The SCAlable & Reconfigurable Electronics plaTforms and Tools project (SCARLETT) was started by European Commission to develop the concept of IMA 2G Avionics. The paper mentions the main objectives of the SCARLET project and presents a sample time-critical application developed for it. The time-critical application is an ARINC 653 based and is a part of a distributed control system working in AFDX network environment. Its task is to control an elevator’s actuator of a typical airliner. The paper presents the structure of the application, its timing schedule, and dedicated control algorithms and error detection mechanism built-in it. Selected application tests conclude the paper.

Nomenclature

| CPM     | = Core Processing Module |
| RDC     | = Remote Data Concentrator |
| REU     | = Remote Electronics Unit |
| IMA     | = Integrated Modular Avionic |
| DME     | = Distributed Modular Electronics |
| AFDX    | = Avionics Full DupleX |
| q       | = pitch rate |
| φ       | = pitch angle |
| de      | = deflection of elevator |
| u       | = airspeed |
| α       | = angle of attack |
| h       | = altitude |

I. Introduction

The Integrated Modular Avionic (IMA) concept has been introduced through the European funded research projects PAMELA, NEVADA and VICTORIA. The result of the projects was so called first generation of IMA (IMA1G) and was successfully implemented onboard A380, A400M and B787 aircrafts. The main achievement of the IMA1G is a general change of modern airliner avionics construction paradigms. The previous federated avionics architecture is being systematically replaced with fewer common processing modules, sharing the necessary power supply and communication links. The current implementation of IMA covers a limited range of aircraft functions but shows that it may bring a significant profits: the aircraft weight reduction and lower maintenance costs.

The SCARLETT project is a natural continuation of abovementioned projects. Its main aim is a definition of a scalable, reconfigurable fault-tolerant driven and secure new avionics platform, namely DME: Distributed Modular Electronics. The new DME should provide:

- Scalability, portability and adaptability,
- Fault tolerance and reconfiguration capabilities,
- A minimum number of types of standardised electronic modules,

---

1 PhD. Eng, Department of Avionics and Control Systems, ul. W. Pola 2 35-959 Rzeszow, Poland.
2 PhD. Eng, Department of Computer and Control Engineering, ul. W. Pola 2 35-959 Rzeszow, Poland.
3 DSc. Eng, Department of Avionics and Control Systems, ul. W. Pola 2 35-959 Rzeszow, Poland.

American Institute of Aeronautics and Astronautics
- Support for the full range of avionics function.

The SCARLETT project is an European enterprise joining 39 companies from 16 countries including large industrial companies, public research centres, industrial research centres, universities, small and medium enterprises. One of the project partners is Rzeszow University of Technology (RUT) Research Team (RUTRT).

This paper presents the current RUTRT’s contribution to the SCARLETT project. The RUTRT takes part in a time critical application development and testing SCARLETT research path. Its task is to prepare a specific hard real-time application that would be executed at several SCARLETT hardware modules. The application should be a part of a control system and make it possible to evaluate whether DME units can be effectively used as hard real-time applications platforms. The preliminary research results of RUTRT were published in7.

The next sections of the paper are organised as follows. Firstly the main SCARLETT DME units will be shortly introduced. Secondly a Pitch Control Application (an illustrative example of the time-critical avionic control system) developed by the RUTRT will be presented. The section will include decryptions of: system specification, application implementation and built-in self testing procedures. The final part of the paper will include our future research plans.

II. SCARLETT DME Units

According to SCARLETT philosophy the future avionics of airliners will consist of a set well-defined hardware modules dedicated to a sets of avionic applications. The following types of hardware modules have been defined in Ref. Błąd! Nie można odnaleźć źródła odwołania.: 1) AFDX - Avionics Full DupleX (AFDX) is a redundant Ethernet network and made reliable, developed and standardized by the European industrialists of the avionics to equip A380 Airbus. It is about a system intended to be used as support for the internal communications with the plane, and not with the communications with outside. The internal communications are primarily the data exchanged between the different components of the avionics.

2) AFDX Switch - Star coupler for AFDX network. Like Ethernet switches, AFDX switches provide packets forwarding, they also have additional features to guarantee the timing and bandwidth allocation of all the network, but they only use multicast addresses (Virtual Links).

3) CPM - The Core Processing Module offers a generic computation capability, an AFDX End System, and a set of generic bus + housekeeping I/Os capability (Field Bus such as CAN or Flexray, Discrete Inputs for pin-programming / servitudes and RS232 / 485 for maintenance in shop). The objective is to suppress the A429 interfaces of the CPM and the adherence between the functional I/O and the applications running on the CPM. As a start SCARLETT identify three types of CPM:
   - High Performance CPM,
   - Time Critical CPM,
   - Avionics Server Function CPM.

