

## Test platform for electronic control units of high-performance safety-critical multi actuator systems

Giovanni Bucci, Fabrizio Ciancetta, Edoardo Fiorucci, Simone Mari

Department of Industrial and Information Engineering and Economics, University of L'Aquila, Italy

---

### Article Info

#### Article history:

Received Jun 10, 2019

Revised Mar 5, 2020

Accepted Mar 15, 2020

---

#### Keywords:

Multi actuator system

Multithreading technique

Synchronous handling

Test of electronic control unit

(ECU)

---

### ABSTRACT

In this paper we are mostly concerned with the problem of testing electronic control units of synchronized electric power actuators. This task is particularly complex for safety critical applications, where it is crucial that the control system properly reacts in response to the faults that are hard to reproduce and verify. A cost-effective flexible and reconfigurable test platform is proposed, discussing its architecture and implementation. The proposed system facilitates the phase of definition and development of the electronic control unit, allowing the interfacing towards both hydraulic and electromechanical actuators, and having a high flexibility as regards the I/O signals. Some results, obtained during the laboratory test activity, are also presented.

Copyright © 2020 Institute of Advanced Engineering and Science.

All rights reserved.

---

### Corresponding Author:

Edoardo Fiorucci,

Department of Industrial and Information Engineering and Economics,

University of L'Aquila,

Via G. Gronchi 18 Campo di Pile 67100, L'Aquila, Italy.

Email: edoardo.fiorucci@univaq.it

---

## 1. INTRODUCTION

Power actuators are essentially systems that provide controlled movements or positioning in response to a control signal. They embody motors, cylinders, or other motion devices to produce rotations or linear translations of an object. The mechanical components of the power transmission connect the actuator to the shifted object. As the process industry continues to achieve more efficient and productive plant, new actuators are designed and applied (e.g. packaging machines, production machinery, etc.). Many actuators are applied in medical equipment, automotive and aerospace systems, and even in civil engineering to move large surfaces (e.g. river flow barriers, roof segments of a stadium).

Regarding the type of power actuator, it is possible to identify two main categories of systems: pressurized fluid and electric motor. In the recent years, many companies are moving towards electric actuators, which show great advantages: (i) reduction of costs and dimensions due to the absence of additional fluid power supplies; (ii) greater efficiency; (iii) greater response speed; (iv) reduced maintenance problems and fire hazards. In recent years the development of electrical actuation systems has driven the tendency to replace the hydraulic devices hitherto used in a wide range of aircraft aerodynamic control surfaces (rudders, ailerons, horizontal stabilizers or flaps), with the aim of converging to the all-electric aircraft [1].

Due to the limited torsional rigidity of the moving device, in some applications more actuators work together to position it, sharing the load. In other cases, there are two or more axes of motion (either working independently or moving together) that must be coordinated (multi-axis synchronization) such as for printing presses, XYZ machines and robots. Multiple electric actuators are applied in industry, for example for drive wheels that are part of the same assembly line (in-wheel technology); they are also used on cars [2] or buses [3-5]. The control of the high-precision movement of the cooperating actuators is a rather difficult task. When several actuators move the same platform, the synchronization of positions and speeds must be

achieved with precision, in order to accurately determine its progression over time [6]. These aspects are also important for twin parallel operating actuators that equally divide the load between them to eliminate any force fighting. In some critical applications, after a failure of the power supply or actuator, the actuator concerned must be configured in a passive damping mode, thus reintroducing the sharing problem of the load [7].

In these advanced systems the position, velocity, acceleration and force/torque are controlled as a function of time, following a required target profile: the actual values derived from the feedback sensor are compared with those of the motion profile, to guide the actuator correctly. Also the closely connected actuators must all be in the same state at all times, accelerating, decelerating and stopping together. Furthermore, if one actuator stops due to a fault, the other must also stop. The performance of these actuators depends more on the coordination of multiple motion actuators than on individual actuator behaviours. Multi-actuator systems with good speed tracking and synchronization performance require electronic coordinating, monitoring and control units capable of precisely controlling both the position and the applied force [8]. Since complex systems may lead to failure and losses, high integrity and failure survival capability are frequently required (e.g. actuation for flight control). Redundancies and fault detections are implemented to quickly detect a fault, in order to avoid system failures. Various techniques are implemented, based on hardware redundancies (sensors, actuators, controllers) [9-13] or on actuator modeling and parameter estimation [10]. These additional components increase the complexity of the actuator controllers.

In the past the test of each individual actuator represented the most demanding test activity. Today these tests are part of routine activities and are defined in detail by specific standards. Currently the assessment of the electronic control units (ECUs) represents the main test activity. As the controllers grow in complexity, even the testing challenges [14-19]. Testing activities for the controller are particularly expensive, due to the high functional density and the numerous degrees of freedom in the controller configuration. Their performance must be validated against all operating conditions of the actuator. Most commonly, this will include position and load control through the complete stroke of the actuator at both minimum and maximum speeds. Furthermore, the I/O signal management section must be evaluated. Since communication interfaces also play an important role, it is necessary to carry out tests to verify the correct functioning of the basic and redundant communication system. In safety-critical applications it is essential that the control system reacts correctly in response to faults. To test these conditions, the test system must be able to emulate various error situations. Extensive tests in these atypical conditions are very important, also because failure events rarely occur, and therefore are difficult to reproduce and verify.

In this paper we proposed a cost-effective test platform for the automated testing of ECUs. To meet the test requirements of a multi-actuator system, the test system is able to simultaneously test a number of interconnected synchronized electric actuators. The proposed test bench is based on the NI compact RIO (cRIO) system, an advanced embedded control and data acquisition system. This test bench is a flexible and reconfigurable test platform, suitable for a wide range of applications and performance requirements. The test platform is able to generate all the signals required by the ECU, both as electrical I/O signals and as communication buses. A PC connected to the system acts as the main actuation controller and user interface. The proposed system is scalable, based on the number of ECUs.

Using the proposed modular system, it is possible to test ECU for commercially available electric actuators; besides, it is possible to efficiently perform functional tests of the ECU during the early stages of development. It can also easily verify strict quality compliance. An important test for multi actuator systems is that related to the correct functioning of the communication interfaces. A key challenge for testing safety-critical systems is the position synchronization test for the tracking error measurement.

Our proposal is essentially motivated by the impossibility of finding specific test systems for these types of applications on the market. Furthermore, similar test systems do not appear to be described in the recent bibliography. Some test devices are commercially available, but specifically for testing control units for internal combustion engines the proposed test platform is cost-effective because it is based on commercially available instruments. It provides an intuitive environment for generating test profiles, alarms, user interface and data acquisition. In the following sections we will discuss the architecture and implementation of the test platform. Furthermore, the results of the laboratory tests are presented.

## **2. THE REFERENCE ARCHITECTURE OF THE HIGH-PERFORMANCE MULTI ACTUATOR SYSTEM**

The system taken as a reference is an advanced and reliable multi actuator system with a number of independent actuators, arranged to track the desired trajectories while keeping their speed the same. As is normally the case, a movement control computer (MCC), interfaced with the operator, sends commands to a supervisor electronic control unit (SECU). The SECU interfaces some electronic control units (ECUs), each of which operates as system controller for the related electro-mechanical actuator (EMA), as shown in Figure 1.

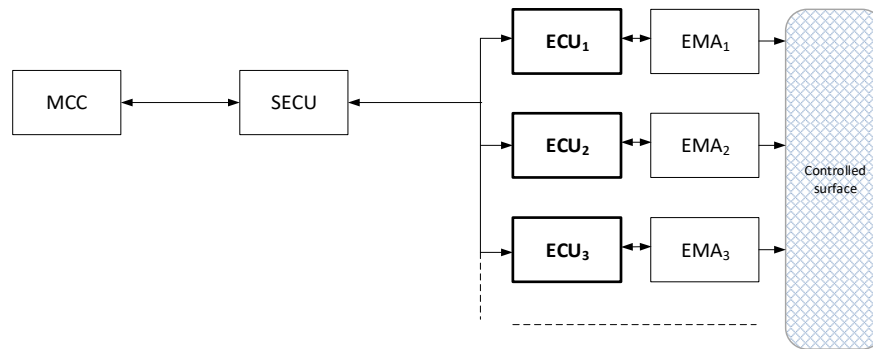


Figure 1. Architecture of a multi actuator system

The MCC receives and processes SECU's data and operator inputs, in order to monitor and control the actuators. ECUs are essentially embedded devices that communicate with the SECU in order to interact and coordinate their behaviour. More in detail, the SECU manages the MCC commands, sets the EMAs operating mode, controls the position and speed of whole surface, and distributes the load setting the force at each actuator. Each ECU receives commands from SECU, performs the EMA close-loop position control, and returns measurement data (including position from a dual LVDT and force) to the SECU. While the SECU can be located away, the ECU resides close to or are mounted on the actuator, to better provide for its local control.

In some safety critical applications, each actuator is controlled by a different SECU, so the loss of a single actuator, or of a SECU, will not mean the overall control loss of the surface. Each ECU is then connected to an associated SECU as shown in Figure 2, a SECU fault causes the out of service of just an actuator, that can be put in the anti-jamming condition. In this way the other actuators can still move the controlled surface in a safe position, even though operating with reduced performance. All SECUs operate in parallel, so there is no switching delay. The data link MCC – SECU is also duplicated.

Besides, in reliable systems a prognostic monitoring system (PMS) is required, in order to performs the diagnosis of the actuator and evaluate its reliability, also in terms of life expectancy. The PMS, embodied in the electronic control and monitoring units (ECMU), controls and analyses the actuator status, starting from actuator parameters evolution and sensor signal behaviour. The SECU obtains ECMU's sensor measurements and monitoring results through the ECMU-SECU communication link.

As well as the benefit of very high reliability, the SECU redundancy allows the overall functionality of SECU to be included in the related ECMU (just a single apparatus), reducing so the hardware dimensions and costs. A fault-tolerant architecture was implemented by replicating the SECU and implementing the ring architecture shown in Figure 2, where each SECU is linked with two other SECUs. This architecture makes possible to share information (SECU's internal status and connected ECMU internal status and measurements) with the two other SECUs linked in the ring chain. The connection Next SECU - Previous SECU is implemented using a private link, to avoid any latency for message transmission and sender identification. In case of an ECMU or EMA failure, the associated SECU puts the failed actuator in fail-safe mode and signals this event to the other SECUs. In this way, the controlled surface can be moved by the other actuators, even if with reduced mechanical performances.

