IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 18, 2021, accepted December 23, 2021, date of publication January 14, 2022, date of current version January 27, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3143239*

# An FPGA-Based Solution for Computing a Local Stationarity Measure From Satellite Data

**DAN CRISTIAN TURICU[1], OCTAVIAN CREŢ[1], MARIUS ECHIM[2,3], AND COSTEL MUNTEANU[3]**
[1]Computer Science Department, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania
[2]Royal Belgian Institute for Space Aeronomy, 1180 Brussels, Belgium
[3]Institute of Space Science, 077125 Măgurele, Romania

Corresponding author: Dan Cristian Turicu (dan.turicu@cs.utcluj.ro)

This work was supported by the Romanian Ministry of Research and Innovation via a PCCDI Grant 18PCCDI/2018 and PROGRAM NUCLEU LAPLAS. The work of Marius Echim was supported by ESA PRODEX CLUSTER and Belgian Solar Terrestrial Center of Excellence (STCE). The work of Costel Munteanu was supported by ESA PRODEX MISION.

**ABSTRACT** The analysis of data variability from in-situ observations is essential for scientists and space mission controllers. Given the limited resources available on-board a spacecraft as well as the presence of the Field-Programmable Gate Arrays (FPGA) devices in modern spacecraft architectures, an efficient real-time monitoring solution should be deployed on these devices to use minimal computational and energy resources, and to reduce the main on-board computer utilization, thus making it available for other tasks. This paper describes the implementation of an algorithm for computing a local stationarity measure (LSM) on FPGA devices. The algorithm tests weak stationarity from the convergence of the partial means of the signal computed on subsets of increasing length, compared to the overall mean of the signal over a fixed-length running window; the window spans the entire signal. The algorithm is designed for an on-board implementation which monitors and detects changes of variables measured in-situ by scientific instruments (*e.g.*, magnetometers). The design was tested with synthetic and real-time signals and provides results in very good agreement with a dedicated data analysis library specifically designed for the analysis of satellite data.

**INDEX TERMS** FPGA-based design, high-level synthesis (HLS), local stationarity measure (LSM), magnetic field monitoring, satellites on-board processing, weak stationarity test.

## I. INTRODUCTION

Understanding data variability recorded in space is of utmost importance for scientists and also for space mission controllers. An estimation of key environmental changes (related to natural or other type of origins) is most needed and can be achieved from the analysis of relevant environmental variables. Consequently, automatic tools designed to extract key-descriptors of variability are extremely useful. Nevertheless, solutions to adapt such algorithms for on-board computers are still rare. In this paper we describe an algorithm used to evaluate stationarity from a time series analysis and its implementation on Field Programmable Gate Arrays (FPGA) devices. The problem of data stationarity is generally relevant for many fields of science and technology – physics, economy, and biology, to mention just a few examples. In our research we focus on the variability of time series collected in-situ by Earth observation satellites.

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

Stationarity can be broadly defined as the property of a stochastic process to be time invariant, in a statistical sense, *i.e.* its statistical properties do not change although the observables that characterize the process are highly variable in time/space. Thus, stationarity is a property of the process itself that is revealed by the analysis of a process realization, resulting in time series of one or more observables. In the study of space plasma turbulence, stationarity is a fundamental assumption for meaningful data analysis, *e.g.* [1], [2]. Data stationarity is a necessary condition for the application of most of the digital analysis techniques, *e.g.* the *Power Spectral Density (PSD)* or the *Probability Density Functions (PDFs)* analyses. If the signal is not stationary the results of PSD and PDF analyses can be spurious and may suggest relationships and structures that are not real.

Perri and Balogh [3] argue that stationarity only holds for the inertial range of scales in turbulent interplanetary plasma. Jagarlamudi *et al.* [4] show that the autocorrelation function has a high variance and fails to converge toward a constant function, even for the longest available intervals of either fast

or slow solar wind. Such findings suggest a scale-dependent non-stationarity of turbulent plasmas in space. Studies based on data provided by state-of-the-art interplanetary missions, *e.g.* [5], make extensive use of the stationarity hypothesis. Nevertheless, more quantitative investigations are required and the solution discussed in this study is such a contribution, being novel both at conceptual and technical levels. It is also important to note that the examples given above are based on satellite data processed on the ground.

The novelty of the proposed solution is twofold:

- The algorithm can be executed on-board thus allowing for a computational and energy efficient in-flight estimation of stationarity.
- It allows real-time testing of stationarity on various scales and frequencies, customizable based on the requirements of the specific application.

Strictly speaking, (strong) stationarity requires that the probability distribution function of the realization of the stochastic process remains invariant. In this study we adopt a less constraining approach and test whether a signal is weakly stationary, *i.e.* if the moments of the distribution function are invariant. More specifically, we restrict the analysis to the first order moments, the mean and the variance.

A closely related problem is the detection of transitions between two states of a system, referred to as change points. Change point detection, also known as edge detection, event detection, and anomaly detection, is the problem of finding abrupt changes in a time series. Such singular events render the signal inherently non-stationary. In [6], a survey of methods for change point detection is presented. Recent developments in the field are published by [7], [8]. Our technology to probe stationarity is tested with a non-stationary signal characterized by a series of abrupt changes, or discontinuities, as detailed below. As will be seen, in addition to characterizing global weak stationarity, the method is also able to localize accurately the abrupt changes of the signal.

This study is part of a broader effort devoted to building a complex semi-autonomous digital signal processing library, able to apply on-board various digital signal processing techniques. Our library already includes modules dedicated to statistical [9] and Fourier [10] analyses. In [9] we described an FPGA-based solution to compute probability distribution functions of fluctuations. In [10] we described the design and implementation of two digital signal processing algorithms on a single FPGA device: (1) the power spectral density and (2) the multiscale probability distribution functions. The new module discussed in the following will serve as a tool to validate the data which satisfy the stationarity condition.

To the best of our knowledge, reports of such FPGA implementations are rare in the scientific literature. The work described in [11], which presents an FPGA implementation of a digital stochastic measurement method, is related to the stationarity problem, but it is applied to the theory of measurement in general and to brain signals in particular.

