

Robust Real-Time Microcontroller-based Control Hardware for a 21.3 m Bow Lifting Body Technology Demonstrator Craft

Robert Knapp, John Elm, and Brian Kays

Navatek, Ltd., Honolulu, Hawaii, USA

ABSTRACT

High-performance boats often require active stabilization using an embedded control system, but the marine environment is hard on electronics. Embedded computers must survive widely-ranging, noisy input power as well as high vibration, high-g shock, widely varying temperatures, and electromagnetic interference. Because typical commercial off-the-shelf (COTS) control technology is not designed for marine use, it is often unreliable and can even be dangerous on boats requiring stabilization. Therefore, to stabilize the 21.3 m technology demonstrator BLB-70 we used hydraulic actuation controlled, not by COTS technology, but instead by custom microcontroller-based hardware and software that we developed. Because microcontrollers are small, tough, and start up within milliseconds, they are often used in anti-lock braking and other demanding automotive applications. We overcame a traditional concern, their lack of floating-point throughput, by mounting five microcontrollers on a single board, all with distinct, specific tasks, in communication with each other using custom hardware and software. This paper explains the features and design principles of a marine-ready, multi-microcontroller ride control system and describes the successful use of such a system in stabilizing vessels such as BLB-70. Future development directions for this system are suggested.

Keywords

Lifting body vessel control system, marine electronics, multi-CPU network.

1 INTRODUCTION

High-performance boats often require stabilization. Electronic control systems that provide such stabilization require real-time data processing to respond to the highly dynamic nature of the ocean environment. Some ships require active stabilization in roll, pitch, and height above the water to maintain dynamic stability; hydrofoils with fully-submerged foils fall into this category. Other ships might employ active stabilization to improve their

seakeeping performance; ships that do not fly their hulls entirely out of the water but partially reduce hull immersion through the lift provided by lifting body technology or hydrofoils fall into this category.

In this latter case of lifting body ships, intact transverse stability decreases with increasing speed due to: i) reduced waterplane area as draft decreases, which can be dramatic when hulls are not wall-sided, and ii) a virtual rise in the center of gravity due to dynamic lift applied at or near the keel. The technology demonstrator BLB-70 (Figure 1) is a 21.3 m, 29 t stabilized monohull with small outboard hulls, or amas, that are designed to remain dry at cruising speed; by design it requires active stabilization in roll, pitch, and height above the water to maximize performance and to achieve superior ride quality in rough water. BLB-70's active stabilization is achieved using an embedded control system similar to so-called fly-by-wire systems used on airplanes.

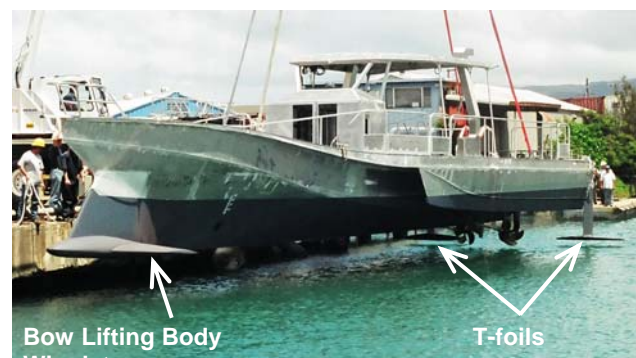


Figure 1: Navatek 21.3 m BLB-70. Hydraulically-actuated control surfaces comprise: outboard winglets on the bow lifting body actuated in tandem; twin, independently-actuated T-foils aft.

Control system hardware with the required capabilities and high reliability necessary for safe, high-performance embedded active stabilization aboard BLB-70 was not available commercially. Consequently, we chose to develop an advanced ride control system (ARCS) architecture based on the use of cooperating

microcontrollers, with some microcontrollers mounted on a single board and others on distant boards, connected via high-speed fiber optic links.

Previously, the use of microcontrollers in such an application would not have been possible due to their lower computing power. However, the capabilities of these devices have been advancing rapidly in recent years and now approach microprocessor floating-point throughput with none of the many disadvantages of microprocessors. These disadvantages, most importantly with regard to marine applications, include: i) susceptibility to shock, ii) a tendency to declock (slow down) when exposed to high ambient temperature, and iii) use of external RAM (Random Access Memory) and concomitant use of an operating system, factors which in turn result in a very slow start-up cycle compared to that of microcontrollers.