Considering the system shown in Figure 2, the main MCC operating tasks are:

- Creation of a position vs. Time profile;
- Sending the related set-position frame to the secus;
- Sending specific ecmu control commands to the secus;
- Receiving and processing of secus internal state;
- Receiving and processing of ecmus internal state.

The secu tasks are instead:

- Receiving the set-position frame from mcc;
- Sending the set-position frame to the ecmu;
- Receiving (continuously) the measurement frame from ecmu;
- Receiving the position measurements from two linear displacement transducers (lvdts) connected to the ecmu;
- Receiving and sending to mcc the internal status of the associated ecmu;
- Checking the status of other actuators;
- Enabling the ecmu operating mode, using dedicate digital outputs;
- Checking the ecmu state via a dedicate digital input.

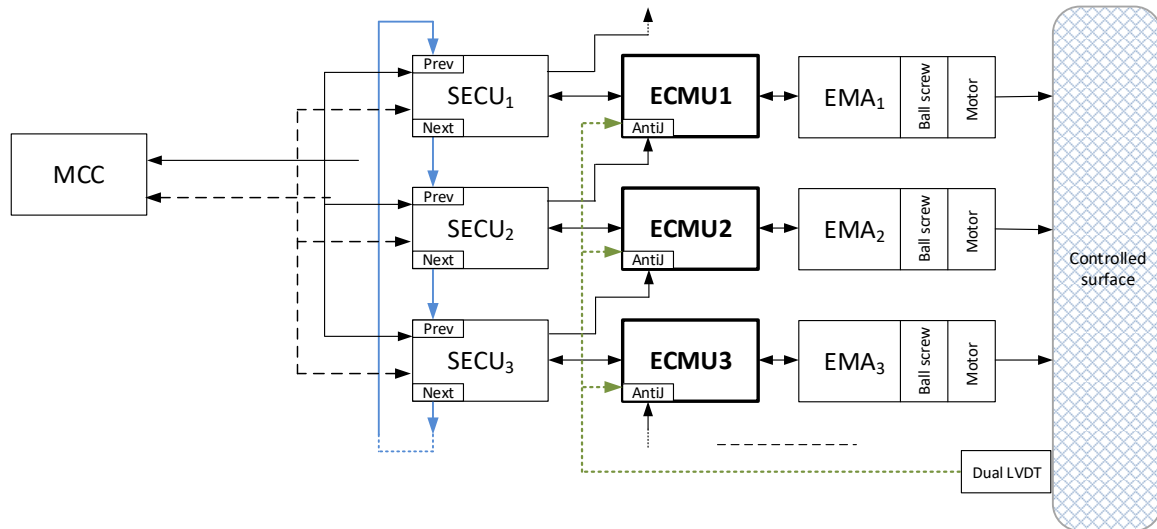


Figure 2. Architecture of a redundant multi actuator system (ECMU: ECU+PMS)

### 2.1. The reference actuator control and monitoring unit (ECMU) under test

The implemented test platform has been developed for testing the controllers (ECMUs) for high-performance actuators. The surface is powered by some independent actuators that operate synchronously. After a power supply or actuator failure the affected actuator is placed into a passive Damping mode in which it is not controlled in position close loop any more. Therefore, this actuator is controlled to act as a damper in which the force is a function of the speed imposed by the other actuators. This mode of operation is possible thanks to the reversible capability of actuator with a low backdrive force. In this way, the actuator can impose the force required to avoid any surface fluttering.

A stiffer actuator failure or a SECU failure can produce a mechanical lock that prevents the shaft to rotate (jamming event), so there is a position error that cannot be automatically compensated. In this case the arise of an imbalance of force can damage the system. In case of a non-recoverable failure (e.g. a broken-tooth in a gear), the previous (in the ring chain) SECU puts the failed actuator in the anti-jamming (AJ) mode, to prevent any possible malfunction. The AJ system is generally located inside the screw and mainly allows disconnecting the actuator from the surface, keeping the control by the other actuators. The Fail-Safe is a feature that, in the event of a specific type of failure, inherently responds in a way that will cause minimal harm to other equipment, or to people. When in the Fail-Safe mode, the SECU operates in order activating the AJ if needed, putting in damping the failed actuators and moving the surface to the home position. The Figure 3 shows the state diagram of a typical ECMU; the states are:

- a) *Off mode*: This is the state assumed by the system at the start up. When in this mode, the ECMU does not send data and ignores all received frames.
- b) *Pre-Run mode*: after the start-up the ECMU executes some tests to check the right and full functionality, and then starts sending data, but ignores all received data.
- c) *Run (Normal) mode*: the ECMU provides the surface position control according to the SECU request, the ECMU performs the following tasks:
  - *Damping*: the ECMU does not control the position;
  - *Performing Control*: the ECMU controls the position;
  - *Homing*: the ECMU performs a position control towards a predetermined referenced (home) position.
- d) *Anti-Jamming mode*: when a jamming condition is detected, the AJ procedure is activated disengaging the EMA (the ECMU does not control the position);
- e) *Fail-Safe mode*: after a failure detection the actuator is put in damping mode (the ECMU does not control the position).
- f) *Maintenance mode*: during this mode the calibration, configuration and maintenance operations are performed.

The ECMU is connected to the SECU with:

- 1) Two RS-485 links (one for receiving set-position frames, the other for sending measurement frames);
- 2) Digital I/O signals:

- a) ECMU inputs:
- *Enable*: is active when the SECU puts the ECMU in control position (*enable state ON*) or in damping state (*enable state OFF*);
  - *Homing*: is active when the SECU requires to put the actuator in the home position;
  - *Anti-Jamming E*: is active when the SECU enables the AJ mode of the ECMU associated with the SECU\_N.
- b) ECMU outputs:
- *Ready*: active when the ECMU is in Ready mode;
  - *Run*: active when the ECMU is in Run mode.
- ECMU connection overview as shown in Figure 4.

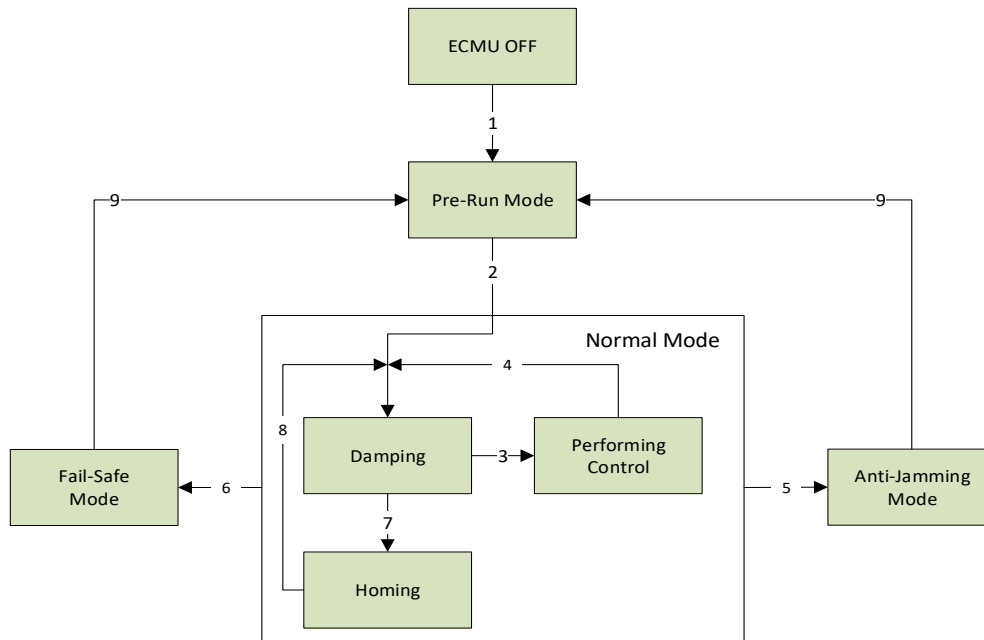


Figure 3. State and transition diagram related to the ECMU under test

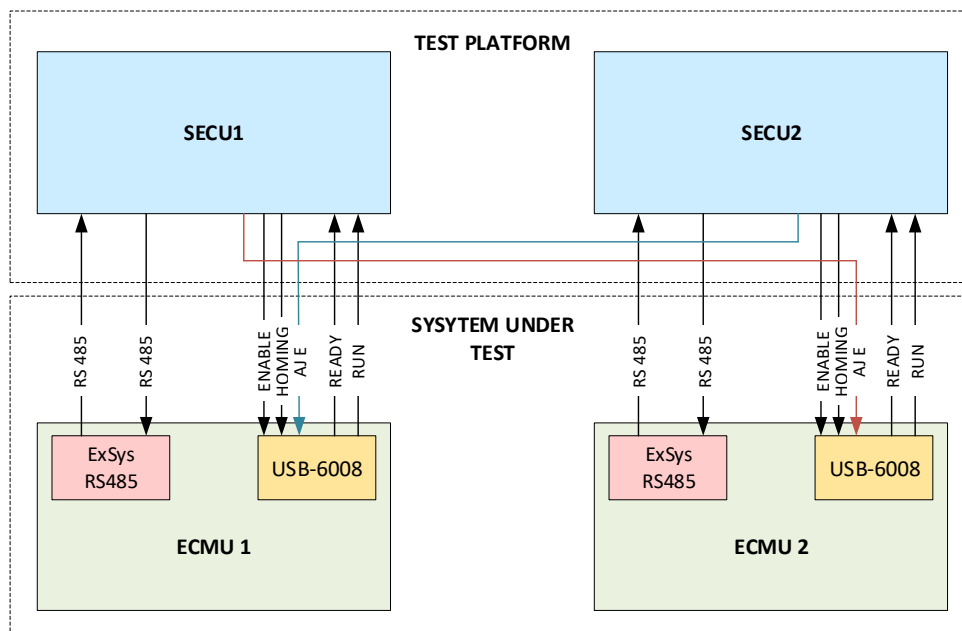


Figure 4. ECMU connection overview

The ECMU executes these main tasks:

- a) Reception of the set-position frame from the associated SECU;
- b) Generation of an internal position that linearly follows the set-positions given by SECU;
- c) Execution of the position control loop;
- d) Measurement of the motor controlled surface position (two lvdts);
- e) Measurement of the motor angular position (resolver);
- f) Measurement of the motor temperature (two Pt100 transducers);
- g) Measurement of the motor torque (two strain gauges);
- h) Measurement of the acceleration (two transducers) for the prognostic system;
- i) Transmission of the measurement frames to the SECU;
- j) Change of its state when required by the SECU;
- k) Setting of the digital output signals based on its internal state.

