Experimental Development of the Fuel Management Distributed Control for Aircraft

J. M. GIRON-SIERRA, C. C. INSAURRALDE, M. A. SEMINARIO, J. F. JIMENEZ
Dep. Arquitectura de Computadoras y Automática
Universidad Complutense de Madrid
Av. Complutense S/N, 28040 Madrid
SPAIN
gironsi@dacya.ucm.es http://www.ucm.es/info/dacya

Abstract: A new generation of the fuel management avionic system is introduced. The idea comes from a European Community project. It implements a distributed control system using CAN bus as communication channel between components. The research proposal is to build smart components and to confirm the viability-reliability for real implementation. The paper describes the applied distributed control concepts, the experimental development and the results obtained in our laboratory prototype system.

Key-Words: Avionics, CAN Bus, Distributed Control, Fieldbus, Fuel Management System, Smart Devices.

1 Introduction

Present Fuel Management avionic systems have a central control, which consists of a point-to-point link between components of the fuel system and a central computer. The components are sensors and actuators. This communication scheme involves a lot of wiring and connectors, which increases the weight and causes many possible connection problems. Our proposal is try to replace this present scheme for another one that is simpler, through the use of the fieldbus as communication channel between components. A fielbus consists in several nodes connected to a bus, which has few wires. Like we will see later, we chosed the CAN bus to make this communication.

Our research is part of an European Community Project. Three universities and six companies are working together to obtain a final product. The main activity of the universities is give scientific support to the companies.

Much information was consulted to contribute to the experimental research development. The fieldbuses documentation was very important [1], and also the knowledge of the actual CAN bus applications, [2] and[3], were necessary to introduce us into the CAN-bus development world and its possible application to avionics.

This paper presents the experimental development of a new generation of fuel management control for helicopters and airplanes. The idea of Distributed Control is a main protagonist along the research development. The main problem to solve is how to devise this distributed control.

In the first part, the paper shows how the distributed control was designed; we present the idea of the automata, the elimination of the central computer, the notion of common code for each component and an example to understand the functional distribution concept. Then we present how the distributed control was implemented, what we wanted to test, and the building of smart components. Then, we explain the experimental development, with our laboratory simulators. Finally, we show the obtained results; we compare our testing prototype results with the real system, and give some conclusions of the work.

Aircraft fuel management systems have some interconnected tanks, were the fuel flows. Main operations of the fuel management systems are refueling, engine supply and aircraft balancing and/or trimming. The fuel transfers from tank to tank, or from/to external fuel supply or engines, are controlled with on/off devices such valves and pumps. From the control point of view this kind of systems is of hybrid nature, combining continuous time processes with discrete-event processes. The main aspects of the fuel system operations can be represented as sequences of states. Thus, it is natural to use finite state machines, automata, for the functions representation and control development.

A main concept to be presented in this paper is the decomposition of the general system operations into interacting local automata. Each system smart component works according with its local automaton.

2 Distributed Control

We got to distribute the control eliminating the central computer, which controlled centrally the previous fuel system. We shared out the control designing smart components with common code software; necessary for certification in avionics.

2.1 Fuel System Operations

There are two basic operations on the aircraft fuel systems. We will present shortly these operations to introduce the working environment. They are:

2.1.1 Pressure Refuelling

It consists in filling the fuel tanks through valves. When the tanks reach the desired level, the valves will have to be closed, and so, the operation finish.

2.1.2 Engine Supply

It consists in supplying fuel from the tanks to the aircraft engines. This operation needs pumps to push the fuel toward the engines. The pump will not start if the fuel level is not appropriate.

2.2 Formalization of Automata

The new fuel management system uses a set of smart components: smart valves, smart pumps, etc. Inside a smart component is a conventional valve, pump, whatever, and a microcontroller giving the smartness. The microcontroller itself is a node of the CANbus distributed system.

A simple analysis of the system functions, suggests the use of automata. The general work of the system can be described with an automaton: The command of operations can also be described by an automaton. And the behaviour of each smart component, according with the operation in course, obeys to a local automaton.

Following this control design scheme, we have three kinds of automata:

- ***Global Automaton:*** Fuel System.
- ***Specific Automaton:*** Fuel Operation Modes.
- ***Local Automaton:*** Component Nodes.