2 THE CASE FOR MICROCONTROLLER-BASED MARINE CONTROL HARDWARE

This section gives a short discussion of embedded control platforms, a summary of the special challenges posed by the marine environment, and the case for an entirely new marine-centric controller design paradigm. In subsequent sections, more detail is presented concerning the ARCS designed for the BLB-70, a brief history of ARCS development, and possible future enhancements to microcontroller-based marine control hardware.

2.1 Embedded Processor Architecture

Early marine flight control systems, such as those on Boeing's high-performance hydrofoil *Jetfoil*, used analog electronics for their computational needs. Modern digital control systems have widely employed Intel's venerable x86-based microprocessors, either in custom designs or in various commercially-available Single-Board Computer (SBC) stackable card formats, most notably the PC104 architecture (Figure 2). Such SBCs typically offer a microprocessor chipset, a solid-state disk drive, communications interface chips, central processor unit (CPU) power conditioning, and limited analog functionality all on a single, densely laid-out card.

Unfortunately, a COTS SBC card is never a complete solution for marine control system use. Using a COTS board in a marine embedded control application requires additional protection from short-circuits, additional analog input filtering, more or different communications capabilities, conversion of widely-ranging shipboard power, and possibly power to distribute to the ship's control surface position-sensing transducers. To accomplish this, a large stack of electronics cards—including some that are custom-designed—connected by ribbon cables and other wiring must inevitably be used. The final cost of such a configuration is always higher than one would expect due to the need for custom cards and the system's required inter-card wiring. On the plus side, COTS SBC systems are able to leverage the high computing power of a modern microprocessor, and

because the computational throughput of these SBCs is generally acknowledged to be in the hundreds of millions of instructions per second (MIPS), in non-marine applications these SBCs have been very successful.

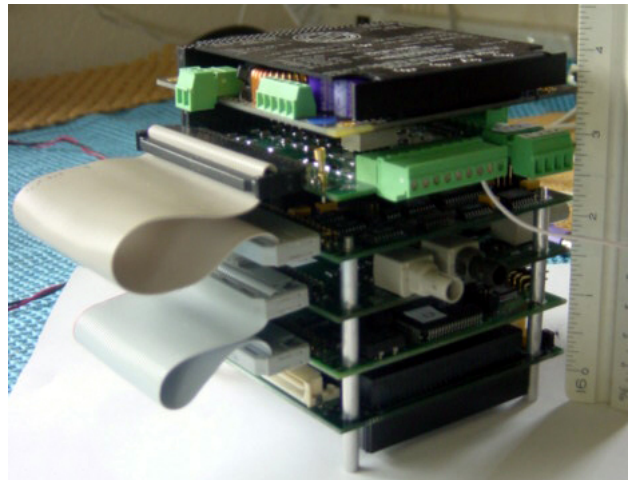


Figure 2: A PC104 embedded control system card stack, comprising a COTS power conditioning card, a custom analog filtering card, a COTS analog I/O card, a custom fiber-optic communications link card, a COTS serial communications card, and a COTS CPU card.

Fortunately, recent 8-, 16-, and 32-bit microcontrollers have attained high computing throughput either in the form of intellectual property 'cores' to be implemented inside other chips, or as standalone processors for use in a wide range of industrial and consumer applications. Microcontrollers are cheap, competitive, sold in great volume, and are relatively well-supported by their manufacturers.

Generally speaking, there are two important trends in the relentless march toward ever-increasing computing throughput:

(i) Microprocessors, driven by the computationally-intensive personal computer industry, are becoming larger; are running hotter, requiring fans and liquid cooling in some cases; and are becoming increasingly power-hungry (5-8 watts is typical just for the processor). At the same time, they are becoming more fragile mechanically, they require operating systems and external RAM, and they have no native analog input/output (I/O) or current-sinking capabilities. As a result, they are arguably less than reliable overkill for embedded control. Simply stated, they are best-suited for low-shock, non-marine applications.