The ECMU periodically sends data to the associated SECU that sends them to MCC, SECU\_P and SECU\_N. In this way SECU\_P and SECU\_N check if SECU is working properly. The ECMU generally has a number of internal bits that identify its internal status; an example related to our system is shown in Table 1.

Table 1. ECMU status bits

bit	Flag	bit	Flag	bit	Flag
0	Pre-Run mode	12	28VDC Left Hand failure	24	Phase error 1 > threshold
1	Active mode	13	Motor temperature 1 error	25	Phase error 2 > threshold
2	Fail-Safe mode	14	Motor temperature 2 error	26	End of travel - Fail Retracted
3	Anti-Jamming mode	15	ECMU temperature error	27	End of travel - Fail Extended
4	Maintenance mode	16	Resolver 1 failure	28	AJS Motor check
5	300VDC motor check	17	Resolver 2 failure	29	RAM Access
6	Max current	18	Resolver 1 input < threshold	30	Firmware checksum error
7	Max current variance	19	Resolver 1 input > threshold	31	Non-volatile memory access
8	300VDC Right Hand failure	20	Resolver 1 amplitude mismatch	32	Resolver to digital access 1
9	300VDC Left Hand failure	21	Resolver 2 input < threshold	33	Resolver to digital access 2
10	300VDC Voltage limit	22	Resolver 2 input > threshold	...	Spare
11	28VDC Right Hand failure	23	Resolver 2 amplitude mismatch	47	Spare

### 3. THE PROPOSED TEST PLATFORM

#### 3.1. The hardware architecture of the proposed test platform

The complexity of ECMU, both in the electronics and software, makes comprehensive tests necessary in all development phases, in order to early detect and correct errors. This is why each ECMU undergoes in-depth testing, followed by integration tests and standard tests that are run on fully equipped test stands. Testing ECMU functionality includes stimulating it via hardware and software interfaces and evaluating its responses. Functional testing of ECMUs, besides testing for basic functionality, must also test the most significant faults at the ECMU's communication and I/O interfaces. These considerations place strict requirements on robust, high-capacity test systems in terms of interfaces and test hardware, test automation, operation of software interfaces. Besides, the proposed test system requires intensive calculations, and precise synchronizations.

To satisfy these requirements, we implemented it using a NI Compact RIO (cRIO) system. This is an advanced embedded control and data acquisition system, designed for applications that require high performance and reliability, as well as flexibility and scalability. A cRIO system essentially consists of four main components: an embedded real-time controller, a high-performance FPGA, I/O modules, and the application program. Each I/O module is directly connected to the FPGA, providing low-level customization of timing and I/O signal processing. The FPGA is connected to the embedded real-time processor via a high-speed PCI bus. This platform makes possible to adopt multicore processing and high-performance FPGAs, in order to improve the test system performance. The proposed test system has a modular architecture, based on a MCC and some SECUs. In this research phase we implemented two SECU units (SECU1 and SECU2) each of which includes as shown in Figure 5:

- a) One CRIO-9039, Embedded crio controller with real-time processor and reconfigurable FPGA (1.91 ghz, quad-core, 8-SLOT, 325T FPGA);
- b) One NI PS-15 power supply, 24 VDC, 5 A;
- c) Two NI 9860 modules: 2-port hardware-selectable NI-XNET CAN/LIN (local interconnect network) compactdaq and crio interface for developing CAN and/or LIN applications, for the connection to SECU\_P and SECU\_N;

- d) One NI 9853 2-port high-speed CAN module (this module, not used for the referenced application, will be used for interfacing hydraulic actuators);
- e) One NI 9871 module: 4 serial port RS-422/RS-485 accessed directly from the FPGA;
- f) Two NI 9401 modules: 5 V/TTL, 8 channels, bidirectional digital I/O.

A Host PC is connected to each SECU, in order to show the test results. As shown in Figure 5 the MCC has been implemented using:

- a) A Personal Computer with Windows installed;
- b) A Komodo CAN Duo Interface linked to a PC USB 2.0 port, for the connection to SECU1 and SECU2;
- c) A Labview application program.

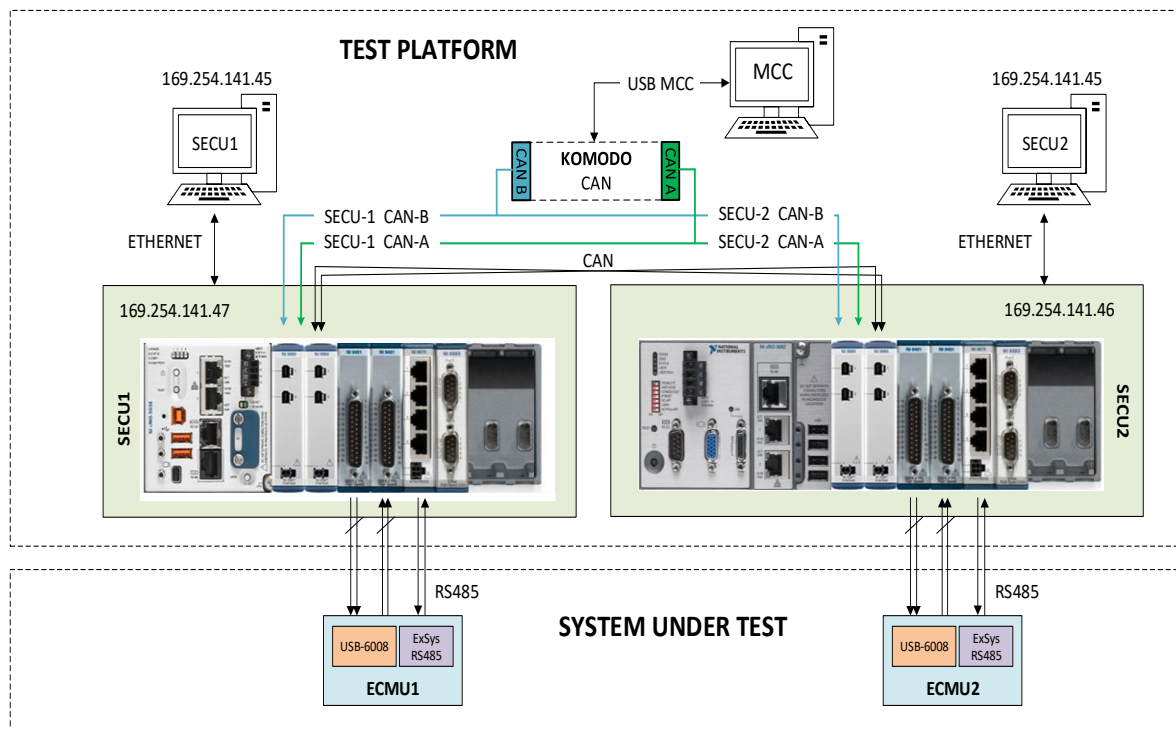


Figure 5. Architecture of the implemented test platform

The MCC writes the same message to both KOMODO CAN output ports at the same time, and receives messages from both SECU1 and SECU2 on the two ports. Using the MCC application program the operator can:

- a) Configure the interface used to communicate with the secus;
- b) Create the set-position profile as shown in Figure 6, the samples are sent to the two secus as a set-position frame;
- c) Start and stop the profile frames transmission (in our application the frame is sent every 20 ms);
- d) Send commands to the secus to control the ecms, by means of the ENABLE digital signal as shown in Figure 4;
- e) Enable/Disable the Anti-Jamming, by means of the AJE digital signal;
- f) Send commands to the secus to perform a position control towards the home position;
- g) Save data to/from secus; the MCC saves a NI TDMS (technical data management streaming) file that contains all data sent/received to/from the secus.

When the MCC sends a command or requires a change on a Configuration Register, the SECUs acknowledge the message sending an ACK message. The MCC displays information about the ACKs received from the SECU on the Main Log window. The MCC receives an ACK only if the transmitted command has been correctly received and validated. If no ACK is received after a timeout (150 ms in our case), the Main Log window displays a warning and requests to resend the message. If the message is sent to both the SECUs (Broadcast) the MCC will wait for two ACKs (from SECU1 and SECU2), if at least one is not received it displays an error anyway.

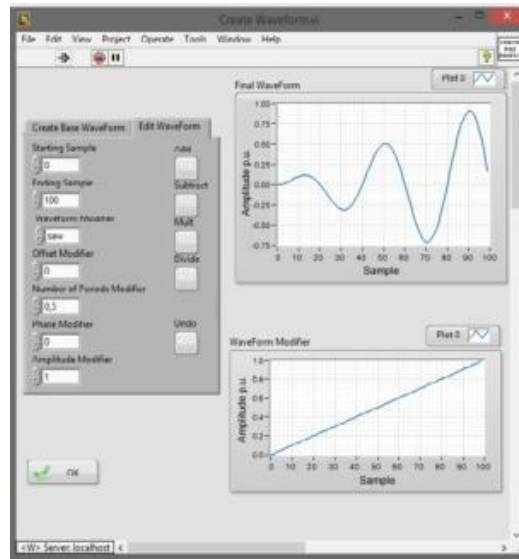


Figure 6. Page for the profile generation

### 3.2. The implemented communication system of the proposed test platform

The communication standard adopted for the MCC-to-SECU and SECU-to-SECU links is the controller area network (CAN) serial standard. Due to the standardization of part of the physical and data link layers of the OSI reference model in ISO 11898 [20] and the openness of the protocol, CAN allows devices, sensors and actuators from different manufacturers to communicate. Originally developed for automotive applications, it attracted the attention of manufacturers in other industries that adopt automation and enhanced sensors/actuators [21]. The remaining physical layer and all of the higher layers are not defined by CAN specification.

