1

Design and Development of a Fly-by-Wireless UAV Platform

Paulo Carvalhal, Cristina Santos, Manuel Ferreira, Luís Silva and José Afonso
University of Minho
Portugal

1. Introduction

The development of unmanned aerial vehicles (UAVs) has become an active area of research in recent years, and very interesting devices have been developed. UAVs are important instruments for numerous applications, such as forest surveillance and fire detection, coastal and economic exclusive zone surveillance, detection of watershed pollution and military missions.

The work described in this chapter is part of a larger project, named AIVA, which involves the design and development of an aerial platform, as well as the instrumentation, communications, flight control and artificial vision systems, in order to provide autonomous takeoff, flight mission and landing maneuvers. The focus of the chapter is on one of the main innovative aspects of the project: the onboard wireless distributed data acquisition and control system. Traditionally, UAVs present an architecture consisting of one centralized and complex unit, with one or more CPUs, to which the instrumentation devices are connected by wires. At the same time, they have bulky mechanical connections. In the approach presented here, dubbed "fly-by-wireless", the traditional monolithic processing unit is replaced by several less complex units (wireless nodes), spread out over the aircraft. In that way, the nodes are placed near the sensors and controlled surfaces, creating a network of nodes with the capacity of data acquisition, processing and actuation.

This proposed fly-by-wireless platform provides several advantages over conventional systems, such as higher flexibility and modularity, as well as easier installation procedures, due to the elimination of connecting cables. However, it also introduces several challenges. The wireless network that supports the onboard distributed data acquisition and control system needs to satisfy demanding requirements in terms of quality of service (QoS), such as sustainable throughput, bounded delay and reliable packet delivery. At the same time, it is necessary to guarantee that the power consumption of the battery powered wireless nodes is small, in order to increase the autonomy of the system. Currently there are many different wireless network technologies available in the market. Section 2 presents an overview of the most relevant technologies and discusses their suitability to meet the above requirements. Based on this analysis, we chose the Bluetooth wireless network technology as the basis for the design and development of a prototype of the fly-by-wireless system. The system was implemented using commercial off-the-shelf components, in order to provide a good tradeoff between development costs, reliability and performance. Some other objectives were also pursued in the development of the system, namely the design of a framework where communication between nodes is effective and independent of the technology adopted, the development of a design approach to model the embedded system and the development of an application oriented operating system with a modular structure.

The following sections are organized as follows: section 2 presents an overview of available wireless network technologies, taking into account the requirements of the application; section 3 presents the global electronics architecture of the UAV platform, while section 4 describes the developed onboard wireless system; section 5 presents experimental performance results obtained with this system, and section 6 presents the conclusions and addresses the future work.

2. Wireless Network Technologies

Most wireless networks technologies available nowadays can be subdivided in a few categories: satellite networks, mobile cellular networks, broadband wireless access, wireless local networks (WLAN) and wireless personal networks (WPAN). The former three differ substantially from the latter two. One difference is that the network infrastructure does not belong to the user, but to the network operator, which charges the user for the services provided. Other difference is that they provide coverage over a large area. On the other hand, WLAN and WPAN are short range technologies in which all the communications equipment usually belongs to the user. These characteristics are more adequate for the intended application, so the remainder of this section will focus on wireless network technologies belonging to these two categories.

The most widespread type of WLAN nowadays is the IEEE 802.11 (IEEE, 2007), also known as WiFi. These networks are available on multiple physical options and operating frequency bands. However, all these versions use the same MAC (Medium Access Control) protocol; a contention based CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) mechanism known as DCF (Distributed Control Function). Given its statistical nature, this protocol is not adequate to provide the QoS guarantees required by the onboard wireless data acquisition and control system due to the probability of collisions.

In order to support real-time traffic, the 802.11 standard defines an alternative MAC protocol know as PCF (Point Coordination Function), based on a polling scheme, which is capable of providing QoS guarantees. However, unlike the DCF protocol, the implementation of PCF is not mandatory, and the availability of products that support it is scarce. More recently, a newer standard, the IEEE 802.11e (IEEE, 2007), designed to improve the efficiency and QoS support of 802.11 networks was released, but its availability on the market is also low.

