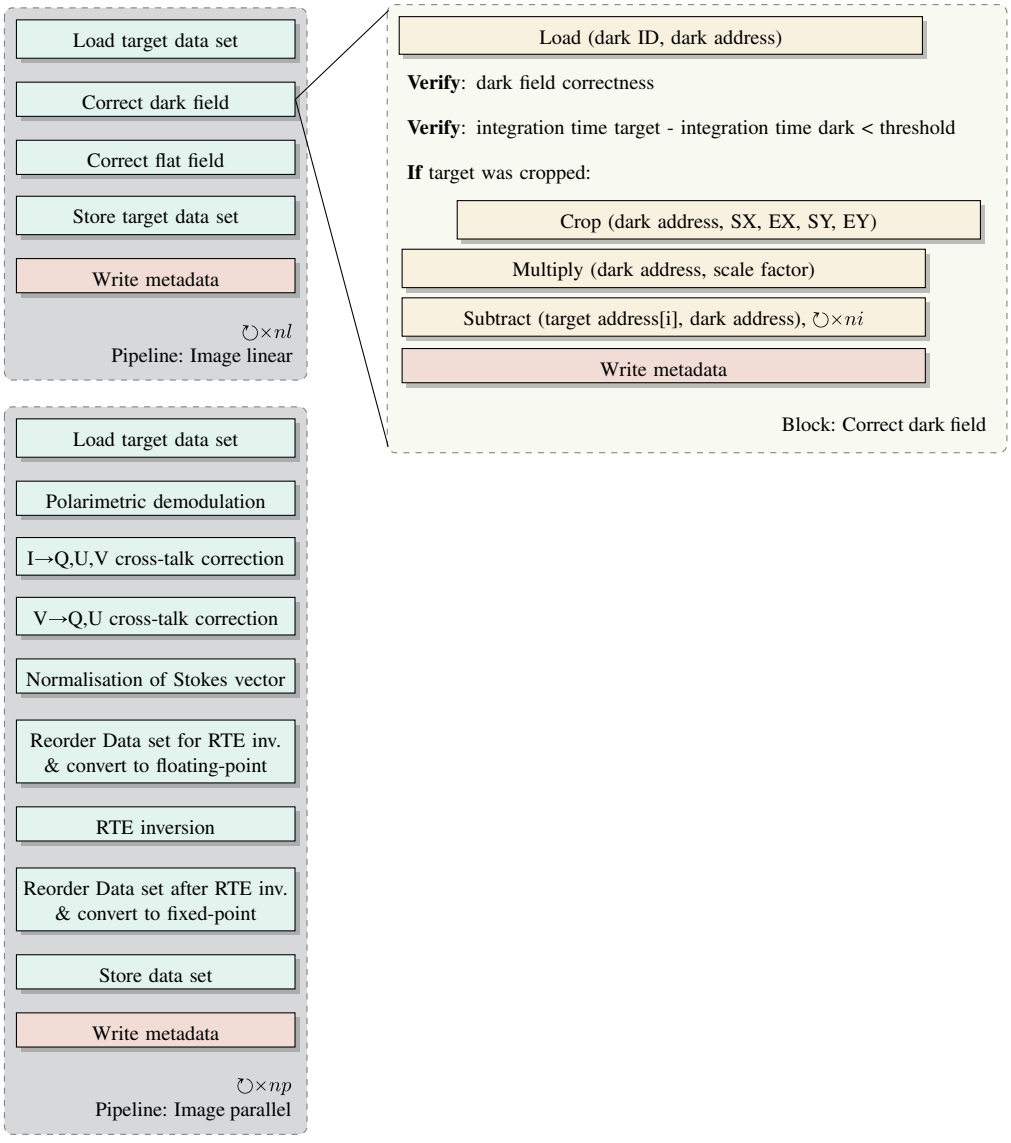

Fig. 4 Example science data processing pipeline. The pipeline is split into two parts to enable its execution with memory budget restriction. The image linear part processes a number of images at a time, the image parallel part processes a number of rows of all 24 images at the same time. The parameters *nl* and *np* are the number of times the image linear and image parallel portion of the pipeline has to run to process the whole data set, respectively. In the implementation detail of the dark field, *ni* denotes the number of images processed.

All pipelines are built by combining pipeline blocks into a processing sequence. They execute the blocks sequentially, i.e. continuing to a new block is possible only once the previous block is finished. Each pipeline also writes metadata specific to it and finally stores the metadata file into the non-volatile memory. The way pipelines are built is also specific to their memory needs and available memory on the two different platforms we run them on (RFPGA or only system controller). In some cases, this means running the sequence of blocks several times, with different subsets of the data set, and in some other cases to split a pipeline into parts, executed one after another.

To show how a pipeline is formed in the system described, we take an example for a science data processing pipeline (see Fig. 4). To cope with processing memory limitations, the pipeline is split into two parts: an image linear part, which processes a number of images of the data set at once, and into an image parallel part, which processes a number of rows of all images of the data set. This is necessary because we have operations that require the full image (e.g., a Fourier transform), and operations that require pixels from several images at once (e.g., polarimetric demodulation, RTE inversion). The two pipelines are executed nl and np times, respectively, both being equal to 1 in the nominal case, when we use the RFPGAs. We also show the implementation details of the dark field correction block, as an example.

4.3 *Data Scaling*

As a method for saving resources, a 24.8 fixed-point notation has been adapted for number representation during data processing, wherever possible (i.e., using a fixed number of 24 bits for the integer part, and 8 bits for the decimal part). Due to this, in all operations performed on the data, the accuracy must be optimized, considering the number of bits on which both the input and the result are represented. This varies from case to case; therefore, a uniform interface is defined for all pipeline blocks (target is scaled up to the most possible bits). All scaling necessary for maintaining accuracy in the process is performed inside the blocks, returning to the same scaling in the end. The obtained accuracy with this scheme is investigated in Ref. 34.

The single frames are obtained from the FPA in 12-bits digital depth, filling the detector well to a predefined level. The correct filling of the detector well is ensured by the exposure time calibration. A number of frames are accumulated into an image (the number of accumulations is determined by the required signal-to-noise ratio), then the image is shifted to the left by 8 bits, which are the decimal part of the numbers, all 0 at this time. Therefore, a raw data set is represented effectively on 12.8-bits digital depth, multiplied by the number of accumulations. This value is written into the metadata, and based on this, at the beginning of the pipeline we calculate the shift necessary to have all its images effectively represented on 23.8 bits (reserving a bit for sign). This scaling is already applied during the loading of the data set into the processing memory, ensuring the correct block interface.

There are three instances during the data processing, where the 24.8 fixed-point notation is abandoned: Fourier domain operations, the RTE inversion, and the compression. The Fourier domain operations are performed in floating-point, as the required accuracy could not be achieved by fixed-point implementation. The same is true for the RTE inversion. Both modules use IEEE 754 single-precision floating-point format. The reason for using 32 bits during the processing is to maximize the computational accuracy in the performed operations (e.g., divisions), however a 16-bit representation of the final results fulfils our requirements. Therefore, the images of the final result are represented on 16 bits, which is also the input to the compression module.

