REAL-TIME HARDWARE-IN-THE-LOOP SIMULATION OF FLY-BY-WIRE FLIGHT CONTROL SYSTEMS

Eugenio Denti*, Gianpietro Di Rito*, Roberto Galatolo*

** University of Pisa – Department of Aerospace Engineering*

Keywords: *Aircraft Systems Integration, Real-Time Simulation, Hardware-In-The-Loop Testing*

Abstract

The paper deals with the development and the set-up of a real-time/hardware-in-the-loop simulation system of the Fly-By-Wire flight control system of a modern aircraft. The plant is designed as a virtual iron-bird, which integrates a parallel computing PC network with hardware components such as a human-pilot interface, the flight control computers and a hydraulic rig for testing flight actuators. The work highlights the challenging problems related to the design of such a complex experimental set-up, from the integration of real-aircraft components to the development of a real-time executable code that includes very complex models of the aircraft systems.

1 Introduction

The design of a modern Fly-By-Wire Flight Control System (FBW/FCS) is a complex multidisciplinary problem, in which the needs of different areas of engineering, such as flight mechanics, automatic control, hydraulics and structural dynamics, must be integrated and harmonized. In order to predict and compensate possible problems at a component as well as at a system level before flight testing, the FBW/FCS must be verified and validated by performing successive phases of ground tests: from software simulations and hardware-in-the-loop simulations, up to the iron-bird facility, related to the highest level of hardware-in-the-loop integration [1], [2].

In the recent years, the Department of Aerospace Engineering of the University of Pisa has been involved in research programs oriented to the study of modern FBW/FCS [3], [4]. One of the main objectives of the research is the set-up of a virtual iron-bird of a FBW aircraft, which can be used for the preliminary validation of the FCS by means of real-time/man-in-the-loop/hardware-in-the loop simulations.

The basic characteristic of the proposed simulation platform is the flexibility. The same plant can be used for the development and the experimental validation of the models of the flight actuators [5], [6], for the development and the testing of an all-software aircraft simulations (both off-line and real-time), as well as for the hardware-in-the-loop testing of one or more aircraft subsystems (flight actuator, sensors, flight control computers).

The paper is divided into two main sections: a first one dedicated to the general description of the simulation plant, and a second one more focused on the simulation code, pointing out the problems related to the real-time execution and proposing an innovative solution based on the use of a low-cost parallel computing network.

2 The Simulation Plant

Figure 1 shows a scheme of the virtual-iron bird at the highest level of hardware-in-the-loop integration, in which the plant includes an Engineering Test System (ETS), a simplified human-pilot interface, the Flight Control Computers (FCC) and a hydraulic rig for testing one of the primary flight actuators.

As a result of the plant flexibility, the architecture shown in Fig. 1 is not the only possible configuration for the virtual iron-bird.