Figure 1 shows the global automaton. It is just the system making initial tasks and waiting for an operation to start. Figure 2 shows the specific automaton, with transitions from one to another system operation mode.

.

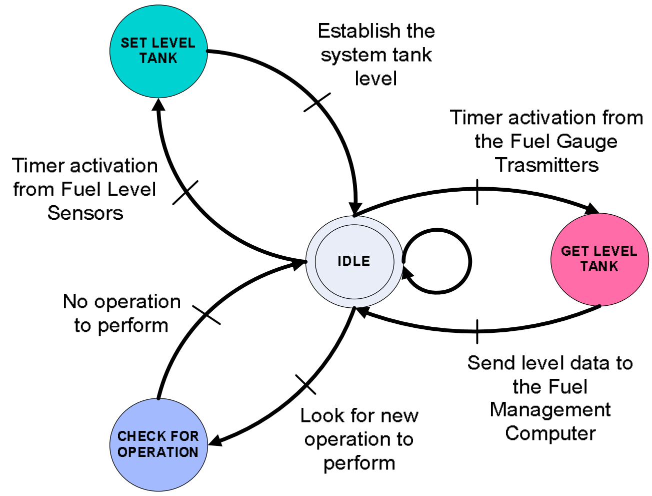

Fig. 1. Global automaton: Fuel system

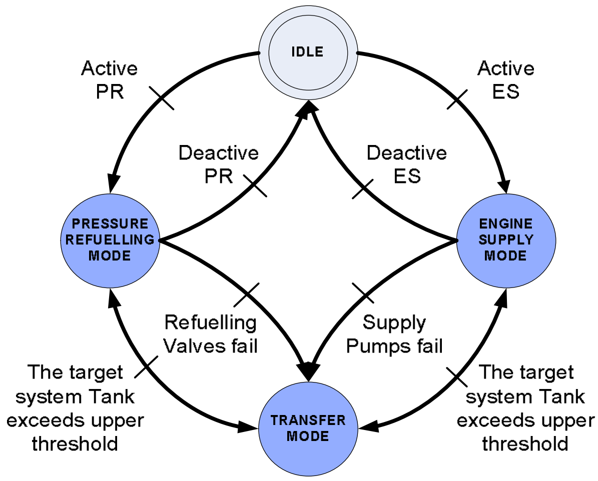

Fig. 2. Specific automaton: System operation modes.

Figure 3 shows an example of local automaton. It corresponds to the a refueling valve.

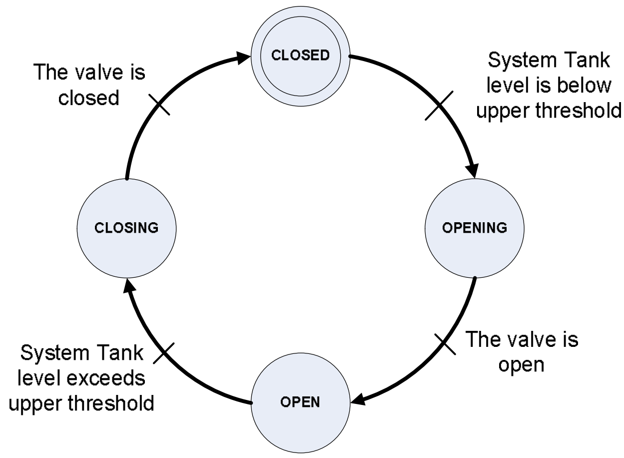

Fig. 3. Local automaton: Refuelling valve.

2.3 Local Automata

Local automata are the main idea of the distributed control. We must think it as several automata cooperating with a global automaton for the system. Every local automaton will take action according to the global system state. We could say that the system behavior emerges form cooperation of local automata.

The local automaton broadcast the information of its states to the other local automata of the system by a data packet; this concept is showed in figure 4. After a local automaton changes its state, all the rest of the nodes will be informed of this change through the communication channel. Of course, we defined a formated data packet by an array of bytes. In this array each smart component is assigned a place to inform about its state. Each component knows where to look in the packet for the pertinent information (to decide a local automaton transition), and where to put the information on its state change.