In the case of calibration data, the optimal scaling is ensured by the pipeline creating it. It is always represented on the fewest bits possible, while maintaining its required accuracy. The scaling is written into the metadata and read from there in the processing blocks that apply them to the science data set.

4.4 *Error Handling*

Error handling is done on all software layers. The guiding principle is to find errors on the lowest possible levels, to isolate them, and to aid error search in case of failures. Table 1 summarizes the detected error types.

Table 1 Error detection and handling on the different software organization layers. Each organization layer (see Fig. 3) implements error detection, with the objective of finding the errors as low in the hierarchy as possible. Based on the severity of the error, they are classified into Errors and Warnings, and different actions are taken at their detection.

Software layer	Scope	Handling	Action
Primitives	Detection of overflow and marking them as NaN-s	Warning	Continue
	Correct handling of NaN-s in operations	None	Continue
Operations	Correct memory addressing	Error	Abort
Application	Correct input parameters	Error	Abort
	Data matching (e.g., focus)	Warning	Continue
	Calibration data quality	Warning	Continue
	Operation errors (e.g., calibration method disturbed by solar scene)	Warning	Continue

On the layer of the Primitives, it is ensured that the image processing functions do not return overflows as valid numbers. This is done by replacing these pixels with a value defined as not a number (NaN), assigned to the lowest negative number on 32 bits (0×80000000, in two-s complement). On the same layer, it is also made sure that these values are not treated as numbers (e.g., division of a NaN with any number results in NaN). NaN-s may be replaced later through the interpolation of surrounding pixels; therefore, we keep track of all generated NaN-s in a bit mask image, along with other information regarding the data set (e.g., magnetic signal strength, pixels outside solar disk). This mask in the end is encoded into the final results of the pipeline through pixel values that could otherwise not appear (e.g., negative or NaN values) to obtain the information on ground without adding to the data volume.

In the operation layer, it is made sure that the inputs to the primitive functions are valid. For example, we check that we do not address any invalid memory positions by calculating the end address of a data set based on the start address and the size of the data.

The application layer is responsible for ensuring that the data sent to the lower layers is meaningful, e.g., that the target image has the same integration time as the dark field or that the flat field had no errors during its generation. It is also on the application layer that the parameters received in the pipeline are verified, e.g., that the target data set of the science data processing pipeline is raw data and contains the expected number of images. Additionally, also on this level, errors related to the solar scene are detected (e.g., when a flat field with the required precision cannot be obtained due to a sunspot in the FOV during the calibration window).

Each function has a return parameter, indicating warnings and errors detected by it. Warnings are small failures that do not affect the execution of the pipeline and are only recorded in the metadata of the data set for evaluation on ground. Errors are problems that make it impossible or meaningless for the pipeline to continue (e.g., no data set with the specified ID was found). In case of error, the execution is interrupted, the metadata is saved, and the instrument continues with the execution of the next command that was received from ground.

4.5 *Metadata Management*

The metadata of a data set comes from different sources.

- The planning process at which time we assign identifiers to the data sets acquired during the calibration campaign.
- The calibration campaign at which time we calculate values that are part of the calibration data.

- The instrument, recording all the current settings at the time of the data set acquisition.
- The data processing, recording all steps performed, their parameters, and return values.

The information from the planning process and calibration campaign is collected in a so-called Processing Environment, describing all information necessary for processing a science data set. This is written into the metadata of the data set at the time of its acquisition, also complemented by the instrument parameters. It is this metadata that is read and appended during the processing of the data set. Furthermore, we may override the Processing Environment written into the data set with the current one set on-board, if the processing plan changes later on.

The metadata generated during processing is created on multiple organization levels to ensure completeness. It is always recorded into metadata entries, each entry being part of one of the following categories.

- Operation entry, recorded by the operation layer functions.
- UDP entry, recorded on application layer, either by a pipeline block or a pipeline.
- Data summary entry, containing target parameters that are dynamic during the execution of the pipeline, recorded at the end of each UDP.

Each metadata entry starts with an ID, marking its category. In the case of the first two categories, the data that follows indicate the operation or UDP executed, its target, the input parameters, and return value. The data summary entry records the start and end indices of processed images within the target data set, and information about which part of the sensor the data belongs to, how it was binned and scaled, and data type and format. At the recording of each metadata entry a time-stamp is added automatically.

More information and an example on the usage of the metadata are detailed in Ref. 35.

5 Execution of a Science Data Processing Pipeline

To illustrate the operation of the data processing system, we run the pipeline presented in Fig. 4. The tests are performed on the flight spare model of SO/PHI. The target data set and all necessary calibration data are loaded into the non-volatile memory, as if they had been acquired previously by the instrument.

The interaction with the on-board software is through the Ground Support Equipment Operating System (GSEOS) software package.[36] GSEOS is used for all interactions with the instrument, modeling telecommands, and monitoring and displaying the values of the housekeeping telemetry packets.

5.1 *Input Data*

Data from the Helioseismic and Magnetic Imager on-board the Solar Dynamics Observatory (SDO/HMI)[12] is used to generate the input data set to the pipeline. SDO/HMI investigates the same absorption line, and the data are further processed with the SO/PHI Software siMulator (SOPHISM),[37] to produce a Stokes vector as similar as possible to one that SO/PHI would obtain, in 2048×2048 pixel resolution, see Fig. 5. There is one important difference, however, that the wavelength sampling of HMI is different, which is not taken into consideration during the pipeline execution. Therefore, the results are expected to contain some errors due to this approximation. Furthermore, the data set used is of a lower level, and neither the filter profiles nor the spacecraft velocity is removed, therefore, we expect an error in the v_{LOS}. These images are further manipulated to represent the raw observables, approximating Eq. (1), according to the following equation:

$$I_m^{\mathrm{input}}(\lambda, x, y) = \left[c \cdot c' \sum_{p=1}^{4} M_{mp}(x, y) S_p(\lambda, x, y)\right] I^{\mathrm{flat}}(x, y) + I^{\mathrm{dark}}(x, y), \tag{5}$$

where $\boldsymbol{S}$ is created with SOPHISM, and c' is the constant that scales the normalized images to represent the number of incident photons in 20-ms exposure time.