CAN supports distributed real-time control with a high level of security. The data signal is normally transmitted on a twisted pair of wires, with a maximum bit rate of 1 Mbit/s (cable lengths < 40 m). Data collisions on the bus are avoided using both the carrier sense multiple access and arbitration based on message priority technologies. The number of terminals connected to the bus is normally limited to 32, to avoid data transmission delay.

Commercial aircraft have already adopted the CAN bus [22]. In order to ensure interoperability between CAN subsystems, the Network Infrastructure and Security group of the airlines electronic engineering committee (AEEC) has defined the ARINC 825 aviation protocol standard that specifies both the communication within CAN-based sub-systems, and between CAN sub-systems [23]. ARINC825 will be the CAN standard for all future Airbus and Boeing aircrafts.

When the MCC sends a frame to the SECU, each SECU is identified in the CAN network with a unique “Base Identifier” (ID). CAN supports two transferred message (frame) formats, as defined in the “Bosch version 2.0 specifications” [20]; the essential difference between them is the length of the ID: 11 bits in the standard frame format (2.0A), and 29 bits in the extended frame format (2.0B). In our case the default values are:

- a) Messages from MCC to SECU1: 0x01;
- b) Messages from MCC to SECU2: 0x02;
- c) Messages from SECU1 to MCC: 0x10;
- d) Messages from SECU2 to MCC: 0x20;
- e) Broadcast messages: 0xff.

The adopted protocol fields, shown in Figure 7, are:

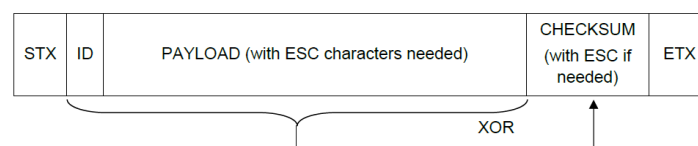


Figure 7. The adopted CAN message format



- a) *STX* (0x02) is a control character that indicates the start of a new frame.
- b) *ID*: frame identification:
  - 'M' (0x4D): frame with measurements from SECU to MCC;
  - 'S' (0x53): position set point from MCC to SECU;
  - 'O' (0x4F): frame received correctly from SECU to MCC (message without payload);
  - 'E' (0x45): frame received with error from SECU to MCC (message without payload);
  - 'J' (0x4A): frame to enable/disable the AJ;
  - 'G' (0x47): frame to start the Homing procedure;
  - 'R' (0x52): frame to notify the end of Homing procedure (from SECU to MCC);
  - 'K' (0x4B): frame to mask the AJ internal register;
  - 'P' (0x50): frame to mask the damping internal register;
  - 'A' (0x41): frame to update the global enable anti-jamming register;
  - 'U' (0x55): frame to update the global enable damping register.
- c) *Payload* is the useful information plus the ESC (0x1B) character. Each byte of information equal to STX, ETX or ESC is preceded by an ESC character.
- d) *Checksum* is a byte containing the result of the exclusive OR logical operation (CRC) of both ID and payload, executed by the transmitter. The receiver repeats the calculation to check if the frame is correct.
- e) *ETX* (0x03) is the control character that indicates the end of the frame.

An acknowledgment technique has been implemented for all the frame, in order to check the received data. Only for both the measurement and set point frames no flow control is applied: if a frame is lost, it is not recovered. The main reason for this choice is that there is not enough time among the frames to repeat them. In case of a problem, the receiver waits for the next frame with new data.

The communication between SECUs is based on the CAN bus. The SECU1 is identified by ID = 0x12, SECU by 0x21. A CRC field is used to check the frame and to request the retransmission. The payload frame is composed of two sub-frames. The first 30 bytes are the last ECMU measurement frame as shown in Table 2. The ECMU transmits data measured from redundant transducers that are checked (validated) by the SECU to see if they are in the range. Values out of the range are not sent to the other SECUs. In this case the valid data are those related to the last validated frame. The last 4 bytes, related to the SECU internal status bits, are shared with the other SECUs. The SECU status bits as shown in Table 3.

Table 2. The measurement frame

Data	Dimension (bytes)
ID	1
validated LVDT measurement	4
validated ECMU DC Bus current	4
validated ECMU DC Bus voltage	4
validated ECMU temperature	4
validated ECMU PCB temperature	2
validated ECMU Torque	4
ECMU Status bits 0-7	1
ECMU Status bits 8-15	1
ECMU Status bits 16-23	1
ECMU Status bits 24-31	1
ECMU Status bits 32-39	1
ECMU Status bits 40-47	1
CRC	1
SECU Status bits 0-7	1
SECU Status bits 8-15	1
SECU Status bits 16-23	1
SECU Status bits 24-31	1

Table 3. The SECU status bits

bit	bit
0 ECMU cable error	11 MCC Cable error
1 ECMU validation error	12 MCC AJ request
2 SECU_N cable error	13 ECMU tracking error
3 SECU_P cable error	14 DAMP for ECMU Fail-Safe
4 Force Fighting	15 DAMP for ECMU AJ
5 ECMU Ready	16 Damp for AJ
6 Enable state	17 De force fighting
7 AJ state	18 Spare
8 ECMU run	.. ...
9 Damping state	.. ...
10 AJ request	31 Spare

The communication between SECU and ECMU is based on two RS-485 differential communication link that is the most commonly used in commercial actuators. One link is used to transmit the position set-point to ECMU and the other to receive the ECMU measurement frame. In our case the bit rate was set at 115.2 kbit/s, the frame period at 20 ms. The main SECU internal registers are:

- a) Global Damp Reg: used to configure the Damping Mode behaviour of the SECU;
- b) Global AJ Reg: used to configure the Anti-Jamming Mode behaviour of the SECU;
- c) Mask Damp Reg: used to set the Damping Mode behaviour based on the ECMU Status bits;
- d) Mask AJ Reg: used to set the Anti-Jamming Mode behaviour based on the ECMU Status bits.

The implemented test system is modular and lets users set up flexible test systems with little integration or wiring effort. It drives ECMU inputs and outputs for functional testing with the CAN links. It lets users set up compact test benches of various complexities. Due to its modular construction, it is suitable for small test setups at developer workbenches as well as for comprehensive test benches in the testing laboratory.

### 3.3. The software architecture of the proposed test platform

The test application program has been developed in the NI LabVIEW environment. This widely applied tool helps to reduce the software development time and to enhance the processing performance. Many high-end computers today offer two or more processor cores for additional computation power. A key approach for taking advantage of multicore processors is the multithreaded programming. In a multithreaded application several threads can simultaneously run on different processors. Multithreading is similar to multiprocessing, the main difference is that a thread has a reduced complexity and is used for a small task. Threads are sequential processes that share memory. They can manage I/O more efficiently because a thread waits for I/O while others are processing the acquired signals.

This technology is not easily applicable: problems comes when trying to write a software that uses these processor cores effectively [24]. Single-threaded applications can run on only a single processor, thus preventing them from taking advantage of the multiple processors to improve performance. Therefore, to achieve maximum performance from multithreaded operating systems and/or multiprocessor machines, an application must be multithreaded. One technique is to partition the application into multiple threads that are scheduled across the multiple processing CPU cores by the operating system. The LabView native parallel programming capabilities automatically run independent parallel sections of code on the different cores [25].

The execution threads generally need to communicate with one another as they work. A solution to this problem, typical of parallel computations, is that each stage adds data to the input queue for the next stage when it is done. To properly work, the input queue needs to be written so that data can safely be added by one thread and removed by another one without corrupting the data structure. In our application we based the implemented thread communication on the producer-consumer paradigm that manages data message from different threads. In this way, the data are protected by multiple accesses and every thread has been developed using a block system: the thread is blocked during the processing of data. If another message has been sent to the thread, it is stored in a FIFO buffer so it can be processed when the thread ends the processing of the previous data.

#### 3.3.1. The main application program threads of the proposed test platform

The *MCC thread* is the thread-based SECU firmware that manages the interaction with the MCC. Security in data transmission is of utmost importance. As the SECU–MCC link is concerned, the security is based on the channel redundancy, so commands or data are validated if they are identical on the two channels. As the SECU-ECMU link is concerned, the SECU check that they are in the permissible range. As shown in Figure 8, this thread is subdivided into more subthreads.

At first the SECU thread opens two bi-directional CAN connections towards the MCC, then three threads are used to receive and send data from/to the MCCs: one thread simultaneously sends data to CAN\_A and CAN\_B interfaces, while two threads are used to receive data from MCC (one for CAN\_A and one for CAN\_B). Unlike transmission, two threads are for the two data reception because they can be not simultaneous. The two received data packets (one for every CAN channel) are then sent to two threads that put the messages in two buffered circular FIFOs. These threads identify the beginning and end of every MCC message. After the identification, the *Validation Data Thread* compares the two data packets to verify they are the same.

After this validation the message is processed by SECU firmware. If the two CAN interfaces supply two different messages, these are discharged. The *MCC Manager Thread* examines the CAN message and sends the received data to the firmware section concerned. Other sub threads are:

- a) Check cable thread: when it detects that the messages were from a single CAN interface, it generates an error message that is handled by the SECU;

- b) Frame counter thread: this thread increments the internal register assigned to count the received CAN messages;
- c) De force fighting thread: when the Force Fighting is detected (actual position is different from the set-points programmed by MCC), the SECU forces an AJ. During the AJ phase, the SECU uses as position set-point the actual position supplied by SECU\_P.