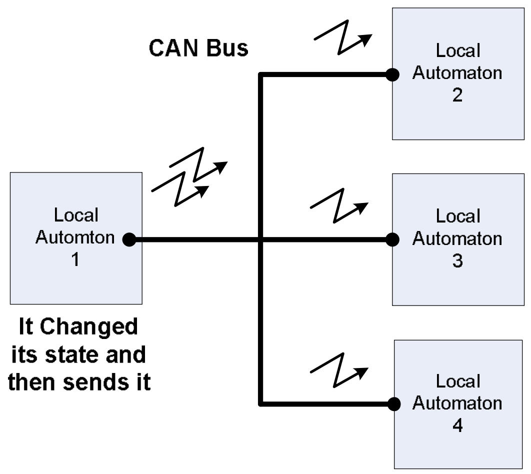

Fig. 4. Communication of the State components.

2.4 Example: Helicopter

For the case of the helicopter fuel system, we have six actuators; two refuelling valves, two supply pumps, one balance valve, and one balance pump. Also, there are two kinds of sensors that give the level tanks. This is how the system work:

- § The refuelling valve state diagram showed in figure 3 has four states; if we suppose that the valve initially is closed and the system is in pressure refuelling mode; the condition "System Tank level is below upper threshold" produces a transition to the next state called opening. It means the valve goes to open to allow the fuel fills the tanks. We defined this inter-middle state because there is a time of opening. When the valve is open physically, it takes the open state. Then, when the "system tank exceeds upper threshold", occurs a transition to the closing state, and finally when the valve is closed physically, it gets the initial state; closed.
- § The supply pump state diagram showed in figure 5 has two states; if we suppose that the valve initially is stopped and the system is in engine supply mode, it provokes a change of state to start the pump in direction A. When the mode operation is suspended, the pump will stop and takes the initial stopped state.
- § The balance valve state diagram showed in figure 6 has four states; it has the similar states than the refuelling valves but it changes the transition conditions.
- § The balance pump state diagram showed in figure 7 has three states; it has similar states than the supply pumps but it changes the transition conditions and the direction to pump the fuel according to the way of the transfer.

The transfer operation mode is automatic; it will be activated in direction A or B, according to the occurred failures in the components.

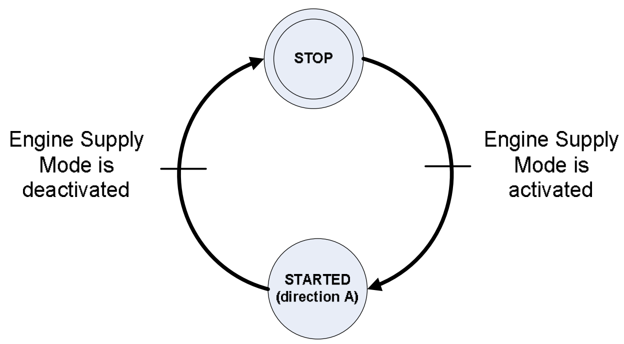

Fig. 5. Local Automaton: Supply Pumps.

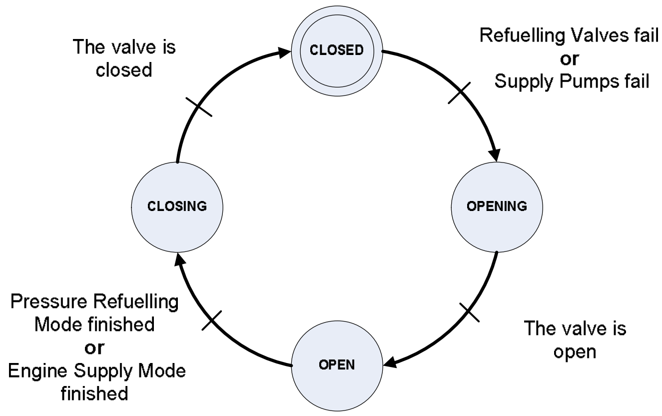

Fig. 6. Local Automaton: Balance Valve.