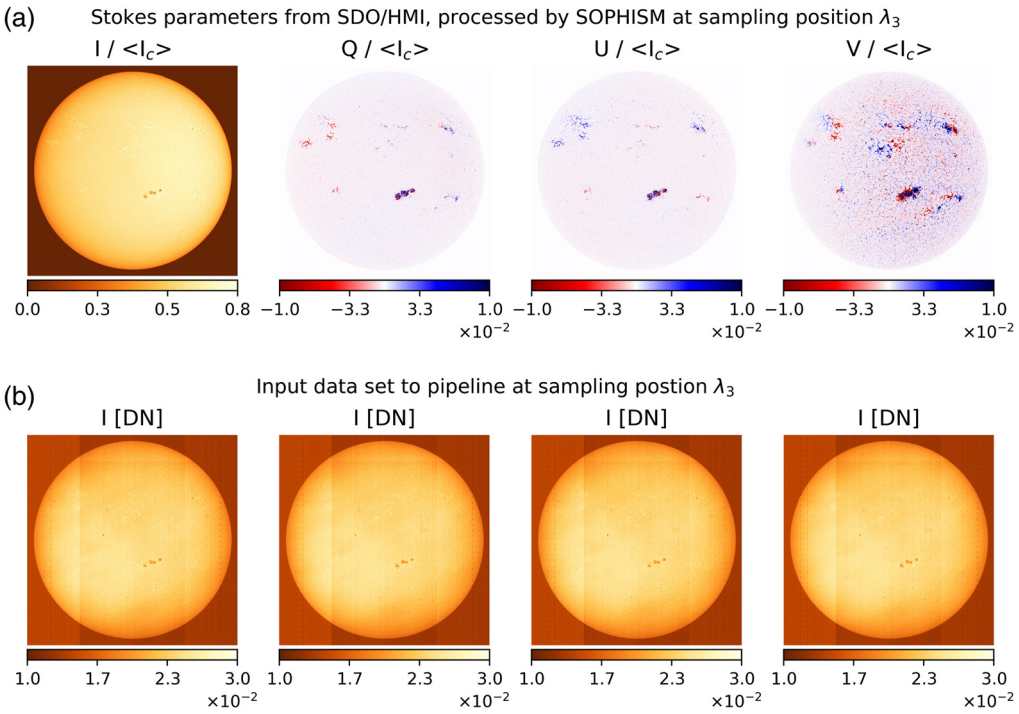

Fig. 5 Test data set at the third wavelength sample. (a) The input to the pipeline are the Stokes images from SO/PHI Software siMulator (SOPHISM).[37] (b) The input images are obtained applying Eq. (5) and show similar light levels in all polarization states as a result of the modulation. The data also show the effect of the dark field (stripes) and the flat field (smudges and dust grains).

The calibration data applied is recorded in the laboratory, using the flight model of SO/PHI, and is shown in Fig. 6. The dark field, acquired in a dark chamber, shows the characteristic sensor pattern: the four distinct horizontal stripes, created using four different channels for image read-out. The flat field has been recorded with the use of a lamp to ensure the most uniform illumination possible. It shows a gradient pattern and a number of dust grains inside the instrument. It has been normalized to its mean intensity, then scaled to be represented on 10 bits total, translating to 2 bits integer and 8 bits decimal part in the fixed-point notation. The demodulation matrix has been determined during the ground calibration campaign with the use of a polarization calibration unit (described in Ref. 38). Its field dependence varies with a standard deviation between $5 \cdot 10^{-9}$ (for $\mathbf{D}_{1,3}$) and 1.12 (for $\mathbf{D}_{4,1}$).

5.2 *Output Data*

We execute the full pipeline on the data set in the nominal configuration, processing all images at the same time. As we do not induce any cross-talk effects into the test data, we set the corresponding parameters to 0, and in consequence, the two cross-talk correction blocks from Fig. 4 do not execute. In this test, we configure the RTE inversion to determine all possible output images it can provide. In regular operations, we set the output of the inversion to be only the four images of interest ($\boldsymbol{B} = (|\boldsymbol{B}|, \gamma, \phi)$, and v_{LOS}), however, the inversion module is able to return nine parameters in total. For comparison, we also execute the image linear part of the pipeline in back-up configuration, processing four images at a time. This run creates six different output data sets, each containing a subset of the full result.

The result of the pre-processing [see Fig. 7, calculated according to Eqs. (2) and (3)] is $\boldsymbol{S}$, which will be the input to the RTE inversion. After the dark- and flat-field correction, the solar scene is undisturbed by instrumental artefacts. In areas of low illumination levels (e.g., dust grains on the sensor, or areas masked by a field stop), NaN-s are produced during the flat-field

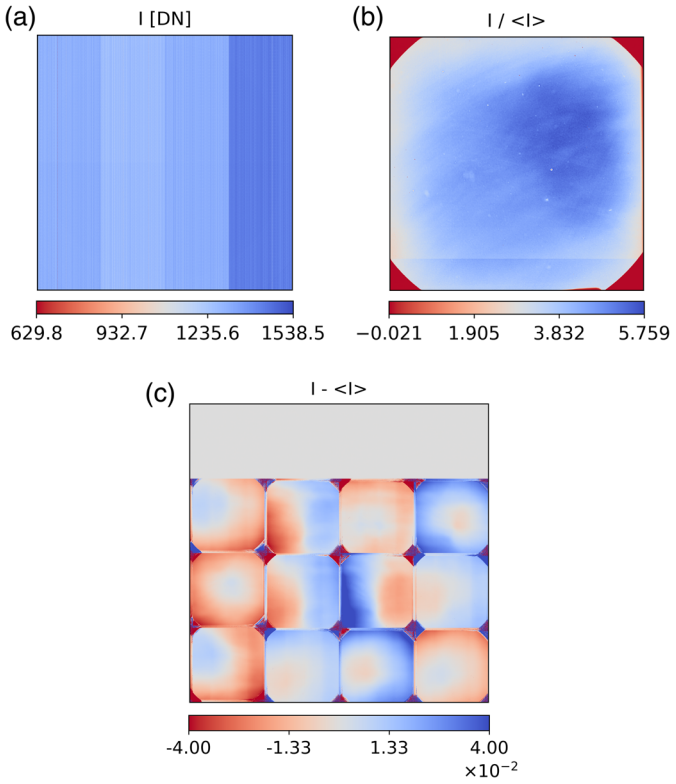

Fig. 6 The calibration data used in the test run were recorded in the laboratory. (a) The dark field shows the characteristic sensor pattern: four distinct vertical stripes due to the four different readout channels. (b) The flat field used in the tests shows a gradient across the FOV, and a few dust-grains. (c) The 4×4 demodulation matrix, $\boldsymbol{D}$, depends on the FOV. To show this dependence in the plot, we subtract from each of the elements their spatial mean.

correction through division by zero. In an ideal flat field, all pixels located behind the field stop would be 0, resulting in an area of uniform NaN-s in the corners after the division. However, due to the imperfection of the dark-field correction, some pixels reach sufficiently large values to not produce a NaN. The Stokes vector, obtained after the polarimetric demodulation, differs slightly from $\boldsymbol{S}$ in Eq. (5) due to the limited numerical accuracy. These images already show the presence of the magnetic field, which will be quantified by the inversion of the RTE.

The final results of the pipeline are the continuum intensity and the results of the RTE inversion: the magnetic field vector $\boldsymbol{B} = (|\boldsymbol{B}|, \gamma, \phi)$, described through its magnitude, azimuth and inclination, and v_{LOS} (see Fig. 8). All values outside of the solar disk are meaningless for the RTE inversion, therefore, it cannot converge, resulting in noise as output. The $|\boldsymbol{B}|$ is stronger in the umbra of the active region and weaker in the penumbra as expected. In γ, we can see the opposing polarities that compose the active region, while ϕ shows the fan-like pattern, originating in the center of the round magnetic features, consistent with the orientation of the magnetic field in such features. The v_{LOS} over the full disk shows the spectral shift due to the rotation of the Sun and the spacecraft velocity, on top of the intrinsic shifts due to plasma motion. The slight bias to one side is due to the lack of filter profile corrections in the used SDO/HMI data set. The v_{LOS} in the active region shows strong flows of opposing directions in the penumbra, known as Evershed flows,[39] driven by the magnetic activity. We analyze the accuracy of the pipeline in Ref. 34.

