AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit
20 - 23 September 2004, Chicago, Illinois

AIAA 2004-6556

Small Adaptive Flight Control Systems for UAVs using FPGA/DSP Technology

Henrik B. Christophersen*, Wayne J. Pickell*, Adrian A. Koller†, Suresh K. Kannan†, and Eric N. Johnson‡
Georgia Institute of Technology, Atlanta GA 30332-0150

Future small UAVs will require enhanced capabilities like seeing and avoiding obstacles, tolerating unpredicted flight conditions, interfacing with payload sensors, tracking moving targets, and cooperating with other manned and unmanned systems. Cross-platform commonality to simplify system integration and training of personnel is also desired. A small guidance, navigation, and control system has been developed and tested. It employs Field Programmable Gate Array (FPGA) and Digital Signal Processor (DSP) technology to satisfy the requirements for more advanced vehicle behavior in a small package. Having these two processors in the system enables custom vehicle interfacing and fast sequential processing of high-level control algorithms. This paper focuses first on the design aspects of the hardware and the low-level software. Discussion of flight test experience with the system controlling both an unmanned helicopter and an 11-inch ducted fan follow.

1. Introduction

Many small UAV guidance, navigation, and control systems today have limited processing power due to stringent size, weight, and power constraints. However, operators want UAVs with enhanced capabilities like seeing and avoiding obstacles, tolerating unpredicted flight conditions, interfacing with payload sensors, tracking moving targets, and cooperating with other manned and unmanned systems. Cross-platform commonality to simplify system integration and training of personnel is also desired. Operators demand high-end performance for larger platforms and ultimately would like to see the same capabilities in smaller vehicles.

Figure 1 A credit card-sized board designed with the processing power and flexibility required for deployment in highly-capable small UAVs.

A small integrated guidance, navigation, and control system, referred to as the Flight Control System 20 (FCS20), has recently been developed, Figure 1. The miniature computer employs Field Programmable Gate Array (FPGA) and Digital Signal Processor (DSP) technology to satisfy the requirements for more advanced vehicle

* Research Engineer
† Graduate Research Assistant
‡ Lockheed Martin Assistant Professor of Avionics Integration, Member AIAA

behavior in a small package. Having these two processors in the system enables custom vehicle interfacing and fast sequential processing of high-level control algorithms. Efficient task sharing and communication between the DSP and the FPGA is essential to reach the full potential of the system.

This paper focuses first on the design aspects of the FCS20 hardware and the low-level software. The system architecture, the superimposed real time operating system (RTOS), inter-processor communication, navigation sensors and hardware integration challenges are discussed. Secondly, the neural network based, adaptive flight control algorithms that are able run on the FCS20 are summarized. Emphasis is placed on platform independence as well as runtime reconfigurability of this high-level software. Discussion of flight test experience with the FCS20 controlling both an unmanned helicopter and a small 11-inch ducted fan follow.

2. Hardware Architecture

A system that is designed for minimum size and weight necessarily entails sacrifices in performance, making the same system undesirable for larger vehicles that do not impose the same size and weight restrictions. Furthermore, a small, inexpensive system effectively becomes expensive if it requires a separate support structure and/or operates with diminished capability. Ideally, a system designed to meet the performance requirements of a larger platform should meet the size, weight, and power demands of smaller vehicles. The goal of this hardware design was to optimize the performance to size ratio for a widely applicable flight control system (good for a large class of vehicle size, including small), a system enabled by Micro Electro Mechanical Systems (MEMS) sensor and embedded processing technologies.

A. FPGA/DSP

As result, the overall form factor of the FCS20 system is a function of the minimum space required to fit a microprocessor, a large FPGA, memory and interface headers on a single board. The result is a credit card-sized board (55mm x 85mm). The FCS20 consists of a minimum of two boards of this size, the processor board and a power/sensor board. Each board has four mounting holes and four 2mm 40-pin headers. The two boards are connected back to back. Additional processor boards or application-specific boards may be added to the stack for increased performance. High-speed differential ports in the FPGA enable dedicated communication channels to other processor boards. Power and ground pins are distributed among the headers to enable multiple board configurations.

Figure 2 shows the baseline hardware configuration for the FCS20. The DSP gives the board its sequential processing power, while the FPGA performs low-level and parallel interface functions for external components. A fast, parallel data bus enables communication between the FPGA and the DSP, minimizing the amount of interruption to the DSP. Sensor data is preprocessed in the FPGA, bundled and transferred to the DSP through a set of First-In First-Out (FIFO) queues in the FPGA. Actuator commands are sent from the DSP to the appropriate servo driver component in the FPGA. The delegation of low-level interface tasks to components in the FPGA increases the overall efficiency of the system, allowing the DSP to dedicate its resources to high-level tasks and signal processing.