## II. THEORETICAL BACKGROUND

### A. WEAK STATIONARITY TEST

There is an extensive literature on hypothesis tests aimed at distinguishing a stationary time series from a non-stationary one [12]–[15], and [16]. Most of these papers test the null hypothesis that a time series is weakly stationary against the alternative hypothesis that it is not. In practice, however, this simple binary distinction is not useful in determining a degree of non-stationarity. Recently, Das and Nason [16] created one of the few indices that estimate the degree of non-stationarity of a time series. Their index is based on measuring the roughness of a statistic estimated from the time series, which is calculated from the roughness penalty associated with a spline smoothing least-squares method. One of the key advantages of a non-stationarity index is that, with two time series, one can state whether one series is more or less non-stationary than the other.

Starting from an idea discussed in [17] to evaluate globally the stationarity of a finite length in-situ data stream, [18] proposes a more general test of stationarity which provides a local quantitative measure of stationarity for a variable A. The qualitative algorithm based on a graphical estimation of the nonlinearity proposed in [17] is further developed in [18], where the author computes a time series of local stationarity scores that characterizes non-stationarity ''events'' and also identifies their ''sources''.

The *local stationarity measure (LSM)* we propose is computed and assigned to a time stamp, $\tau_k$, corresponding to the center of a time window $\Delta W$ of fixed length, $L$. The window sweeps the entire signal and an *LSM* value is computed for each instance.

The mathematical kernel for the computation of the local stationarity measure is available from a scientific data analysis toolbox, the Integrated Nonlinear Analysis (INA) library [19]. INA cumulates some of the most used time series analysis methods: the power spectral density (PSD), the multiscale probability distribution functions (PDFs), and the structure functions and wavelets.

In order to compute the local stationarity measure $LSM(\tau_k)$, one first computes the mean value of the variable A, $< A > (\tau_k)$, from all the samples within the window $\Delta W$ centered on $\tau_k$. Then, a series of partial means $< A_{left} >_i$ is computed for subsets of the data samples within the window, with increasing lengths $l_i = l_1, l_2, l_3 \ldots l_L$. Next, we calculate the series $\delta_i$, defined as:

$$\delta_i =< A > -< A_{left} >_i \tag{1}$$

A signal is considered stationary if $\delta_i$ converges towards zero. To illustrate the procedure, we applied it on a time series of 900 synthetic samples as shown in Fig. 1. The test described above is based on partial means calculated by starting from the left-side of the window $\Delta W$ and progressing toward the right-side. This implementation has a major limitation: on the one hand, a non-stationary feature (*e.g.* the discontinuity in Fig. 1) produces large differences between $< A >$ and $< A_{left} >_i$ only if it is localized

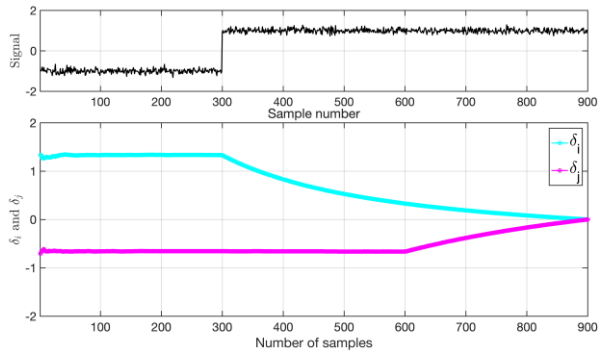



**FIGURE 1.** Weak stationarity tests using the Integrated Nonlinear Analysis (INA) [19]. Top panel: a synthetic time series obtained by introducing a sharp discontinuity on a white noise background. Bottom panel: $\delta i$ as a function of $i$ (in cyan) and $\delta j$ as a function of $j$ (in magenta).

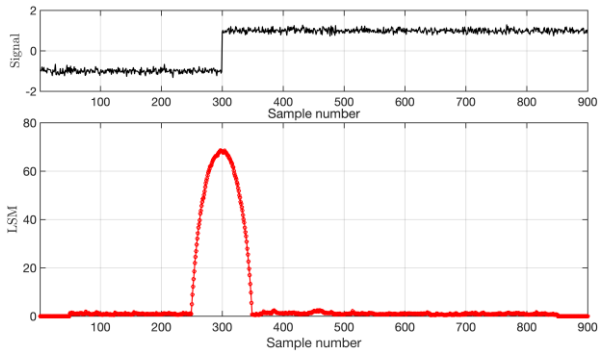



**FIGURE 2.** The local stationarity measure (LSM) using INA. Top panel: the synthetic time series depicted in Fig. 1. Bottom panel: the local stationarity measure (LSM) computed for the signal in the top panel, and depicted as a function of sample number.

closer to the beginning (left side) of the window $\Delta W$; on the other hand, if the non-stationary feature is localized closer to the end (right side) of the window, the differences will be much smaller, and the test might fail to identify the data non-stationarity.

This limitation can be removed by applying the stationarity test two times: the first time as described above, *i.e.* by computing the partial means from left to right, and the second time by computing the partial means from right to left. This second series of partial means (denoted as $< A_{right} >_j$) is then used to compute $\delta_j$, defined as:

$$\delta_j =< A > - < A_{right} >_j \tag{2}$$

The results of the second stationarity test are also depicted in Fig. 1. Instead of a qualitative visual examination of the convergence towards zero of $\delta_i$ and $\delta_j$, we introduce a quantitative local stationarity measure $LSM(\tau_k)$, defined as:

$$LSM\ (\tau_k) = \frac{sum\ (abs\ (\delta_i)) + sum\left(abs\left(\delta_j\right)\right)}{2} \tag{3}$$

The method proceeds by moving the window $\Delta W$ by one point and the entire procedure is executed again to provide the value at the next time step, $LSM(\tau_{k+1})$. Note that the procedure bears some similarities with the computation of the *local intermittency measure (LIM)* based on wavelet analysis described in [20].

The results of a test based on this approach for a window $\Delta W$ of length $L = 100$ samples are described in Fig. 2. One notices that *LSM* exhibits large peaks precisely for those parts of the signal with discontinuous changes. Thus, a continuous monitoring of *LSM* values provides a good estimation of drastic changes in the signal.