On ground, the image data and associated metadata are converted into files according to the Flexible Image Transport System.[40] While the metadata on-board is written into a continuous unit, the converter separates the header entries into separate American Standard Code for Information Interchange tables based on their origin. The metadata associated with the processing, recorded from UDP level, reflects the steps taken by the pipeline, its parametrization, and shows the success of the blocks (see Tables 2 and 3). Based on this table, we also have an

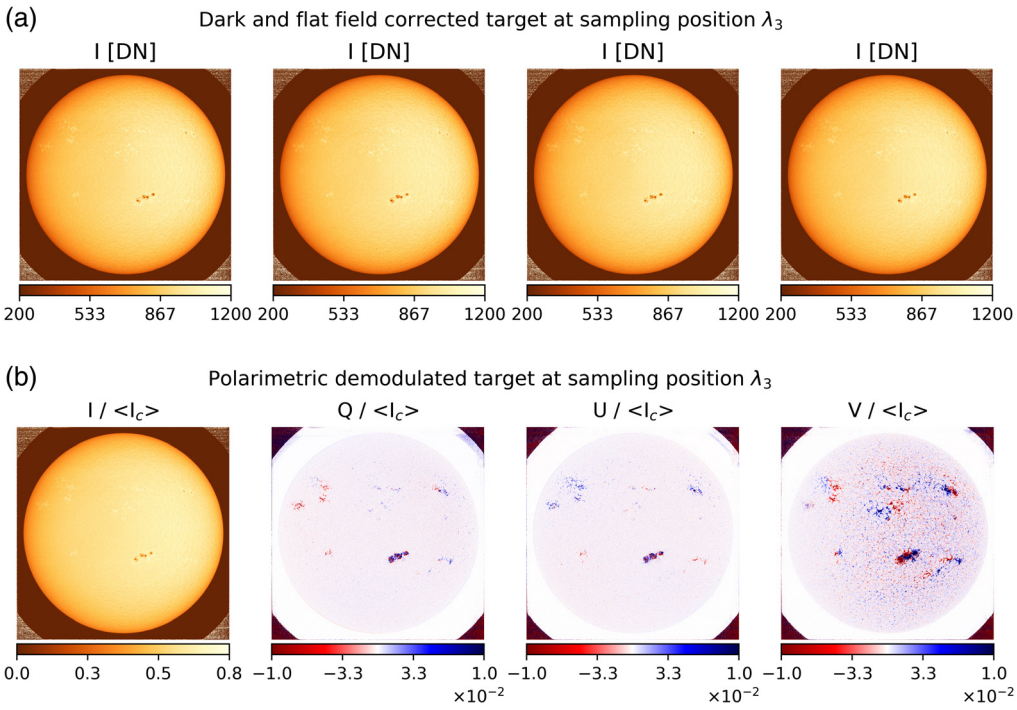

Fig. 7 Step-wise results of the pre-processing on the input data set from Fig. 5. (a) The third wavelength sample in the four different polarization states (4 out of 24 images) after dark- and flat-field correction of $I_\lambda^{\text{recorded}}(x, y)$, see Eq. (2). The white pixels in the image corners are NaN-s produced during the processing through division by sufficiently small numbers to produce an overflow. (b) Result of the pre-processing, the third wavelength sample of the normalized Stokes vector (4 out of 24 images), see Eq. (3).

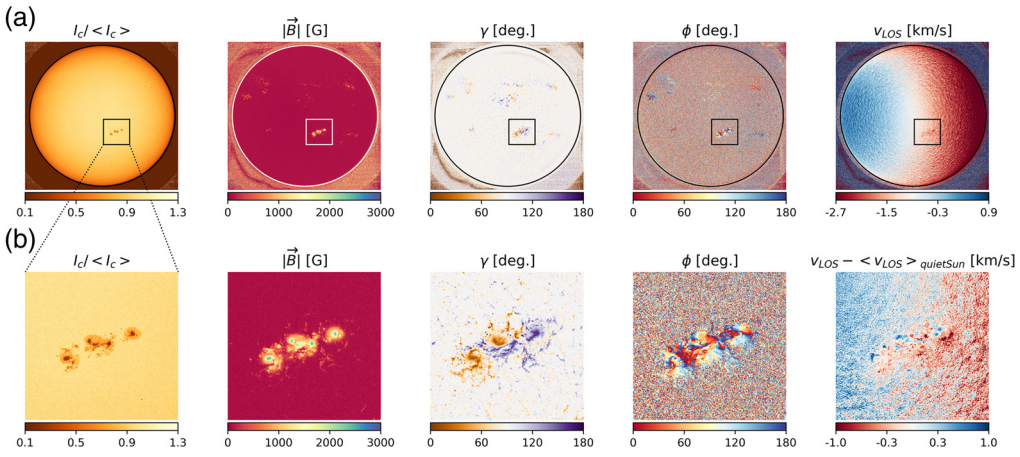

Fig. 8 (a) The five resulting images from the full pipeline, continuum intensity (I_c), the components of the magnetic field vector $\boldsymbol{B} = (|\boldsymbol{B}|, \gamma, \phi)$, and the LOS velocity, v_{LOS}. v_{LOS} is plotted on a scale centered around the spacecraft velocity, −0.9 km/s ($\langle v_{\text{LOS}} \rangle$ at disk center). The circular contour marks the solar limb, and the square shows the region magnified in the next row. (b) Detail of the full FOV, showing the active region. The v_{LOS} was corrected for the mean quiet Sun intensity in this region.

overview of any warnings or failures during the execution. The metadata recorded during the dark fielding block from operation level reflects the implementation shown in Fig. 4, giving an insight into the lower-level information that we can access (see Table 4). See Ref. 35 for more detail on the usage of metadata.

Table 2 Excerpt of metadata recorded from UDP level, during nominal execution of the pipeline, showing all steps taken, and their parameters. In cases where we have more parameters than what can be recorded in one entry, several entries are added by the same UDP. The timestamp shows the on-board time during the execution. The return value shows success in most cases (0), with some steps creating NaN pixels in the result (10). The start and end indices show that the full data set is processed at once, and how the number of images changes during the processing through extension with the mask, the inversion, and the encoding of the mask into the I_c image.