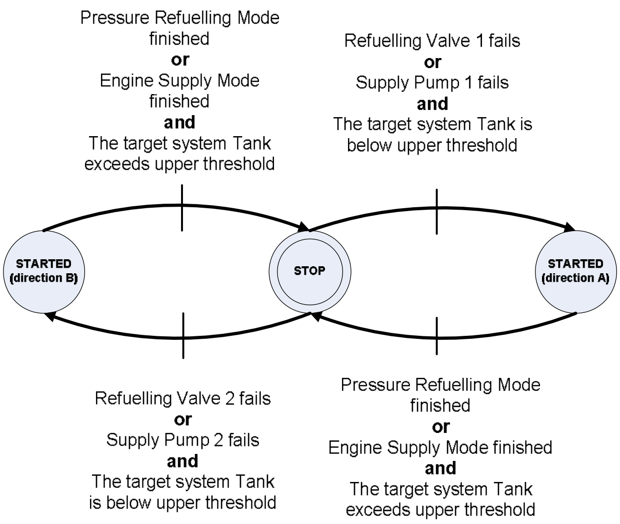

Fig. 7. Local Automaton: Balance Pump

2.5 Common Code

The idea is that all the smart components have the same software code. According to their identification numbers (ID's); detected by hardware, they will follow different processing flow inside the microcontroller.

Figure 8 shows a code switch done for the selection of the code part to be executed by each node-component, according to its ID.

An important objective of the Project is to have common software for all components. This is for an easier certification.

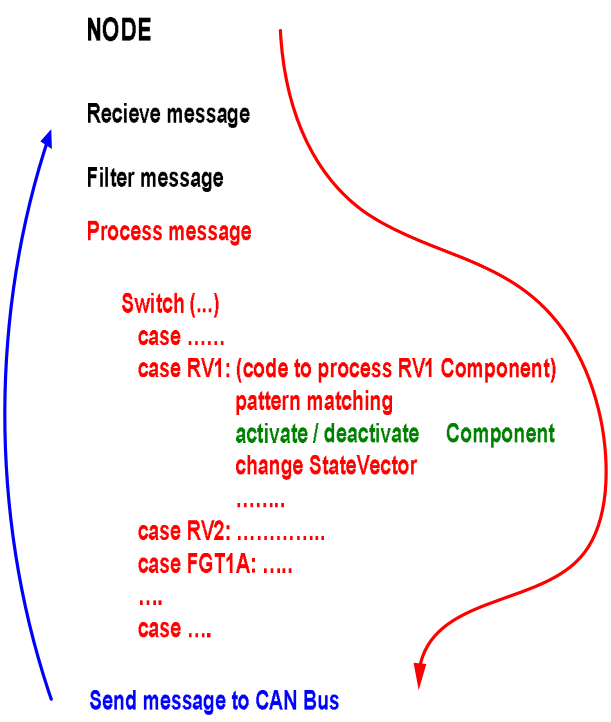

Fig.. 8. State machine processing.

3 Experimental Development

3.1 General Development Strategy

We made a laboratory simulator of the fuel system with conventional components. A microprocessor has been associated to each component. There are as many microcontrollers as components. In this way, the equivalent to the future smart components has been made. It represents a scaled model of the real fuel system.

The common code, to be put in every microcontroller, is developed using transportable C software.

CANbus has been selected as the fieldbus to be used by the distributed system. The bandwidth of the CANbus 1 megabit per second in distances shorter than forty meters.

With this entire working scheme, there are two important aspects to see. The first one is message collisions and the other one is the message overload on the CAN bus.

3.2 Fuel Systems Considered in the Research

We have considered two fuel systems. The helicopter and the airplane fuel systems. See figure 9 and figure 10 to see the physical distribution of the fuel tanks inside airborne vehicles.

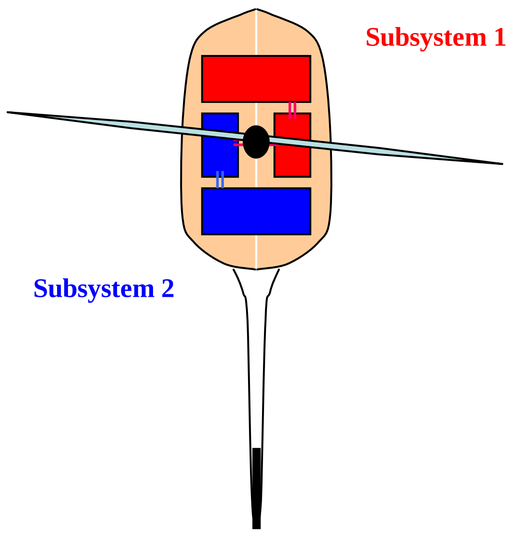

Fig. 9. Top-view schematic of the helicopter fuel system.