Concurrently to the development of the 802.11, the European Telecommunications Standards Institute (ETSI) has developed another WLAN standard: HIPERLAN/2 (ETSI, 2002). HIPERLAN/2 networks are designed to operate at the 5 GHz band using OFDM (Orthogonal Frequency Division Modulation). Its physical layer is similar to the one used by the IEEE 802.11a due to agreements made by the two standard bodies. On the other hand, the MAC protocols used by these networks are radically different. HIPERLAN/2 uses a demand based dynamic TDMA (Time Division Multiple Access) protocol, which is able to provide extensive support of QoS to multiple types of traffic, including those generated by data acquisition and control systems (Afonso & Neves, 2005). However, the 802.11 standard won the battle for the wireless LAN market and as such no available HIPERLAN/2 products are known at the moment.

Due to its design characteristics, 802.11 and HIPERLAN/2 modules present relatively high power consumption. Although these networks can be suitable to interconnect devices like computers, there is an enormous potential market to provide wireless communication capabilities to smaller and cheaper devices running on batteries without the need of frequent recharging. Such devices include computer peripherals, biomedical monitoring appliances, surveillance units and many other sensing and actuation devices. To provide communication capabilities to such devices, various low cost short range networks, known collectively by the term wireless personal area network (WPAN), are being developed.

At the IEEE, the task of standardization of WPAN networks is under the scope of the IEEE 802.15 group. One of these standards, the IEEE 802.15.4 (IEEE, 2006) defines the physical and MAC layer of ZigBee (ZigBee, 2006), which aims to provide low power and low bit rate WPANs with the main purpose of enabling wireless sensor network applications. At the physical layer, the IEEE 802.15.4 relies on direct sequence spread spectrum (DSSS) to enhance the robustness against interference, and provides gross data rates of 20/40 kbps, at the 868/915 band, and 250 kbps, at the 2.4 GHz band. As in 802.11 networks, the basic ZigBee MAC protocol is a contention based CSMA/CA mechanism. A complementary mechanism defined in the 802.15.4 standard, the guaranteed time slot (GTS), enables the provision of some QoS guarantees to real-time traffic.

Bluetooth (Bluetooth, 2003) is another WPAN technology. It operates in the 2.4 GHz band using frequency hopping spread spectrum (FSSS) and provides a gross data rate of 1 Mbps. Bluetooth operates using a star topology, called piconet, formed by one master and up to seven active slaves. Transmissions can achieve a range of 10 or 100 m, depending of the class of the device. At the MAC layer, the Bluetooth devices uses a polling based protocol that provides support for both real-time and asynchronous traffic.

Bluetooth provides better overall characteristics than the other networks discussed here for the desired application. It drains much less power than 802.11 and HIPERLAN/2, uses a MAC protocol that provides support for real-time traffic, and provides a higher gross data rate than ZigBee. Bluetooth spread spectrum covers a bandwidth of 79 MHz while ZigBee operates in a band of less than 5 MHz, what makes the former more robust against interference. Moreover, Bluetooth provides an adaptive frequency hopping mechanism that avoids frequency bands affected by interference. Given these characteristics and the availability of the technology at the time of development, Bluetooth was chosen as the supporting wireless network technology for the development of the prototype of the system described in the following sections.

3. Global Electronics Architecture

The global view of architectural model of the onboard computing and communication system of the AIVA fly-by-wireless UAV platform is presented in Figure 1. It is a multitasking/multiprocessor based system connected by an asynchronous local bus that allows for speed adaptation of different tasks/processors. The system architecture supports one processing unit for a Bluetooth piconet master node, one flight controller unit, one data logger and earth link, and one embedded vision system (EVS). In each of these nodes many critical processes are permanently running.