Consequently, the output of the algorithm can be used as input for other algorithms running on-board or from ground. It can be used, for instance, to trigger the execution of other complex analyses on the satellite's main on-board computer by comparing the stationarity parameters with a selected threshold value.

### B. FPGA TECHNOLOGIES FOR SATELLITES

The large amount of data generated by the complex active and passive satellite's on-board instruments requires increasing on-board processing capabilities.

The limited computation capabilities and the high power consumption of the spacecraft and satellites processors led to the integration of the FPGA technology in modern architectures [21]–[25], and [26].

In space applications, the electronic systems are susceptible to failures caused by the radiations emitted by a large variety of high-energy particles. The FPGA devices are even more vulnerable since radiations can cause bit-flipping in memory elements, which can affect the applications data and also the configuration logic itself in both SRAM and Flash technology configuration cell FPGA devices.

Specific hardware (*e.g.* hardware shielding) and software (*e.g.* triple module redundancy, safe state-machines, etc.) mitigation techniques are required for space-flight qualified FPGA designs to mitigate the effects of radiation on registers and memories. Although laboratory test results [27] performed on the latest-generation of commercial FPGA devices show an improved resistance to radiations, the radiation tolerant FPGA devices from Xilinx [28]–[30] or Microsemi [31] mitigate the effect of the radiations and eliminate the requirement of using dedicated mitigation techniques.

The low availability and the high price of the radiation hardened FPGA devices drove us to the implementation of the system on a commercial off-the-shelf device, whose internal structure is similar to the one of the radiation-tolerant devices.

The traditional development flow for FPGA devices is based on a *Register Transfer Level (RTL)* description of the circuits using a hardware description language, *e.g.* VHDL, Verilog. The *High-Level Synthesis (HLS)* technology simplifies the implementation of the digital signal processing algorithms in FPGA devices by allowing the description of the functionality using classical programming languages, *e.g.* C++. The high level synthesis generates the RTL implementation of the design, which is integrated into the traditional development flow.

Our proposed solution uses high-level synthesis to generate the RTL description for the weak stationarity test and an RTL description for the rest of the system components.

## III. SYSTEM OVERVIEW AND IMPLEMENTATION

The satellites main on-board computers execute many critical tasks. Considering that the satellites platforms already include FPGA devices, our prototype demonstrates that some of these tasks can be retargeted to be executed on those FPGA devices, thus saving time and energy.

The system demonstrates that the weak stationarity test can detect significant changes of the measured physical variable. The laboratory tests were performed by continuously monitoring the measurements of a magnetometer and calculating the local stationarity measure. The information provided by the system can be used by the on-board computer as triggers for performing more complex analyses.

### A. SYSTEM IMPLEMENTATION

A design conceived for space applications usually is first developed on a regular FPGA device. The same design will then work as well on a space-grade FPGA device by rerunning the implementation flow for the new target device. This is precisely the strategy we used for designing our prototype.

Our system is implemented on Digilent Nexys 4 DDR development board featuring a Xilinx Artix-7 FPGA device. This set of devices is the low-end of the 7-series FPGA family of devices from Xilinx and consequently has fewer programmable logic resources compared to FPGA devices generally targeting space applications. The block diagram of the system is shown in Fig. 3.

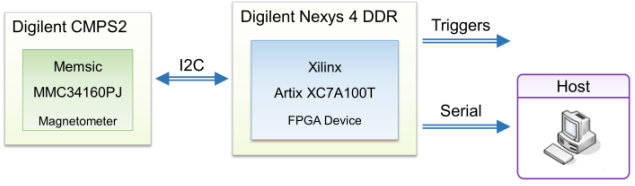



**FIGURE 3.** System block diagram.

To measure the intensity of the magnetic field, a Digilent CMPS2 module was attached to the development board. The module includes a Memsic MMC34160PJ magnetometer [32] able to perform magnetic field measurements on 3 axes. The sensor has a measurement range of $-16/+16$ Gauss with a resolution of 0.5 mG and can take a measurement for the 3 axes every 10 ms when the resolution of the ADC is set to 16 bits. The system communicates with the magnetometer over an I2C channel to configure and control the magnetometer.

The *Triggers* outputs notify the on-board computer when significant changes of the magnetic field are detected. To visualize the measurements of the magnetic sensor for the 3 axes and the calculated local stationarity measures, the system uses a serial connection with the host computer.

In a real-life scenario, when the system is deployed on the satellite's on-board FPGA device, the *Triggers* are the only outputs of the system, and the serial connection can be replaced with dedicated interfaces for communication with other experimental instruments or the main computer.

### B. FPGA SYSTEM DESIGN

The hardware system implemented in the FPGA device includes the following functionalities: an I2C communication interface with the magnetometer, the weak stationarity test computing module for each magnetic field axis, a trigger module which notifies the on-board computer and a serial unit for communicating with the host computer in order to validate the system. The architecture of the system is shown in Fig. 4.

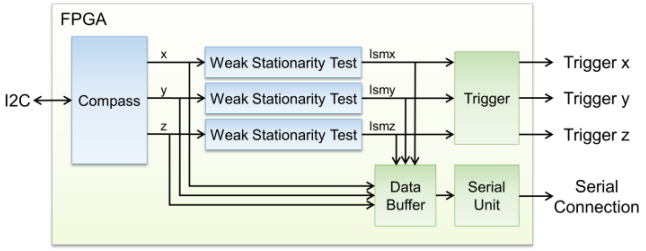



**FIGURE 4.** FPGA system architecture.

#### 1) COMPASS INTERFACE

The *Compass* component implements the communication interface with the magnetic sensor. The implementation is based on an I2C controller which provides a generic implementation of the I2C communication protocol.

The *Commands* and *Compass state machines* implement the operations specific to the Memsic magnetic sensor, *e.g.* take one measurement, activate the continuous measurement mode, and send calibration commands. The read measurements for each axis are returned on the X, Y, and Z outputs. The full architecture of the *Compass* component is shown in Fig. 5.