4) REU - Remote Electronics Unit is an electronics box dedicated to a function and geographically close to this function. REU are not generic units and therefore not part of the IMA perimeter. However the interfaces of REUs to the IMA world shall be standardised.

5) RDC - Remote Data Concentrator: equipment supporting the exchange of information between sensors / actuators (digital, discrete and analogue data) and aircraft digital communication networks (ADCN). The RDCs are located in pressurized areas close to sensors and effectors, which may be potentially remote from the associated processing resources, then not in the avionics bay.

The abovementioned hardware modules should mainly communicate between each other using AFDX. The message format should follow ARINC 664 specification. The software modules executed on the hardware units should be developed according ARINC 653 specification.

III. RUTRT Research Objectives

In SCARLET project RUTRT was obliged to prepare a hard real-time application (time-critical application) which parts would be distributed over several devices developed according DME paradigm. The SCERLETT
partners specified the following requirements for RUTRT's application. An application demonstrates possibility of the usage of SCARLETT philosophy to control an aircraft in the pitch control channel. The application should demonstrate the possibility of the use of two synchronously working actuators deflecting elevator’s surfaces. An external controller computes position of the elevator on the basics of flight parameters, pilot’s inputs and, implemented control procedures. The application communicates with actuator’s controllers and units collecting data from indicators by specially dedicated REU units connected to ADFX bus. The application should be prepared for two kinds of hardware configurations. For the first hardware configuration (called the Time Critical Demonstrator) the application should assess the possibility of correct execution of a distributed hard real-time application loaded on the elements of the demonstrator. For the second one (called the Reconfiguration and Maintenance Demonstrator) the application should be so called background application. It should be possible to detect whether the control systems works correctly even if the reconfiguration of other applications occurs or has recently occurred. Figure 1 shows an example RUTRT's application distribution over the SCARLETT hardware modules.

A part of the control application is executed on CPM module. Two other parts are allocated to two REUs. The REUs are directly connected to actuators. The Pitch Control Application modules communicate via AFDX.

A. Control System Architecture

Practically, RUTRT’s Pitch Control Application (RUTRT PCA) should control two actuators (brushless motors) connected to a load (an elevator). Each actuator is controlled by a separate cascade of controllers as in fig. 2. The actuator control system includes an internal current control loop, a velocity control loop, and a position control loop. Flight Control Algorithm is a superior module that generates the position demand signal for both control subsystems. To adjust the dynamics of the flight control application to the realistic values it includes a real-time aircraft simulator that reflects a typical airliner behaviour and produces reference signal for Flight Control Algorithm module. The Flight Control Algorithm module collects signals from the aircraft simulator, pilot simulator and the actuator and produces the position demand signal for both actuator control subsystems.

B. Control Application Distribution Scenarios

![Diagram of RUTRT's application distribution](image)
One of the SCALLET programme goals is evaluation of quality of distributed control applications where some parts of the application are hosed on a different devices. Therefore the RUTRT’s Pitch Control Application was developed having in mind the possibility of distribution of some its parts on separate hardware modules. Figure 2 shows the elaborated control application distribution variants. In the first variant (CPM1_2+REU1_2+REU2_2) Pilot, Aircraft, Flight Control Algorithm, and two position controllers are hosted on CPM module, whereas velocity control algorithms are hosted on separate REUs. In the second variant ((CPM1_1+REU1_1+REU2_1)) the CPM module executes the superior part of the control system (Pilot, Flight Control Algorithm, and Aircraft) and both position and velocity controllers are hosted on REUs.

### IV. RUTRT’s Pitch Control Application Implementation

The main goals of the RUTRT’s Pitch Control Application software development were as follows:

1) Position controller must be movable. It should be possible to host it both on CPM or REU hardware modules.
2) Communication data must be adjusted to both external (inter-module) and internal (intra-module) data exchange.
3) Some verification procedures should be built-in the application software. They should give the information about the quality of control system and the correctness of system structure.

RUTRT’s Pitch Control Application was developed according ARINC 653P1-2 specification\(^1\). The final source code was prepared in two variants: for VxWorks 653\(^5,8,9,10\), and for PikeOS\(^1\). Data packages that are sent between partitions are prepared according ARINC 664P7 requirements\(^2\).