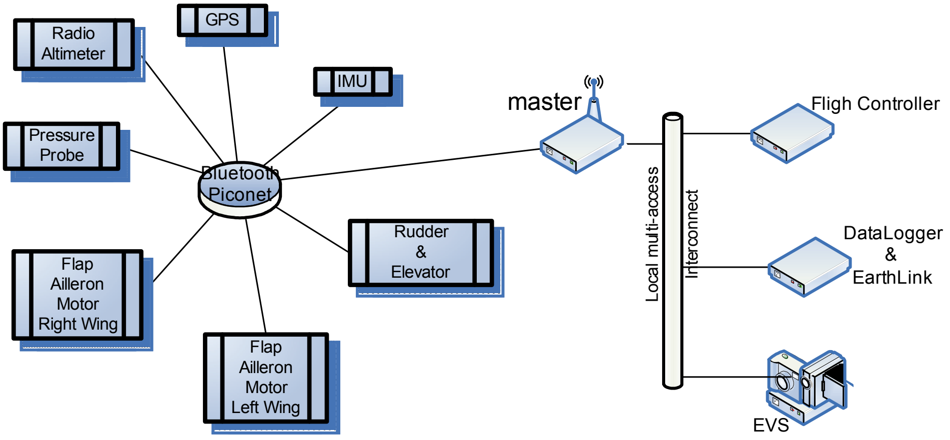

Figure 1. Global electronics architecture of the AIVA UAV platform

This architecture allows an easy way to introduce or remove processing units from the platform. For instance new sensors or new vision units can be included. In the first case a new module must be connected to the Bluetooth piconet, and in the second case the new module is connected to the local multi-access bus.

4. Onboard Wireless System

The AIVA UAV platform implements an onboard wireless distributed data acquisition and control system based on Bluetooth (BT) wireless network technology, represented by the Bluetooth piconet of Figure 1. The general architecture of a wireless node is presented on Figure 2. Each node is composed by a commercial off-the-shelf Bluetooth module that contains the radio electronics, a microcontroller that runs the code that controls the behavior of the node, and a local bus that provides interfacing between the node components, as well as specific sensors and/or actuators according to the purpose of the node.

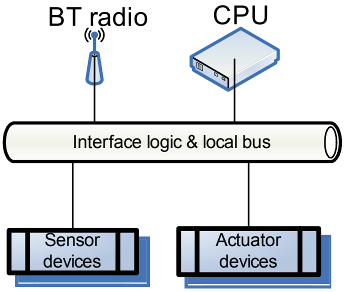

Figure 2. Architecture of a Bluetooth wireless node

4.1 Physical Architecture

The physical part of the platform is built around a low power Texas Instruments MSP430 microcontroller, a Von-Neumann 16 bit RISC architecture with mixed program, data and

I/O in a 64Kbytes address space. Besides its low power profile, which uses about 280 µA when operating at 1 MHz @ 2.2 Vdc, MSP430 offers some interesting features, like single cycle register operations, direct memory-to-memory transfers and a CPU independent hardware multiplication unit. From the flexibility perspective, a flexible I/O structure capable of independently dealing with different I/O bits, in terms of data direction, interrupt programming, and edge triggering selection; two USARTs supporting SPI or UART protocols; an onboard 12 bit SAR ADC with 200 kHz rate; and PWM capable timers, are all relevant features.

The Bluetooth modules chosen for the implementation of the wireless nodes are OEM serial adapter devices manufactured by connectBlue. The master node uses an OEMSPA33i module and the slave nodes use OEMSPA13i modules (connectBlue, 2003). These modules include integrated antennas; nevertheless, we plan to replace them with modules with external antennas in future versions of the platform, to be able to shield the modules in order to increase the reliability of the system against electromagnetic interference.

While the module used on the master (OEMSPA33i) allows up to seven simultaneous connections, the module used on the slaves (OEMSPA13i) has a limitation of only three simultaneous connections. However, this limitation does not represent a constraint to the system because the slaves only need to establish one connection (to the master).

The connectBlue modules implement a virtual machine (VM) that enables the provision of a serial interface abstraction to the microcontroller, so Bluetooth stack details can be ignored and focus can be directed to the application. The manufacturer's virtual machine implements a wireless multidrop access scheme where the master receives all frames sent by the slaves and all slaves can listen to the frames sent by the master, in a point-to-multipoint topology.

The AIVA onboard wireless system is composed by one Bluetooth piconet containing seven nodes: one master (MM - Master Module) and six slaves (SAM - Sensing & Actuation Modules). The nodes are spread over the aircraft structure, as shown in Figure 3. The master node (MM) is placed at the fuselage body, and acts as the network and flight controller, onboard data logger, and communications controller for the link with the ground station. On each wing, there is a SAM node for an electrical propulsion motor and for control surfaces (ailerons and flaps). These wing nodes are responsible for motor speed control and operating temperature monitoring, as well as control surfaces actuation and position feedback.