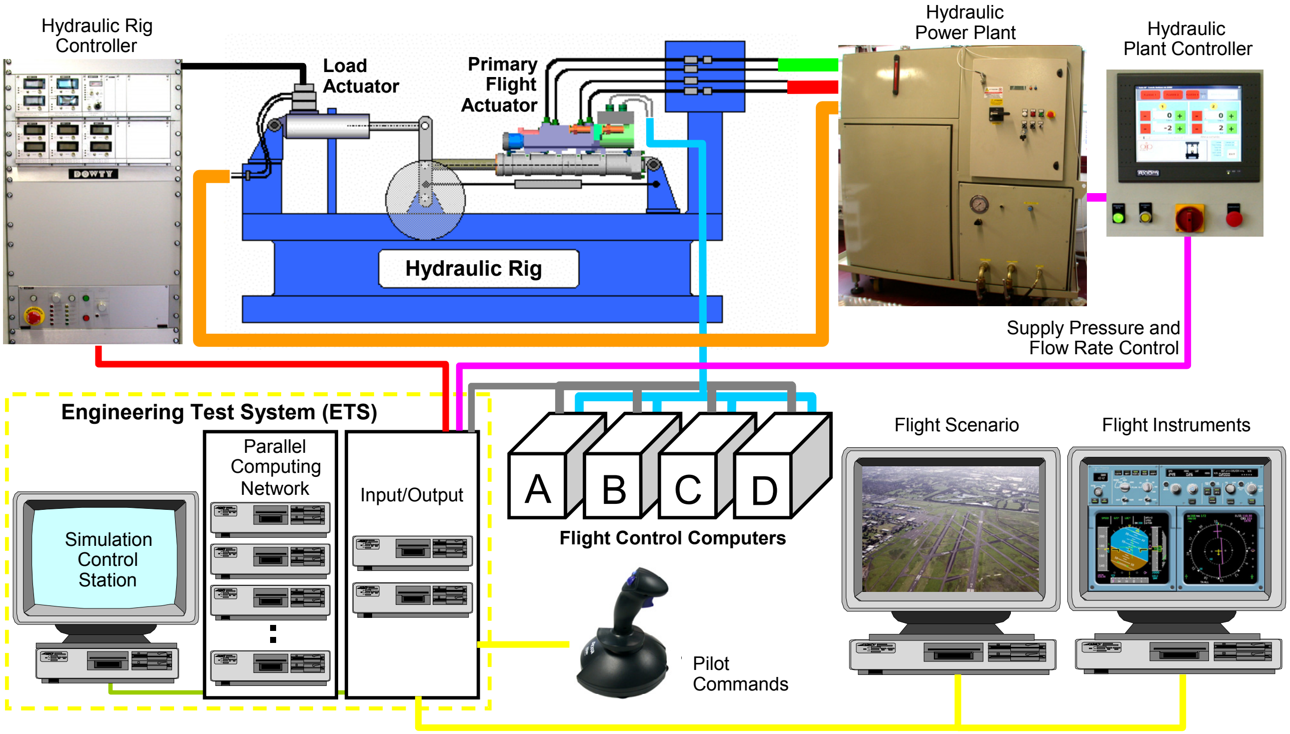

Fig. 1. Virtual Iron-Bird for Real-Time Hardware-In-The-Loop Simulations.

All-software real-time simulations as well as simplified hardware-in-the-loop simulations can also be performed. This feature has been a major requirement during the plant design, since the ability to perform integration studies with increasing levels of complexity is a crucial topic in the development and the verification of a FBW/FCS [1].

2.1 Engineering Test System

The core of the simulation plant is the ETS, whose architecture is schematically reported in Fig. 2. The ETS is basically composed of a Input/Output (I/O) section and a parallel computing network for the real-time execution of the FCS model, § 3.

The I/O section of the ETS provides the hardware-in-the-loop components with the appropriate electrical interfaces and operates (Target PC 7, Fig. 2) the closed-loop controls on both the load applied to the flight actuator and the hydraulic plant parameters (§ 2.2-3).

As already pointed out, one of the most interesting features of the simulation plant is the flexibility. Different versions of the virtual iron-bird can be obtained according to the desired hardware-in-the-loop test. For this reason, Fig. 2 shows the ETS architecture with reference to two configurations of the virtual iron-bird: the one in which the FCC are integrated in the simulation, and the other in which the FCC are not present (but their functions are simulated), and the only hardware-in-the-loop is the flight actuator.

2.1.1 Simulation with the FCC

When the virtual-iron bird integrates the FCC, all the electrical interfaces between the FCC and the aircraft systems must be reproduced by the ETS. Basically, the ETS must receive from the FCC the commands for the simulated subsystems and must send to them the appropriate output signals (positions of the simulated actuators, signals of the aircraft sensors, etc.).