### A. Pitch Control Application Structure

The released variant of RUT Flight Control Application was a VxWorks 653 and PikeOS based real-time simulator of complete system. Figure 3 includes its structure. The simulator consists of four ARINC 653 partitions. P1, P2, and P3 are main software modules that RUTRT had to provide. P4 partition includes software real-time simulators of SCALLET REU modules. The software emulation of REU modules made it possible to develop the RUTRT’s application independently of the hardware module that was under development.
Figure 3. Pitch Control Application structure.

The first (P1) partition includes some real-time simulators of pilot and aircraft. It also includes Flight Control Algorithm (FCA) block that collects signals from pilot, aircraft and actuator modules and produces the desired pitch angle signals for controllers. The last module built-in the P1 partition is Error Estimator. It makes it possible to monitor both communication channels and the quality of control during the system run-time. This module will be presented in detail in next sections. Error Estimator, Pilot, Aircraft and FCA modules are separate real-time tasks. The second (P2) partition includes the first position controller algorithm (CPx1) running as a separate real-time task. Identically, the third (P3) partition includes the second position control algorithm (CPx2). The fourth (P4) partition includes 2 real-time REU simulators: the velocity controllers modules joined with actuators attached to the first (CVx1+Actu1) and the second (CVx2+Actu2) control loops.

For the intra-partition communication ARINC 653 blackboards were applied, whereas the inter-partition communication is based on ARINC 653 sampling ports and channels.

Figure 3 includes both ARINC 653 port names and communication channels names referenced in the next part.
of this paper. The same port and channel names were encoded in the application XML configuration file created according to ARINC 653 specification.

B. Application Timing

RUTRT’s Pitch Control Application was developed to fulfil the timing constraints shown in fig. 4. Application major frame and partition windows were defined according to ARINC 653 specification and encoded in the application XML configuration file. The timing constraints for each of partitions were derived from project partners and make it possible to develop properly executing digital Pitch Control System for typical airliners. Moreover other partitions with third-party software will be executed on the same CPM module.

C. Built-in quality of application procedures

According to system’s requirements the RUTRT’s Pitch Control Application should provide a set of test procedures informing the user about the quality of the system before or at the runtime. Therefore the following extensions of the basic control system have been applied.

1) Separate error (ERR) port was included in P1 partition structure.
2) New Error Estimator function block was introduced in P1 partition.
3) It has been decided that the quality of service control system’s subsystems would be run simultaneously with the control procedures.
4) The quality of service procedures has been divided into two subsystems:
   a. Channel connection detector that permanently monitors the channels of the system and informs whether all the links are properly connected. This subsystem gives the guarantee that all system components send and receive data from the proper ports and software modules.
   b. Control system error detector that signals to the system operator that the quality of control is below the assumed acceptable level. It may suggest some problems with communication or control system’s parameters should be refined to compensate delays in the communication.

Channel connection detector checks whether all channels are configured according to the assumed structure. During the runtime of the application, apart from control application data, a separate set of values is sent via the channels. Some additional procedures included in RUTRT’s application control blocks make it possible to detect whether the application’s ports receive data from the assumed sources. The detector makes it also possible to reveal data transmission faults. It produces a 16-bit word, where each bit value means the correctness of the related channel. If the bit value equals 0 the channel works properly, on the other hand the channel is badly established or the function block connected to the channel produces incorrectly formulated data packages.

Control System Error Detector is a piece of software implemented into P1 partition. Figure 5 presents location of the Error Detector in the general scheme of the application. Its task is to monitor quality of control realised by monitored control channel. The algorithm of the error detector works according to schema presented in figure 6. Symbols and abbreviations located on this figure mean:

- PD – desired position of the elevator,
- ED – measured position of the elevator,
- VFx – measured velocity of the elevator (actuator output),
- Filter – low-pass filter,
- Err_1 – signalling of the actuator angular velocity oscillations,
- Err_2 – signalling of the elevator position error,
- cVFx – estimated value of the elevator angular acceleration,
- cerr_E – estimated value of the elevator position error,
- CVFxgr – angular acceleration threshold value,
- cerr_Egr – elevator position error threshold value.
Error estimator’s task is to indicate which actuator control system operation cases do not meet assumed expectations. It is assumed that the incorrect operation is caused by data transmission delay using AFDX network, or by the damage of control system elements that produce unacceptable errors of elevator’s movement. Therefore, the control system should be insensitive to standard transport delays. In order to examine the effect of signals transport delays in AFDX network on control system operation, control system properties should be chosen in such a way that, the real delay values in data transfer would cause a indictable change in elevator deflection control quality.