Once the *Compass* component is activated by the assertion of the *Compass Start* input, the operations performed by the magnetic sensor, according to the procedure described in the sensor's datasheet [32], are:

1) Executes the magnetometer self-check test.
2) Measures and saves the calibration offsets.
3) Activates the continuous measure mode:
   - Takes a measure on each axis every 12 ms.
   - Adjusts the measurements with the saved offsets.

#### 2) WEAK STATIONARITY TEST COMPONENT

The weak stationarity test described in section II.A is implemented in C++ and the RTL implementation is generated using high-level synthesis. Since the sampling rate of the magnetometer is only 100 Hz, which is very slow compared to the 100 MHz clock signal driving the FPGA device, the

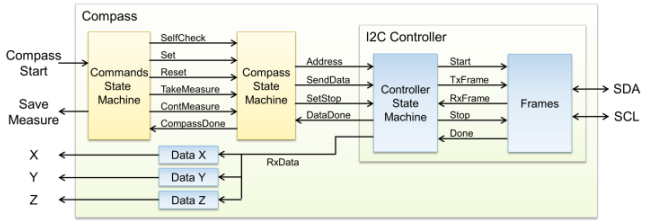



**FIGURE 5.** Compass component architecture.

goal set for the high-level synthesis optimizations is to reduce the area of the design. The design can be adapted to support sensors with higher sampling rates by applying different high-level synthesis optimization techniques, *e.g.* unroll the loops or pipeline the design.

The raw measurements received from the magnetometer are converted to Gauss before they are passed to the function which computes the weak stationarity test. The pseudocode for this function is described in Fig. 6. The $W$ parameter of the function holds the last $L$ measurements received from the magnetometer. It is implemented as a buffer in which the most recent measurement is saved, and the oldest one is discarded.

```
function Weak_Stationarity_Test(W, L):
1   sum_left = 0
2   sum_right = 0
3   for i in 1 to L do
4     sum_left = sum_left + W[i]
5     sum_right = sum_right + W[L - i + 1]
6     A_left[i] = sum_left / i
7     A_right[i] = sum_right / i
8
9   window_mean = sum_left / L
10
11  lsm_left = 0
12  lsm_right = 0
13  for i in 1 to L do
14    lsm_left = lsm_left + |window_mean - A_left[i]|
15    lsm_right = lsm_right + |window_mean - A_right[i]|
16
17  lsm = (lsm_left + lsm_right) / 2
18  return lsm
```

**FIGURE 6.** Weak stationarity test function pseudocode.

The first loop, lines 3 – 7, computes the partial means vectors of the measurements received within the window $\Delta W$. Each position $< A_{left} >_i$ and $< A_{right} >_j$ holds the mean of the measurements received up to the sample stored in that position, $t_i$ and $t_j$. The mean for the measurements $< A > (\tau_k)$ is calculated by line 9.

The second loop, lines 13 – 15, calculates $\delta_i$ and $\delta_j$, the differences between the mean value $< A > (\tau_k)$ and the partial means vectors $< A_{left} >_i$ and $< A_{right} >_j$ calculated by the first loop.

Finally, line 17 calculates the stationarity parameter $LSM(\tau_k)$ for the samples within the window $\Delta W$ as the average between $\delta_i$ and $\delta_j$. When a new measurement is received, it is saved in the buffer $W$, the oldest measurement is discarded, and the entire procedure is executed again to provide the local stationarity measure at the next time step, $LSM(\tau_{k+1})$.

#### 3) TRIGGER COMPONENT & SERIAL UNIT

The *Trigger* and *Serial Unit* components generate the outputs of the system. The *Trigger* component compares the local stationarity measure for each axis with a predefined threshold value. Once the calculated local stationarity parameter $LSM(\tau_k)$ for an axis exceeds the threshold value, the corresponding *Trigger* output will be asserted to notify the main on-board computer. Based on the requirements of the on-board computer, the component can be further customized by implementing a handshake mechanism or maintaining the output asserted for a predefined period of time. While only the *Trigger* component is present in the system deployed in a real- life scenario, for the validation of the implementation the *Serial Unit* sends the magnetic sensor's measured values and the calculated local stationarity measures to the host computer for results visualization.

## IV. EXPERIMENTAL RESULTS

The implementation of the system in the FPGA device started with the generation of the RTL description for the weak stationarity test using high-level synthesis and the corresponding techniques. The following sections describe several analyses we performed for the implementation of the weak stationarity test, specifically the datatype analysis, and the device utilization and timing analyses for different sizes of the window timeframe.

Based on these analyses, two configurations for the weak stationarity test were selected for the validation of the system. The generated RTL descriptions for the weak stationarity test components were integrated with the rest of the system. The validation results for the complete system implemented in the FPGA device are also described in the following sections.

### A. WEAK STATIONARITY TEST IMPLEMENTATION

The initial proof-of-concept for the weak stationarity test algorithm was implemented in MATLAB running on a regular desktop computer [18]. The algorithm was validated with series of signals which simulate different scenarios.

For the implementation of the algorithm in FPGA devices with Xilinx Vivado HLS 2019.1 [33], the weak stationarity test algorithm had to be rewritten in C++. To test and to validate the implementation, the MATLAB results for the local stationarity measures calculated for the synthetic signals and for the measurements received from the magnetometer were compared to the results calculated by the C++ implementation for the same input signals.