In order to do that, two Target PCs of the ETS are used, Fig. 2. Target PC 6 is equipped with digital cards reproducing the data bus protocols MIL-1553B and ARINC-429, and it simulates the digital signals coming from the aircraft sensors (air-data sensors, inertial sensors, etc.). Target PC 7 is equipped with

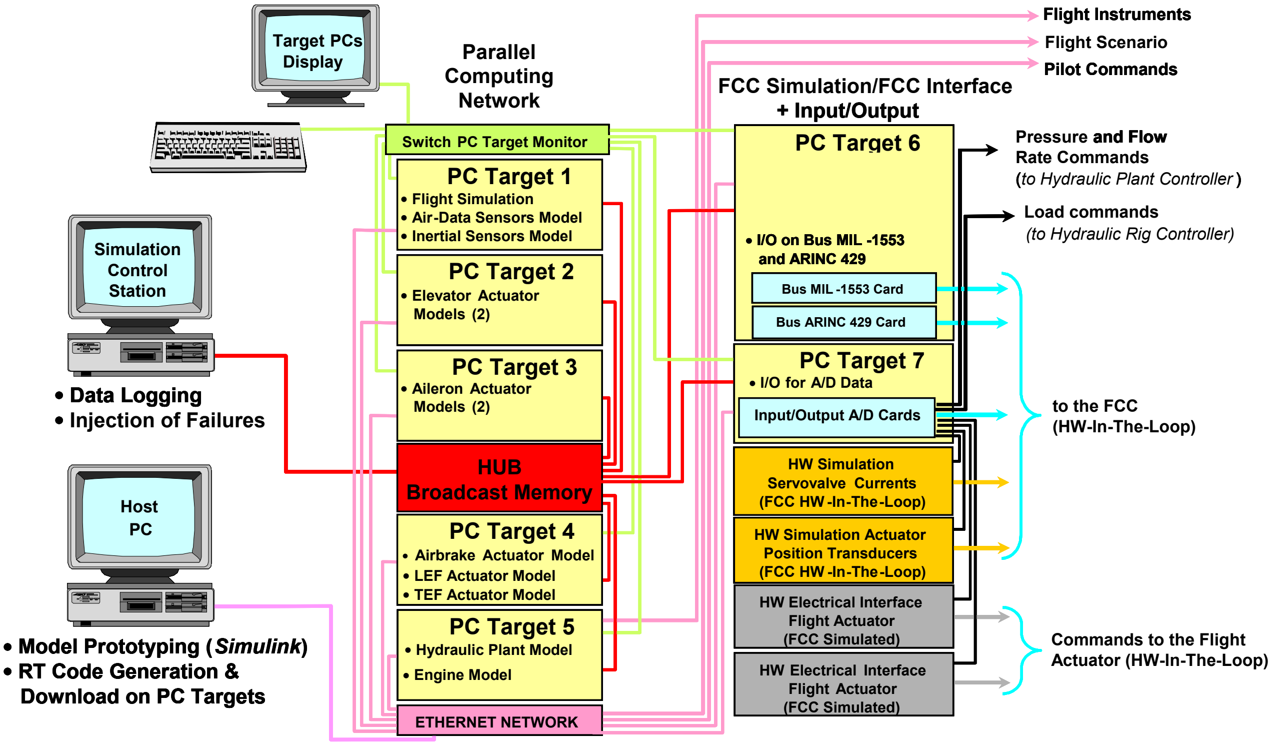

Fig. 2. Engineering Test System for Real-Time Simulations.

A/D cards, and it converts into the analog format the signals coming from the simulated flight actuators (currents, positions, etc.).

2.1.2 Simulation without the FCC

When the simulation is performed without the FCC, the hardware-in-the-loop flight actuator is controlled by a specific Actuator Control Unit (ACU), which has been realised and integrated into the Hydraulic Rig Controller, Fig. 1. The ACU communicates in real-time with the I/O section of the ETS (Fig. 2), and (only for this plant configuration) it operates the closed-loop control on the actuator, according to the commands generated by the FCC model running in real-time in the ETS.