	Timestamp											
Explan.[a]	December 20, 2019	Re.[b]	UDP[c]	Data ID	In 1[d]	In 2	S #[e]	E #[f]	S row[g]	E row	S col.[h]	E col.
Load[i]	13:45:21	0	1351	90030	512	0	0	23	0	2047	0	2047
Dark c.[j]	13:45:28	0	1363	90070	8388608	0	0	24	0	2047	0	2047
Flat c.[k]	13:46:07	10	1365	90080	4	0	0	24	0	2047	0	2047
Lin. p.[l]	13:46:38	0	7002	90030	90070	90080	0	24	0	2047	0	2047
Lin. p.	13:46:38	0	7002	90030	0	24	0	24	0	2047	0	2047
Lin. p.	13:46:38	0	7002	90030	24	65535	0	24	0	2047	0	2047
Load	13:52:44	0	1351	92030	1	0	0	24	0	2047	0	2047
Demod.[m]	13:55:06	10	1366	90060	512	0	0	24	0	2047	0	2047
Norm.[n]	13:55:42	10	1371	0	49868192	2048	0	24	0	2047	0	2047
NaN en.[o]	13:55:55	0	1356	0	0	0	0	24	0	2047	0	2047
Reord.[p]	13:56:18	0	1372	0	0	0	0	24	0	2047	0	2047
Inv.[q]	15:05:06	0	1374	0	511	1	0	10	0	2047	0	2047
Reord.	15:05:28	0	1375	0	0	0	0	9	0	2047	0	2047
Par. p.[r]	15:05:36	0	7003	92030	90060	25	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	2048	5	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	−3803904	1779456	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	73216	4096	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	−256	−445	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	128	49868192	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	2048	511	0	9	0	2047	0	2047
Par. p.	15:05:36	0	7003	92030	1	115	0	9	0	2047	0	2047

Note: UDP identifiers: 1351, Load data set; 1363, Subtract dark field; 1365, Divide by flat field; 1366, Polarimetric demodulation; 1371, Stokes vector normalization; 1356, NaN encoding; 1372, Reorder images for RTE inversion; 1374, RTE inversion; 1375, Reorder images after RTE inversion; 7002, Image linear pipeline; 7003, Image parallel pipeline. *Data set ID-s*: ID 90030, the science data; $\boldsymbol{I}_{\lambda}^{\text{input}}(x, y)$; ID 90070, $I^{\text{Dark}}(x, y)$; ID 90080, $I^{\text{Flat}}(x, y)$; ID 90060, demodulation matrix; $\boldsymbol{D}(x, y)$.

[a]Additional explanation (not part of metadata).
[b]Return value.
[c]UDP identifiers.
[d]First input parameter (var. content).
[e]Start index.
[f]End index.
[g]Start row.
[h]Start column.
[i]Load data set.
[j]Dark field correction.
[k]Flat field correction.
[l]Linear pipeline.
[m]Polarimetric demodulation.
[n]Stokes vector normalization.
[o]NaN encoding.
[p]Image reordering (before and after RTE inversion).
[q]RTE inversion.
[r]Parallel pipeline.

Table 3 Excerpt of metadata recorded from UDP level, during the execution of the linear part of the pipeline in back-up configuration. In this configuration we process four images at a time and obtain six output data sets. Here we show the metadata of the first one of the six. The entries indicate that the first four images have been loaded (indices 0 to 3) with full FOV (0 to 2047), extended during the processing with the mask image, index 4.

	Timestamp											
Explan.[a]	December 21, 2019	Re.[b]	UDP[c]	Data ID	In 1[d]	In 2	S #[e]	E #[f]	S row[g]	E row	S col.[h]	E col.
Load[i]	11:15:52.0000	0	1351	90010	512	0	0	3	0	2047	0	2047
Dark c.[j]	11:18:10.0000	0	1363	90070	8388608	0	0	4	0	2047	0	2047
Flat c.[k]	11:21:39.0000	0	1365	90080	4	0	0	4	0	2047	0	2047
Lin. p.[l]	11:23:14.0000	0	7002	90010	90070	90080	0	4	0	2047	0	2047
Lin. p.	11:23:14.0000	0	7002	90010	0	24	0	4	0	2047	0	2047
Lin. p.	11:23:14.0000	0	7002	90010	4	65535	0	4	0	2047	0	2047

Note: UDP identifiers: 1351, Load data set; 1363, Subtract dark field; 1365, Divide by flat field; 7002, Image linear pipeline; Data set ID-s: ID 90030, the science data; $I_{\lambda}^{\text{input}}(x,y)$, ID 90070, $I^{\text{Dark}}(x,y)$, and ID 90080, $I^{\text{Flat}}(x,y)$.
[a]Additional explanation (not part of metadata).
[b]Return value.
[c]UDP identifiers.
[d]First input parameter (var. content).
[e]Start index.
[f]End index.
[g]Start row.
[h]Start column.
[i]Load data set.
[j]Dark field correction.
[k]Flat field correction.
[l]Linear pipeline.

5.3 *System Metrics*

The CPU load during the processing reflects the pipeline implementation and the underlying on-board software (see Figs. 9 and 10). These profiles are dominated by the memory transfers and the image processing operations, together with the necessary RFPGA related actions. Whenever reading the non-volatile memory, the DPU used 100% due to a background task that polls the queue of the non-volatile storage to transmit the next data chunks. As soon as everything is in the queue, the load drops to a lower level, until the transfer is finished. In case of writing to the non-volatile memory, the system controller is in full control of the data transfer, eliminating the need for synchronization. Therefore, it can write at 100% capacity, which is also the reason that it takes shorter time than reading. This is the reason for the nominal configuration to show 100% load at memory transfers between the non-volatile storage and RAM #2. Transferring from non-volatile memory to RAM #1 shows a different load profile due to the different implementation of the data transfer. In this case, the non-volatile memory controller waits for a confirmation from the system controller before transmitting a new chunk, resulting in a slower transfer (the case of the back-up configuration). During RFPGA configurations, the CPU load is high (∼95%), creating peaks at each reconfiguration. The system controller runs the image processing pipeline in parallel with a scrubbing task, that uses all idle time to rewrite the RFPGA configurations part by part to defend against radiation effects. This keeps the CPU load high even during times when a low load would be expected otherwise (e.g., during the RTE inversion, performed by the RFPGA, the combination of the scrubbing and the memory transfers keeps the load high). In the case of the back-up configuration, image processing functions run on the DPU, and use its full capacity.

The run-time of the pipeline is significantly shorter in the nominal configuration than in back-up mode. We measure it by extracting the number of processor ticks occurred during the

Table 4 Excerpt of metadata recorded on operation level by the dark field subtraction block, during nominal execution of the pipeline, see Fig. 4. The return values all show success (0). The processing memory addresses show that the dark field is loaded to the address 419430400, scaled to match the target in the same location, then it is subtracted from each image of the target data set (loaded to address 0). There are 24 entries for subtraction in total (four shown here), each recording the increment of addresses from image to image in the target data set. The scalar input values show the dark-field ID and the scale factor.

	Timestamp									
Explan.[a]	December 20, 2019	Operation	Re.[b]	Addr. 1[c]	Addr. 2	Addr. 3	Im. row	Im. col.[d]	Sc. 1[e]	Sc. 2
Load[f]	13:45:24	PROC_LOAD	0	0	0	419430400	0	0	0	90070
Scale[g]	13:45:24	PROC_F_MUL_S	0	419430400	0	419430400	2048	2048	131072	0
Sub. 1[h]	13:45:24	PROC_F_SUB_I	0	0	419430400	0	2048	2048	0	0
Sub. 2	13:45:24	PROC_F_SUB_I	0	16777216	419430400	16777216	2048	2048	0	0
⋮										
Sub. 23	13:45:28	PROC_F_SUB_I	0	369098752	419430400	369098752	2048	2048	0	0
Sub. 24	13:45:28	PROC_F_SUB_I	0	385875968	419430400	385875968	2048	2048	0	0

Note: Operations: PROC_LOAD, loading from volatile memory to processing memory; PROC_F_MUL_S, multiplication by scalar; PROC_F_SUB_I, subtraction of images.
[a]Additional explanation (not part of metadata).
[b]Return value.
[c]Memory address 1.
[d]Image columns
[e]Scalar input 1.
[f]Load dark field.
[g]Scale dark field to match target data set.
[h]Subtract dark field from image 1 of data set.

