POLITECNICO DI TORINO
Repository ISTITUZIONALE

Creation of a Support Software for the Development of a System for Sending and Visualizing FBG Sensor Data for Aerospace Application

(Article begins on next page)

14 November 2024

Creation of a Support Software for the Development of a System for Sending and Visualizing FBG Sensor Data for Aerospace Application

Antonio Costantino Marceddu
*Department of Control and
Computer Engineering
Politecnico di Torino*
Turin, Italy
antonio.marceddu@polito.it

Alessandro Aimasso
*Department of Mechanical and
Aerospace Engineering
Politecnico di Torino*
Turin, Italy
alessandro.aimasso@polito.it

Antonio Scaldaferri
*Department of Control and
Computer Engineering
Politecnico di Torino*
Turin, Italy
antonio.scaldaferri@studenti.polito.it

Paolo Maggiore
*Department of Mechanical and
Aerospace Engineering
Politecnico di Torino*
Turin, Italy
paolo.maggiore@polito.it

Bartolomeo Montrucchio
*Department of Control and
Computer Engineering
Politecnico di Torino*
Turin, Italy
bartolomeo.montrucchio@polito.it

Matteo D. L. Dalla Vedova
*Department of Mechanical and
Aerospace Engineering
Politecnico di Torino*
Turin, Italy
matteo.dallavedova@polito.it

Abstract—Optical fiber-based sensors have rapidly increased their application fields across multiple engineering sectors. Their physical characteristics, including low weight, electrical passivity, immunity to electromagnetic disturbances, and high sensitivity, make them highly suitable for aerospace applications. A flying test bench was recently developed at the Politecnico di Torino to evaluate the performance of optical sensors applied to an aircraft model. To allows their fruition in near real-time, it was necessary to develop a pipeline of software capable of transmitting, saving, and displaying them on the ground. To carry out in-depth tests on their reliability it is necessary to carry out more tests with different sensors and configurations. However, doing flight tests for the sole purpose of testing the data acquisition system is too expensive and impractical. Therefore, it was developed an emulator software that can generate data like that obtained from the real system, saving time and resources. Thanks to it, it is now possible to exploit the complete test bench on the aircraft for the final verification campaign only. The results are therefore very positive and demonstrate the potential of the emulator also for the most recent applications.

Index Terms—Computer graphics, Data visualization, Databases, Graphical user interfaces, Middleware, Optical fiber sensors, Remotely piloted aircraft, Wireless communication

I. INTRODUCTION

Fiber Bragg Grating (FBG) sensors are optical devices used to measure a variety of physical quantities such as temperature,

pressure, strain, velocity, and force [1] [2]. They consist of an optical fiber containing a reflection grating, known as a Bragg grating, inscribed within the fiber itself. It is obtained through a laser technique which creates a periodic re-modulation of the refractive index of the inner layer of the fiber. When light is sent through the optical fiber, a specific wavelength is reflected by the Bragg grating while the others pass through it. If the physical quantity being measured by the sensor changes, the reflected wavelength will also change. This can be used to determine the physical quantity in question.

FBG sensors have several features that make them particularly useful in many applications, such as monitoring the structure of buildings and bridges, environmental conditions, human health, and industrial process control [3] [4] [5]. For example, they are very sensitive and can be used to measure very small changes in physical quantities. They are also very reliable and have a long lifespan, making them ideal for long-term applications. Furthermore, they are resistant to electromagnetic interference, making them ideal for environments with strong electromagnetic fields. Finally, they are small and lightweight, making them easy to install and transport. Even the aerospace industry has recently begun to consider the use of these fiber optic sensors advantageous due to the physical characteristics mentioned above. In particular, as described before, they are really sensitive to the physical quantities which are essential for system monitoring. Moreover, the increasing use of electro-mechanical components, in line with the evolution towards the concept of *"more electric aircraft"* [6], makes it fundamental to have sensors that are immune to electromagnetic interference [7]. FBG sensors are therefore particularly suitable for instrumenting components that have
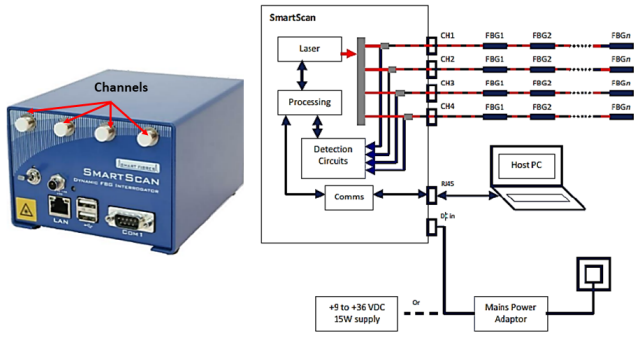
Fig. 1. A SmartScan interrogator with a scheme of an optical sensors data acquisition system.