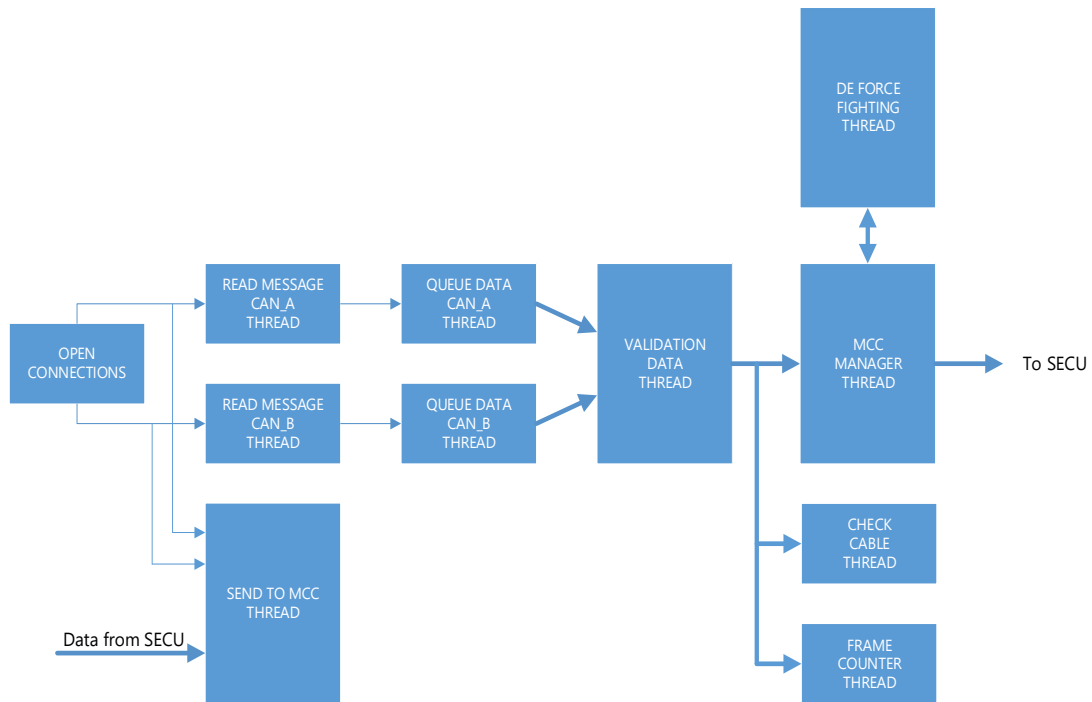


Figure 8. SECU threads related to the interaction with the MCC

The *ECMU thread* is the SECU firmware that manages the interaction with the associated ECMU. As shown in Figure 9, as the first operation the SECU opens two half-duplex RS-485 connections towards the ECMU. These connections are managed by two different threads. One thread is used to read data from SECU and the other thread is used to send data to SECU.

The received bytes are queued in a circular FIFO by the *Queue Data Thread*. When the correct number of ECMU bytes is recognized, the *Validation Data Thread* checks if the measures supplied by ECMU transducers are in the range, otherwise they are rejected. After validated, the message is processed by the *ECMU Manager Thread* that mainly transfers the message to the concerned firmware section. The other ECMU subthreads are:

- a) Check coherence thread: This thread verifies the “coherence” between the actual position (measured by the EMA transducer) and the expected (set-point) position: if their difference is outside a specific range the tracking error is generated.
- b) Check cable thread: when this thread detects the absence of ECMU messages, it generates an error message that is handled by the SECU.
- c) Frame counter thread: this thread increments the internal register assigned to count the received RS-485 messages.

The *SECU to SECU thread* manages the interaction with both SECU\_N and SECU\_P.

In our application, we set to automatically activate this thread every 50 ms, even if this time interval can be changed uploading a new configuration file. As shown in Figure 10, the first operation performed is relative to the opening of two bidirectional CAN connections towards two SECUs. Besides, two threads send and two receive data to/from the SECUs.

The SECU receives data from ECMU that are validated and then transmitted to the other SECUs.

- a) The SECU\_N (and SECU\_P) Manager Thread receives and processes the SECU data.
- b) The Force Fighting Manager checks if the ECMU is in the Force Fighting condition.
- c) The AJ Manager checks if an AJ request has been received from SECU\_P.

The *ECMU Manager Thread* processes the ECMU message and sends it to the concerned firmware section. Other threads internal to the *SECU to SECU thread* are:

- a) Check cable thread: when this thread detects the absence of SECU messages, it generates an error message that is handled by the SECU.
- b) Frame counter thread: this thread increments the internal register assigned to count the received CAN messages from SECU.

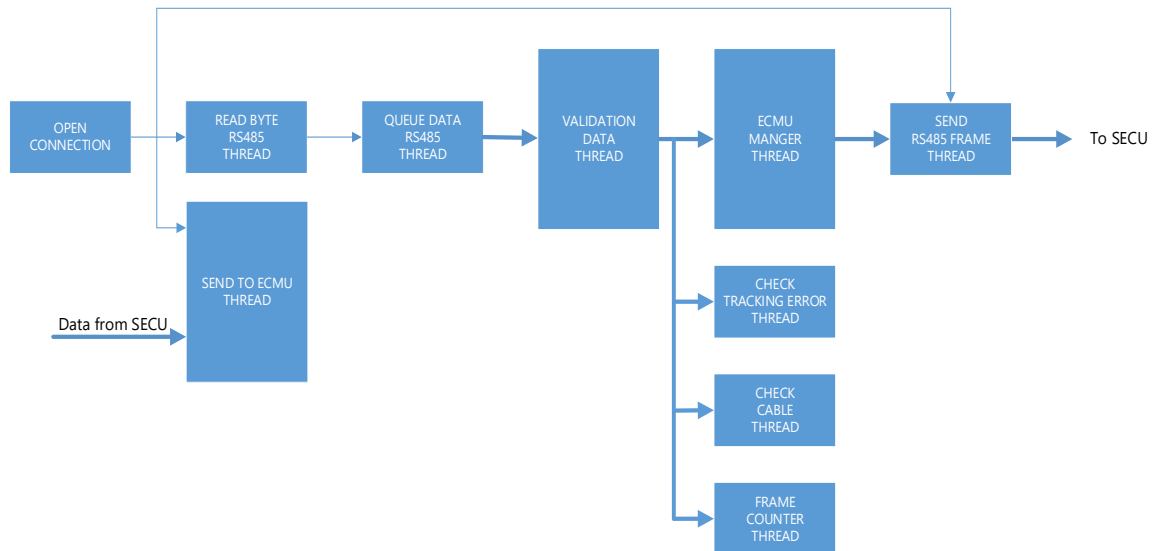


Figure 9. SECU threads related to the interaction with the ECMU

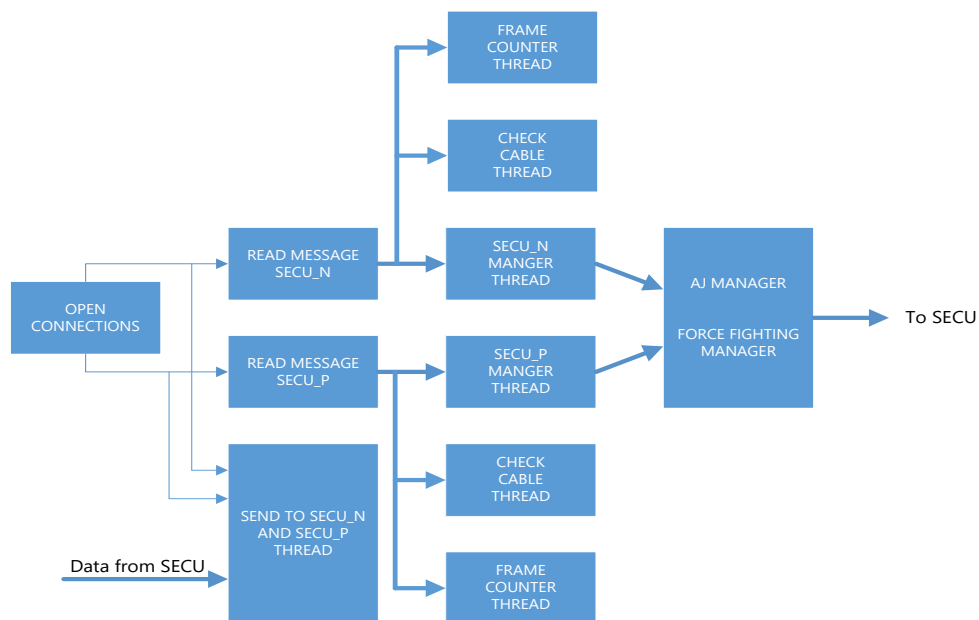


Figure 10. SECU threads related to the interaction with the other SECUs

The *DAMP thread* activates or deactivates the ECMU damping state. The load damp (disconnection) mode is active when the SECU handle a Fail-Safe error and the mode is enabled (enable bit is on). In damp mode the motor is disconnected from the power electronic system. The *AJ thread* generates an AJ request when the SECU handles an error and mode is enabled. This request is directed to *SECU\_N* that enables the ECMU AJ system as shown Figure 2.

#### 4. THE POSITION SYNCHRONIZATION VERIFICATION

As previously introduced, a key challenge for high-performance multi actuator systems is the position synchronization test, for the tracking error measurement. This test allows to verify the ability of the ECMU to correctly detect the tracking error and to implement the necessary corrective actions, for the purpose of the integrity and safety of the controlled surface (system).

In our implementation the SECU can handle many error sources, in order to adapt the implemented test system to the actual actuators under test. To give more flexibility the final user selects (from the MCC) the errors that put the SECU in damping mode and those that force the AJ request to the SECU\_N. To this aim we used two mask registers: the AJ GLOBAL ENABLE mask and the DAMP GLOBAL ENABLE, whose bits enable the relative error to activate the Anti Jamming or Damping mode, respectively. When the ECMU status bits are in a suitable configuration, the relative ECMU AJ request or ECMU DAMP request are enabled. The ECMU status bit that enables the ECMU AJ or DAMP request is selectable via another mask.

As previously introduced, multiple actuators that move the same platform require a precise synchronization in order to both accurately control its positioning trend over time (minimizing the tracking error) and to equally divide the load between them (minimizing the force fighting). Each SECU sets the actuator operating mode, to control the instantaneous position of whole surface and distributes the load. Each SECU controls the ECMU minimizing the tracking error and force fighting.