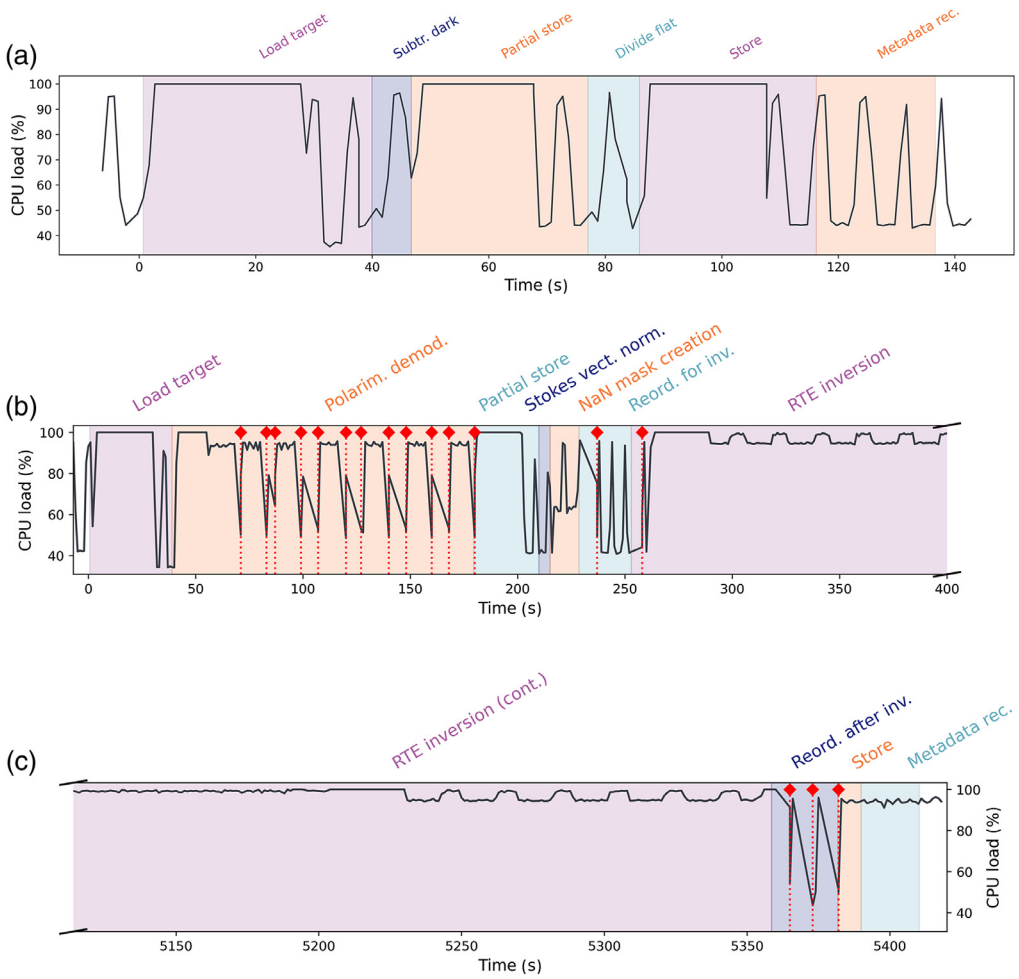

Fig. 9 CPU load during the execution of the (a) image linear and (b, c) image parallel pipelines in nominal configuration. The colours show the execution of different pipeline blocks (see Fig. 4). The red dotted lines with diamonds indicate the times of RFPGA reconfigurations. During the polarimetric demodulation, there are 12 reconfigurations, which will be entirely removed in the future. The full pipeline takes little over 1.5 h to complete.

execution, where one tick corresponds to 2.5 ms. To execute the image linear pipeline in nominal mode takes just over 2 min, while the image parallel pipeline takes a little over 90 min. The majority of the processing time is made up by RTE inversion block, the duration of which varies based on the science mode it operates in. The configuration used during this test writes all nine possible outputs, while if we choose only the four results of interest (as aimed to be configured in nominal mission phase), it runs ∼30 min faster. Moreover, the desired implementation of the inversion is to read the data from the RAM #2 and write the results to the non-volatile memory, from where they will be read back to the RAM #2 to continue their processing. However, the current implementation writes the data in chunks from the RAM #2 to RAM #1, the RTE inversion module reads the input from here and writes the results to the same memory, which are then read back to the RAM #2. This implementation introduces additional data transfer operations of the input data, transfers between different memories which have longer transfer time and prohibits parallel read and write operations which would be possible if the input and output were transferred between different memories. In this setup, the time of the inversion itself (together with reading and writing RAM #1) is 71.41 min, the time of the input data transfer from RAM #2 to RAM #1 is 10.88 min, while the time of the output data transfer from RAM #1 to RAM #2 is 2.7 min.

The RFPGA configurations are optimized to reduce the execution time by minimizing the number of reconfigurations during science data processing (each reconfiguration takes between

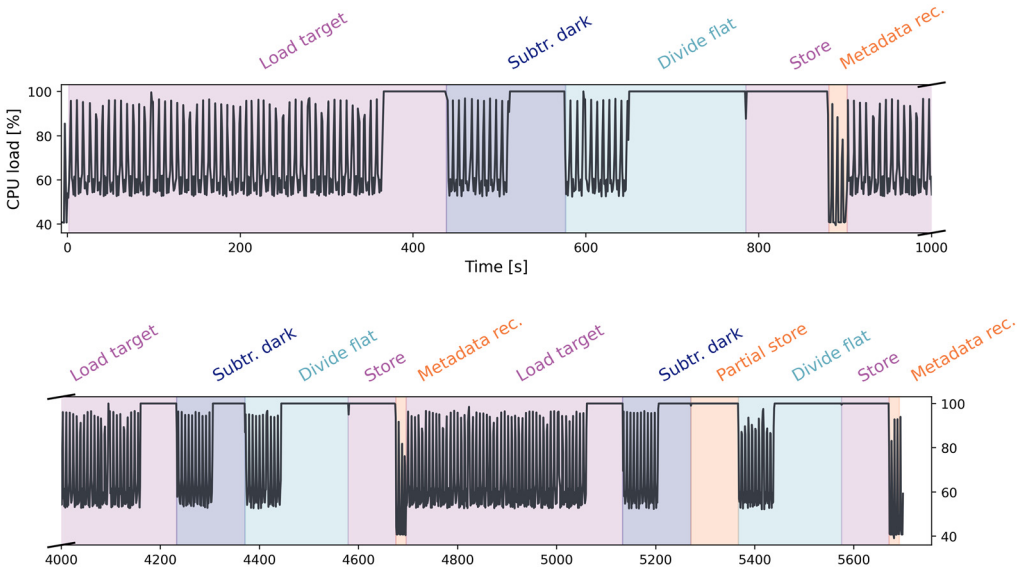

Fig. 10 CPU load during the image linear pipeline execution in back-up configuration. The colours indicate the execution of different pipeline blocks (see Fig. 4). In back-up mode, we only process four images of the full data set at a time, due to the size limitation of RAM #1. Therefore, the pipeline executes a loop, repeating the block sequence "Load target, Subtr. dark, Divide flat, and Store, Metadata rec." six times in total (only three shown). The image linear part runs in a little over 1.5 h in back-up mode, a significant time increase compared to nominal mode.

