FPGA Based Flexible Autopilot for Unmanned Systems

W. Alvis*, S. Murthy*, K. Valavanis*, W. Moreno*, M. Fields**, S. Katkoori*
*CSE and EE, Unmanned Systems Laboratory, University of South Florida, Tampa, FL, USA
**Army Research Lab, Aberdeen, MD, USA
Motivation

• Unmanned systems research bridges several areas of engineering, from the mechanical design of new platforms, system identification of the vehicle utilized in research, developing new controllers for improved response of the unmanned system, new forms of sensor integration, vision algorithms, swarm formation and mission planning.

• The available hardware that is currently available as finished products and proposed in research, while sufficient for finished unmanned systems products, falls short of being able to accommodate the needs of those in the research community that must be able to quickly implement new ideas on a flexible, easy to use platform.
CONTRIBUTION

The realization of a hardware platform that will allow rapid implementation of newly developed theory by providing an autopilot that provides Simulink as a high level programming language and a previously unrealized flexibility in sensor selection.
IMPLEMENTATIONS IN STATE OF THE ART

- Full computing system
  - Very popular [10-16]
- Microprocessor and Digital Signal Processor (DSP)
  - Very popular [17-33]
- DSP and Field Programmable Gate Array (FPGA) Hybrid
  - Very few publications [34-37]
- FPGA
  - Very few publications [38-39]
  - Research in preliminary stages
OVERVIEW FULL COMPUTING SYSTEM

• Mini-itx or PC104 type system
• Good processing power
• Not all unmanned systems have payload capacity
• Requires external hardware for interaction with many sensors that are analog or have 1.8, 3.3 or 5 volt logic interface—Proposed autopilot will have this capabilities and more
• High level functions must be programmed in such a way as not interfere with low level controls timing
  – Requires knowledge in real time operating systems
  – Not easy to modify when controls and sensor integration is modified
OVERVIEW
MICROPROCESSOR/DSP

• Ideal for low payload capacity vehicles because of low power and lighter package
• Requires background in embedded systems design
• Lacks parallel processing capabilities
  – Programming can be difficult because care must be taken to ensure that each of the asynchronous sensors are sampled at the correct time
DSP/FPGA HYBRID

- Improvement over DSP by including some parallel processing capabilities
- Majority of processes can not run in parallel
- Not easily implemented because knowledge is required in both DSP and FPGA programming
FULL FPGA

• Benefits micro air vehicle research
  – FPGA design is more easily converted to custom Integrated Circuit (IC) design
• Parallel processing lends itself more easily to modular software design for more easily modifying of one specific algorithm
• Very high frequency and parallel computational structures can provide computational speeds as much as 10 times sequential processor [40-42]
• Hardware allows variable word length [40-42]
  – Higher sampling rates
  – Better accuracy
  – Higher computational speed with less power consumption
# Autopilots: Market Products and Research

<table>
<thead>
<tr>
<th>Product</th>
<th>Proprietary</th>
<th>Hardware-In-The-Loop</th>
<th>Integration with Simulink</th>
<th>Take Over by External Processor</th>
<th>Processor Type</th>
<th>Programmable Analog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation II by BAI</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rotomotion</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kestral by Procerus</td>
<td>Some</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>DSP</td>
<td>No</td>
</tr>
<tr>
<td>MP2028 by Micropilot</td>
<td>Some</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>DSP</td>
<td>No</td>
</tr>
<tr>
<td>Ezi-Nav by Autonomous Unmanned Air Vehicles</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>8-µCONTROLLERS</td>
<td>No</td>
</tr>
<tr>
<td>Phoenix by O-Navi</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>DSP</td>
<td>No</td>
</tr>
<tr>
<td>Piccolo II by Cloudcap</td>
<td>No</td>
<td>Yes *with CAN card</td>
<td>Yes *Real Time Workshop</td>
<td>No</td>
<td>DSP</td>
<td>No</td>
</tr>
<tr>
<td>Microbot by Microbotics</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Maybe *FPGA may allow it</td>
<td>FPGA/DSP</td>
<td>No</td>
</tr>
<tr>
<td>PCM/Controller Board by Air Force (Wright Patterson AFB)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Maybe *FPGA may allow it</td>
<td>FPGA/DSP</td>
<td>No</td>
</tr>
<tr>
<td>FCS20 by GA Tech</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>FPGA/DSP</td>
<td>No</td>
</tr>
<tr>
<td>Virginia Commonwealth Sazuki V Board</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>FPGA</td>
<td>No</td>
</tr>
<tr>
<td>USF Proposed Design</td>
<td>No</td>
<td>Yes *USB connection</td>
<td>Yes</td>
<td>Yes</td>
<td>FPGA</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Ability to communicate with and follow commands of second processing system & Take over by either “master” computer or human pilot
On board pressure sensors for altitude & forward velocity
Flexible inputs, both digital and analog
Additional memory for data acquisition, kept separate from program memory.
Virtex II Pro FPGA
- 2 PowerPC™ 405
- Full integration with MATLAB / Simulink through System Generator Toolbox