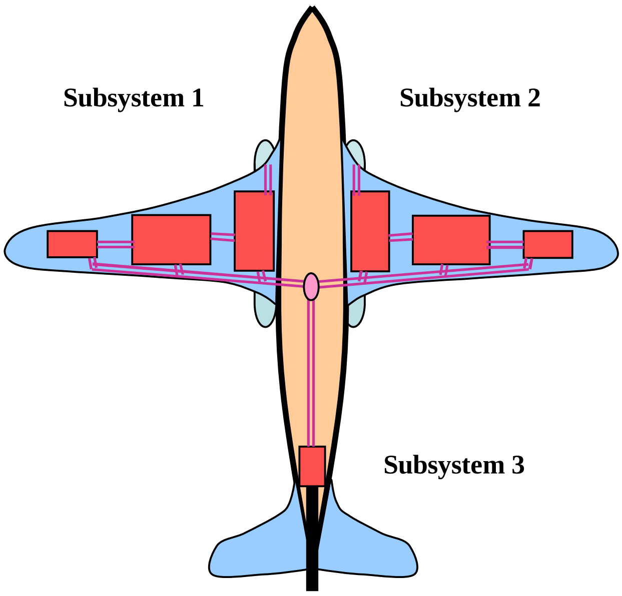

Fig. 10. Top-view schematic of the airplane fuel system.

The helicopter has two subsystems called subsystem 1 and subsystem 2. See figure 11. The same happens with the airplane, it has two wings, but with a third subsystem at the tail.

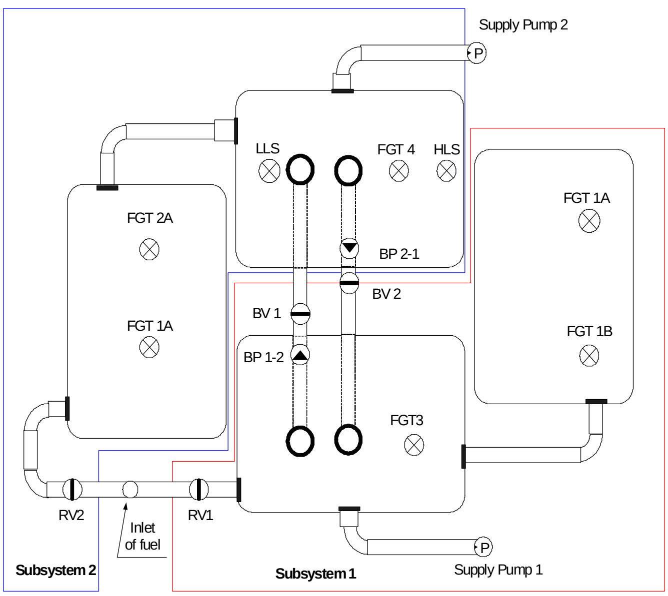

Fig. 11. Details of the helicopter fuel system

We can find some problems in the different operation modes.

For example, during the pressure refueling, one of refueling valves (RV) might fail; we have to solve this bad operation activating the transfer operation automatically to compensate this blocked fuel input. It will transfer fuel from a fuel subsystem to the other one through the Balance Valve (BV) and the Balance Pump (BP). The sensors are represented by FGT's and FLS's and for redundancy, some of them are repeated.

The direction to transfer has two ways called direction A and direction B respectively. The first one is to transfer fuel from subsystem 2 to subsystem 1. The other one is to transfer fuel from subsystem 1 to subsystem 2.

3.3 Node Architecture

Figure 12 shows the internal architecture of a smart component., and the layers considered by the software development. There are three layers, the component layer corresponds to the physical component (valve, pump, sensor), the software layer is the common code, which runs on the microcontrollers, and the hardware layer corresponds to the electronic circuit (CANbus enabled microcontroller and interfaces) associated to the component.

.