(ii) Microcontroller evolution, in contrast, is driven by a myriad of low-cost, low-power embedded applications in tough environments such as automobiles, consumer appliances, medical equipment, and cell phones. Microcontrollers consume less power (one watt, typically), have small die sizes, and are mechanically tough. Additionally, they start up rapidly, require no external storage capability or external RAM, offer analog and discrete I/O.

encased in solder-resist mask) is far more reliable than wiring. It is also desirable to use tough, environmentally sealed connectors.

c) *Vibration and high-g shock commonly cause electrical problems aboard fast ships.* These conditions can cause heavy components to rattle off boards and electronics to experience connector problems. To minimize this, the control system's connectors and chips themselves must be tough and well-mounted; the boards must be physically thick; and the overall housing should be shock-mounted.

d) *Power for shipboard systems is not SBC-ready.* All commercial SBCs require a clean +5 VDC power supply, but power for shipboard electronics is usually provided as either 12 VDC or 24 VDC. All shipboard systems must be able to tolerate quick power spikes and general voltage variation on each bus ranging from 20% undervoltage to 50% overvoltage. Hence, considerable power isolation and conditioning is needed before a COTS SBC can operate effectively in a shipboard environment.

e) *Temperatures onboard ships can range from below freezing to higher than 50 C.* In the face of extraordinary heat, microprocessors will either summon more cooling power, if available, or declock (slow down the system clock). We once experienced a failure in which the temperature sensor for a CPU on an SBC failed; the microprocessor subsequently declocked and was then too slow to communicate serially with any other system component, let alone control the vessel. Fortunately, this COTS SBC was part of a triply-redundant control system designed to handle many types of failures, so there was no impairment of boat safety.

2.3 A New Paradigm: Microcontroller-based Marine Advanced Ride-Control System Hardware

The main reasons to use microcontrollers in safety-critical or high-performance marine control applications such as the stabilization of a lifting body ship are the following:

a) *Reliability.* Microcontroller-based hardware is reliable in marine applications. Our experience with COTS-based control systems has involved many COTS board failures, whereas we have not yet experienced a microcontroller-based board failure.

b) *Rapid Start/Restart.* A microcontroller can start up and be available for controlling a vessel within tens of milliseconds. By contrast, a microprocessor necessarily starts up slowly after a power-up or brownout due to its need to load an operating system and then a program into RAM; this process may take many seconds to complete.

c) *Strong Configuration Control.* Microcontroller-based marine control hardware contains no source code that is not visible to the application developer. The firmware and operating systems used by COTS microprocessor-based boards may contain bugs and/or malicious code, and are subject to frequent manufacturer's revisions that can be costly and may be of suspect integrity. The fact that microcontrollers don't

require an operating system means that 100% of the code used in the controller can be inspected and verified, further enhancing reliability. Data-encryption routines such as AES, XTEA, or SKIPJACK could be easily implemented to secure the system's external communications interfaces. Note also that if desired, code and data could be secured inside the microcontrollers such that they could not be uploaded or in any way examined. This feature was originally provided by chipmakers to avoid reverse engineering.

The need for reliability, safety, and security in the control of fast ships suggests the need for a new kind of marine-centric ride control system design involving microcontrollers. The ride control system on BLB-70 uses two microcontroller-based PentaCon boards, one in the forward part of the boat and one aft, which communicate with each other via high-speed fiber-optics to share sensor information. We'll examine the details of this system in the next section.

3 BLB-70 CONTROL SYSTEM DETAILS

As noted earlier, the BLB-70 technology demonstrator craft is a 21.3 m 30-knot stabilized monohull employing patented bow lifting body technology (Loui et al. 2007) to maximize hull efficiency and to provide superior ride quality. The BLB-70 is designed to take advantage of the lift provided by its bow lifting body and aft control foils to partially reduce hull immersion, but the main hull does not fly completely out of the water. Consequently, the ride control system does not require the fail-safe redundancy typical of fully-flying hydrofoil control systems.

As noted in Figure 1, the control surfaces on BLB-70 are comprised of a pair of winglets on the bow lifting body, and twin, independently-actuated T-foils mounted on the transom. The BLB winglets are mounted on a common shaft and are operated in tandem by means of a single hydraulic cylinder in the bow. The T-foils, which rotate longitudinally about pivots near the top of their struts to provide incidence control, are actuated by individual hydraulic cylinders at the stern. All three servo-valves open proportionally to an input control voltage generated by the system's two PentaCons, and given normal hydraulic pressure the rams can fully extend or retract within approximately one second. The servo-valves offer at least 30 Hz bandwidth, but considerable damping occurs due to water displacement by the T-foils, thus a control loop issuing new servo commands at 100 Hz is smoothly commanding these servo-valves; the system thereby approximates continuous control.

Hull immersion depth, which we refer to as 'flying height' to be consistent with the terminology used in our fully-flying configurations, is principally controlled by means of the BLB winglets; pitch and roll control are accomplished through the T-foils.