B. Sensors and Actuators

Depicted in Figure 3, the sensor board contains three Analog Devices ADXR300 Rate Gyros. In addition, two Analog Devices ADXL210E accelerometers complete the inertial measurement unit set. This is augmented with absolute and differential pressure sensors and a μBlox GPS OEM module. The board contains four voltage regulators and level converters for the input/output pins used to interface with servos/actuators and other components.

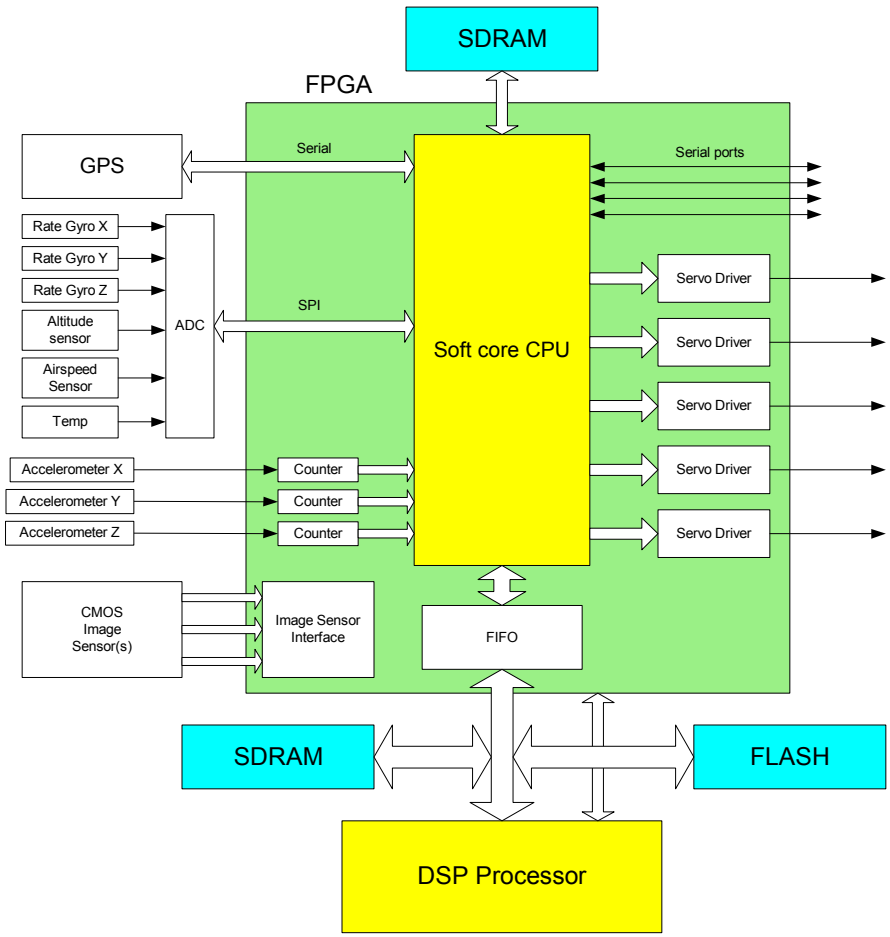

Figure 2 In the baseline hardware configuration, the DSP is mainly focused on the high-level tasks and signal processing, while the softcore CPU in the FPGA handles all I/O functionality.

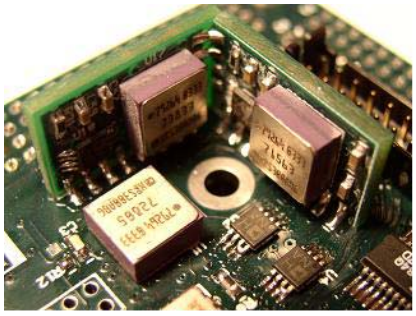

Figure 3 Three rate gyros are mounted orthogonally to provide three-dimensional angular rate information.

3. Software

The DSP in the FCS20 utilizes the MicroC/OS-II real-time operating system[1]. The operating system is a preemptive, real-time multitasking kernel setup to manage up to 64 separate tasks. This kernel includes basic operating system services such as semaphores, mutual exclusion semaphores, event flags, message mailboxes, message queues, task management, fixed size memory block management and time management functions. The execution times for most services in the operating system are both constant and deterministic and do not depend on the number of tasks running in the system. In the FCS20, the RTOS manages the data transfer from and to the FPGA.