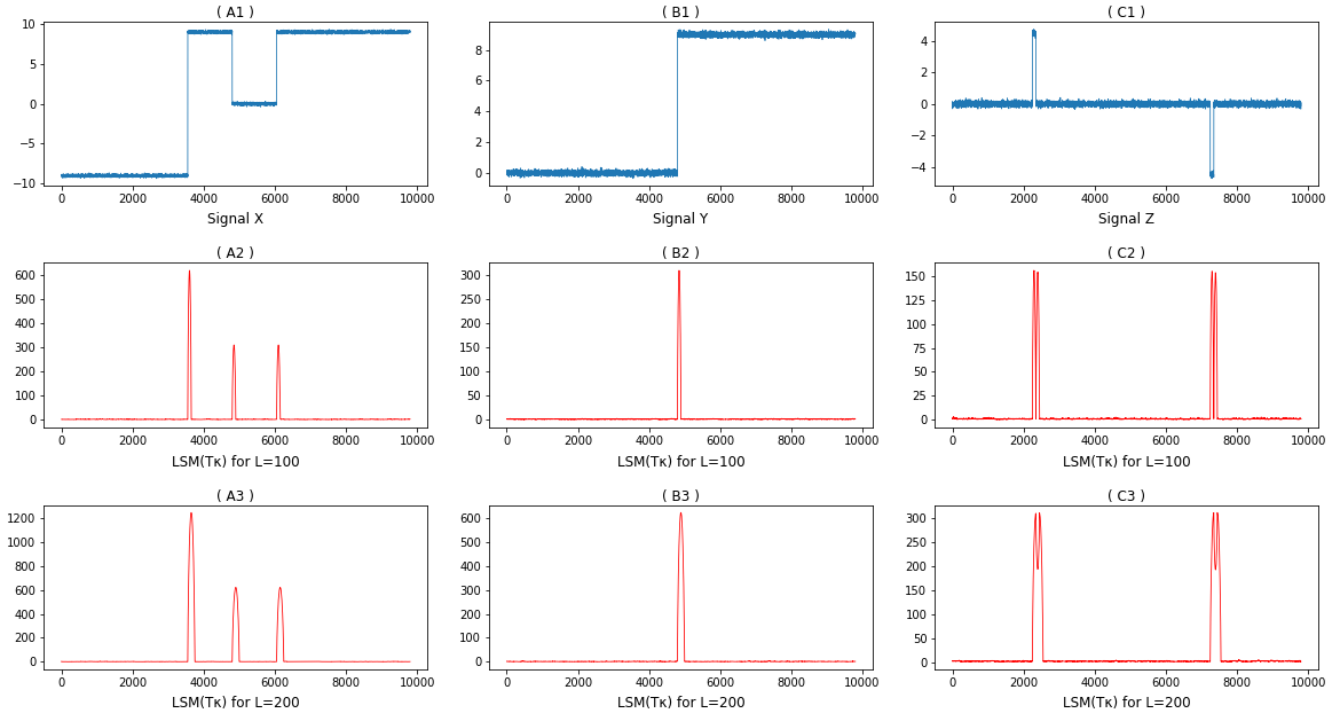



**FIGURE 7.** Weak stationarity test with synthetic signals. Row 1 shows three synthetic signals of length 10000 samples containing various sharp discontinuities. Rows 2 and 3 show the corresponding LSM values computed using windows of 100 and 200 samples, respectively.

Several different datatypes were analyzed for the implementation of the weak stationarity test algorithm. The Vivado HLS supports the standard C++ datatypes and it also provides alternative datatypes that can provide a more efficient hardware implementation of the algorithms, specifically the arbitrary precision integer and fixed-point datatypes. For the representation of non-integer numbers, the fixed-point datatype supports an arbitrary number of bits for the representation of the integer and the fractional part (compared to the standard float and double datatypes compliant with the IEEE-754 standard). Also, for the non-integer numbers, the half precision datatype provides another alternative.

Table 1 shows the analysis of different datatype configurations. For the synthetic signals shown in Fig. 7 and for the measurements captured from the magnetic sensor shown in Fig. 8, the table describes the maximum differences between the results of the MATLAB algorithm and the C++ implementation in Xilinx Vivado HLS.

The *Max Error* columns describe the maximum absolute difference for each axis between the local stationarity measures computed by the MATLAB implementation and the C++ implementation. It provides an idea of the order of magnitude of the error for each datatype.

Based on these results, we selected the float datatype for the implementation of the weak stationarity test algorithm. Even though the differences between the two programming languages have the potential to introduce errors, by tuning the C++ implementation we obtained a maximum absolute error smaller than 0.001 between the two implementations.

**TABLE 1.** Tested datatypes and resulting errors.

| Test Signal | Datatype | Max Error X-axis | Max Error Y-axis | Max Error Z-axis |
|---|---|---|---|---|
| Synthetic | Float | 0.000244 | 0.000092 | 0.000046 |
| Capture | Float | 0.000006 | 0.000001 | 0.000006 |
| Synthetic | Half | 3.984375 | 3.439453 | 1.200195 |
| Capture | Half | 1.334961 | 1.361328 | 1.736328 |
| Synthetic | FXP<37,17> | 0.000063 | 0.000045 | 0.000046 |
| Capture | FXP<37,17> | 0.000043 | 0.000044 | 0.000042 |
| Synthetic | FXP<36,17> | 0.000099 | 0.000090 | 0.000090 |
| Capture | FXP<36,17> | 0.000088 | 0.000088 | 0.000086 |
| Synthetic | FXP<34,17> | 0.000351 | 0.000359 | 0.000359 |
| Capture | FXP<34,17> | 0.000359 | 0.000343 | 0.000351 |
| Synthetic | FXP<34,16> | 400.9719 | 458.0677 | 474.5886 |
| Capture | FXP<34,16> | 1092.654 | 1099.496 | 1098.8443 |
| Synthetic | FXP<32,16> | 400.9719 | 458.0577 | 474.5886 |
| Capture | FXP<32,16> | 1092.654 | 1099.496 | 1098.8442 |

### B. WEAK STATIONARITY TEST ANALYSES

Vivado HLS provides performance and utilization estimates for the generated circuits. Table 2 presents the performance estimates for the implementation of the weak stationarity test with different sizes of the window timeframe. The FPGA device uses a 100 MHz external oscillator, therefore the targeted clock period of the circuit is 10 ns. For all sizes of the window timeframe $L$ this requirement is met.