The environmental sensor suite for the BLB-70 ARCS includes: i) millimeter-radar height sensor, ii) inertial measurement unit (6-axis accelerometer), iii) three single-

axis accelerometers, iv) Global Positioning System receiver. Additional sensors provide data to the ARCS for winglet position feedback, port and starboard T-foil position feedback, rudder position feedback, and port and starboard engine RPM.

3.1 PentaCon Features and ARCS Details

The BLB-70 advanced ride control system uses two types of control loops:

(i) An inner ‘servo-stabilization’ feedback control loop controls each of the craft’s three flow-rate servo-valves as output, and uses as feedback the position of the control surface. A commanded control surface position is achieved using this inner loop.

(ii) An outer ‘overall control’ feedback control loop processes the sensor inputs of IMU, GPS, and Height and generates the control surface commands; these commands are issued to the appropriate control surface servo-stabilization loops with the assumption that they will be followed.

Rudder position is automatically provided to the control system for purposes such as banking the craft during turns, but the rudder itself is manually controlled. Thus, if the BLB-70’s control system should fail rudder control would still be available.

Figure 4 shows a block diagram of the overall BLB-70 advanced ride control system layout. Each PentaCon comprises a dominant ‘Core’ processor that is starved out to four subservient processors called Peripheral Processors (PPs). Each PP (designated PP0 through PP3) is assigned a specific task, and it performs only that task. For example, a GPS parser is assigned to PP3 of the aft PentaCon node, and the software for that PP is wholly devoted to this task. Partitioning tasks by physical processor gives the system an object-orientation that yields a robust code and permits recovery from certain peripheral failures. Table 1 shows the assignment of tasks to the Peripheral Processors.

Table 1: Peripheral Processor task assignments.

Task	Peripheral Processor			
	PP0	PP1	PP2	PP3
Monitor display	B			
Inter-node bus		B		
IMU data parser			A	
GPS data parser				A
External computer interface			F	
DAQ data parser				F

B=Both nodes **A**=Aft node **F**=Fwd node

These tasks are described in more detail as follows:

- The monitor display task provides selectable data sets on system health and current IMU information to a display unit mounted on the captain’s console.

- The inter-node bus task relays all system information between the two PentaCon nodes.

- The Inertial Measurement Unit (IMU) parsing task parses the data stream from the IMU, condenses relevant information, and relays this information at high speed to the Core, which uses this information in its control scheme.

- The GPS parser takes any NMEA-0183 standard \$GPRMC sentence; the information is condensed and relayed to the Core for use in its navigation algorithms.

- The external computer interface allows a laptop computer to collect performance data. This is particularly useful when tuning the control algorithm.

- The DAQ data parsing task interfaces with a custom data acquisition (DAQ) board that functions as an interface card to an operator control panel, collecting operator input such as Pitch Angle Set or the acknowledgement of alarms.

The Cores in both PentaCons used on BLB-70 function identically; control inputs are fully shared between the two nodes. It should be noted that while BLB-70 requires only two nodes, it would be possible to connect as many nodes as needed to each other with certain software changes. Each Core computes the overall control law and outputs digital-to-analog voltages in order to control servo- valves, directed to the appropriate control surfaces.

The system uses all-custom communications protocols designed specifically for high throughput, in which both software protocols and classic hardware Ready-To-Send (RTS) and Clear-To-Send (CTS) control lines are used. The software validates communications messages wherever possible using a 16-bit Cyclic Redundancy Check (CRC-ITT). COTS peripherals, unfortunately, do not always include CRC generation, so further error checking is always employed in software to ensure that erroneous values are not fed into the control algorithm.

Note that PP functions need not be allocated in this order. On other vessels, we have used PentaCons in different topologies, parsed different types of IMUs, GPSs, and height sensors, and used the PentaCon’s five-processor network as needed. We consider the PentaCon as a building block to suit our needs, whether used for data collection or in embedded control as it is used onboard BLB-70.

Internally, PPs communicate with the Core at 2 Mbaud; externally, the baud rates are matched to the target peripheral, a technique referred to as baud-stepping. For example, many GPS units output messages at either 4800 or 9600 baud. A GPS-parsing PP can slowly gather a NMEA sentence, validate it, condense its information, attach a CRC, and then forward essential information to the Core at 2Mbaud. This kind of baud-stepping to and from the outside world is one of several throughput-enhancement services provided by a PP to the Core.