The flight actuator used for the research is a Direct-Drive servoValve (DDV) tandem actuator with quadruple electrical redundancy. Thus, the ACU is equipped with four servo-amplifiers (for supplying the four DDV coils), eight LVDT conditioners (four ones for the DDV and the others for the actuator), and two logic switches (related to the two by-pass valves of the tandem actuator). In addition, a specific electrical circuitry has been designed and realized for the artificial injection of electrical failures, in order to verify the failure management logics as well as to measure the failure transients.

2.2 Hydraulic Rig

As shown in Fig. 1, the hydraulic rig includes two actuators: the flight actuator (which is position-controlled) and a high-bandwidth force-controlled actuator [7]. A specific control unit, called Hydraulic Rig Controller (HRC), provides the load actuator with the appropriate electrical interface. The HRC communicates in real-time with the I/O section of the ETS, which operates the closed-loop force control, in order to reproduce the aerodynamic loads calculated by the model of the control hinge moments running in real-time in the ETS.

In order to study possible structural coupling phenomena, the supports connecting the flight actuator to the workbench have been designed reproducing the values of the structural stiffness of the aircraft constraints.

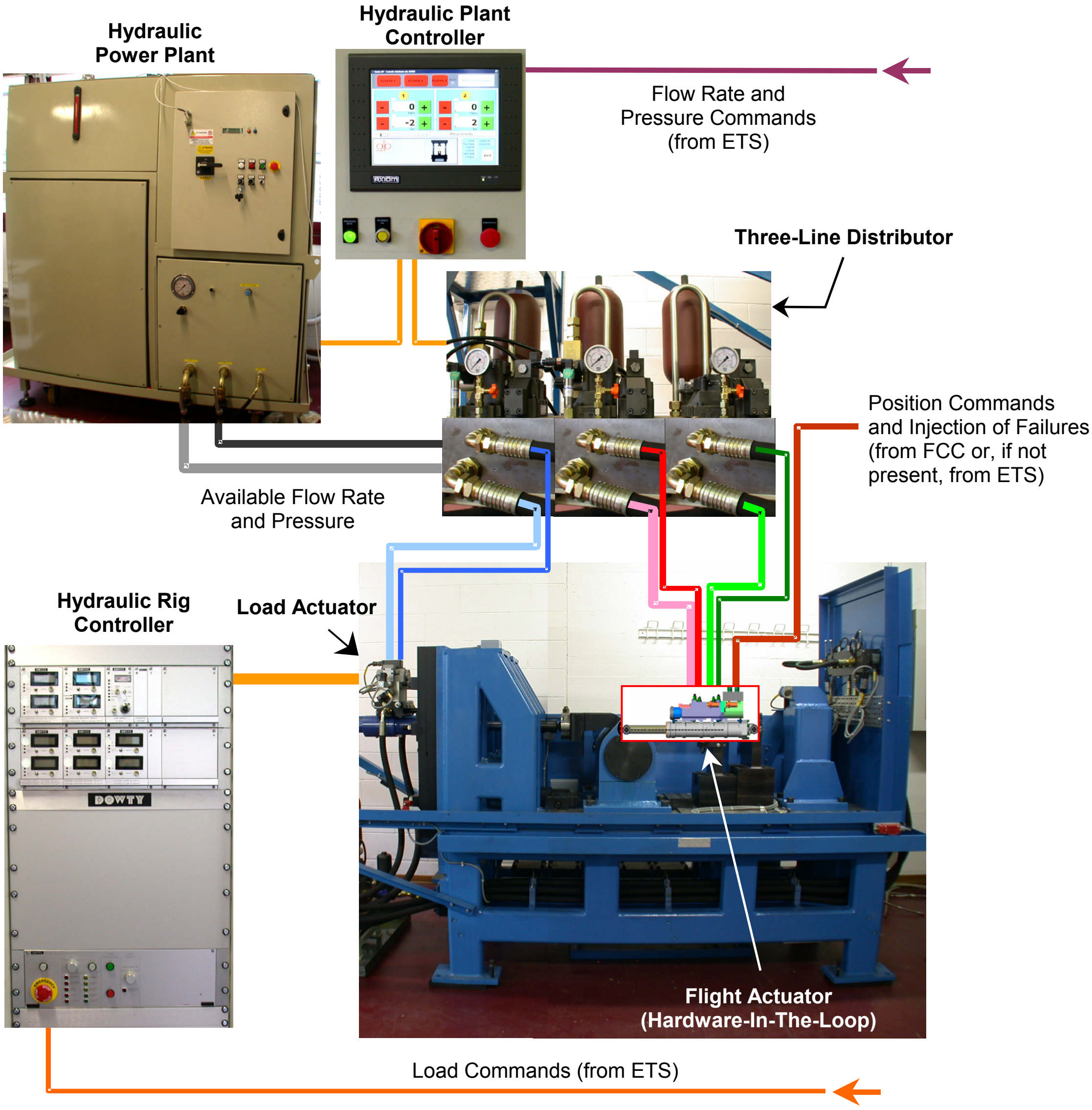

Fig. 3. Generation and Distribution of the Hydraulic Power in the Simulation Plant.

The rig is generally supplied by three hydraulic lines, two ones for the flight actuator, and one for the load actuator. This architecture allows to test tandem flight actuators, which are typical for the actuation of the primary controls in modern FBW aircrafts (EFA, Aermacchi M346, etc.).

2.3 Generation and Distribution of the Hydraulic Power

The hydraulic power is generated by a variable displacement axial piston pump, capable of providing a flow rate of 100 litres/min at a pressure of 210 bar. Higher pressures can also be achieved, but this induces a diminution of the flow rate.

The hydraulic power generated by the pump is supplied to the hydraulic rig by means of a three-line distributor, Fig. 3. In the distributor, each of the high-pressure lines of the flight actuator is equipped with a pressure-regulating servovalve, a flow-limiting servovalve, a flow rate transducer and a pressure transducer. A specific control unit, called Hydraulic Plant Controller (HPC), provides all these components with the appropriate electrical interface. The HPC communicates in real-time with one of the I/O section of the ETS, which operates the closed-loop control on the flow rate and the pressure supplying the flight actuator, according to the results of the model of the aircraft hydraulic plant running in real-time in the ETS.