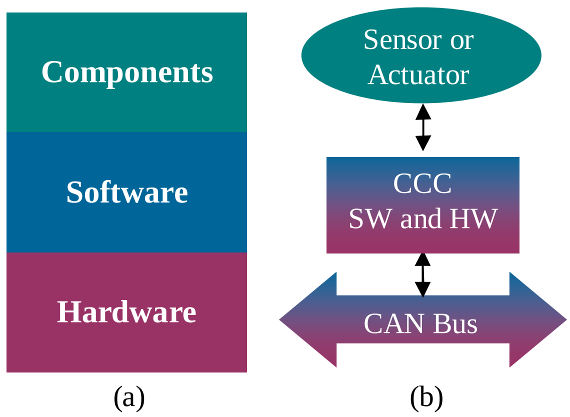

Fig. 12. (a) Architecture layers. (b) Block diagram of a smart component.

3.4 Fuel System Blocks Architecture

Figure 13 shows how the smart components, the CAN Modules, interact with the fuel system physical plant (formed by tanks and pipes). This is the fundamental architecture of the laboratory simulators we built for the helicopter and the airplane.

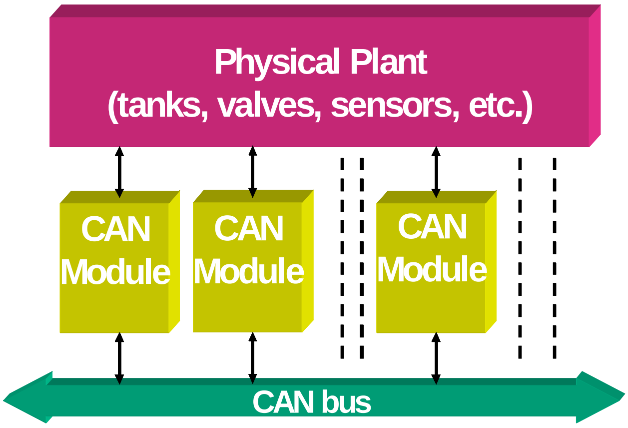

Fig. 13. Block diagram of the laboratory system.

3.5 Fuel System Prototypes

Figure 14 shows a photograph of the helicopter laboratory system and figure 15 shows a photograph of the airplane laboratory system.

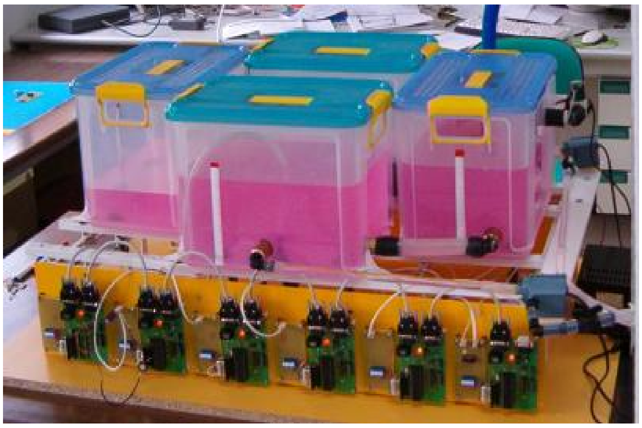

Fig. 14: *Helicopter laboratory system.*

Fig. 15: *Airplane laboratory system.*

4 Experimental Results

The laboratory simulators have been used for testing the software development. After some tests and modifications, an operational version of the common code has been obtained. This common code is put in every microcontroller of the simulator (as many as components). In particular, the following experimental results were obtained:

- We saw that our fuel prototype system follows the sequence of the scheduled control.
- The time constants of the laboratory system are faster than the time constants in the real system. So the laboratory conditions for sequences, for instance in terms of CANbus overload, are more difficult than in reality. A good behaviour in the laboratory means a good, better, behaviour in reality.
- We monitored the system doing a trace of CANbus data; we could see no collision of messages and no overload of messages on the bus. The trace showed us the packet of data from each smart component when it was sent.

5 Conclusion

A new distributed system, using smart components, has been developed. It is the target of an European Research Project. For certification purposes, all smart components have the same common code

In this paper the main ideas to obtain the control distribution with smart components having the same internal code, have been presented. It is a matter of cooperating local automata.

The ideas for distribution have been implemented and tested in laboratory simulators we made.

In the future, more powerful functionality is intended to be embedded into the smart components, so the universe of local automata will increase, requiring more developments and laboratory testing.

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] J. R. Jordan, *Serial Networked Field Instrumentation*, Wiley, 1995.

[2] Wolfhard Lawrenz, *CAN System Engineering*, Springer, 1997.

[3] *CAN in Automation* (www.can-cia.com).