Two diagnostic methods were used:
1) Detection the actuator angular velocity oscillations (Err_1).
2) Signalisation the elevator’s position error that exceed the assumed threshold value (Err_2).

Properties of the errors estimator module has been tested using MATLAB-Simulink package. The diagnostic was carried out as it is illustrated in fig.7 and fig.8.

Figure 7 presents the general simulation scheme. The actuator and the elevator assumed as a real integral term are controlled by PID controller. The feedback signals of the elevator position and its angular velocity are delayed in the Delay Generator module of about a value that represents the transport delay of AFDX network.

Oscillation detection in the control system depends on average absolute value of the actuator angular acceleration and comparing it with the assumed threshold value (Err_1). Tracking error detection depends on the comparison of absolute value of the deviation between desired and measured value of the elevator deflection angle and the assumed threshold value (Err_2).

For the operator the most important is value of ERR signal produced by error estimator (fig.5). It clearly informs the operator about quality of control loop. In general, 0 informs that the control loop works correctly. Values between 1 and 3, include any errors of the monitored control loop (Table1).

Supplementary signals cVFx and cerr_E (fig. 6) can be used for scaling of errors estimates and applied as an elevator control system “quality indexes”.

For the given actuator properties, PID controller’s parameters should be chosen in such a way that the assumed control system sensitivity to delays bigger than the allowable ones will be reached. Figure 8 illustrates the operation of the diagnostic system for an exemplary configuration of elevator control system. Transport delay is modeled in 5-8 ms interval (Fig. 8a) and a periodic sinusoidal elevator deflection is introduced (Fig. 8b). Figure 8c presents the tracking error and the value of the function that is used to evaluate tracking quality. In figure 8e, signal Err_2 takes the value “true” when the tracking error is bigger than 0.003 rd. Figure.8d presents the angular velocity of the actuator while, Fig. 8f presents control quality evaluation function. Signal Err_1 in fig. 8e takes the value “true” when the transport delay of feedback signals exceeds 7.7ms.

Threshold values are set arbitrary on the basics of experiments ant expert knowledge. They must be readjusted for real-hardware tests.

<table>
<thead>
<tr>
<th>Signal name</th>
<th>Signal value</th>
<th>Signal interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR</td>
<td>0</td>
<td>Control system works correctly</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Deflection velocity oscillations exist</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Tracking error exists</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Tracking error and control system oscillations exist</td>
</tr>
</tbody>
</table>

Table 1. Control system error detector signals interpretation.
The application controls two actuators moving the elevator simultaneously so there are two error detectors implemented into the application in fact. One for each of elevator’s control channel. They produce ERR1 and ERR2 signals respectively. If both ERR1 and ERR2 are zeros the entire system works properly. Else some errors exists and can be decoded on the basics of Table 1.

Figure 6. MATLAB-Simulink scheme for diagnostic module testing.
Figure. 8. Results of the errors estimator module testing. a) transport delay of the actuator velocity, b) angular deflection of the elevator, c) tracking error and error estimator function, d) angular velocity of the actuator, e) signalization of oscillations (Err_1) and tracking error (Err_2), f) estimation function of velocity.
V. Application tests

Two stages of RUTRT Pitch Control Application (RUTRT PCA) tests were conducted. The first groups of tests assessed the application from the control engineering point of view. The second stage of test covered testing of so called self-test procedures built-in the RUT PCA.

A. Pitch Control Application Evaluation

RUTRT’s task in SARLETT project was to prepare time critical application indeed to download to one of CPMs provided by other SCARLETS partners. Application’s goal is to work in AFDX environment and control actuators using data incoming from the network. To guarantee more realistic test conditions the actuator’s controllers software was put into simulated flight control system application environment. To adjust the dynamics of the flight control application to the realistic values it includes a real-time aircraft simulator that reflects a typical airliner dynamics and produces reference signal for Flight Control Algorithm module. Simulations uses model of longitudinal dynamics of DC-8 airplane flying at the speed of 251[m/s] at flight level 330. Step responses for elevator step input ($\delta_e=1$ [deg]) of the model described in the state spaces by vector $x$, state matrix $A$ and $B$ presents Figure 9.

$$A = \begin{bmatrix} 0 & -0.014 & -0.043 & 0 & -9.81 \\ -0.0735 & -0.806 & 251.2 & 0 \\ -0.0025 & -0.0351 & -1.33 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -10.55 \\ -4.57 \\ 0 \end{bmatrix} \quad (1)$$

Simulation tests was performed with the use of simple model of the actuator deflecting elevator’s surface. Its dynamics defines formula(2).

$$G(s)^e = \frac{K}{s^2 + 2T\xi s + 1}$$

$$K = \frac{0.25}{2\pi V/\text{rad}}$$

$$T = 0.05[\text{s}]$$

$$\xi = 0.85[-] \quad (2)$$

The preliminary tests of the RUTRT Pitch Control Application concerned typical properties of the whole control system. The complete real-time control system simulator was executed and states of its state variables were collected and assessed form the control engineering point of view. Typical test scenario looked as follows.