2.4 Simplified Human-Pilot Interface

The human-pilot interface is basically composed of a joystick and by two off-line PCs: one dedicated to the visualisation of the flight instruments (by means of a panel developed in the *LabView* environment), and the other to the visualisation of the flight scenario (by means of *Flight Simulator 2002*).

The signals necessary to visualise the flight instruments are generated by the aircraft model running in the ETS and sent to the off-line PC via Ethernet (with UDP protocol). The same data link is used to send the signals of latitude, longitude, altitude, Euler angles and velocity to the PC dedicated to the flight scenario.

3 The Parallel Computing Network

3.1 Real-Time Simulation of a FBW/FCS

The reliability of the results of a hardware-in-the-loop simulation of a system basically depends on the capability to reproduce the actual operative conditions of the system itself [1]. This implies the set-up of appropriate hardware interfaces (§ 2) as well as the development of a real-time simulation code, able to reproduce, at a highest level of accuracy, the dynamic interaction of the hardware with its working environment.

The real-time simulation of a FBW/FCS is a very complex topic, since accurate results can be obtained only with highly sophisticated models of the FCS subsystems, which need very small integration steps and high computing resources. As an example, a comparative study [8] on models characterised by different levels of complexity pointed out that the maximum allowable integration step for a satisfactory model of one of the flight actuators is of about 10^{-4} seconds. Considering that the simulation of a FBW/FCS must include many actuators as well as the hydraulic plants, the sensors and the aircraft dynamics, the only low-cost solution is to realise a parallel computing network of common off-the-shelf PCs connected by a high-speed data link.

3.2 Network Architecture

As already mentioned in § 2.1, a section of the ETS is dedicated to the real-time simulation, Fig. 2. This section is composed of a parallel computing network of PCs (named Targets) for the real-time execution of the simulation code, a PC (named Host PC) for the prototyping and the preparation of the real-time code, and a PC (named Simulation Control Station) for the data logging and the simulation management.