Software running on FPGA
- Provided VHDL libraries
- Standard System Generator building blocks,
- Functions running on a user selectable operating system within the PowerPCs
GENERAL AUTOPILOT REQUIREMENTS

- Integration with MATLAB/Simulink for both simplicity of programming and hardware-in-the-loop simulation capabilities
- Adequate memory for data collection for use with system identification research
- Analog design to allow for reconfigurable cross platform/sensor capabilities
- Ability to act as a slave to a master processor
  - support of vehicles implementing full computing system for high level software development
  - Process high level navigation commands for autonomous operation
- Parallel processing capabilities & hardware level timing control.
- Emergency take over of servos for additional layer of safety
Support to High Level Software Development

- Design of high level algorithms independent of Hardware
- Implemented on full computing systems for vehicles with payload capacity
- Usually familiar with C-programming for implementation.
- Desire robot to follow high level navigation commands automatically.
DUAL BOARD DESIGN

STACKED BOARD DESIGN WITH MOST LIKELY NEEDED COMPONENTS ON MAIN BOARD BUT WITH UPPER BOARD TO ALLOW MORE FLEXIBILITY

DAUGHTER BOARD CUSTOM COMPONENTS
ON-BOARD SENSORS FOR STAND ALONE FUNCTION,
PRIMARILY MICRO AIR VEHICLES

MAIN BOARD WITH COMMON COMPONENTS SUCH AS PROCESSOR, SENSOR INPUTS, PRESSURE SENSOR, CLOCK, MEMORY, SIMULINK INTEGRATION
DUAL BOARD DESIGN

STACKED BOARD DESIGN WITH MOST LIKELY NEEDED COMPONENTS ON MAIN BOARD BUT WITH UPPER BOARD TO ALLOW MORE FLEXIBILITY

DAUGHTER BOARD CUSTOM COMPONENTS
EMERGENCY TAKE-OVER BY MASTER COMPUTER WILL BE DEVELOPED

MAIN BOARD WITH COMMON COMPONENTS SUCH AS PROCESSOR, SENSOR INPUTS, PRESSURE SENSOR, CLOCK, MEMORY, SIMULINK INTEGRATION
**I/O Ports and Sensors**

- Two On-board pressure sensors for altitude and forward speed.
- Two large signal, single ended analog inputs.
- Two small signal differential analog inputs.
- Twenty-four variable voltage logic inputs.
  - Input voltage set in blocks of four.
  - 1.8V to 5V range
- 5 Tx and 5 Rx RS232 lines.
  - Allows for 1 full duplex or two half duplex connections
- Three USB ports.

**Capabilities**

- Integration with master computer.
  - requires second board for emergency take over of sensors and servos
- On-board data acquisition memory.
- Twelve servo outputs.
- Safety switch for servo control.
  - Six critical servos can be taken over by pilot
  - All twelve, selectable in sets of six by second board (if present)
- Simulink programming and hardware-in-the-loop.
- 60 I/O FPGA connections to daughter board.
Predefined connections to second board. FPGA can be reconfigured around this, won’t limit functionality.
AUTOPilot MAIN BOARD

