Simulation Decomposition for the Computer Based Study of Aircraft Fuel Management Distributed Control

J. M. GIRON-SIERRA, M. SEMINARIO
C. INSAURRALDE, J. F. JIMÉNEZ.
Department: Arquitectura de Computadores y Automatica,
Universidad Complutense de Madrid.
Av. Complutense S/N, 28040 Madrid
SPAIN

Abstract: The paper describes part of an EU research project, aiming at the development of a new distributed aircraft fuel management system, using smart components. Along the research is necessary to develop and check candidate solutions for the distributed control. A simulation system has been created, using two interconnected computers. One of the computers simulates in real time the networked electronic part of the smart components. The other computer simulates the fuel plant, with the tanks, valves, pipes, etc. Both computers interact with the same analog or discrete signals that are present in the real aircraft fuel system. The paper describes this simulation system, and presents some experiments which reproduce the main operations of the fuel management system.

Key-Words: - Aircraft subsystem modeling, Aircraft subsystem simulation, Aircraft Fuel Management System, Distributed simulation.

1 Introduction

The objective of this research is to develop a computer simulation system to support the development of a new distributed fuel management system for aircrafts. The purpose of the simulation is to test the system in a set of experimental conditions, including normal operations and pertinent reactions when there are malfunctions. Computer simulations have the advantage of allowing for non dangerous experiments, the kind of experiments that should not be done on real aircrafts.

This paper is concerned with part of an European Community Research Project, denoted as "Smartfuel". Conventional aircraft fuel management systems use a central specific computer, connected point-to-point to each component, such sensors, valves, and pumps. The target of the Smartfuel project is to develop a new fuel management system, distributed along a fieldbus, and using smart components. Digital processing capabilities are included in each component, sensor, valve, pump, making it a smart component. By using such new components, the central computer is no longer needed for control, only for human interface. Our new simulation system corresponds to the new fuel management system.

The project Smartfuel involves four industrial partners makers of components, two companies making aircrafts, and three universities providing scientific support for the research. The CANbus has been selected as the fieldbus. The research started with a requirements phase, then proceeded with new ideas for distributed control, then provided a communication protocol between smart components, and finally came to implementation and experimental testing.

Aircraft fuel systems are described in books such [1] and [2]. Very few scientific articles deal with such systems. We found only an experimental reference in [3] and [4]. There is an overview of avionic data buses in [5]. In this overview there is no reference to fieldbuses. The application of fieldbuses in avionics is a new issue. The book [6] contains a general treatment of fieldbuses. The CANbus is fully described, together with applications, in [7] There are not many references about smart components, although they are now in common use in certain industrial sectors; let us quote [8] and [9] as books on this topic.

Due to the characteristics of the system to be simulated, a decomposition of the simulation into two interconnected computers has been decided. This is the main point of this paper. One of the computers is devoted to the simulation of fast phenomena, which take place around the CANbus: the exchange of messages, the smart component microcontrollers operation. The other computer simulates the fuel plant: tanks, pipes, valves, pumps, and fuel flow. The phenomena on the fuel plant are slow. The development of the simulation in each computer has specific requirements, having specific solutions: not the same for fast processes than for slow processes.

The paper starts with a description of the problem, the system to be simulated, and then describes the solution, in terms of simulation decomposition and how the simulation is developed for each computer. Experiments with the simulations are described, for the main fuel system operations. Finally some conclusions are presented.

2 Problem Formulation

2.1 Fuel Management Systems

The research considers two types of aircrafts: helicopters and airplanes. Generic cases have been depicted for each type.

Figure 1 shows a top view of a helicopter fuel system. The four tanks are connected to form two subsystems. One consists of the front and right hand tank, the other consists of the rear and left hand tank. All tanks are at the same horizontal level, under the pilot. The connections between tanks allow for simple gravity transfers between them.