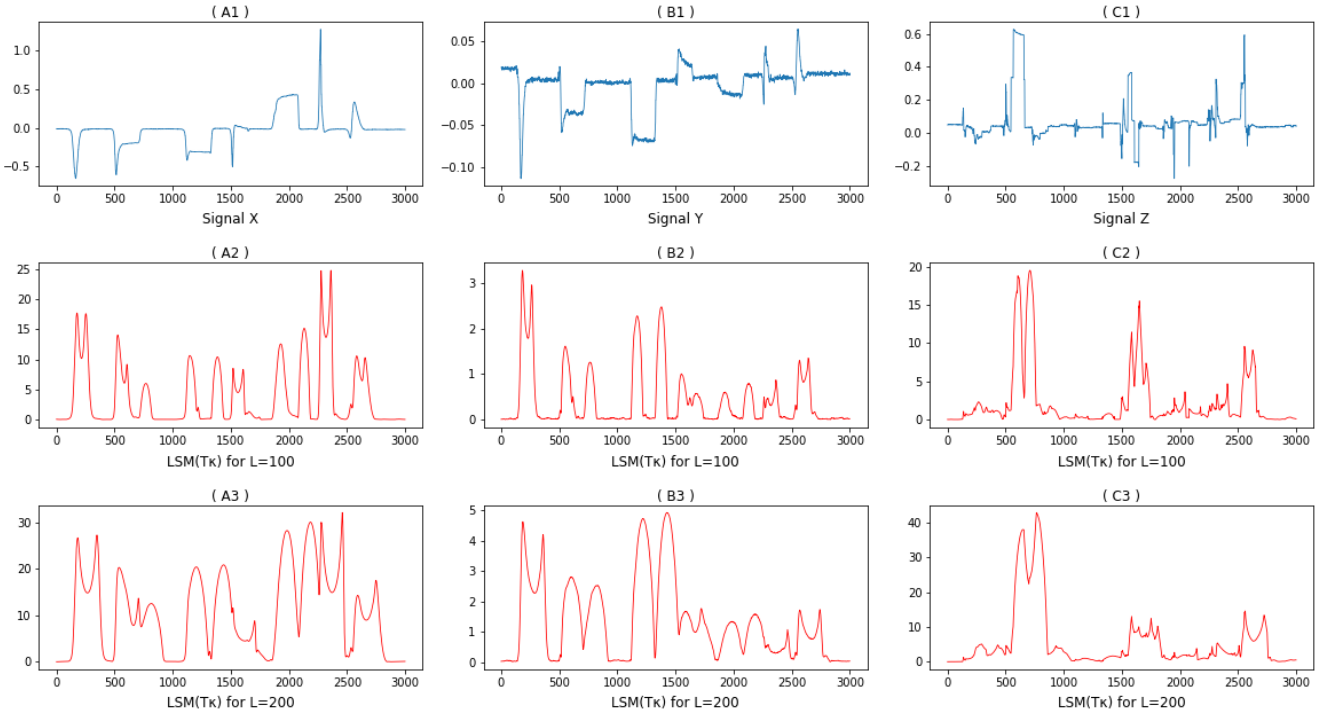



**FIGURE 8.** Weak stationarity test results obtained by the FPGA device connected to real-time magnetic field data measured by a Memsic MMC34160J. The figure setup is the same as in Figure 7, the first row shows 3000 measurements received from the magnetic sensor.

**TABLE 2.** Timing estimates for the weak stationarity test.

| Window Timeframe | Clock Target | Clock Estimated | Initiation Interval | Latency |
|---|---|---|---|---|
| *samples* | *ns* | *ns* | *clock cycles* | *clock cycles* |
| 100 | 10 | 8.263 | 3650 | 3650 |
| 200 | 10 | 8.300 | 7250 | 7250 |
| 500 | 10 | 8.304 | 18050 | 18050 |
| 1000 | 10 | 8.308 | 36050 | 36050 |
| 2000 | 10 | 8.312 | 72050 | 72050 |
| 5000 | 10 | 8.319 | 180050 | 180050 |
| 8000 | 10 | 8.319 | 288050 | 288050 |
| 10000 | 10 | 8.323 | 360050 | 360050 |

**TABLE 3.** Utilization estimates for the weak stationarity test.

| Window Timeframe | BRAM | DSP | FF | LUT |
|---|---|---|---|---|
| 100 | 2 | 7 | 2839 | 3019 |
| 200 | 2 | 7 | 2845 | 3019 |
| 500 | 2 | 7 | 2851 | 3026 |
| 1000 | 6 | 7 | 2803 | 2905 |
| 2000 | 12 | 7 | 2810 | 2908 |
| 5000 | 48 | 7 | 2824 | 2915 |
| 8000 | 48 | 7 | 2824 | 2915 |
| 10000 | 96 | 7 | 2831 | 2918 |

The *Initiation Interval* defines the number of clock cycles before the circuit which implements the function can accept new input data, while the *Latency* specifies the number of clock cycles required to compute all output values. Considering that we set the magnetic sensor to take a new measurement every 12 ms, these requirements are satisfied.

Table 3 presents the reconfigurable logic resources estimated to be required for the implementation of the weak stationarity test in FPGA device for different window timeframe sizes.

The results show that the size of the window timeframe only influences the required number of Block-RAM resources of the FPGA device, while the amount of reconfigurable logic resources remains constant. The implementation of the weak stationarity test with window timeframes $L$ of 100 and 200 measurements were exported from Vivado HLS and they were integrated with the rest of the system for testing and validation.

### C. SYSTEM VALIDATION

The system implemented in the Artix-7 FPGA device using the Xilinx Vivado 2019.1 [34] suite was tested with synthetic signals and with magnetic field measurements received from the attached magnetometer. The calculated local stationarity parameters were compared with the results calculated by the MATLAB implementation for the same measurements. The maximum absolute error was smaller than 0.001.

Fig. 7 shows the weak stationarity test results for a synthetic signal with 10,000 measurements and two window timeframes with $L$ of 100 and 200 measurements.

Fig. 8 shows a series of 3000 measurements captured with the magnetic sensor and the weak stationarity test results for

**TABLE 4. FPGA device resource utilization.**

| Resource | Utilization | Available | Utilization% |
|---|---|---|---|
| LUT | 6984 | 63400 | 11.02 |
| LUTRAM | 469 | 19000 | 2.47 |
| FF | 7422 | 126800 | 5.85 |
| BRAM | 6 | 135 | 4.44 |
| DSP | 21 | 240 | 8.75 |

window timeframes $L$ of 100 and 200 measurements. We used a magnet to artificially introduce changes to the magnetic field measured by the sensor. The magnet was moved closer and further away from the sensor to produce changes to the magnetic field. By comparing panels B2 and B3, one can easily observe the effect of using analysis windows of different lengths. There are two sharp discontinuities in the interval from 1000 to 1500 measurements depicted in panel B1. Using a window of 100 measurements (B2), the two discontinuities are clearly detected as distinct non-stationary features, separated by a stationary interval of measurements. However, using a larger analysis window containing 200 measurements (B3), one can observe that the two non-stationary features are barely distinguishable from one another and the stationary interval between them is completely absent from these results.