A. Overall Structure

All software and FPGA configuration data are stored in non-volatile Flash memory. When power is supplied to the board, a small boot program is loaded and executed in the DSP. The boot file initializes the DSP hardware and successively configures the FPGA from Flash, loads software for the softcore processor in the FPGA, and finally loads the main DSP application into SDRAM and/or internal memory. From then on, the softcore CPU on the FPGA initiates every flight control cycle by sending the most recent sensor data to the DSP, effectively becoming the system driver.

Efficient internal and external communication is essential in a multi-processor system. The DSP communicates with the FPGA through a 32-bit parallel bus with a maximum throughput of 250 Mbps. Multiple, parallel First-In First-Out (FIFO) components inside the FPGA enable direct communication with the DSP from dedicated hardware components in the FPGA, avoiding information bottlenecks and reducing time delays. The FIFOs also act as buffers between the different components, which helps ensure data integrity. As shown in Figure 4, some information bypasses the softcore CPU and enters the DSP directly through hard-mapped UARTs in the FPGA to reduce the workload on the smaller softcore processor. The flexibility the FPGA allows for future addition of communication interface logic, image sensor interfaces, Ethernet, USB, and encryption or other protocols.

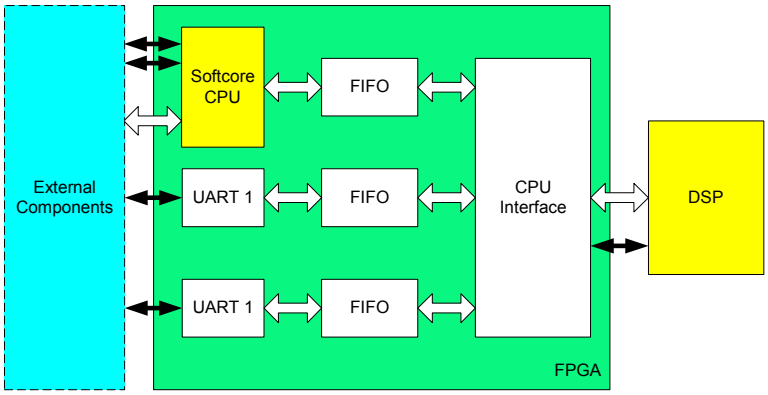

Figure 4 The schematic of internal and external communication highlights the fact that some information bypasses the softcore CPU and alleviating its workload

The guidance, navigation, control and communication software has a platform-independent structure. All the hardware or operating system dependent software has been pulled into a separate software library, called the Standard Vehicle Interface Library (SVIL). Data communication ports that are created in SVIL handle all communication in and out of this flight controller software. Figure 5 depicts the overall software architecture of the complete flight controller system with the FCS20 at its base. This allows the flight controller code to be common across a number of different operating systems and hardware choices.

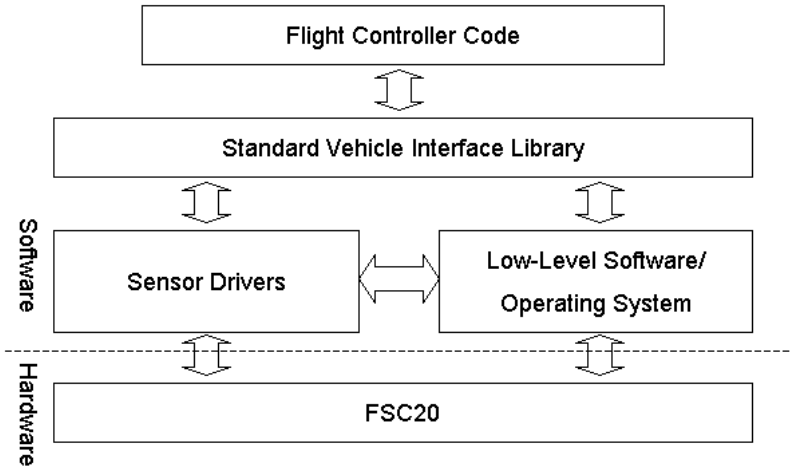

Figure 5 The controller software is platform independent. Hardware or Operating System-specific software is located in the Standard Vehicle Interface Library (SVIL).