2 and 4 s). There is none necessary during the linear pipeline, while in the parallel pipeline, there are five reconfigurations outside of the polarimetric demodulation block, which has twelve. This latter block requires such a high number, because the matrix multiplication is in a separate configuration. However, this operation will be re-implemented in the future as a UDP, calling other low-level operations, eliminating all reconfigurations in the polarimetric demodulation block. Time wise, this will save the reconfiguration times (12 times 3 s on average), this final implementation has been measured to execute in 92 s for a full data set.[41]

Running the full pipeline in the nominal mode with all nine possible outputs, on the full FOV, requires a little over 1.5 h. This time is reduced to close to 1 h when we store only the four outputs of interest: the three components of the magnetic field vector and the flow velocities. This is the time required for processing each of the acquired data sets, which may amount to several hundred in each orbit. This time may further be reduced by cropping and binning, depending on science case.

In contrast, the linear pipeline runs in over 90 min in back-up configuration. This increase originates both from the longer execution time of the primitive functions and the longer memory transfer times (between the non-volatile memory and RAM #1). This large increase indicates that the execution of the full pre-processing in back-up configuration would have a significant influence on the science operations.

6 Summary and Conclusions

We have detailed the implementation of the on-board data processing system of SO/PHI: the requirements and the different parts that implement them. To show the synergy of its components, we defined and ran a science data processing pipeline. The partial and final results are as expected from the input data set, reducing the 24 raw images into five images of scientific interest. The recorded metadata show all steps performed on-board, which is used both on-board as integral part of the system, and on ground: for verification, possible error search and during scientific analysis.

We also showed system metrics for a more detailed view of the system, and we assess the time frame in which different parts of the pipeline run. The processing of a full data set takes a

little over 1.5 h, the majority of the time being used for the inversion of the RTE. Several points of improvement have also been pointed out, planned for the next version of the on-board software.

The main challenge that SO/PHI's on-board data reduction system faces is to implement complex scientific data analysis in the lack of human interaction. The most important tools in overcoming this challenge are the metadata- and the error handling systems. Moreover, it was necessary to adapt algorithms that are routinely ran on computer clusters on ground, to a space-qualified computing system. This is achieved by the custom hardware design, the use of distributed processing between a CPU and RFPGA-s, the custom firmware that the RFPGA-s use, and the software architecture. Due to the limitations, we use fixed-point number representation wherever possible, which has proven to require significant effort to maintain the required accuracy during scientific processing.

While the lower layers of the software are hardware specific, the two highest layers can be embedded into different systems. We are currently working on integrating it with our instrument simulator, SOPHISM[37] to enable the running of our pipelines on a desktop. Furthermore, the flexibility of the pipeline definition promises the possibility of using the system for future missions.

This data processing system takes the full science data reduction of a solar spectropolarimeter for the first time on-board the spacecraft. It enables new avenues for future missions with challenging orbits that can provide new points of views, at the cost of reduced telemetry volumes.

Acknowledgments

This work was carried out in the framework of the International Max Planck Research School for Solar System Science at the Max Planck Institute for Solar System Research. Solar Orbiter is a mission led by the European Space Agency with contribution from the National Aeronautics and Space Administration (NASA). The Polarimetric and Helioseismic Imager instrument is supported by the German Aerospace Center (DLR) under grant Nos. 50 OT 1201 and 50 OT 1901. The Spanish contribution has been partly funded by the Spanish Research Agency under projects under grant Nos. ESP2016-77548-C5 and RTI2018-096886-B-C5, partially including European FEDER funds. IAA-CSIC members acknowledge and funds from the Spanish Ministry of Science and Innovation "Centro de Excelencia Severo Ochoa" Program under grant No. SEV-2017-0709. The solar data used in the tests are the courtesy of NASA/SDO HMI science team. Parts of the work shown in this paper have been introduced at the SPIE Astronomical Telescopes + Instrumentation conference.[42]

References

1. D. Müller et al., "The Solar Orbiter mission. Science overview," *Astron. Astrophys.* **642**, A1 (2020).
2. S. K. Solanki et al., "The polarimetric and Helioseismic imager on Solar Orbiter," *Astron. Astrophys.* **642**, A11 (2020).
3. J. C. del Toro Iniesta, *Introduction to Spectropolarimetry*, Cambridge University Press, Cambridge (2003).
4. J. P. C. Carrascosa et al., "The RTE inversion on FPGA aboard the Solar Orbiter PHI instrument," *Proc. SPIE* **9913**, 991342 (2016).
5. J. P. C. Carrascosa et al., "Scientific computing and fault mitigation on FPGA aboard the Solar Orbiter PHI instrument," in *NASA/ESA Conf. Adapt. Hardware and Syst.*, IEEE, pp. 1–8 (2015).
6. J. P. C. Carrascosa et al., "SIMD architecture on FPGA for scientific computing aboard a space instrument," *J. Syst. Archit.* **62**, 1–11 (2016).
7. R. Castano et al., "Oasis: onboard autonomous science investigation system for opportunistic rover science," *J. Field Rob.* **24**(5), 379–397 (2007).
8. M. Neugebauer et al., "Genesis on-board determination of the solar wind flow regime," *Space Sci. Rev.* **105**, 661–679 (2003).