- Static RAM for program memory
- Flash for data acquisition

- Analog Signal Conditioning A/D Converter
- Voltage Translator Circuitry
- +5Vcc Supply CMN
- 3.3 Logic Lines
- Data Acquisition Memory
- Program Memory
- Actuators
- Human Pilot
- Receiver/Servo Control
- PWM to Servos
- 1 PWM for Control
- 6 PWM to Servos
- 12 PWM
- Clock
- Data Line
- Control Line
- SIMULINK Programming & Hardware-in-Loop
Digital sensor inputs such as: RS232, i²c, SPI
Level conversion needed for different voltage levels.
USB Communication
Need library for USB devices
Code provided for Simulink by Mathworks
Two on-board pressure sensors for altitude & forward velocity: standard sensors that requires custom analog signal conditioning & filtering.
**Autopilot Main Board Design**

- **FPAA used for programmable analog signal conditioning & 8-bit A/D**

- **Large Signal Analog Inputs:** 0 to 20 Volts, Single Ended
- **Small Signal Analog Inputs:** -2 to 2 Volts, Differential

FPAA used for programmable analog signal conditioning & 8-bit A/D
AUTOPilot MAIN BOARD

Safety Switch for Emergency Take-over by Circuitry on Second Safety Board or Human Pilot
- Twelve output servos
- Pilot input only six, these are the servos that control vehicle flight
- Allows the pan and tilt to be controlled by the computer while vehicle is flown by human pilot
- CPLD acts as a three way switch
**SECONDARY SAFETY BOARD SPECIFICATIONS**

<table>
<thead>
<tr>
<th>I/O PORTS WITH MASTER COMPUTER TAKE OVER</th>
<th>CAPABILITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two large signal, single ended analog inputs</td>
<td>Communication with master computer</td>
</tr>
<tr>
<td>Two small signal differential analog inputs</td>
<td>Takeover of control of vehicle dynamics</td>
</tr>
<tr>
<td>Twenty-four variable voltage logic inputs</td>
<td>Takeover of six servos for camera pan and tilt control</td>
</tr>
<tr>
<td>• Input voltage set in blocks of four</td>
<td>Separate circuit design for additional failsafe mechanism</td>
</tr>
<tr>
<td>• 1.8V to 5V range</td>
<td></td>
</tr>
<tr>
<td>5 Tx and 5 Rx RS232 lines</td>
<td></td>
</tr>
<tr>
<td>• Allows for 1 full duplex or two half duplex connections</td>
<td></td>
</tr>
<tr>
<td>Three USB ports</td>
<td></td>
</tr>
</tbody>
</table>
Master computer can not “talk” to TTL components, so a second chip & clock will be needed for intermediate communication between the sensors and the actuators. Looking at Texas instrument’s DSP chips rather than an FPGA because the master computer wont have real time control & the goal is to maintain just enough control for emergency return home or landing.
Sensor input now goes through multiplexer that is under the control of the DSP chip.
If master computer determines autopilot failure, a signal is sent to servo switch on main board & PWM to servos is sent from DSP chip to servos.
Software Overview

• Software within the autopilot is built from both provided libraries and the standard System Generator building blocks
  – Sensor and actuator interfaces to the FPGA are written in VHDL code to efficiently manage the internal hardware clock & made building blocks within the provided library
  – Previously developed Matlab m-code functions can be converted to System Generator fixed point blocks utilizing AccelDSP software
  – High level signal processing functions, such as filtering, sensor integration, and controllers can be developed using standard System Generator functions

• Various functions can be run on a user selectable operating system within the PowerPCs
HARDWARE-IN-THE-LOOP CAPABILITIES

- Simulate a System Generator design in Simulink
  - Simulate the control algorithm for the Mettler linearised VTOL model using system generator and verify the performance.
  - System Generator is capable of simulating hardware behavior
- Hardware-in-the-loop through USB
  - Downloads controller design in hardware using the JTAG connection to the target FPGA.
  - Simulink transmits data to FPGA for processing and receives the processed samples that can be viewed on Simulink sinks
  - System generator synchronizes the FPGA clock frequency to Simulink
- Hardware-in-the-loop with data acquisition card
  - Card not selected
  - Simulink model must be synchronized to system clock
  - Becomes hard real time simulation
The proposed design presented not only improves upon the processing capabilities of small DSP/microcontroller designs, but it also provides programming and hardware-in-the-loop capabilities in an environment familiar to researchers in many areas of engineering that will allow for rapid prototyping of new ideas. The design also presents an unrivaled flexibility due to the programmable analog interface and the inherent flexibility of the FPGA processor. The combined functionality and flexibility of this research will lead to a novel and well-needed processing platform for the unmanned systems community.