The model of the whole FBW/FCS is developed on the Host PC in the Matlab-Simulink environment. The executable code is generated by means of the xPC Target tools of Matlab, and automatically distributed via Ethernet to the Target PCs (§ 3.3) by means of a specifically designed software tool.

Each Target PC is provided with a broadcast memory card, which is connected by optic fibres with a central hub, Figs. 2 and 5. The broadcast memory card is basically a shared RAM card, containing the local data as well as the data of all the cards connected with the hub. When a Target PC writes a data in its local card, the central hub writes a copy of such a data in the cards of all the other Target PCs, with a time delay of few nanoseconds. In such a way, each Target PC works with the same set of data at each time step.

3.3 Main Characteristics of the Simulation Code

The simulation code used in the virtual iron-bird starts form the development of a model of the FCS of a modern FBW aircraft in the Matlab-Simulink environment. The model is split into a number of conditionally-executable blocks that is equal to the number of Target PCs used in the parallel computing network (e.g. Fig. 4). In the top-level of the Simulink diagram, an *Enable* block (which basically generates Boolean signals) is inserted and

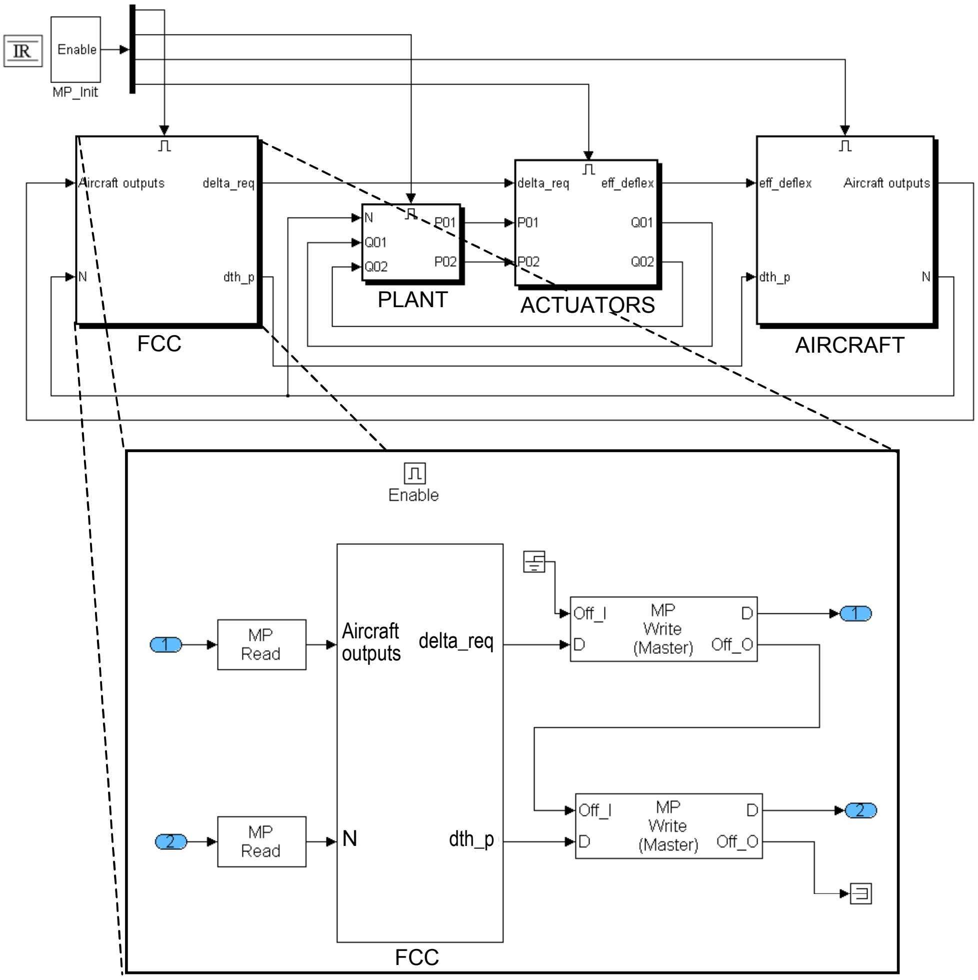

Fig. 4. Simulink Model of the FBW/FCS for a Parallel Computing Network with four PCs.

connected to the conditionally-executable blocks of the model (e.g. Fig. 4).