### D. REAL-TIME VALIDATION WITH MAGNETOMETER

For the validation of the system in a setup close to the deployment on a relevant environment, a real-time scenario was tested. The measurements received from the magnetic sensor are sent to the host together with the computed local stationarity measures. An application running on the host computer was developed to read and plot the received measurements and the stationarity test results in real-time.

### E. ENERGY EFFICIENCY AND DEVICE UTILIZATION

The resource utilization and power consumption analyses of the system implemented in the Artix XC7A100T-CSG324 FPGA device were performed. The results are provided for the system that integrates a weak stationarity test with a window timeframe $L$ of 100 measurements.

Table 4 shows the amount of reconfigurable logic resources used by the implementation of the complete system in the FPGA device. The system uses a small part of the device resources, allowing other functionalities to be implemented in the same device.

The power estimation for the system implemented in the Artix XC7A100T FPGA device is shown in Table 5. The power analysis for the routed design was performed using a vectorless approach and default environmental settings in Vivado. It is obvious that the power consumption, which is a critical aspect in computations performed on satellites, is two orders of magnitude less than in a general-purpose processor, which is one of the main reasons for targeting an implementation of this analysis method in FPGA devices.

**TABLE 5. FPGA device power analysis.**

| | | |
|---|---|---|
| Total On-Chip Power | 0.261 W | 100 % |
| Device Static Power | 0.098 W | 37 % |
| Dynamic Power | 0.163 W | 63 % |

## V. SUMMARY AND CONCLUSION

We designed, implemented, and tested an FPGA device to collect data from a scientific instrument on-board spacecraft and to characterize the weak stationarity properties of the measured variable. The algorithm uses the concept of a local stationarity measure computed as a function of time by moving a window of fixed length over the entire time series.

The laboratory tests performed with the FPGA design provide accurate results for synthetic and real-time signals, fully compatible with independent calculations performed on the same data with a scientific data analysis software.

The solution is based on an Artix XC7A100T-CSG324 FPGA device. It can also be easily adapted for space tolerant systems and it demonstrates the feasibility of this prototype to perform on-board weak stationarity tests. The FPGA design facilitates a more efficient utilization of the limited computational and energy resources available on-board.

As the resource utilization is relatively low (less than 15%), several instances of this architecture can be used in the same FPGA device. All instances will work in parallel, thus significantly increasing the overall performance compared to a processor-based implementation.

The proposed FPGA-based solution for estimating the weak stationarity is part of a broader effort meant to build a library able to perform on-board time series analyses. An immediate future development is to use the components employed in this study to design an FPGA-based solution for the detection of discontinuities from in-situ data – an important aspect for solar system plasma exploration, particularly in the interplanetary space. The end goal is to integrate all the solutions we developed in recent years in a single design for on-board applications.

## REFERENCES

[1] W. H. Matthaeus and M. L. Goldstein, "Stationarity of magnetohydrodynamic fluctuations in the solar wind," *J. Geophys. Res. Sp. Phys.*, vol. 87, no. A12, pp. 10347–10354, 1982.

[2] M. Echim, T. Chang, P. Kovacs, A. Wawrzaszek, E. Yordanova, Y. Narita, Z. Vörös, R. Bruno, W. Macek, K. Mursula, and G. Consolini, "Turbulence and complexity of magnetospheric plasmas," in *Space Physics and Aeronomy Collection: Magnetospheres in the Solar System, Geophysical Monograph 259*, vol. 2, 1st ed. R. Maggiolo, N. André, H. Hasegawa, and D. T. Welling, Eds. Washington, DC, USA: American Geophysical Union, 2021.

[3] S. Perri and A. Balogh, "Stationarity in solar wind flows," *Astrophys. J.*, vol. 714, no. 1, p. 937, 2010.

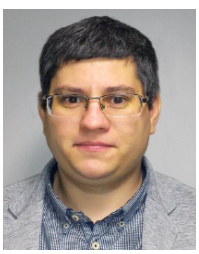
[4] V. K. Jagarlamudi, T. Dudok de Wit, V. Krasnoselskikh, and M. Maksimovic, "Inherentness of non-stationarity in solar wind," *Astrophys. J.*, vol. 871, no. 1, p. 68, Jan. 2019.