The *tracking error* is checked by evaluating the difference between the validated MCC reference position and the ECMU measured position. When this difference is greater than a fixed threshold  $\Delta_T$ , the associated SECU generates the tracking error, as shown in Figure 11. This difference is evaluated by each SECU for the corresponding actuator ( $\Delta_1$  for ECMU1 and  $\Delta_2$  for ECMU2).

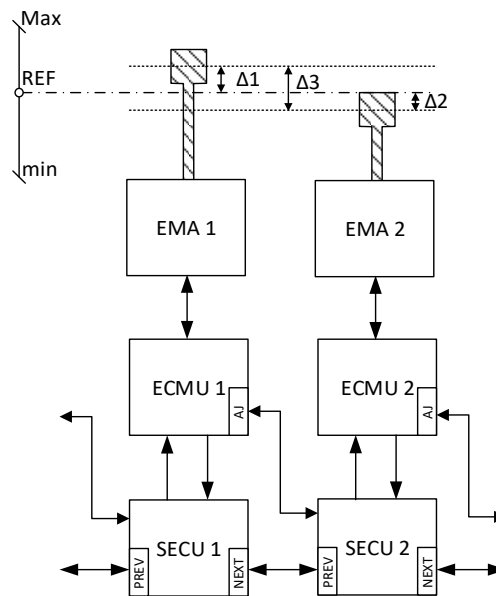


Figure 11. Tracking error

When all the actuators move at the same speed, all apply the same force. A number of reasons (manufacturing tolerance, difference between mechanical or electronic components, disturbances) may lead to a fighting between various actuators in opposite direction [26]. If the speed of an actuator deviates from the correct value and a Force Fighting occurs, the MCC can enable the AJ system to stop the ECMU that is operating normally. More in detail, each SECU handles a Force Fight error if there is a difference between SECU1 and SECU2 positions (supplied by ECMU1 and ECMU2). As reported in Figure 11, the Force Fight error can be expressed as  $\Delta_3 > \Delta_{FF}$  where  $\Delta_3$  is  $\text{abs}(\text{ECMU1}_{\text{position}} - \text{ECMU2}_{\text{position}})$  and  $\Delta_{FF}$  is a definite threshold, where

$$\begin{aligned}\Delta_1 &= \text{abs}(\text{ECMU1}_{\text{position}} - \text{MCC}_{\text{ref}}) \\ \Delta_2 &= \text{abs}(\text{ECMU2}_{\text{position}} - \text{MCC}_{\text{ref}}) \\ \Delta_3 &= \text{abs}(\text{ECMU1}_{\text{position}} - \text{ECMU2}_{\text{position}})\end{aligned}$$

Since the SECUs are ring connected, each SECU has knowledge of all the information about its SECU\_N and SECU\_P. The SECU1 determines its Force Fight state using the following relation:

$$\begin{cases} \Delta 3 > \Delta_{FF} \\ \Delta 1 > \Delta 2 \end{cases}$$

The SECU2 executes the same test.

When a Force Fight error is detected, the De Force Fight mode operates as shown in the specific case of Figure 11, where ECMU1 is forced in AJ mode and EMA 2 that is free to move, is forced by ECMU2 to the EMA 1 position as shown in Figure 12. In this final position  $\Delta 3 = 0$ , that is on the primary surface do not act mechanical forces. The Table 4 describes the structure of the file by means of groups and channels related to our application. The elements of this structure can be modified according to the system under test.

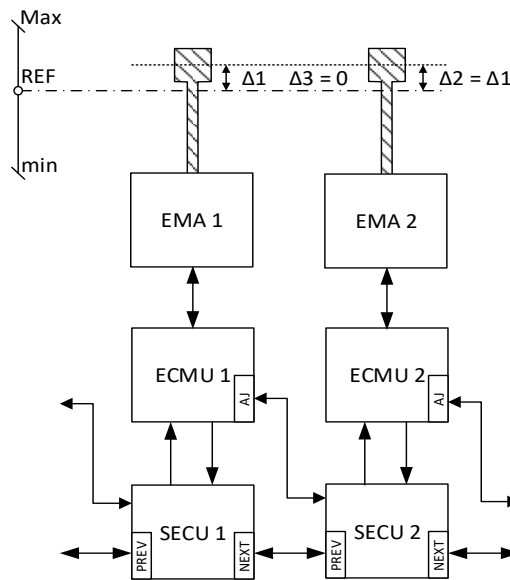


Figure12. The de force fight mode

Table 4. Structure of the saved file

Group	Channel	Measurement units	Description
Time			Timestamp group
	Time	DD/MM/YYYY hh:mm:ss,ssss	Timestamp at the time when the values are saved (resolution of 0.1ms)
MCC	Ref Position	[mm]	All data sent by the MCC The sample of the position profile sent
SECUX MeasData from ECMUX			Sensors data from SECUX
	LVDT	[mm]	Position of ECMUX
	DC Bus current	[A]	DC Bus current of ECMUX
	DC Bus voltage	[V]	DC Bus voltage of ECMUX
	Pt100	[°C]	Motor Temp of ECMUX
	PCB Temperature	[V]	PCB Temp of ECMUX
	Torque	[V]	Torque of ECMUX
	Torque Excitation	[V]	Torque Excitation of ECMUX
SECUX StatusBits from ECMUX			ECMUX Status Bits group
	Status Bits 7-0	-	ECMUX status bits
	Status Bits 15-8	-	ECMUX status bits
	Status Bits 23-16	-	ECMUX status bits
	Status Bits 31-24	-	ECMUX status bits
	Status Bits 39-32	-	ECMUX status bits
	Status Bits 47-40	-	ECMUX status bits
SECUX StatusBits			SECUX Status Bits group
	Status Bits 7-0	-	SECUX status bits
	Status Bits 15-8	-	SECUX status bits
	Status Bits 23-16	-	SECUX status bits
	Status Bits 31-24	-	SECUX status bits

## 5. EXPERIMENTAL RESULTS

As part of the development phase, the implemented system has been tested in the laboratory. In more detail, Figure 13 shows the test platform consisting of: (i) the supervisor control units based on cRio (SECU1 and SECU2); (ii) the two SECU monitoring systems; and (iii) the movement control computer (MCC). The two electronic control and monitoring units under test are not visible in the Figure 13, because they are placed near the two actuators, installed on the test benches. The operator interfaces for the two monitor programs are instead visible (ECMU1 and ECMU2). Several tests have been implemented to simulate both static and dynamic conditions, as discussed below.

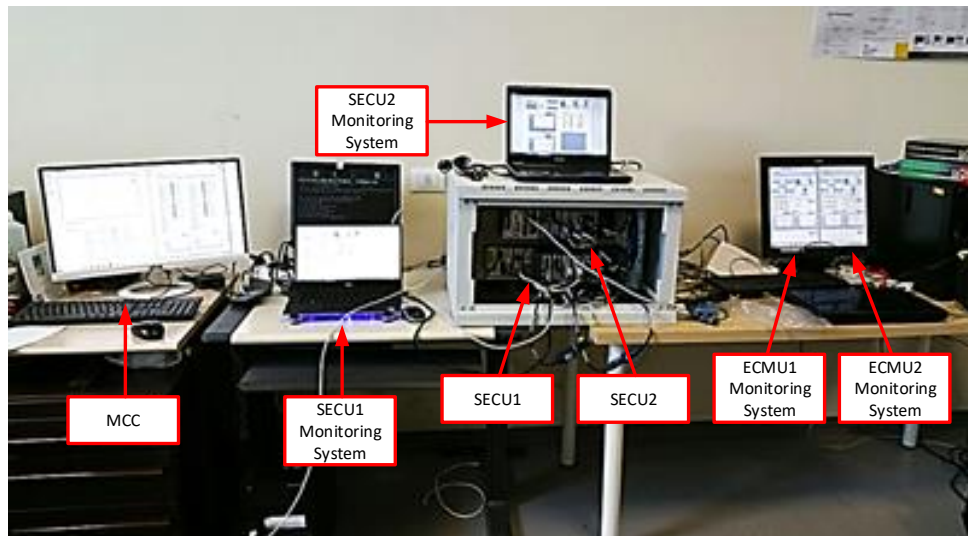


Figure 13. The implemented system during the lab tests activity

Test of the communication interfaces. In the initial phase the correct functioning of the proposed test platform has been validated, verifying the communication system. First of all, we verified that MCC sends the same position setpoints to all the SECU's present in the ring architecture. We also verified that every time a SECU receives a position set-point from MCC, it sends it to the ECMU under test. The first test was related to the full-duplex CAN bus link that connects the SECU with the MCC. As shown in Figure 14, we verified that the bit rate was set to 1Mbit/s and that both the SECU to MCC and MCC to SECU frames are present. As shown in Figure 15, we measured a MCC to SECU position set-point frame duration of 107  $\mu$ s, and a frame rate of 20 ms. The SECU to MCC measurement frame has a duration of about 992.8  $\mu$ s.

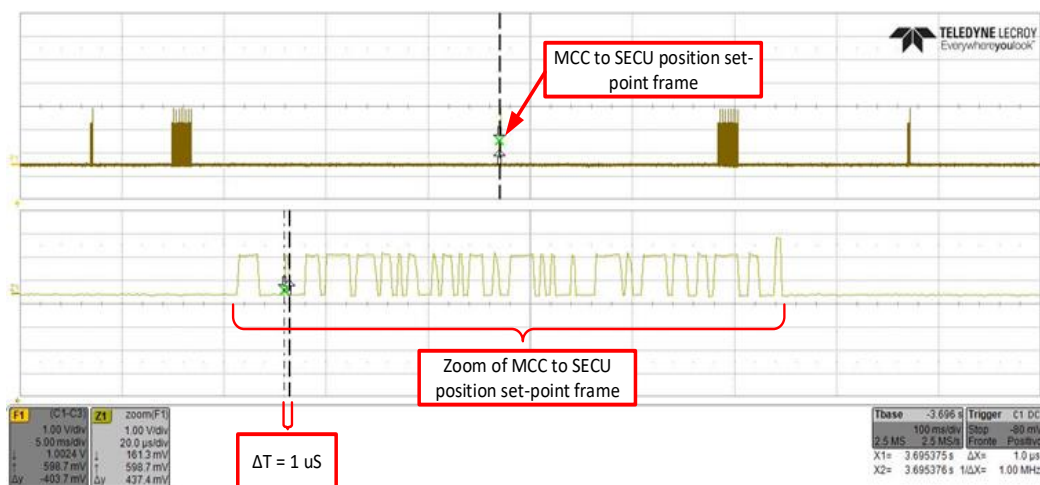


Figure 14. MCC to SECU and SECU to MCC communications at 1Mbit/s



Figure 15. SECU to MCC measurement frame

The successive test was related to the inter SECU links (two full duplex CAN buses). The measured bit rate was of 1Mbit/s, the frame duration was about 663  $\mu$ s, while the frame rate was 50 ms. As concerning the MCC to SECU communication, we carried out the Cable Error test, disconnecting the cable from MCC to SECU as shown in Figure 16. We verified that when the SECU detects a difference between the two MCC CAN frames (for example the frame is received just from a single interface), the SECU notifies the MCC *Cable Error*. The SECU to SECU communication has been tested carrying out the SECU\_N cable test, by disconnecting this cable and verifying the notification of this error. These results show the correct functioning of the communication interfaces of the test system.