If a real-time simulation is to be performed, the following procedure is applied:

1. the executable code is generated by means of the xPC Target tools and sent via Ethernet to each Target PC;
2. on each Target PC, the *Enable* block activates only one conditionally-executable block and the data flow between the disabled blocks and the active one is interrupted and automatically redirected on the broadcast memory card (e.g. Fig. 5);
3. one of the Target PC (named Master) operates the synchronization of the others (named Slaves).

Two Simulink blocks, specifically designed for the scope, obtain the data flow interruption mentioned at point 2: the *MP Read* block and the *MP Write* block, related to the input and the output data respectively, Fig. 4.

On the other hand, in order to perform an off-line simulation, the model can be normally executed in the Simulink environment, as in this configuration the *Enable* block activates all the conditionally-executable blocks and a normal data flow is allowed between the blocks.

The major advantages provided by the proposed solution are:

- flexibility;
- modularity;
- possibility to be tested and updated by non-expert users (e.g. students);
- possibility to run off-line simulations on a common off-the-shelf PC;
- possibility to run real-time simulations on the parallel computing network.

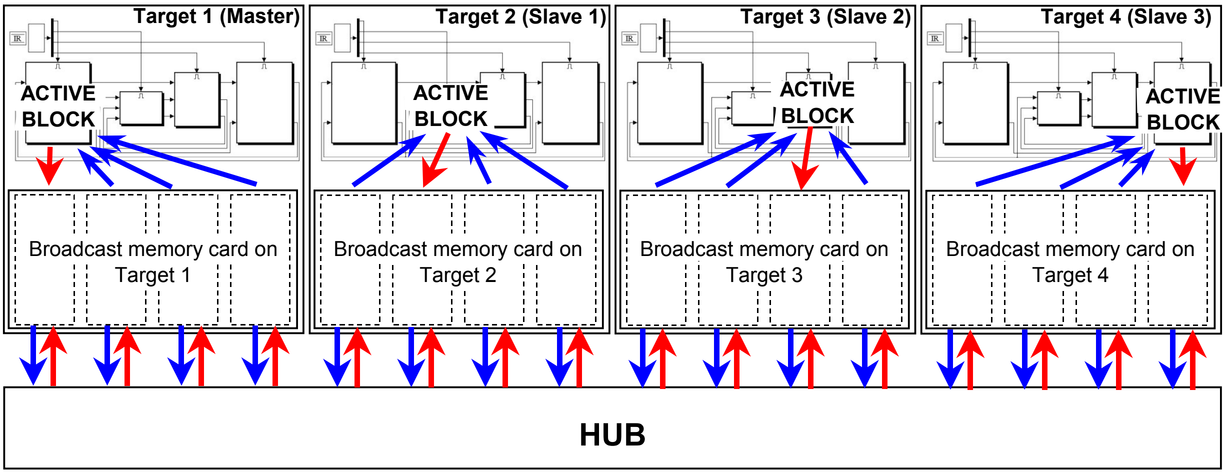

Fig. 5. Real-Time Simulation on the Parallel Computing Network.

These features are of basic importance in order to harmonize the development of the virtual iron-bird with other research activities, more focused on the study of specific components and subsystems.