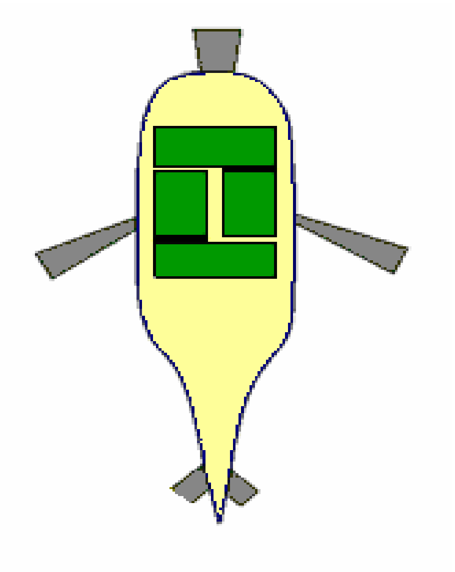


Fig 1. Top view of the helicopter fuel management system

Figure 2 shows in more detail the helicopter fuel management system. The pilot must know the fuel quantity, so there are fuel gauging sensors in each tank. The engine supply is made from the left and right hand tanks. The pressure refuelling is done through two valves (near bottom left in the figure), one is connected to the rear tank and the other connected to the right hand tank.

There is a balancing system that can be activated in certain cases. It consists of a connection between left and right hand tanks, through a bidirectional pump and a valve.

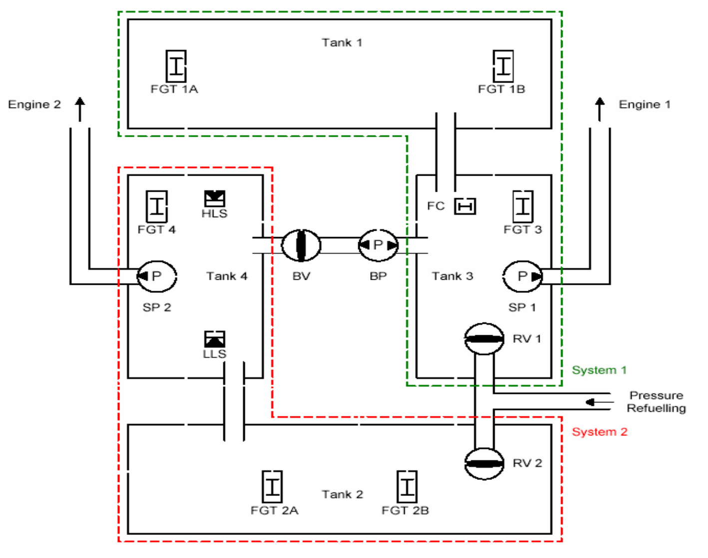


Fig. 2. The helicopter fuel management system in detail

Figure 3 shows a simplified top view of the airplane fuel management system. There are three tanks in each wing, and another tank in the tail. The wings inclination allows for gravity transfers of fuel, to guarantee engine supply.

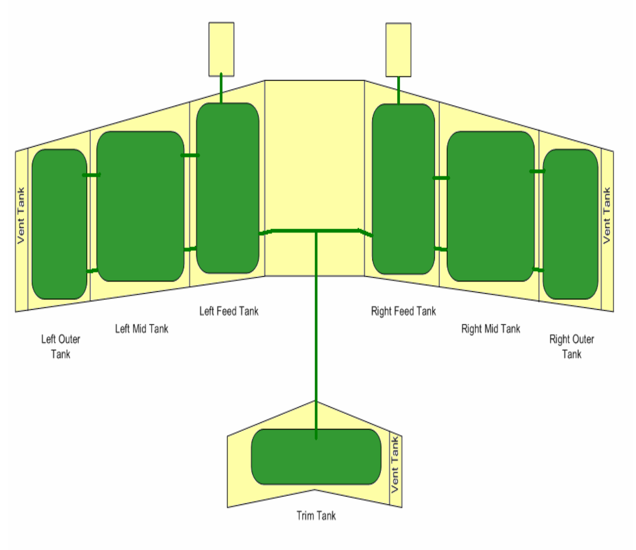


Fig 3. Top view of the airplane fuel management system

Balancing connections through valves and pumps are provided. The tail tank is used for the position control of the CoG. There are transfers from and to the wing tanks, to keep the CoG in an optimal region along the several phases of a flight.

2.2 The New Distributed Proposal

Conventional fuel management systems have a central computer and point-to-point wirings to the sensors and actuators. Figure 4 depicts the concept.