In the fuselage body, there are other two SAM nodes, one for a GPS module and other for an inertial measurement unit (IMU), which provide information assessment for navigational purposes. At the tail, there is another SAM node for elevator and rudder control, and position feedback. Finally, at the nose there is a SAM node connected to a nose probe consisting of a proprietary design based on six independent pressure sensors that give valuable digital information for flight control. This node also contains an ultrasonic probe that provides information for support of the automatic take-off and landing system. Figure 4 displays the physical layout of the nose node. The Bluetooth module is in the lower corner, the microcontroller is on the left hand side and the sensor hardware on the right hand side of the board.

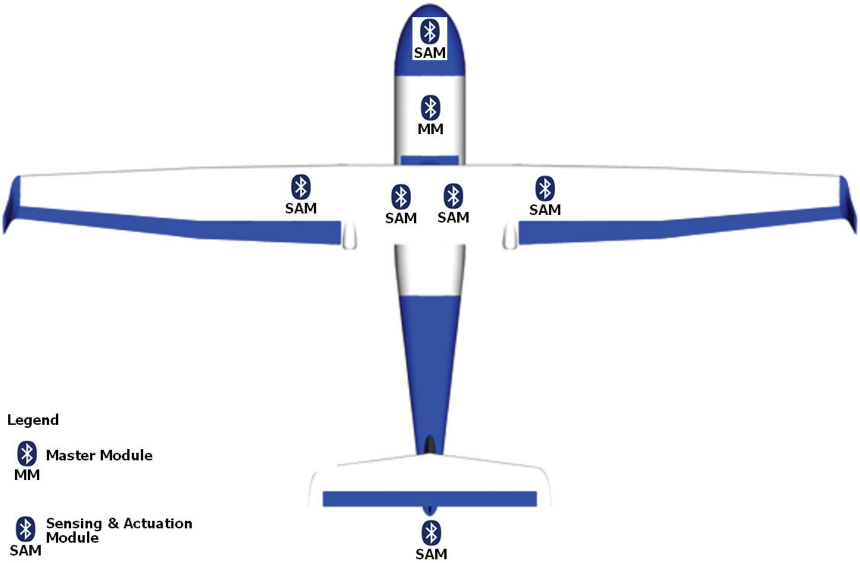

Figure 3. Node distribution on the aircraft structure

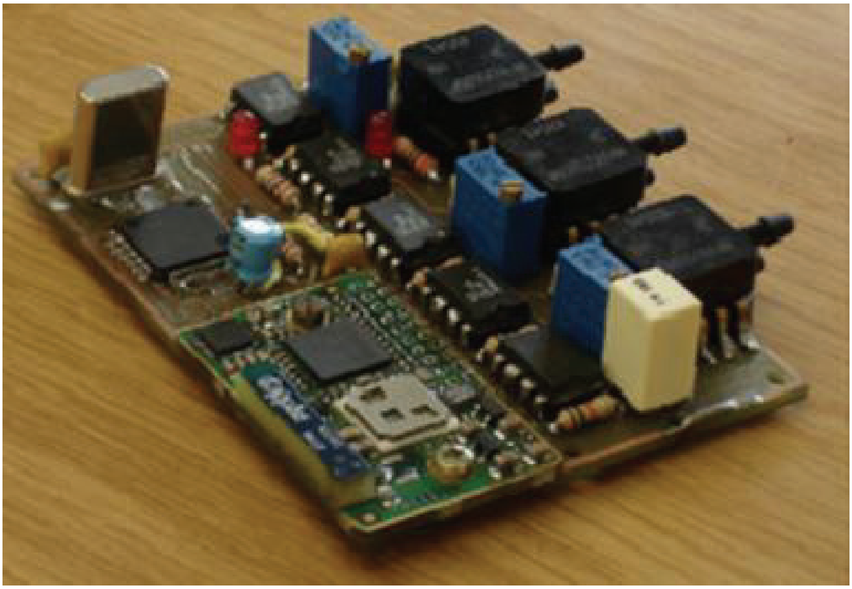

Figure 4. Physical layout of the nose node

4.2 Logical Architecture

The logical architecture of the developed system is a two layered state machine implementation, composed by a transport layer and an application layer. The transport layer provides a packet delivery service, under control of master node, capable of transparent delivery of data packets across the network.