3.4 Simulation Management

In real-time procedures, the operations of reading and writing on the hard disk are generally not allowed, so the data logging of a simulation parameter as well as the updating of a parameter during the simulation cannot be obtained in real-time. For these reasons, the real-time section of the ETS includes an off-line PC containing a broadcast memory card: the Simulation Control Station (SCS), Fig. 2. From this PC, the user can manage the simulation, saving on the hard disk any output parameter or modifying any input parameter (e.g. failure events).

3.5 Limitations of the Code

In the present version of the code, all the Target PCs of the parallel computing network use the same integration step (about 10^{-4} seconds), a fact that induces a significant waste of resources. Actually, the FBW/FCS model includes the simulation of physical systems with quite different dynamic characteristics. If an integration step of 10^{-4} seconds is necessary for the simulation of the actuators, it is excessive for the simulation of the aircraft dynamics.

A differentiation of the integration steps among the Targets would diminish the number of PCs of the simulation plant, and a more optimised network would be obtained. However, this approach is not applicable with the present network architecture. Actually, the fact that all the Targets run with the same integration step allows only one of them (the Master) to operate the synchronization task. If the integration steps were different among the Targets, each PC should be equipped with a timer-card, and each card should be synchronized with the others.

Conclusion

The research led to the development of a simulation plant of a modern FBW/FCS that can be defined a "virtual iron-bird", since the simulation is performed using real-aircraft hardware components and complex experimentally-validated models of subsystems.

The most relevant result of the work is the development of a flexible and modular low-cost real-time parallel computing network, which is capable of executing models of aircraft systems characterised by an integration step of about 10^{-4} seconds. The network is basically composed of common off-the-shelf PCs equipped with broadcast memory cards and connected by an optic fibres data link.

The simulation code of the whole FCS has been developed in the Matlab/Simulink environment, and it can be either executed off-line on a single PC, or automatically distributed and executed in real-time on the parallel computing network. The off-line simulation gives the possibility to update and test the models of aircraft subsystems before using them for real-time simulations. As far as the real-time simulation is concerned, the executable code is generated by means of the xPC Target tools of Matlab, while the distribution and execution of the model on the network is obtained by means of a specifically designed software tool.

The present configuration of the plant, composed of seven PCs, is capable to simulate the dynamics of a whole FBW/FCS integrating, as hardware-in-the-loop, the flight control computers and one of the primary flight actuators.

References

[1] Webster F. and Smith T. D. *Flying qualities flight testing of digital flight control systems*. RTO-AG-300, Flight Test Techniques Series, Vol. 21, 2001.

[2] Fielding C. *The design of fly-by-wire flight control systems*. BAE Systems (Operations) Limited, 2000.

[3] Denti E. and Galatolo R. A research laboratory for the study of fly-by-wire flight control systems (in Italian). *Proceedings of the 17th AIDAA National Congress*, Rome, Italy, pp. 997-1004, 2003.

[4] Bernardini M., Cecchelli G., Denti E. and Galatolo R. A parallel computing network for the real-time simulation of fly-by-wire flight control systems (in Italian). *Proceedings of the 17th AIDAA National Congress*, Rome, Italy, pp. 549-558, 2003.

[5] Di Rito G. and Galatolo R. Modelling the dynamics of a direct-drive servovalve for fail-operative primary flight control actuators. *Proceedings of the 3rd International Conference on Recent Advances in Aerospace Actuation Systems and Components*, Toulouse, France, 2004, pp. 67-72.

[6] Di Rito G., Denti E. and Galatolo R. Experimental validation of the model of a fault-tolerant servo-hydraulic flight actuator for FBW/FCS (in Italian). *Proceedings of the 18th AIDAA National Congress*, Volterra, Italy, 2005.

[7] Di Rito G., Denti E. and Galatolo R. Robust force control in a hydraulic workbench for flight actuators. *Proceedings of the IEEE Joint Conference CCA, ISIC and CACSD*, Munich, Germany, 2006.

[8] Denti E., Di Rito G. and Galatolo R., Modelling hydraulic actuators for aircraft primary control surfaces (in Italian). *Proceedings of the 16th AIDAA National Congress*, Palermo, Italy, 2001.