The application controlling actuators were connected to software modules simulating pilot’s control and aircraft’s dynamics. Pilot’s control (fig. 10) is interpreted inside FCA module and translated into desired aircraft’s pitch angle (compare fig. 2). FCA module also uses simulated pitch angle regulator to control flight of the simulated plane with the use of tested control channel of actuators.

Figure 9. Plane’s model responses to elevator step input.
Test of applications consists of two steps. The first step of tests involved simulations of properly working system. Values of flight parameters satisfied desired control precision. Figure 11 presents sample time graphs of pitch angle and pitch rate responses of the simulated model of plane to pilot’s control signal depicted in fig. 10 recognised as good enough.

During those simulations the programmed controller of actuator working with simulated model of actuator guaranteed the position of elevator matched desired position of elevator calculated by FCA module good enough (according to expert’s opinion). Figure 12 presents desired (calculated by FCA module) position of elevator and position of simulated elevator’s mechanism controlled by controller implemented in the application.

Graphs from figure 13 present sample errors values in control process satisfying desired control precision. Final error’s value (ERR) produced by error estimator is zero. Values of variables informing about elevator’s position error and actuator’s angular velocity oscillations error are in ranges recognised as proper ones. Ranges of acceptable values of these errors are set arbitrary on the basics of experiments and areas follow: 0.05 for elevator’s position error and 0.3 for actuator’s angular velocity oscillations error (fig. 13).

Figure 10. Pilot’s control programmed in tests

Figure 11. Flight parameters of simulated model of plane, a) pitch angle, b) pitch rate.

Figure 12. The time domain graph of desired and final positions of the elevator.
B. Build-in self test subsystems evaluation

The RUTRT Pitch Control Application was developed as a self testing system that during its runtime checks both its communication configuration and its quality of control.

The channel connection detector subsystem of the application permanently monitors all the channels existing in the application and produces 16-bit word that reports the channel’s state. The values of subsequent bits in the word reflect the relevant channels state. If a relevant bit value is 0 the channel is recognised as properly working. If a relevant bit equals 0, the channel is incorrectly established or physically destroyed. The channel connection detector subsystem was tested in details during long-term tests of the RUT PCA. All the possible channels malfunctions were emulated and properly detected.

The control system error detector subsystem of the application permanently calculates the quality of the control system by collecting all the possible control state variables and assessing their values according the algorithm described section III C. The subsystem tests were conducted simultaneously with the long-term RUTRT PCA execution tests. Typical test scenario looked as follows.

The group of tests consisted of a set of simulations of broken down actuator system. Figures below (fig.14, fig. 15, fig. 16) present results of sample tests of when the actuator’s control channel was degraded. 0.1 [s] time delay was modelled in this control channel. That time the control precision was not recognised as good enough.

Figure 13. Errors signalised by Error estimator module, a) values of position and oscillation errors, b) Final error’s value.

Figure 14. Flight parameters of simulated model of plane, a) pitch angle, b) pitch rate. – broken down control channel
Because of existing oscillations, the position of simulated elevator didn’t match desired elevator’s position calculated by FCA algorithm.

According to expectations due to the control system quality lowered below assumed values the control system error detector produced the error signal.

VI. Conclusions and Future Research

The paper reports RUTRT contributions to European Community SCARLETT project. At the current state of the development RUTRT PCA the software quality was evaluated by conducting a set of real-time simulations. The results of simulations presented in sections IV and V prove that software modules may execute according to system specification. Moreover the built-in test procedures may effectively detect errors on both communication and quality of control system levels.

Currently, the preliminary application test are performed on the target hardware platforms (CPM and REU). Final tests are planed during spring and summer of year 2011.

Acknowledgments

Programme SCARLETT and research reported in the paper are founded by 7th European Framework Project. The part of hardware components applied in the research published in this paper was founded by European Union Operational Programme - Development of Eastern Poland, project no. POPW.01.03.00-18-012/09.

References.

11. http://www.scarlettproject.eu