Fig. 4. Concept of conventional fuel management systems.

In this research a new distributed architecture is pursued. A CANbus is used together with CAN enabled smart components. Figure 5 shows the concept.

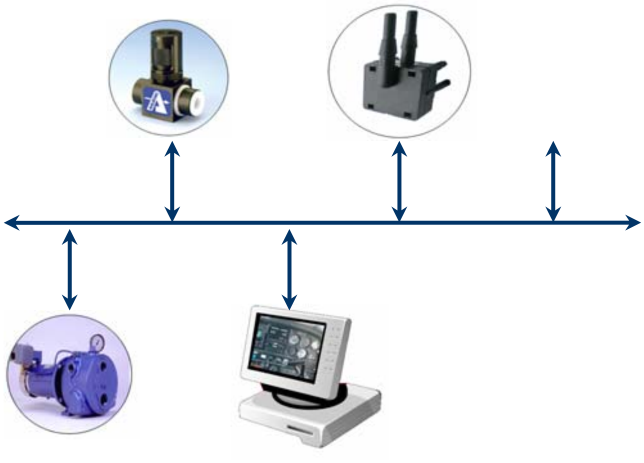


Fig. 5. New concept, a distributed fuel management system.

With the new system, there is no central computer in charge of fuel transfers control. The only computer in the system is used for pilot interface. Advantages of the new system are that less cabling is required, simpler connectors, simpler architecture, and more powerful maintenance facilities.

A functional solution has been developed for distributed control. Each smart component has the same control code into its microcontroller. There is an I.D. code for each component, telling if this is a pump, a valve, etc. Using the I.D. each component select its own behaviour from the set of behaviours included in the common code.

2.3 Smart Component Architecture

Figure 6 shows a block diagram of the smart components. From top to bottom it has the sensor of actuator itself, then an interface to a digital microcontroller, then a CAN module, which is the CAN enabled microcontroller, and then a connection to the CANbus.

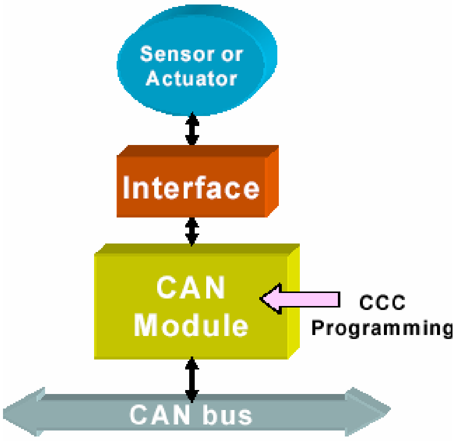


Fig. 6. Block diagram of a smart component

A key for our simulation development is to focus in the connection between the interface block and the sensor or actuator. Here we have either analog signals from the sensor, or digital signals for on/off control of the actuators (to open or close a valve, to make a pump run or stop). These signals will be reproduced in our simulation system.

2.4 Fuel System Levels

The fuel management system can be seen as two interacting subsystems. One is formed by the tanks, pipes, valves, pumps, sensors, etc., with fuel flow. Let us denote that level as the physical plant. The other level is formed by the microcontrollers, one for each component, which are connected to the CANbus. Let us denote that as the electronic level. Figure 7 shows a block diagram of fuel system highlighting the two levels. The connection of both levels is made with the analog or digital signals between the interface and the actuator or sensor in figure 6.

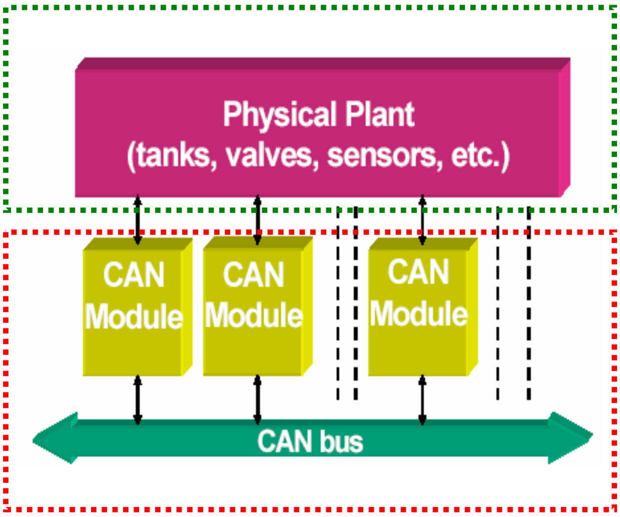


Fig. 7. Two levels in the fuel system

3 Problem Solution

A simulation system, using two interconnected computers, has been developed. The objectives of the simulation are the following:

- To check the correct behaviour of the common code to be embedded in the real smart components. With the simulation it is possible to detect any problem before the smart components were built.
- To study the behaviour of the CANbus message traffic during the fuel system operations. See whether there are problems such collisions, overload or information missing.
- To study malfunctions in the physical plant or in the electronic level. Non destructive experiments can be done in simulation.

3.1 Structural solution for the simulation

To implement the simulation we preferred the decomposition of the simulation into two computers exchanging the analog and digital signals of the real two-level system. The reason for this decomposition is that the fuel system has two types of dynamic response. The physical plant is slow, and the electronic level is fast.

One of the computers simulates the physical plant. The other computer simulates the electronic level. USB ports, and A/D and D/A converters are used for the interconnection between computers [10]. Figure 8 shows a block diagram of the simulation system

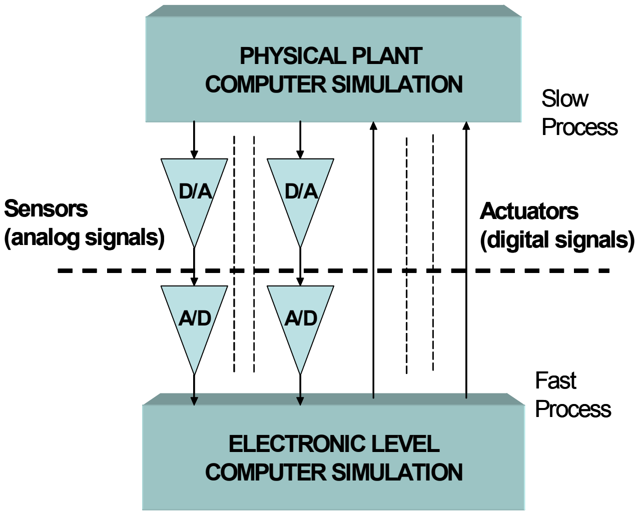


Fig.8. Decomposition of the simulation into two interconnected computers

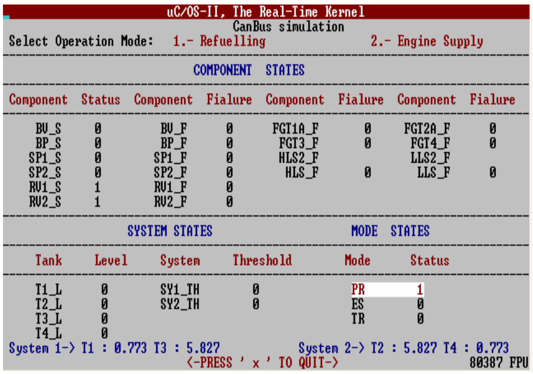
3.2 Simulation of the electronic system and the communication channel

The electronic level of the fuel system is made with several nodes, each one with a microcontroller. Ecah node has the same code. The simulation of this level is done with a real-time multitasking operating system. A set of tasks are devoted to model the nodes, one task per node. The tasks work together exchanging information according with the procedures of the CANbus.

With this simulation concept, issues pertaining to communications (collosions, overload, etc.) can be studied.

Regarding the implementation, many options for the operating system are available, for instance VxWorks, Linux Real Time, RTI and mCOS-II (The real time kernel). We selected mCOS-II [11] because the programming is simple in C++, and applications run in real time over DOS, so they are deterministic.