The transport layer is application independent, and interfaces with the top level application layer by means of a data space for buffering and a set of signaling control bits that allow total synchronization between the two layers. The hierarchy and signaling between the two layers is represented in Figure 5.

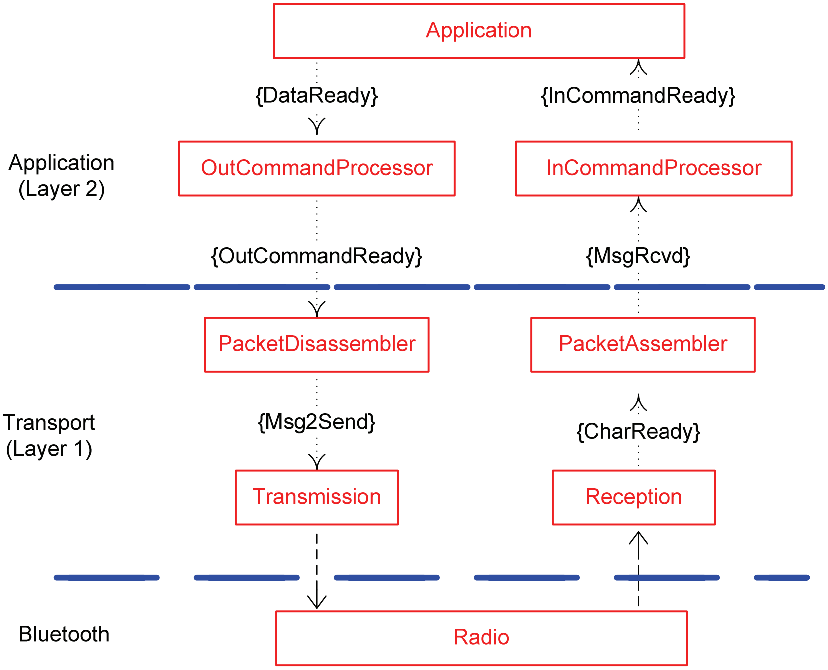

Figure 5. Hierarchy and signaling at the logical level of the platform

The asynchronous reception process delivers characters to upper processes. Analyzing the hierarchy from the lower level to the upper level, CharReady condition goes TRUE every time a new character arrives to the interface. The next process in the hierarchy is PacketAssembler, a state machine that performs packet re-assembly, reconstructing the original packet from a group of segments, and delivers packets for the next process in the hierarchy. When MsgRcvd (message received) goes TRUE, a new message is ready for processing. Thus, for incoming data, the model at layer 1 receives characters and delivers ready-to-process messages to the application layer. When the application layer understands that the message is ready to process, a command processor for incoming messages is activated in order to decode the embedded command and semantics contained in the message, to eventually execute some action, and to pass relevant information for the final application.

For outgoing data, the resident application eventually makes available some data to transmit to the master, signaling this event with a DataReady signal. This causes the output command processor to execute its cycle, preparing one message to be sent. When the message is ready, OutCommandReady goes TRUE, signaling to the lower layer that there is a message to send to the network. At this phase, frame segmentation starts (if needed) by means of a state machine for packet disassembly. This state machine breaks the original message in smaller segments prepared to be serialized. Each time a segment is ready to be sent, Msg2Send goes TRUE and serialization is triggered. So, for outgoing data, the transport layer receives messages from application layer, and sends segments to the radio module in order to be sent over the wireless medium.

Layer 2 is application dependent, and has no knowledge of the lower layer internal mechanisms, just the services and interfaces available from it. That means that its logical architecture can be used in other applications. For the fly-by-wireless application, its main goal is to replicate a system table among all network nodes, at the maximum possible frequency. This system table maintains all critical system values, describing the several sensors and actuators, status parameters, and loop feedback information. Each network node is mapped to a table's section, where all related variables from sensing, actuators and metering are located. This layer is responsible for cyclic refreshing the respective table contents, based on local status, and also for cyclic actuation according to data sent from the master node (flight controller orders). This way, the whole system is viewed as a resident two-dimensional array located at master, with different partial copies distributed and synchronized among the slave nodes.