B. Guidance, Navigation, Control and Communications Software

The FCS20 navigation system is a 17 state extended Kalman filter. The states include: vehicle position, velocity, attitude (quaternion), accelerometer biases, gyro biases, and terrain height error[2]. The system is all-attitude capable and updates at 100 Hz. The baseline flight controller is an adaptive neural network trajectory following controller with 18 neural network inputs, 5 hidden layer neurons, and outputs for each vehicle degrees of freedom, 6 for the case of the small ducted fan UAV[3]. The baseline flight controller and navigation system, which coupled with the simple baseline trajectory generator, is capable of automatic takeoff, landing, hover, and aggressive maneuvering. This approach allows a given UAV to adapt to even drastic changes in configuration, providing a great deal of flexibility during operation and a reduction in development cost.

Generic and highly-capable data communication software has been developed to support a large number of potential flight and simulator test configurations[4]. First, these routines support serial data reading and writing as necessary for the Commercial Off The Shelf (COTS) sensors and other custom components used. These same routines can also be used to re-route any data through Ethernet (if available) or as memory within a single executable. These data routings can be modified in real-time, by software switch. It should be noted that almost all operating system specific software is limited to these routines, including for the microC operating system. These data communication routines are used to interface with all sensors, the wireless serial data link, and to repeat all wireless serial data over the wireless Ethernet (for redundancy).

Additionally, UAV obstacle detection and avoidance algorithms require systems that use other sensors to detect and interpret the environment around the aircraft. These sensors normally need real-time signal processing capabilities, which are generally available on the DSP. Hence the FCS20 with its FPGA providing custom interfaces to aircraft hardware such as servo mechanisms and built in inertial sensors, and the DSP providing processing capability for navigation, control and sensor processing algorithms allows a wide range of application domains to be implemented.

4. Results

To date, the FCS20 has been flight-tested on two research aircraft. The first is an unmanned helicopter, the second is a small ducted-fan UAV. Results from these tests are summarized here.

A. Unmanned Helicopter

The GTMax research UAV, figure 6, consists of four major elements: the basic Yamaha R-Max airframe, a modular avionics system, baseline software, and a set of simulation tools[4]. The Yamaha R-Max industrial helicopter airframe, has a 10.2 ft rotor diameter. The hardware components that make up the baseline flight avionics include general purpose processing capabilities and sensing, and add approximately 35 pounds to the basic airframe, leading to a total weight of approximately 157 pounds. The digital interface to the vehicle is via a modified Yamaha

Attitude Control System (YACS) interface that allows raw servo commands to be given without modification by this stability augmentation system. The YACS also has other sensor outputs. The current baseline research avionics configuration includes: two Intel processors, an IMU, differential GPS, 3-axis magnetometer, sonar altimeter, and wireless data links.

Figure 6 – GTMax Research UAV

An FCS20 system was added to the GTMax for testing purposes. This enabled detailed comparisons with the GTMax' more capable sensors. Wiring was also modified to allow the FCS20 to control the GTMax servos, allowing the FCS20 to act completely independently of the existing avionics installed in the GTMax. Comparisons were also enabled by the fact that the FCS20 guidance, navigation, and control software is the same as that running on the existing flight control processor.

Several flights were conducted with this architecture, including several flights where the FCS20 was given full control of the helicopter in April 2004. Performance was inferior to the existing GTMax system primarily due to the less accurate GPS unit, which produces a larger position command tracking error. However, the full controller bandwidth and system update rates were available.

B. Small Ducted Fan Results

Following initial checkout on the GTMax, the system was utilized to control a small ducted-fan UAV. In this case, the MASS HeliSpy UAV, shown in Figure 7, was used. It has a maximum takeoff weight of 5.5 lbs and about 1 lb avionics payload capacity. Space and weight restrictions, vibration, power and cooling were some of the challenges faced during the integration effort. In addition, the unstable nature of the vehicle required gyro stabilization even during manual flight, a task the FCS20 is well suited for. Figure 8 shows a block diagram of the gyro stabilization system in the FPGA: The softcore CPU (Nios) calculates servo commands based on received commands and rate gyro data.