Figure 9 shows one of the screens of the electronic system simulation. Screens are textual, to avoid the delays caused by graphical visualizations.



```
                        uC/OS-II, The Real-Time Kernel
                              CanBus simulation
Select Operation Mode:   1.- Refuelling            2.- Engine Supply
--------------------------------------------------------------------------
                          COMPONENT  STATES
--------------------------------------------------------------------------
Component  Status  Component  Fialure  Component  Fialure  Component  Fialure
--------------------------------------------------------------------------
   BV_S      0        BV_F       0      FGT1A_F      0      FGT2A_F      0
   BP_S      0        BP_F       0       FGT3_F      0       FGT4_F      0
  SP1_S      0       SP1_F       0       HLS2_F              LLS2_F
  SP2_S      0       SP2_F       0        HLS_F      0        LLS_F      0
  RV1_S      1       RV1_F       0
  RV2_S      1       RV2_F       0
--------------------------------------------------------------------------
                  SYSTEM STATES                      MODE  STATES
--------------------------------------------------------------------------
    Tank      Level     System     Threshold      Mode      Status
--------------------------------------------------------------------------
    T1_L        0       SY1_TH         0           PR          1
    T2_L        0       SY2_TH         0           ES          0
    T3_L        0                                  TR          0
    T4_L        0
System 1-> T1 : 0.773 T3 : 5.827             System 2-> T2 : 5.827 T4 : 0.773
                          <-PRESS ' x ' TO QUIT->                   80387 FPU
```

Fig. 9. Screen of the electronic system simulation

3.3 Simulation of the physical plant

Since the physical plant works as a slow process, there is more freedom for the operating system and the graphical visualization. Window XP was chosen as operating system, and Microsoft Visual C++.net for the development of the simulation.

An important reason for this selection is that the same GUI environment developed for the simulation will be used as the pilot interface for the real system.

In the following section some simulation experiments will be described, showing the screens displayed by the physical plant simulation computer.

3.4. Simulation experiments

The simulation system is used for the study of normal operations, and abnormal situations derived from malfunctions. Let us show some examples of simulation experiments.

3.4.1 Simulation of the helicopter

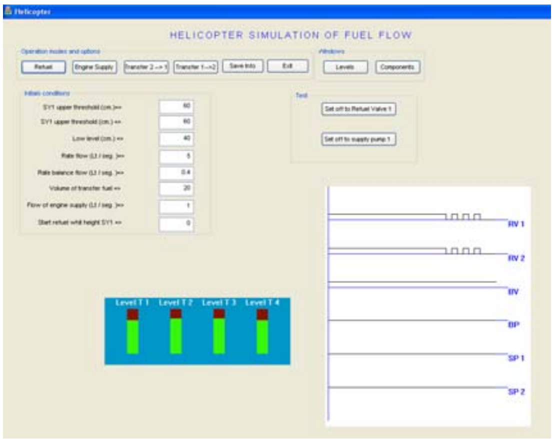
Figure 10 shows the main screen to define and visualize simulation experiments for the helicopter fuel system. The left diagram is a top view of the fuel system. The level of the tanks move in real time to visualize the fuel levels evolution along the operations (refueling, engine supply, balancing).

Tank 1	Tank 2	Tank 3	Tank 4
0.0	5.0	5.0	0.0
1.4	8.6	8.6	1.4
3.1	11.9	11.9	3.1
5.0	15.0	15.0	5.0
7.0	18.0	18.0	7.0
9.1	20.9	20.9	9.1
11.3	23.7	23.7	11.3
13.5	26.5	26.5	13.5
15.8	29.2	29.2	15.8
18.1	31.9	31.9	18.1
20.5	34.5	34.5	20.5
22.8	37.2	37.2	22.8
25.2	39.8	39.8	25.2
27.6	42.4	42.4	27.6
30.1	44.9	44.9	30.1
32.5	47.5	47.5	32.5
35.0	50.0	50.0	35.0
37.4	52.6	52.6	37.4
39.9	55.1	55.1	39.9
42.4	57.6	57.6	42.4
44.8	60.2	60.2	44.8
47.3	57.7	57.7	47.3
49.4	60.6	60.6	49.4
51.5	58.5	58.5	51.5
53.2	61.8	61.8	53.2
55.0	60.0	60.0	55.0