4.3 Other Design Issues

All the Bluetooth modules in the developed platform are configured in non discoverable mode, which contributes to the security of the system. The node discovery process of Bluetooth is a slow process, in the order of seconds, however it is not a problem since the master stores the addresses of all slave nodes that should participate in the piconet, so this process is avoided. The piconet formation is performed on the ground before the takeoff procedure, so the associated delay does not constitute a problem as well.

The use of Bluetooth technology limits the piconet operation to a maximum of seven active slaves; however, this limitation is not of major concern on the developed system, since only six slaves are used, and could only impose some restrains if node number should be raised.

The number of slaves in the network could be increased by interconnecting a number of piconets to form a scatternet. That way, a device participating in two piconets could relay traffic between both piconets. However, this architecture would probably have a negative impact in the performance of the network, making it more difficult to provide QoS guarantees to the application. Moreover, currently there are very few actual implementations of scatternets available.

Given that free space propagation loss is proportional to the square of the distance, it is not expected that the onboard wireless network will either suffer or induce interference on other networks operating in the same frequency band, such as the widely deployed WiFi networks, since the former operates in the sky most of time, while the later are normally based on the ground.

5. Experimental Results

The performance of the developed wireless system was evaluated in laboratory. The experimental setup used to achieve the results presented in this section is composed by 6 slaves sending data periodically to the master (uplink direction) at the same predefined sampling rate. Each sampling packet has a length of 15 octets, which is the maximum packet length supported by the transport layer due to a limitation imposed by the virtual machine used by the Bluetooth module.

Figure 6 presents the aggregated uplink throughput that reaches the master node as a function of the sampling rate used by the 6 slaves. Since Bluetooth uses a contention-free MAC protocol, the uplink transmissions are not affected by collisions, so the network throughput increases linearly with the offered load until the point it reaches saturation, which in this scenario corresponds to the situation where the slaves transmit data at sampling rates higher than 200 Hz. As this figure shows, the maximum throughput available to the application is about 160 kbps, which is significantly lower than the gross data rate provided by Bluetooth (1 Mbps). This difference can be explained by the overhead introduced by the Bluetooth protocol and the virtual machine, including the gap between the packets, the FEC (Forward Error Correction) and ARQ (Automatic Repeat reQuest) mechanisms, the packet headers, as well as the overhead introduced by control packets such as the POLL packet, that is sent by the master to grant permission to slaves to transmit.