[5] D. Perrone, R. Bruno, R. D'Amicis, D. Telloni, R. D. Marco, M. Stangalini, S. Perri, O. Pezzi, O. Alexandrova, and S. D. Bale, "Coherent events at ion scales in the inner heliosphere: Parker solar probe observations during the first encounter," *Astrophys. J.*, vol. 905, no. 2, p. 142, Dec. 2020.
[6] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 339–367, May 2017.
[7] G. Montanez, S. Amizadeh, and N. Laptev, "Inertial hidden Markov models: Modeling change in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 1819–1825.
[8] Y. Kawahara and M. Sugiyama, "Sequential change-point detection based on direct density-ratio estimation," *Statist. Anal. Data Mining*, vol. 5, no. 2, pp. 114–127, 2012.
[9] N. Deak, O. Cret¸, M. Echim, E. Teodorescu, C. Negrea, L. Văcariu, C. Munteanu, and A. Hângan, "Edge computing for space applications: Field programmable gate array-based implementation of multiscale probability distribution functions," *Rev. Sci. Instrum.*, vol. 89, no. 12, Dec. 2018, Art. no. 125005.
[10] L. Opincariu, N. Deak, O. Cret¸, M. Echim, C. Munteanu, and L. Văcariu, "Edge computing in space: Field programmable gate array-based solutions for spectral and probabilistic analysis of time series," *Rev. Sci. Instrum.*, vol. 90, no. 11, Nov. 2019, Art. no. 114501.
[11] P. Sovilj, V. Vujičić, N. Pjevalica, D. Pejić, M. Urekar, and I. Župunski, "Influence of signal stationarity on digital stochastic measurement implementation," *Electron. ETF*, vol. 17, pp. 45–53, Jun. 2013.
[12] Y. Dwivedi and S. S. Rao, "A test for second-order stationarity of a time series based on the discrete Fourier transform," *J. Time Ser. Anal.*, vol. 32, no. 1, pp. 68–91, Jan. 2011.
[13] G. Nason, "A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series," *J. Roy. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 75, no. 5, pp. 879–904, Nov. 2013.
[14] M. Basta, "Wavelet-based test for time series non-stationarity," *Statistika*, vol. 95, no. 4, pp. 29–46, 2015.
[15] L. Jin, S. Wang, and H. Wang, "A new non-parametric stationarity test of time series in the time domain," *J. Roy. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 77, no. 5, pp. 893–922, Nov. 2015.
[16] S. Das and G. P. Nason, "Measuring the degree of non-stationarity of a time series," *Stat*, vol. 5, no. 1, pp. 295–305, 2016.
[17] C. Pagel, "Radial dependence of intermittency in the fast polar solar wind magnetic field using Ulysses," *J. Geophys. Res.*, vol. 108, no. A1, p. 1012, 2003.
[18] C. Munteanu, "Turbulent fluctuations and discontinuities in the solar wind: Statistical properties and possible effects on the terrestrial plasma environment," Ph.D. dissertation, Doctoral School Phys., Univ. Bucharest, Fac. Phys., Magurele, Romania, 2017. [Online]. Available: http://gpsm.spacescience.ro/gpsm/core/wan/en/thes.php
[19] (2015). *Solar System Plasma Turbulence: Observations, Intermittency and Multifractals—A European FP7 Project*. [Online]. Available: http://www.storm-fp7.eu
[20] M. Farge, N. Kevlahan, V. Perrier, and E. Goirand, "Wavelets and turbulence," *Proc. IEEE*, vol. 84, no. 4, pp. 639–669, Apr. 1996.
[21] S. Habinc. (2020). *Suitability of Reprogrammable FPGAs in Space Applications*. [Online]. Available: http://www.gaisler.com/doc/fpga_002_01-0-4.pdf
[22] P. J. Pingree, "Advancing NASA's on-board processing capabilities with reconfigurable FPGA technologies," in *Aerospace Technologies Advancements*, 2010, ch. 5, pp. 69–86.
[23] T. Kuwahara, "FPGA-based reconfigurable on-board computing systems for space applications," Thesis, Inst. Space Syst. Universitat Stuttgart, Oct. 29, 2009.
[24] M. French, "SpaceCubeX2: Heterogeneous on-board processing for distributed measurement and multi-satellite missions," in *Proc. Earth Science Technology Forum (ESTF)*, 2018.
[25] F. Huber, P. Behr, H.-P. Röser, and S. Pletner, "FPGA based on-board computer system for the flying laptop micro-satellite," in *Proc. Data Syst. Aerosp. Conf.*, Naples, Italy, 2007.
[26] A. Hanafi, M. Karim, I. Latachi, T. Rachidi, S. Dahbi, and S. Zouggar, "FPGA-based secondary on-board computer system for low-earth-orbit nano-satellite," in *Proc. Int. Conf. Adv. Technol. Signal Image Process. (ATSIP)*, May 2017, pp. 1–6.
[27] D. S. Lee. (May 2015). *Commercial Field-Programmable Gate Arrays for Space Processing Applications*. United States. Accessed: Oct. 10 2021. [Online]. Available: https://www.osti.gov/biblio/1458107
[28] *Space-Grade Virtex-4QV Family Overview*, document DS653 (v2.1), 2014.
[29] *Radiation-Hardened, Space-Grade Virtex-5QV Family Data Sheet: Overview*, document DS192 (v1.6), 2018.
[30] *RT Kintex UltraScale FPGAs for Ultra High Throughput and High Bandwidth Applications*, document WP523 (v1.0), 2020.
[31] *Mitigation of Radiation Effects in RTG4 Radiation-Tolerant Flash FPGAs*, document WP0191 White Paper, 2015.
[32] *±16 Gauss, Ultra Small, Low Noise3-Axis Magnetic Sensor*, document MMC3416xPJ, 2013.
[33] *Vivado Design Suite UserGuide High Level Synthesis*, document UG902 (v2019.1), 2019.
[34] *Vivado Design Suite UserGuide Synthesis*, document UG901 (v2019), 2020.

**DAN CRISTIAN TURICU** received the B.S. and M.S. degrees in system and computer engineering from the Technical University of Cluj-Napoca, Romania, in 2007, where he is currently pursuing the Ph.D. degree with the Computer Science Department. Since 2012, he has been a Teaching and Research Assistant with the Technical University of Cluj-Napoca. His research interests include the reconfigurable computing, digital systems design, computer systems security, machine learning, and deep learning.

**OCTAVIAN CRET¸** received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in reconfigurable computing systems from the Technical University of Cluj-Napoca, Romania, in 1995, 1996, and 2004, respectively. He is currently a Professor with the Computer Science Department, Technical University of Cluj-Napoca. His research interests include reconfigurable computing and digital systems design.

**MARIUS ECHIM** received the B.S. and M.S. degrees in plasma, lasers and nonlinear optics from the University of Bucharest, in 1992, and the Ph.D. degree in space physics from Catholic University Louvain, Belgium, in 2004. Since 1992, he received academic degrees, from an Assistant Scientist to a Senior Scientist 1st degree, with the Institute of Space Science, Măgurele, Romania. Since 2005, he has been a Postdoctoral Scientist, and then a Senior Scientist Level SW2 with the Space Physics Department, Royal Belgian Institute for Space Aeronomy, Brussels, Belgium.



**COSTEL MUNTEANU** received the B.S. and M.S. degrees in physics from the University of Iasi, Romania, in 2008 and 2010, respectively, the Ph.D. degree in physics from the University of Oulu, Finland, in 2015, and the Ph.D. degree in physics from the University of Bucharest, Romania, in 2018. Since 2010, he has been a Researcher with the Space Plasma and Magnetometry Laboratory, Institute of Space Science, Romania.