Fig. 10. Main screen of the helicopter simulation

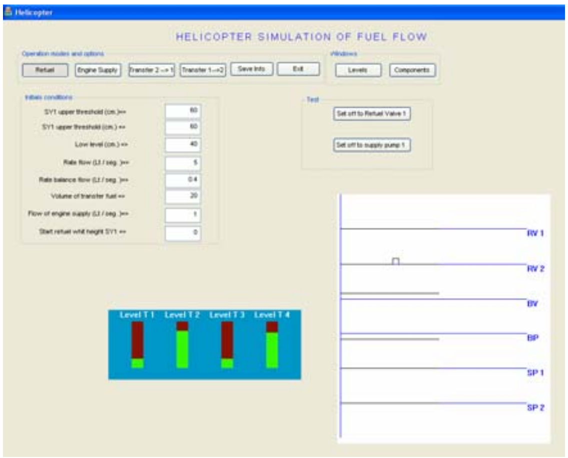
There are other screens available to see the evolution vs. time of the fuel levels and the on/off action of the actuators. Figure 11 shows one example for the refueling.

Figure 12 shows the reaction of the system for a malfunction of one of the refueling valves. This is a malfunction that has been considered my the smart components code. In this case, the balancing system, between the two fuel subsystems, is used.



SY1 upper threshold (cm.)>= 60
SY1 upper threshold (cm.) <= 60
Low level (cm.) <= 40
Rate flow (Lt / seg.)>= 5
Rate balance flow (Lt / seg.)>= 0.4
Volume of transfer fuel <= 20
Flow of engine supply (Lt / seg.)>= 1
Start refuel whit height SY1 <= 0

Fig. 11. Screen showing the evolution of events along time during a refueling operation



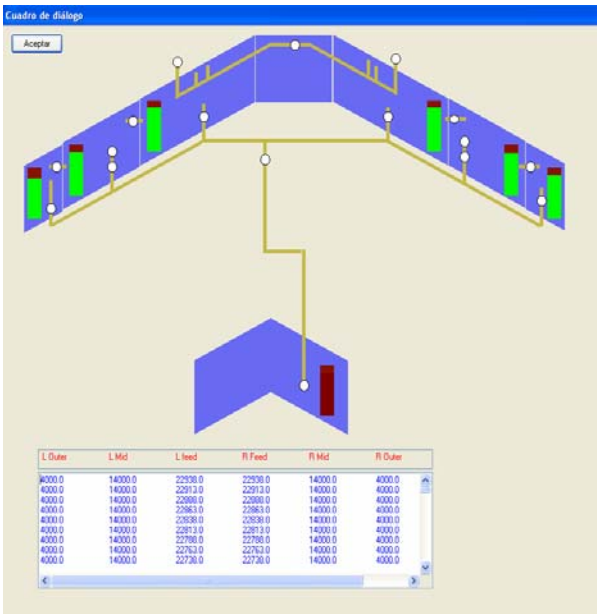
SY1 upper threshold (cm.)>= 60
SY1 upper threshold (cm.) <= 60
Low level (cm.) <= 40
Rate flow (Lt / seg.)>= 5
Rate balance flow (Lt / seg.)>= 0.4
Volume of transfer fuel <= 20
Flow of engine supply (Lt / seg.)>= 1
Start refuel whit height SY1 <= 0

Fig. 12. Screen showing the evolution of events along time during a refueling with a malfunction

3.4.2 Simulation of the airplane

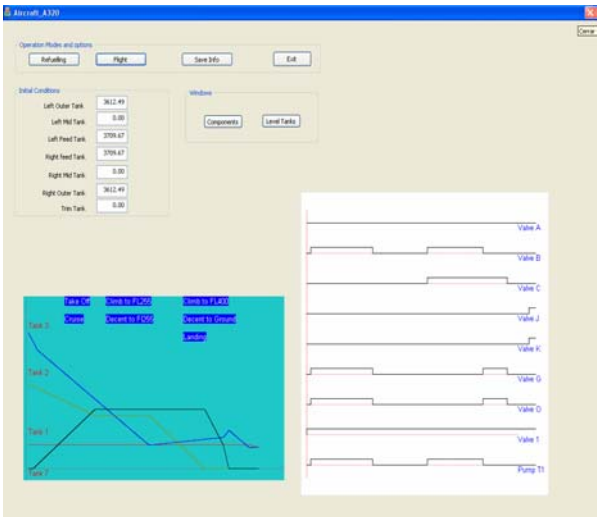
Figure 13 shows the main screen of the airplane simulation. It displays the tank levels during refueling operation, and during the different phases of a flight.