9. G. Rabideau et al., "Mission operations of earth observing-1 with onboard autonomy," in *2nd IEEE Int. Conf. Space Mission Challenges Inf. Technol.*, Vol. 7, p. 373 (2006).
10. D. W. Curtis et al., "On-board data analysis techniques for space plasma particle instruments," *Rev. Sci. Instrum.* **60**, 372–380 (1989).
11. L. Amoruso et al., "Solar wind analyzer-the Solar Orbiter milestone towards on-board intelligent decision making systems," in *Proc. Int. Astronaut. Cong.*, p. A7, The International Astronautical Federation (IAF) (2018).
12. P. H. Scherrer et al., "The Helioseismic and Magnetic Imager (HMI) investigation for the Solar Dynamics Observatory (SDO)," *Sol. Phys.* **275**, 207–227 (2012).
13. S. Couvidat et al., "Observables processing for the Helioseismic and Magnetic Imager instrument on the Solar Dynamics Observatory," *Sol. Phys.* **291**(7), 1887–1938 (2016).
14. P. H. Scherrer et al., "The solar oscillations investigation—Michelson Doppler Imager," *Sol. Phys.* **162**, 129–188 (1995).
15. J. de la Cruz Rodríguez et al., "CRISPRED: a data pipeline for the CRISP imaging spectropolarimeter," *Astron. Astrophys.* **573**, A40 (2015).
16. A. Gandorfer et al., "The Solar Orbiter mission and its polarimetric and helioseismic imager (SO/PHI)," *J. Phys. Conf. Ser.* **271**, 012086 (2011).
17. A. Gandorfer et al., "The high resolution telescope (HRT) of the Polarimetric and Helioseismic Imager (PHI) onboard Solar Orbiter," *Proc. SPIE* **10698**, 106984N (2018).
18. A. Alvarez-Herrero et al., "The polarization modulators based on liquid crystal variable retarders for the PHI and METIS instruments for the Solar Orbiter mission," *Proc. SPIE* **10563**, 105632Z (2017).
19. C. Dominguez-Tagle et al., "Filtergraph calibration for the polarimetric and helioseismic imager," *Trans. Jpn. Soc. Aeronaut. Space Sci.* **12**, Tk_25–Tk_27 (2014).
20. E. L. Degl'Innocenti and M. Landolfi, *Polarization in Spectral Lines*, Vol. 307, Kluwer Academic (2004).
21. F. Bubenhagen et al., "Reconfigurable platforms for data processing on scientific space instruments," in *NASA/ESA Conf. Adapt. Hardware and Syst.*, pp. 63–70 (2013).
22. B. Fiethe et al., "Reconfigurable system-on-chip data processing units for space imaging instruments," in *Proc. Conf. Des., Autom. and Test Europe*, EDA Consortium, pp. 977–982 (2007).
23. B. Fiethe et al., "Adaptive hardware by dynamic reconfiguration for the Solar Orbiter PHI instrument," in *NASA/ESA Conf. Adapt. Hardware and Syst.*, IEEE, pp. 31–37 (2012).
24. B. Osterloh et al., "Socwire: a network-on-chip approach for reconfigurable system-on-chip designs in space applications," in *NASA/ESA Conf. Adapt. Hardware and Syst.*, IEEE, pp. 51–56 (2008).
25. B. Osterloh et al., "Architecture verification of the SoCWire NoC approach for safe dynamic partial reconfiguration in space applications," in *NASA/ESA Conf. Adapt. Hardware and Syst.*, IEEE, pp. 1–8 (2010).
26. "RTEMS real time operating system (RTOS)."
27. D. O. Suárez and J. C. del Toro Iniesta, "The usefulness of analytic response functions," *Astron. Astrophys.* **462**(3), 1137–1145 (2007).
28. M. Semel, "Contribution à létude des champs magnétiques dans les régions actives solaires," *Ann. Astrophys.* **30**, 513–513 (1967).
29. D. E. Rees and M. D. Semel, "Line formation in an unresolved magnetic element: a test of the centre of gravity method," *Astron. Astrophys.* **74**, 1–5 (1979).
30. D. H. Expósito et al., "Image compression on reconfigurable FPGA for the SO/PHI space instrument," *Proc. SPIE* **10707**, 107072F (2018).
31. "ZLIB."
32. T. Lange et al., "On-board processing using reconfigurable hardware on the Solar Orbiter PHI instrument," in *NASA/ESA Conf. Adapt. Hardware and Syst.*, IEEE, pp. 186–191 (2017).
33. T. Wittrock, "Flexible Operational Sequencing of Complex Spaceborne Instruments – TheSoftware System OCL," *54th Int. Astronaut. Congr. Int. Astronaut. Fed., Int. Acad. Astronaut., Int. Inst. Space Law*, Bremen, Germany (2012).

34. K. Albert et al., "Performance analysis of the SO/PHI software framework for on-board data reduction," in *Astron. Data Anal. Software and Syste. XXVIII, ASP Conf. Ser.*, Astronomical Society of the Pacific (2019).
35. K. Albert et al., "Metadata and their importance in SO/PHI's on-board data processing," in *Astron. Data Anal. Software and Syste. XXVIII, ASP Conf. Ser.*, Astronomical Society of the Pacific (Submitted).
36. "IDA GSESOS V software package," http://www.gseos.de/index.htm (accessed 1 September 2018).
37. J. B. Rodríguez et al., "SOPHISM: an end-to-end software instrument simulator," *Astrophys. J. Suppl. Ser.* **237**, 35 (2018).
38. J. Schou et al., *Polarization Calibration of the Helioseismic and Magnetic Imager (HMI) Onboard the Solar Dynamics Observatory (SDO)*, pp. 327–355, Springer US, New York (2012).
39. J. Evershed, "Radial movement in sun-spots," *Mon. Not. R. Astron. Soc.* **69**, 454 (1909).
40. D. C. Wells, E. W. Greisen, and R. H. Harten, "FITS—a flexible image transport system," *Astron. Astrophys. Suppl.* **44**, 363 (1981).
41. T. Lange et al., "Evaluation of a hardware accelerated on-board processing pipeline for Solar Orbiter's PHI instrument," in *ASD-Eurospace Conf. Data Syst. Aerosp.*, DASIA, Oxford (2018).
42. K. Albert et al., "Autonomous on-board data processing and instrument calibration software for the SO/PHI," *Proc. SPIE* **10707**, 107070O (2018).

Kinga Albert is a PhD candidate at the Max Planck Institute for Solar System Research, working is on the on-board data reduction of the PHI on the Solar Orbiter mission. She received her MSc degree in spacecraft design from the Luleå University of Technology in Sweden in 2014 and has been a young graduate trainee at the European Space Agency from 2014–2015. Her research interests include solar spectropolarimetry, science data processing pipelines, and autonomous operations.

Johann Hirzberger is a senior scientist at the Max Planck Institute for Solar System Research and working on instrumentation projects. He leads the operations and calibration of the PHI on the ESA/NASA Solar Orbiter space mission. After receiving his PhD from the University of Graz/Austria, he has worked on ground based solar observations at the Institute for Astronomy in Göttingen/Germany and at the Institute of Physics in Graz/Austria. His main research topics are high resolution solar astrophysics, solar spectro-polarimetry and image processing.

Martin Kolleck is a software engineer who has been working on instrument control software for 15 years at the Max Planck Institute for Solar System Research. His work encompasses the design and implementaion of on-board software, and ground support software and instrument operations for the Sunrise I and II missions and the PHI instrument for the ESA-NASA Solar Orbiter mission.

Juan Pedro Cobos Carrascosa received his MSc and PhD degrees in computer architecture from the University of Granada in 2007 and 2016, respectively. He is computer engineer at the Instituto de Astrofísica de Andalucía (IAA-CSIC) from 2008. He participated in the PHI instrument for ESA's Solar Orbiter mission, specifically in the RTE inversion on FPGA. His research interests include high-performance scientific computing on FPGA and embedded computer architectures for space instrumentation.

David Orozco Suárez is currently a Ramón y Cajal research fellow at IAA-CSIC working primarily on the PHI for the ESA-NASA Solar Orbiter mission and on the stratospheric balloon-borne mission Sunrise. He has worked at the National Astronomical Observatory of Japan after receiving his PhD from the University of Granada (Spain), with a grant from the Japanese Society for the Promotion of Science and at the Instituto de Astrofísica de Canarias (IAC) with an ERC's Marie Curie fellowship. His field of interest is polarimetry, solar magnetic fields, and the development of space-based vector polarimeters in the solar physics field.

David Hernández Expósito is an electronic engineer at IAC, currently working in the development of the on-board data processing system for the instruments SCIP and TuMaG of the Sunrise mission. He has worked at IAA-CSIC with major responsibilities in the development of the FPGA image compression core for the PHI instrument aboard the ESA-NASA Solar Orbiter mission. His work focuses on FPGA development for astrophysics instrumentation, specifically image processing and compression.

Biographies of the other authors are not available.