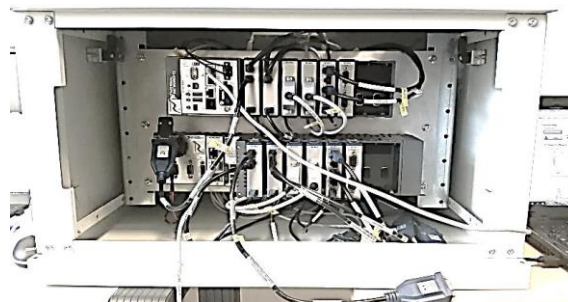


Figure 16. Disconnection of cable between MCC and SECU

At the end of this phase we tested the communication between the test platform and the two ECMUs. The SECU to ECU communication has been tested carrying out the ECU cable test, by disconnecting the RS-485 cable from ECU1 and verifying the error notification. The second test was related to the ECU to SECU connection (2 half-duplex RS-485 links). The measured duration of the position set-point frame was about 440  $\mu$ s as shown in Figure 17. As regards the ECU communications, we verified that the ECU performs the measurements and sends the related frame to the SECU. These results show the correct functioning of the communication interfaces of the two ECMUs under test.

The tracking error test. The tracking error is detected when the difference between the MCC reference position and the ECU measured position is greater than a fixed threshold. This test is executed defining a squared position profile on the MCC, as shown in the left side of Figure 18, where instantaneous transitions of 60 mm are highlighted. Due to the physical limitations of the system, this profile is not feasible: no real actuator can allow such a movement in a null time. The reason for this test is to verify the ability of the ECMUs to handle this event. This test also allows verification of the performance of the proposed test platform, in terms of speed of data acquisition and analysis. The right side of Figure 18 shows the trends of the positions of the two actuators measured with the two position transducers LVDT1 and LVDT2. The two curves are perfectly superimposed, therefore the two drives are controlled by the ECMUs in perfect



synchronism, but lagging behind the required profile. We verified that both the SECUs handle the error and puts the related ECMU in damping mode. The Force Fight test. From MCC we initially generated a sinusoidal profile for both the actuators. This profile is followed by both actuators, as shown by the plot of the outputs of the two LVDT position transducers shown in Figure 19.

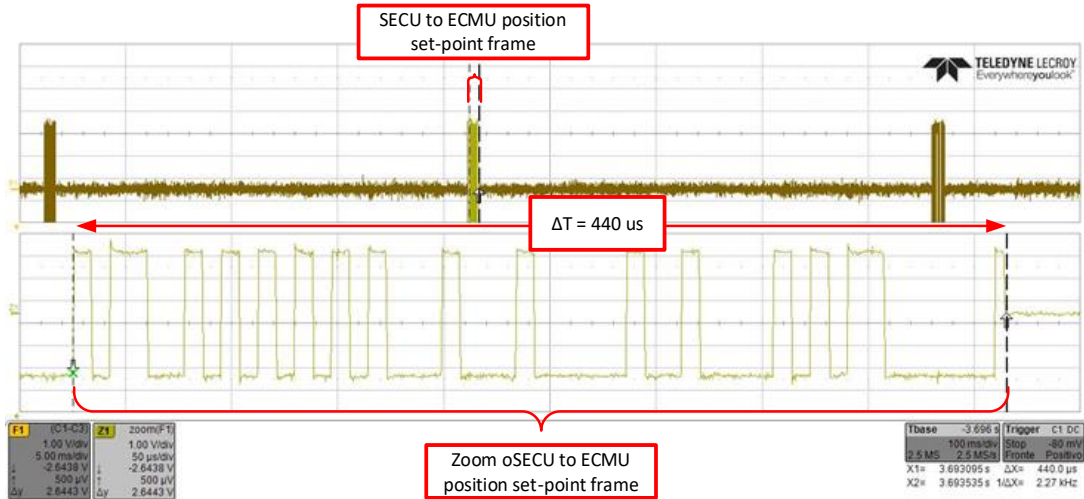


Figure 17. SECU to ECMU position set-point frame duration

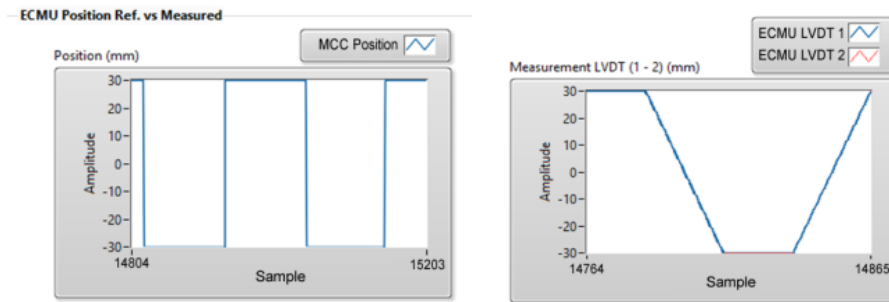


Figure 18. Squared position profile and ECMU measurement data

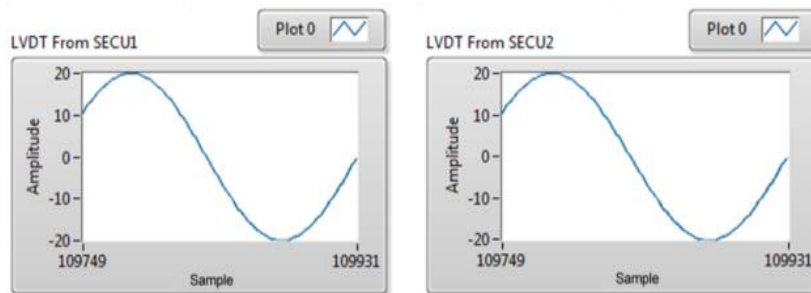


Figure 19. MCC creates a sinusoidal profile for SECU1 and SECU2

Successively, the test platform forced a speed reduction for ECMU2 only (from 20 mm/s to 6 mm/s), creating a Force Fighting. The resulting position error, forced by the test system, cannot be automatically compensated by ECMU2. After the detection of this error, we verified the following two possible behaviors.

- a) When the De Force Fighting is disabled, the test system manages this irreversible fault by putting the failed actuator controlled by ECMU2 in Anti-Jamming (AJ) mode to prevent possible malfunctions.

In more detail, the ECMU2 drives the AJ system, mechanically disconnecting the actuator. The ECMU1 goes into damping mode (a type of neutral) and therefore no longer drives its motor. This behavior can be observed in Figure 20 which shows the trends of both the required sinusoidal profile received by the ECMUs from the SECUs (target position) and the profile measured by the ECMUs (internal position), for both the actuators. The measured profiles are expanded along the time axis. From the figure it is possible to see the slowdown of the second actuation with respect to the first, which instead continues to follow the imposed profile. ECMU2 detects the error and activates the AJ by disconnecting from the controlled surface and keeping the position reached constant. ECMU1 then goes into dumping mode, and waits for subsequent commands. The two final positions are different, because the De Force Fighting is disabled.

- b) To carry out the next test, we enabled the De Force Fighting and generated the same error. The effects are similar to those illustrated in the previous test. The ECMU2 drives the AJ and disconnects the actuator. ECMU1 drives the motor to align with the position of ECMU 2, in order to reduce the stress on the moved surface as shown in Figure 21.

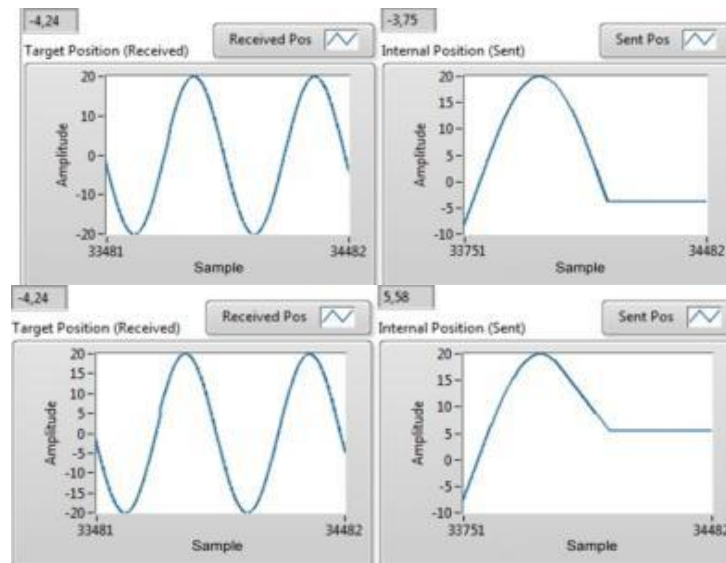


Figure 20. ECMU1 (up) is in damping mode, ECMU2 (down) is in AJ mode

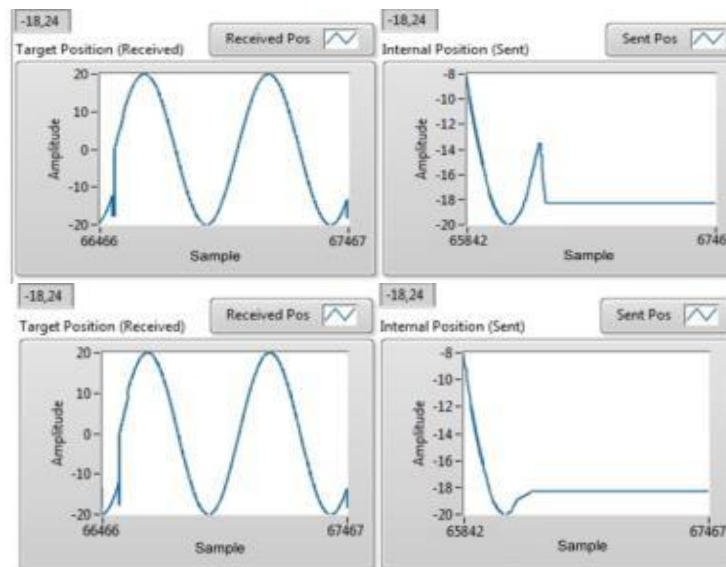


Figure 21. ECMU2 (down) reduces the speed generating a force fighting, SECU1 drives ECMU1 (up) to ECMU2 position

The Force Fight test verifies the ability of the two ECMUs to check the operation of the two drives, which must operate in perfect synchronism with each other. With this test, the ability of the two ECMUs to react to the error situation in the two different possible operating modes was verified.