There are other screens that focus on specific aspects of the operations. For instance, figure 14 shows the evolution of fuel levels and on/off switching of the actuators during a flight. The panel at the bottom left displays the fuel levels in a manner that helps to understand the sequence of transfers to guarantee engine supply and a good CoG position.



L Outer	L Mid	L feed	R Feed	R Mid	R Outer
4000.0	14000.0	22938.0	22938.0	14000.0	4000.0
4000.0	14000.0	22913.0	22913.0	14000.0	4000.0
4000.0	14000.0	22888.0	22888.0	14000.0	4000.0
4000.0	14000.0	22863.0	22863.0	14000.0	4000.0
4000.0	14000.0	22838.0	22838.0	14000.0	4000.0
4000.0	14000.0	22813.0	22813.0	14000.0	4000.0
4000.0	14000.0	22788.0	22788.0	14000.0	4000.0
4000.0	14000.0	22763.0	22763.0	14000.0	4000.0
4000.0	14000.0	22738.0	22738.0	14000.0	4000.0

Fig. 13. Main screen of the airplane simulation



Left Outer Tank 3612.49
Left Mid Tank 0.00
Left Feed Tank 3709.67
Right feed Tank 3709.67
Right Mid Tank 0.00
Right Outer Tank 3612.49
Trim Tank 0.00

Fig. 14. Screen showing the events evolution along time during a flight.

4 Conclusion

A decomposed simulation system has been developed to support an European Research Project on a new CANbus-based avionics system, for fuel management in aircrafts.

An analysis of the aircraft fuel systems reveals two types of processes, and a structure of two levels. The physical plant, the level where the fuel flow is controlled, has slow processes. The electronic level, that includes a set of a microprocessors interconnected with a CANbus in the new avionic system, has fast processes.

To simulate these two interacting levels, two simulations were developed: one for the fast processes, the other for the slow processes. Each simulation runs in a different computer. The two computers are interconnected, using the same signals than in the real plant.

This decomposition of the simulation makes easy to study several aspects of the research: the behaviour of the CANbus, the correct operations of the fuel system, and the reactions when there are malfunctions.

Part of the development can be reused for the pilot computer, with similar screens.

Next future will be to include some “hardware in the loop features” to the simulation, using the signals between the two computers.

Acknowledgments: The authors thank to the European Community support, through the Research Project “Smartfuel”. Likewise, the authors thank the collaboration of the research partners

References:

[1] D.A. Lombardo, *Advanced Aircraft Systems*, McGraw-Hill, 1993

[2] I. Moir, A. Seabridge, *Aircraft Systems*, AIAA, 2001

[3] Y. Papadopoulos, *Safety Directed System Monitoring Using Safety Cases*, Doctoral Dissertation, University of York, 2000

[4] Y. Papadopoulos, Model-Based System Monitoring Using Statecharts and Fault Trees, *Reliability Eng. & System Safety*, Elsevier, Vol.81, 2003, pp. 325-341

[5] Jian-Guo Zhang, A. Pervez, A. B. Sharma, Avionics data Bus: An Overview, *IEEE Aerospace and Electronic Systems,* Vol.18, Num.2, February, 2003

[6] J.R. Jordan, *Serial Networked Instrumentation*, John Wiley & Sons, 1995

[7] L. Wolfhard, *CAN System Engineering*, Springer-Verlag, 1997

[8] F. Randy, *Understanding Smart Sensors*, Artech House, 2000

[9] N.V. Kirianaki, Y. Yurish, O. Shpak, V.P. Deynega., *Data Acquisition and Signal Processing for Smart Sensors*, John Wiley & Sons, 2002

[10] Myke Predko, *PC Ph.D.: Inside PC Interfacing,* McGraw-Hill, 2000.

[11] Jean J. Labrosse, *MicroC/OS-II, The Real Time Kernel*, CMP Books, 2002.