to work even in particularly hostile conditions [8]. The testing phases involving components instrumented by FBGs could potentially be very complex and expensive. Moreover, to carry out accurate post-processing analysis of the data coming from the optical sensors, it is necessary to develop specific software for saving, managing, and visualizing the data. The creation of this computer system requires the availability of large amounts of optical data. However, constructing multiple experimental setups with different sensor configurations just to extensively test the data acquisition system is not easily achievable and time-consuming. For this reason, this work describes the development and implementation of a software, called *Interrogator Emulator*, able to simulate the overall output coming from a variable network of real optical sensors. It was mainly designed to support the development of a flying test bench for optical sensors, which will be summarized in the following section, but its use can be extended to other scenarios.

II. REAL DATA ACQUISITION SYSTEM

The FBGs *Interrogator* is the device designed to automatically recognize and query the optical sensors present in the fibers connected to various channels, collecting and analyzing their responses. This device communicates individually with each FBG, minimizing the risk of data misunderstandings from multiple sensors. The interrogator sends a laser beam through the fiber and measures the wavelengths reflected back. The working of the interrogator is schematically shown in Fig. 1.

The application of interest for this paper concerns the continuous improvement of a flying test bench for FBG optical sensors [9] [10]. The readings are possible through Smart Fibres's SmartScan laser interrogator, which runs a data acquisition by sampling at a frequency of 25 kHz. The software that comes with SmartScan itself is quite limited. For this reason, the authors of this paper recently presented a new system that can be defined as a "flying test bench" which relies on ad-hoc software capable of doing much more than the original software could do. This bench consists of a model aircraft, called Anubi, whose half-wing has been instrumented with FBG sensors. Raw data coming from such sensors are
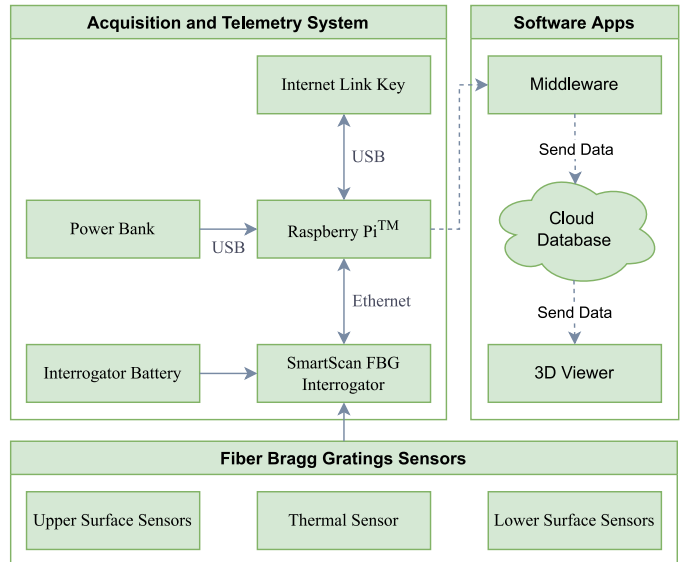

Fig. 2. Complete block diagram of the flying test bench. Taken from [9].

read by a SmartScan interrogator, which is connected via Ethernet with a *Raspberry Pi™ 3 Model B+* System-On-Chip (SOC). Both the interrogator and the SOC are powered by batteries mounted on the model aircraft itself. The *Raspberry Pi™* executes an application called *Middleware* which can receive data from the interrogator and transmit it to a Cloud Database using a *4G Internet Link Key* attached to it [9] [11] [12]. The *Cloud Database* is based on *MongoDB®*, a fast non-relational database that can take advantage of the Change Stream feature to forward the data received to any listening data visualization applications. These visualization applications, called *3D Viewer*, are located at the end of the system pipeline and are responsible for showing the received data to users on the ground in a simple and readable way. This is possible through the simultaneous use of a graph showing the variation in the value of the sensors with respect to their base value ($\Delta\lambda$) and a 3D heat map visualization consisting of a 3D version of the aircraft model that changes the color of the wing instrumented more or less markedly according to the forces acting on the wing itself [9] [13] [14]. The block diagram of the full flying test bench is depicted in Fig. 2.