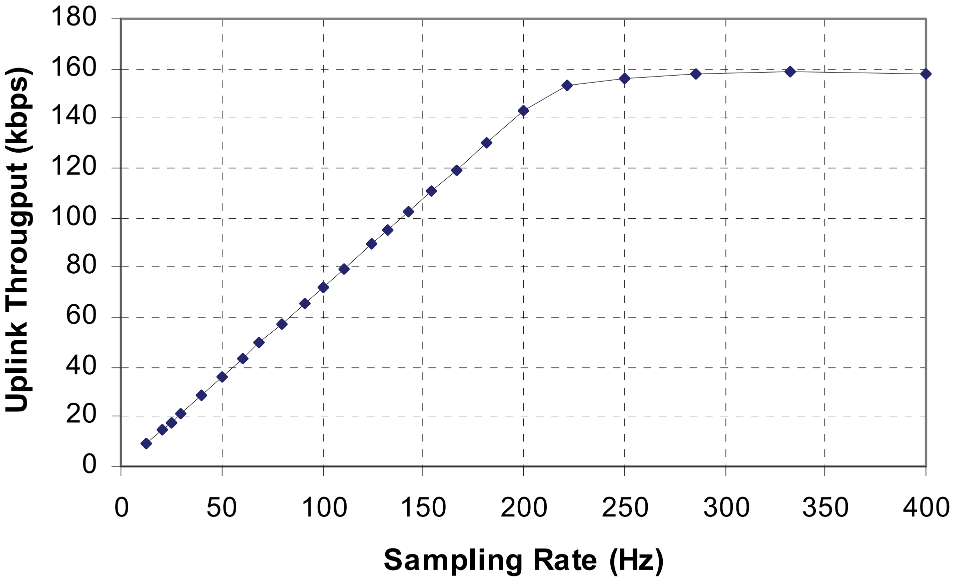

Figure 6. Uplink throughput as a function of the sampling rate

Figure 7 presents the packet loss ratio (PLR) averaged over the 6 slaves as a function of the sampling rate. As the figure shows, the PLR is limited to less than 0.5 % in the region where the network is not congested, but increases rapidly after the saturation point. The flight control application should be able to tolerate such small losses; otherwise a change in the supporting wireless technology should be made in the attempt to obtain higher link reliability.