## 6. CONCLUSIONS

In the introduction we illustrated the developments taking place in various application sectors. The main application for which we developed the proposed system is related to the movement of flight control surfaces. This is mainly because in the avionic field it is in place a transition from hydraulic to electromechanical actuators that will be slow and will require several comparative tests. Hybrid systems with dual implementation will initially be developed, therefore those with the only electromechanical actuation. Simultaneously with the development of the actuators, the ECMUs are being studied. Also for these devices the functions and the interfaces will be defined first, then the tests will follow and then the standards will be defined. We worked in the past in the implementation of automatic test equipment for avionics electro-mechanical actuators [8]. In this paper we proposed a cost-effective test platform for the automated testing of electronic control units (ECMUs) of synchronized electric power actuators. The test system facilitates the phase of definition and development of the ECMUs, allowing the interfacing towards both hydraulic and electromechanical actuators, and having a high flexibility as regards the I/O signals.

The implemented system, based on commercially available instrument modules, shows some hardware redundancies that make it more flexible and reconfigurable, in order to be adopted for a wide range of applications and performance requirements. The proposed system is oriented to maximum flexibility to support the development and testing phase of these units. In the paper, the test platform architecture and implementation have been discussed, showing also some satisfactory results obtained during the laboratory test activity.

## ACKNOWLEDGEMENTS

The present research is inherent to the program PON01\_02886, Objective OR3, for the characterization in different configurations of electric generators driven by linear actuators. The authors would like to thank UmbraGroup S.p.A (Foligno, Italy) for making possible this research activity and for the technical support during the experimental activities.

## REFERENCES

- [1] All Electric Aircraft (AEA), "Electric Aircraft Propulsion and Power System," [Online]. Available: <http://allelectricaircraft.com/>
- [2] P. Harrop, "In-Wheel Motors for Electric Vehicles Gaining Market Acceptance," 2011. [Online]. Available: <http://evworld.com/news.cfm?newsid=25224>
- [3] K. Bullis, "Wheel Motors to Drive Dutch Buses," *MIT Technology Review*, 2009.
- [4] Hong Sun, and G.T.-C. Chiu, "Motion synchronization for multi-cylinder electro-hydraulic system," *Proc. of Advanced Intelligent Mechatronics*, vol. 1, pp. 636-641, 2001.
- [5] Hogan, P. and Burrows, C.R., 1994, "Synchronizing Unevenly Loaded Hydraulic Cylinders," *Proceedings of the ASME Fluid Power and Systems Technology Division*, vol. 1, pp. 75-80, 1994.
- [6] M. H. Cheng, and E. G. Bakhoun, "Adaptive robust control of tracking and synchronization for multi-axis motion system," *Proc. of the 2011 American Control Conference*, pp. 1-6, 2011.
- [7] G. Bucci, F. Ciancetta, and E. Fiorucci "A Testing System for the Performance Evaluation of Fault-Tolerant Electromechanical Drives for Aircraft Applications," *Proceedings of 12th IMEKO TC10 Workshop on Technical Diagnostics: New Perspective in Measurements, Tools and Techniques for Industrial Applications*, pp. 34-39, 2013.
- [8] M. G. Antonelli, *et al.*, "Automatic test equipment for avionics Electro-Mechanical Actuators (EMAs)," *Measurement* vol. 57, pp. 71-84, 2014. ISSN: 0263-2241,
- [9] M. Villani, *et al.*, "Multi-phase permanent magnet motor drives for fault-tolerant applications," *Proc. of IEEE International Electric Machines & Drives Conference (IEMDC)*, pp. 1351-1356, 2011.
- [10] A. Credo, *et al.*, "An Open-Loop Starting Procedure for Sensorless Control of Synchronous Reluctance Motors," *2019 IEEE 10th International Symposium on Sensorless Control for Electrical Drives, SLED*, pp. 1-6, 2019.
- [11] M. Tursini, *et al.*, "Sensorless control of a synchronous reluctance motor by finite elements model results," *2017 8th International Symposium on Sensorless Control for Electrical Drives, SLED 2017*, pp. 19-24, 2017.
- [12] A. Di Tullio, *et al.*, "Five phase multi-layer drive with fault tolerant neuro-fuzzy features for safety critical applications," *2017 20th International Conference on Electrical Machines and Systems, ICEMS*, pp. 1-6, 2017.
- [13] M. Tursini, *et al.*, "A switched-reluctance motor for aerospace application: Design, analysis and results," *Electric Power Systems Research*, vol. 142, pp. 74-83, 2017.
- [14] D. Ossmann, and F. L. J. van der Linden, "Advanced sensor fault detection and isolation for electro-mechanical flight actuators," *Proc. of NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp. 1-8, 2015.

- [15] V. Tešić, *et al.*, "Automated testing of electronic control units in a hardware-in-the-loop simulation environment," *2015 23rd Telecommunications Forum Telfor (TELFOR)*, pp. 1012-1015, 2015.
- [16] X. Sun, *et al.*, "Real-Time HIL Emulation for a Segmented-Rotor Switched Reluctance Motor Using a New Magnetic Equivalent Circuit," *IEEE Transactions on Power Electronics*, vol. 35, no. 4, pp. 3841-3849, 2019.
- [17] Y. Yuan, *et al.*, "An investigation into factors determining the metering performance of a fuel control unit in an aero engine," *Flow Measurement and Instrumentation*, vol. 71, no. 101672, 2020.
- [18] V. Ili, S. Popi and M. Kovači, "Data flow in automated testing of the complex automotive electronic control units," *2016 Zooming Innovation in Consumer Electronics International Conference (ZINC)*, pp. 1-3, 2016.
- [19] E. Bagalini and M. Violante, "Development of an automated test system for ECU software validation: An industrial experience," *2016 15th Biennial Baltic Electronics Conference (BEC)*, pp. 103-106, 2016.
- [20] ISO 11898-1:2015. Road vehicles -- Controller area network (CAN) -- Part 1: Data link layer and physical signalling. [Online]. Available: <https://www.iso.org/>
- [21] O. Acevedo, *et al.*, "Towards optimal design of avionics networking infrastructures," *Proc. of IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, pp. 7A3-1, 2012.
- [22] D. Briere, and P. Traverse, "Airbus A320/A330/A340 electrical flight controls - a family of fault-tolerant systems", *Proceedings of 23 IEEE Int. Symp. On Fault-Tolerant Computing (FTCS-23)*, pp. 616-623, 1993.
- [23] ARINC Specification 825, "General Standardization of CAN (Controller Area Network) Bus Protocol for Airborne Use," *ARINC Industry Activities*, pp. 191, 2015.
- [24] D. Patterson, "The Trouble with Multicore," *IEEE Spectrum*, vol. 47, no. 7, pp. 28-32, 2010.
- [25] "Multicore Programming with LabVIEW," *Multicore Programming Fundamentals Whitepaper Series*, 2014.
- [26] Y. Fu, *et al.*, "Force Fighting Research of Dual Redundant Hydraulic Actuation System," *International Conference on Intelligent System Design and Engineering Application*, vol. 1, pp. 762-766, 2010.

## BIOGRAPHIES OF AUTHORS



**Giovanni Buccì** received a degree in electrical engineering in 1985 from the University of L'Aquila. He worked with Selenia Spazio as a designer of automatic test equipment until 1989. He is now Full Professor in Electrical Measurement at the University of L'Aquila. His current research interests include smart sensors, multiprocessor-based measuring systems, digital measurement stations operating in real-time, and power measurements. He is an author of more than 200 scientific papers in these fields.



**Fabrizio Ciancetta** received the M.S. degree in electronic engineering and the Ph.D. degree in electrical and information engineering from the University of L'Aquila, L'Aquila, Italy, in 2003 and 2009, respectively. In 2003, he joined the Department of Electrical Engineering, University of L'Aquila, where he worked on the development of digital and distributed measurement systems. Since 2009, he has been holding a postdoctoral position at the same university. His research and technical interests cover several aspects related to distributed measurement systems, signal processing, and multilevel inverters; in these fields, he authored about 60 journal and conference papers.



**Edoardo Fiorucci** received the Ph.D. degree in Electrical and Information Engineering from the University of L'Aquila in 2004. He is now Associate Professor in Electrical Measurement at the University of L'Aquila. His research interests include: power measurements, testing of electrical machines and power systems, pc-based instrumentation, measurement systems for power quality evaluation, smart web sensors for distributed measurement applications. He is the author of more than 100 scientific papers in these fields.



**Simone Mari** received the bachelor degree in Industrial Engineering from the University of L'Aquila in 2017. He is currently a final year student of the master course of Electrical Engineering, and he is developing his thesis on topics concerning electrical measurements and instrumentation.