III. EMULATOR

Interrogators are very expensive and delicate devices. Furthermore, the realization of experimental setups can be rather complex and difficult to implement. For this reason, it was decided to create an interrogator emulator, simply called *Emulator*, to facilitate the testing of software applications under development without having to employ the real test bench. An image showing the launch of the software is shown in Fig. 3.

The purpose of the *Emulator* is to send random raw and peak wavelength values or other kinds of messages to any *Middleware* application connected to the local network. It can work through two abstraction libraries, called *LibUtils* and

```
SmartScan board initalized.
Emulator started.
Open diagnostic socket on 0.0.0.0:30011.
Open maintenance socket on 0.0.0.0:30012.
Open send socket on 127.0.0.1:30071.
█
```

Fig. 3. The simple Graphical User Interface (GUI) of the Emulator when starting the program.

LibSmartScan. They have been developed with the aim of creating a single generalized library that can be extended to future interrogators of different models and brands. Internally, these libraries deal with converting the commands given by the user, which are managed in a language common to all interrogators, into commands that allow driving a particular emulator. The *LibSmartScan* library was created based on the datasheets of the SmartSoft kindly provided to the authors of the paper by Smart Fibres. It contains information about the customized UDP (User Datagram Protocol) used by the interrogator to send real raw and peak wavelength values, to vary its behavior, and more. So, if it is wanted to emulate a SmartScan, which is currently the only interrogator supported by the *Emulator*, messages will be sent by the software itself in the format typically used by this interrogator.

The *Emulator* supports the modification of several operating parameters, including:

- Current configuration, i.e. the number of sensors for each channel.
- Start and stop of data transmission.
- Transmission speed.

They can be set through the use of another software, called *Configurator*, which can be run on the same computer or the

```
Configurator started.
Menu thread started.
Open tx socket on 127.0.0.1.

* SMARTSCAN CONFIGURATOR MENU *
*
* - a: send diagnostic message
* - b: send default maintenance message
* - c: send custom maintenance message
*
* - z: exit
* - Choice: █
```

Fig. 4. The Configurator menu allows the user to send different types of messages based on the custom UDP developed by Smart Fibres.
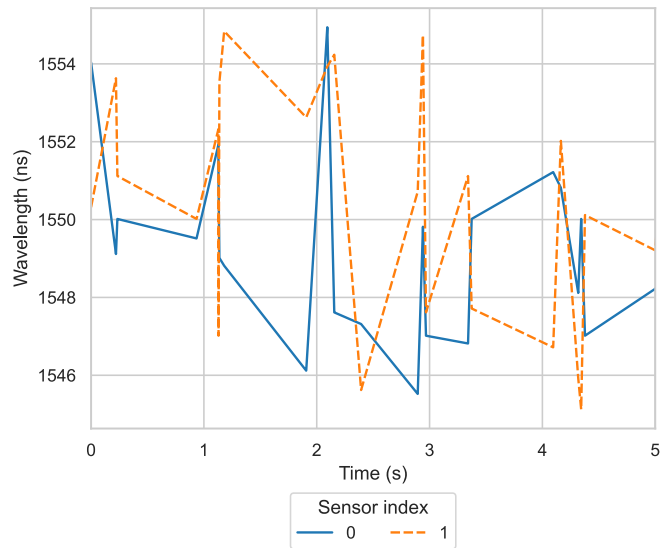

Fig. 5. Example of output provided by the emulator in case it is requested to simulate the output of 2 sensors connected to a single interrogator channel.

same local network. An image of the *Configurator* menu is shown in Fig. 4. The *Emulator* [15] and the *Configurator* [16] software and the *LibUtils* [17] and *LibSmartScan* [18] libraries are open-source, so their complete code is available on GitHub under the GNU General Public License v3.0.