Figure 7 HeliSpy 11 inch ducted-fan UAV, weighs between five and seven pounds

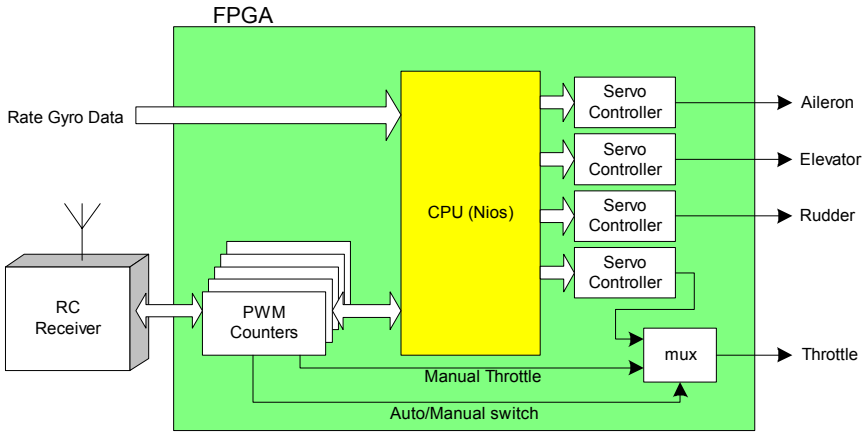

Figure 8. The Nios CPU in the FPGA uses rate gyro data to adjust the pilot's servo control signals in the Helispy. A throttle "kill" function is implemented in case the CPU locks up.

The following is a discussion of some of the specific challenges encountered during the integration process for the small ducted-fan:

1. Space: The entire avionics system had to fit inside a 3.5" by 8" long tube. This played an important role during the initial steps in the integration effort, but the final version of the avionics module used actually less than the available space.
2. Vibration: The vehicle is powered by a hobby-type combustion engine (OS32), which is mounted to the lower fuselage tube. As a result, there is severe vibration present in the lower half of the vehicle, particularly at lower RPM. Thanks to the unique design of the Helispy, there is even less vibration present in the top (forward) section, where the avionics is installed (Onboard video shows a remarkably stable image during flight).
3. Interference: Issues related to RF interference are difficult to avoid when attempting to package high frequency digital processors with sensitive analog circuits, transmitters and receivers. The goal is to achieve normal performance from all components/systems, while minimizing the total package volume and the use of (heavy) shielding materials. Here are some of the issues we encountered:
    a. The transmitter power from the FreeWave 900Mhz wireless serial transceiver had to be reduced to avoid interference with the nearby RC receiver.
    b. The oscillator on both the FCS20 and the video camera interfered with the GPS receiver. Replacing the oscillator and selecting a different camera solved the problem.
    c. Antennae were separated as much as practical on the vehicle.
4. Cooling: A few strategically placed holes in the upper fuselage of the vehicle allow for air to flow through the avionics during flight. During extended ground operations with the engine not running, an external cooling fan is attached to the upper fuselage, instead.
5. Center of Gravity: In order to increase effectiveness of the vehicle's control surfaces, it is important to keep the CG relatively high. Additional ballast had to be added to the nose (100g), to enable flight in up to 10kts of wind.

A number of tethered and approximately 10 un-tethered flights using this autopilot have been conducted in July and August of 2004, and are continuing. These flights included operations in some wind gusts (up to about 15 knots airspeed), automatic takeoffs and landings. Videos of some of these flights can be found at **`uav.ae.gatech.edu`**.

5. Conclusions and Future Work

Utilizing the FCS20 DSP/FPGA system, it was possible to demonstrate a highly capable processor combination (2 GFLOPS) together with an IMU, differential GPS, data link, video camera, video transmitter and batteries in a space slightly larger than a soda can (60 cubic inches), and weighing less than one pound. The system was developed and tested in a small ducted fan, but is suitable for many types of vehicles and system requirements. Work to further miniaturize the overall system volume is ongoing, particularly with respect to sensors and wireless communication. Processor utilization is also likely to increase to include image processing, obstacle avoidance, data encryption and any other feature required for advanced vehicle behavior.

Acknowledgements

The authors would like to acknowledge some of the many other contributors to this work: Jeong Hur, Phillip Jones, Daniel Schrage, George Vachtsevanos, and Brent Yates. This work is supported in part by the DARPA Software Enabled Control (SEC) Program under contracts #33615-98-C-1341 and #33615-99-C-1500.

References

[1]Labrosse, J. J., *MicroC/OS-II, The Real-Time Kernel*. CMP Books, 2003.

[2]Dittrich, J., and Johnson, E. N., "Multi-Sensor Navigation System for an Autonomous Helicopter," *AIAA/IEEE Digital Avionics Systems Conference*, 2002.

[3]Johnson, E. N. and Kannan, S. K., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," in *AIAA Guidance, Navigation and Control Conference*, AIAA-2002-4439, Monterey, CA, August 2002.

[4] Kannan, S. K., Koller, A. A. and Johnson, E. N., "Simulation and Development Environment for Multiple Heterogeneous UAVs" in *AIAA Modeling and Simulation Technology Conference*, AIAA-2004-5041, Providence, Rhose Island, August 2004.