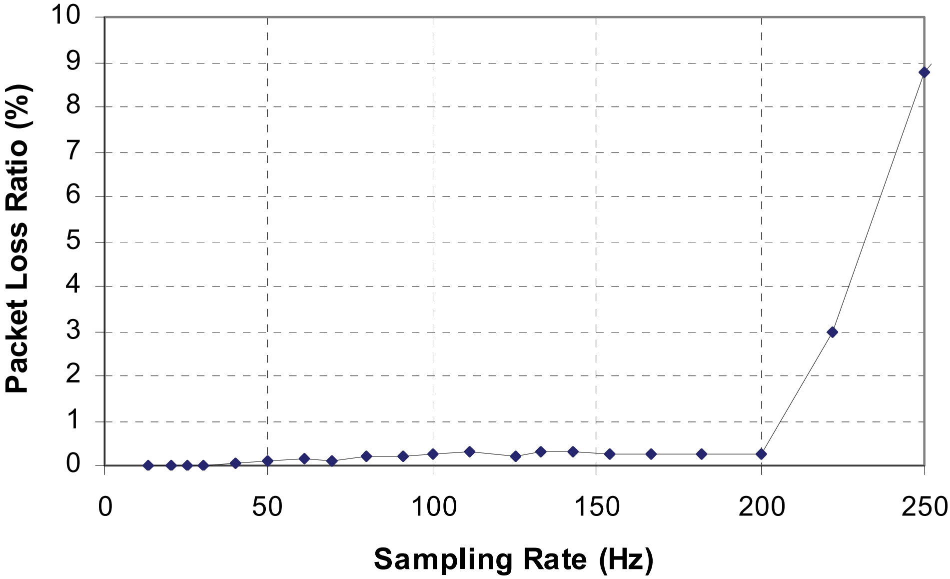

Figure 7. Packet loss ratio as a function of the sampling rate

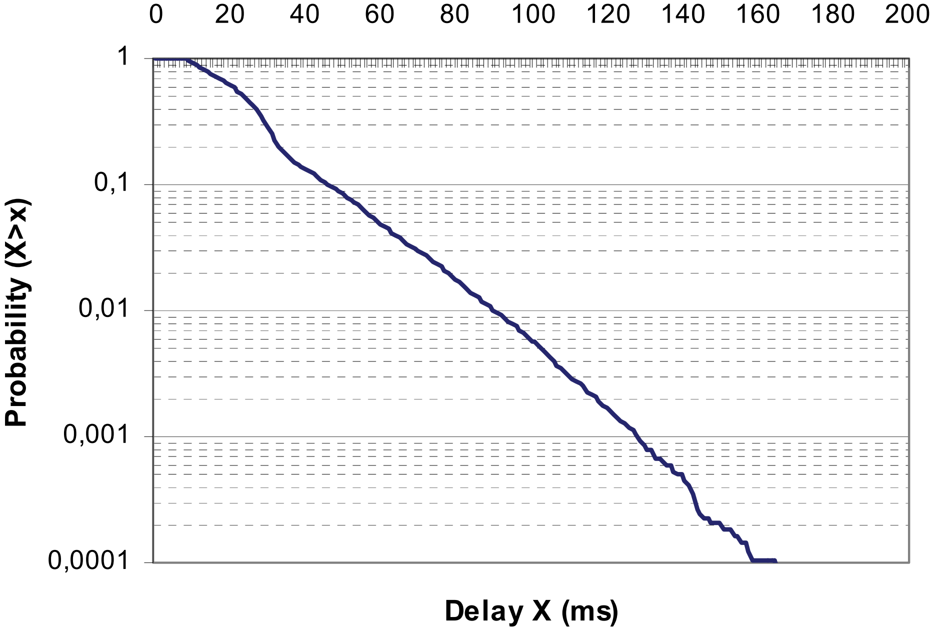

Figure 8. Complementary cumulative distribution of the delay

Concerning to the delay experienced by the packets as they travel from the slaves to the master, results showed that the delay is not adversely affected by the rise in the offered load, as long as the network operates below the saturation point. For sampling rates up to 200 Hz, the registered average delay was 27 ms and the standard deviation was 16 ms.

Figure 8 presents the complementary cumulative distribution (P{X>x}) for the random variable X, which represents the delay for all the samples collected using sampling rates in the range from 0 to 200 Hz. With this chart, it is possible to see the probability that the delay exceeds a given delay bound, which is an important metric for real-time applications such as the one considered in this chapter. The chart shows, for instance, that less than 1 % of the sample packets suffer a delay higher than 90 ms, while less than 0.1 % of the packets suffer a delay higher than 120 ms.

Experimental tests were also made with a varying number of slaves in the piconet (from 1 to 6), both in the uplink and downlink direction. The average delay measured in the downlink direction (from the master to the slaves) was slightly higher than the one registered in the uplink direction, but below 40 ms, for the measurements made with up to 4 slaves. However, the average master-to-slave delay with 5 slaves in the network ascended to 600 ms, while with 6 slaves the performance was even worse, with the average delay reaching 1000 ms.

6. Conclusion

This chapter presented the design and development of a fly-by-wireless UAV platform built on top of Bluetooth wireless technology. The developed onboard wireless system is composed by one master node, connected to the flight controller and six slave nodes spread along the aircraft structure and connected to several sensors and actuators.

In order to assess the suitability of the developed system, several performance evaluation tests were carried out. The experimental results showed that, for the slave-to-master direction, the system prototype is able to support a sampling rate of up to 200 Hz for each of the 6 slaves simultaneously without significant performance degradation in terms of throughput, loss or delay. On the other hand, although the master-to-slave delay with 1 to 4 slaves in the network is low, its value increases significantly with 5 and 6 slaves, which is unacceptable given the real-time requirements of the desired application. This problem is caused by implementation issues related to the proprietary embedded virtual machine provided by the manufacturer of the Bluetooth module that is used in the master node of the prototype.

The approach of relying on the virtual machine provided by the manufacturer, which hides the Bluetooth protocol stack functionality, allowed the development focus to be directed to the application, reducing the development costs. The disadvantage, however, is the lack of control of the behavior of the system at the Bluetooth stack level, which impedes the optimization of the performance of the system at this level and the correction of problems such as the verified with the master-to-slave delay. The solution to the detected problem can pass either by the replacement of the Bluetooth module by a newer version (already available) from the same manufacturer or by the direct interaction with the Bluetooth stack, with the bypass of the virtual machine.

Despite the limitations of the current prototype, the overall results provided by the experimental tests are satisfactory. Nevertheless, further tests are needed in order to evaluate the behavior of the system under more harsh interference conditions, as well as in a real scenario onboard the aircraft.

7. References

Afonso, J. A. & Neves, J. E. (2005), Fast Retransmission of Real-Time Traffic in HIPERLAN/2 Systems, *Proceedings of Advanced Industrial Conference on Telecommunications (AICT2005)*, pp. 34-38, ISBN 0-7695-2388-9, Lisbon, Portugal, July 2005, IEEE Computer Society.

Bluetooth SIG (2003), Specification of the Bluetooth system. Available at http://www.bluetooth.org/.

connectBlue (2003), Serial Port Adapter – 2nd Generation, *User Manual*. Available at http://www.connectblue.com/.

ETSI TR 101 683 V1.1.1 (2000), Broadband Radio Access Networks (BRAN)—HIPERLAN Type 2—Data Link Control (DLC) Layer—Part 1: Basic Data Transport Functions.

IEEE Std 802.11 (2007), IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

IEEE Std 802.15.4 (2006), IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).

ZigBee Standards Organization (2006), ZigBee Specification. Available at http://www.zigbee.org/.