IV. OUTPUT TESTS

This section shows some tests regarding the output data from the emulator. Various configurations of sensors were therefore reproduced and, for each of them, data were captured for about 5 minutes. It was decided to select rather high extreme wavelength values, between 1545 and 1555, and a
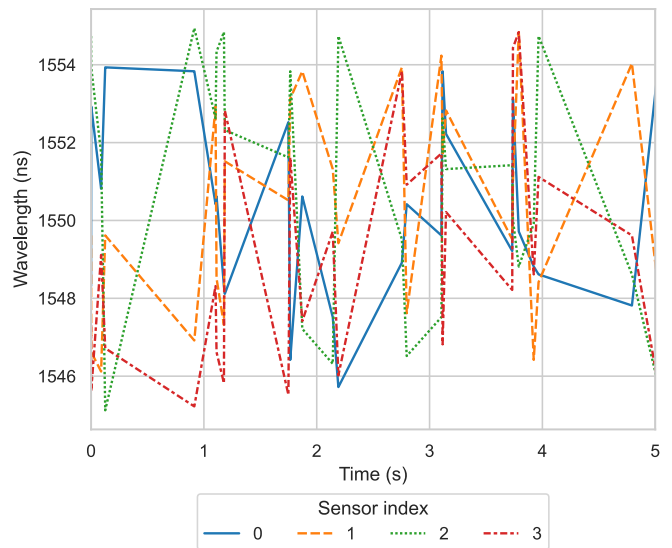

Fig. 6. Example of output provided by the emulator in case it is requested to simulate the output of 4 sensors connected to a single interrogator channel.
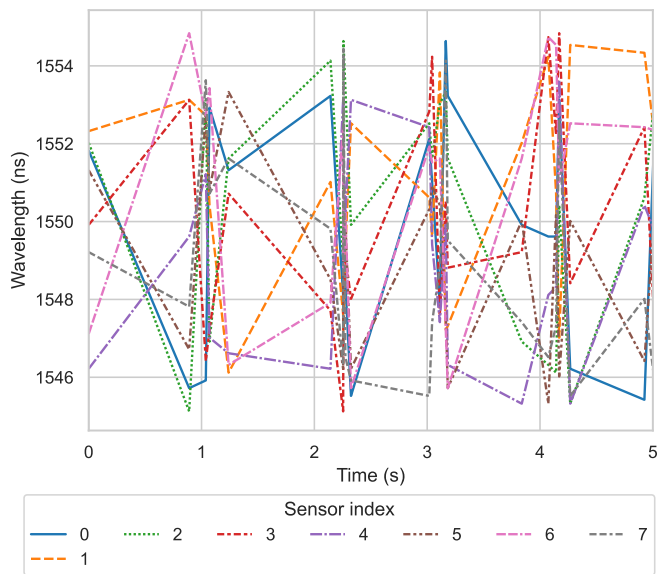
Fig. 7. Example of output provided by the emulator in case it is requested to simulate the output of 8 sensors connected to a single interrogator channel.

Fig. 9. Example of output provided by the emulator in case it is requested to simulate the output of 32 sensors connected to two interrogator channels.

sampling frequency of 5Hz per sensor. The configurations are summarized below:

- 2 sensors connected to the same channel.
- 4 sensors connected to the same channel.
- 8 sensors connected to the same channel.
- 16 sensors connected to the same channel.
- 32 sensors connected to two different channels.
- 64 sensors connected to four different channels, which is the maximum number of sensors that can be connected to the SmartScan interrogator.

For the sake of understanding, Fig. 5 to 10 show only the first 5 seconds of data. As can be seen from the images themselves, as the number of sensors increases, it becomes more difficult to understand the output data.
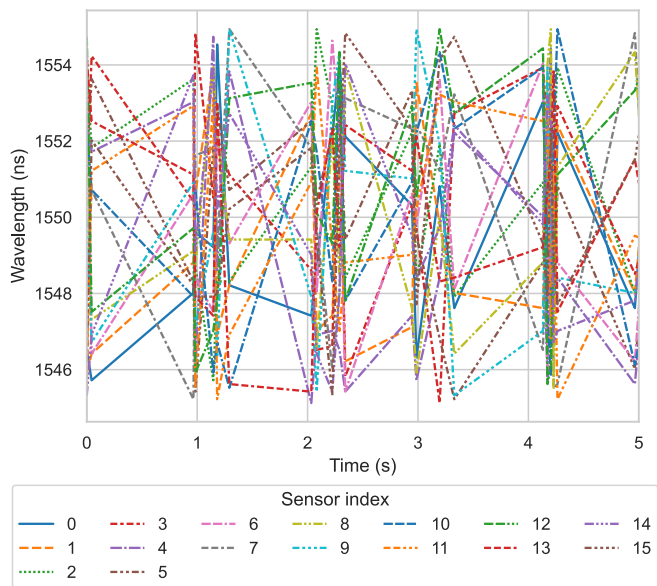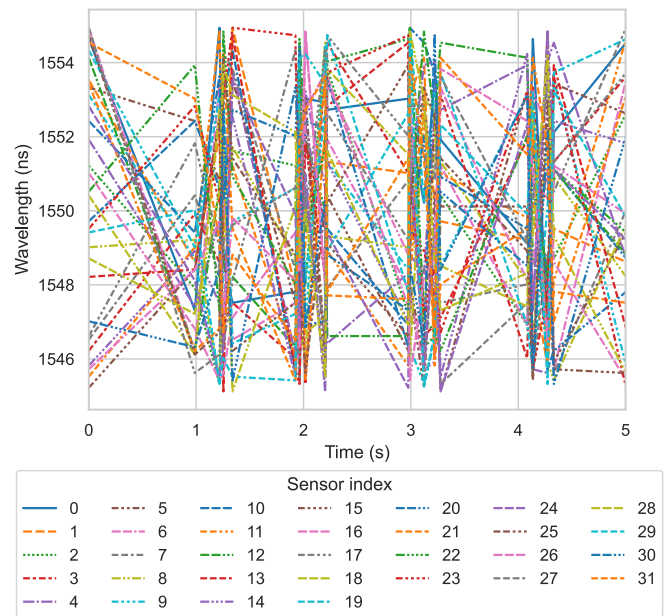

Fig. 8. Example of output provided by the emulator in case it is requested to simulate the output of 16 sensors connected to a single interrogator channel.
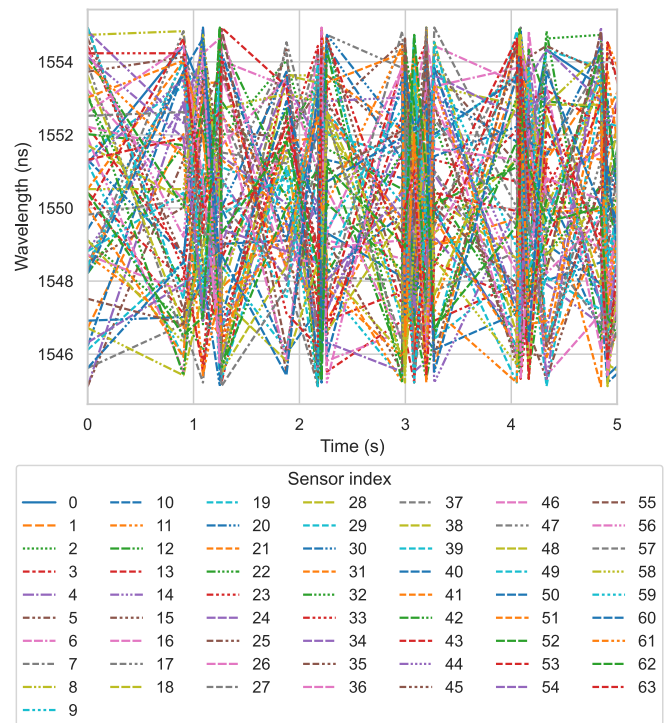

Fig. 10. Example of output provided by the emulator in case it is requested to simulate the output of 64 sensors connected to four interrogator channels.

V. Conclusions

This paper presented a novel software that is extremely useful for the continuous improvement of the flying test bench itself. Having an emulator of the real interrogator allows the developer of the system to avoid using it in the development phases of all the software that make up the system and to simulate complex configurations without having to physically build them. This is extremely useful for increasing the speed of test implementation and for being able to reconstruct even unusual situations for the current context to verify the reliability of the software developed up to now.

Acknowledgment

References

[1] A. Aimasso *et al.*, "Study of FBG-Based Optical Sensors for Thermal Measurements in Aerospace Applications," *Journal of Physics: Conference Series*, vol. 2293, no. 1, 2022.

[2] M. D. L. Dalla Vedova, P. C. Berri, and A. Aimasso, "Environmental Sensitivity of Fiber Bragg Grating Sensors for Aerospace Prognostics," *Proceedings of the 31st European Safety and Reliability Conference, ESREL 2021*, p. 1561 – 1567, 2021.

[3] A. Aimasso, M. D. Dalla Vedova, and P. Maggiore, "Analysis of fbg sensors performances when integrated using different methods for health and structural monitoring in aerospace applications," in *2022 6th International Conference on System Reliability and Safety (ICSRS)*. IEEE, 2022, pp. 138–144.

[4] A. Aimasso, M. D. Dalla Vedova, and P. Maggiore, "Innovative sensor networks for massive distributed thermal measurements in space applications under different environmental testing conditions," in *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE, 2022, pp. 503–508.

[5] A. Aimasso *et al.*, "Fiber bragg grating sensor networks enhance the in situ real-time monitoring capabilities of mli thermal blankets for space applications," *Micromachines*, vol. 14, no. 5, p. 926, 2023.

[6] R. E. Quigley Jr., "More electric aircraft," *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC*, pp. 906–911, 1993.

[7] D. Belmonte, M. D. L. Dalla Vedova, and P. Maggiore, "Prognostics of Onboard Electromechanical Actuators: A New Approach Based On Spectral Analysis Techniques," *International Review of Aerospace Engineering*, vol. 11, no. 3, pp. 96–103, 2018.

[8] C. Byington *et al.*, "A Model-Based Approach to Prognostics and Health Management for Flight Control Actuators," *IEEE Aerospace Conference Proceedings*, vol. 6, pp. 3551–3562, 2004.

[9] A. C. Marceddu *et al.*, "Air-To-Ground Transmission and Near Real-Time Visualization of FBG Sensor Data Via Cloud Database," *IEEE Sensors Journal*, pp. 1–1, 2022.

[10] A. C. Marceddu and B. Montrucchio, "Storage and Visualization On-The-Ground and in Near Real-Time of the Data Measured by the Optical Sensors Connected to a Flying Test Bench," *3rd Aerospace PhD Days*, Apr. 2023.

[11] Politecnico di Torino, "PhotoNext Middleware," https://github.com/CARDIGANSPoliTo/PhotoNext_Middleware, [Online]. Accessed on Jan. 12, 2023.

[12] M. Guerrera, "Algorithms and methods for Fiber Bragg Gratings sensor networks," Master's thesis, Politecnico di Torino, 2018.

[13] Politecnico di Torino, "PhotoNext 3D Viewer," https://github.com/CARDIGANSPoliTo/PhotoNext_3D_Viewer, [Online]. Accessed on Jan. 12, 2023.

[14] M. G. Canu, "Mixed Real-Time Visualization Framework for FGB IoT sensors," Master's thesis, Politecnico di Torino, 2019.

[15] Politecnico di Torino, "PhotoNext Emulator," https://github.com/CARDIGANSPoliTo/PhotoNext_Emulator, [Online]. Accessed on Jan. 12, 2023.

[16] Politecnico di Torino, "PhotoNext Configurator," https://github.com/CARDIGANSPoliTo/PhotoNext_Configurator, [Online]. Accessed on Jan. 12, 2023.

[17] Politecnico di Torino, "PhotoNext LibUtils," https://github.com/CARDIGANSPoliTo/PhotoNext_LibUtils, [Online]. Accessed on Jan. 12, 2023.

[18] Politecnico di Torino, "PhotoNext LibSmartScan," https://github.com/CARDIGANSPoliTo/PhotoNext_LibSmartScan, [Online]. Accessed on Jan. 12, 2023.