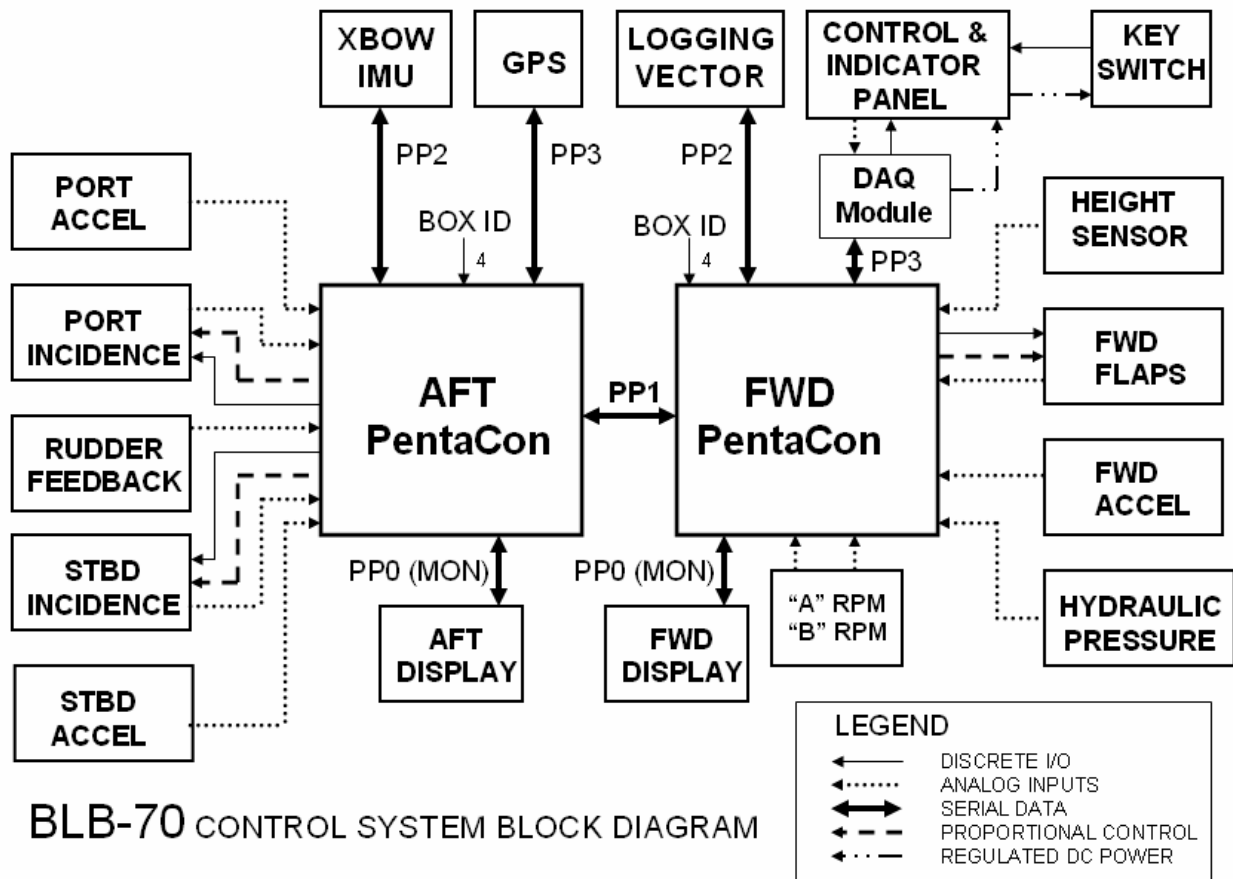


Figure 4: BLB-70 control system component-level block diagram. Two Pentacon share sensor information; the FWD Pentacon controls the BLB winglet control surfaces; the AFT Pentacon controls port and starboard incidence T-foils.

All five processors on a Pentacon share a common board-traced clock signal, but they run their software asynchronously and communicate via buffered serial data streams using either internal UARTs (Universal Asynchronous Receiver/Transmitters) or SPI (System Peripheral Interconnect) buses. The Core maintains strict timing and implements ‘real-time frames,’ but the PPs typically run freely; their input is solicited by their associated Core at the Core’s convenience. The PPs are thus typically subservient to the Core, though the hardware does not require this. As each PP is an independent CPU, its programming can be quite different from that of any other PP or the Core. PPs know their own identity via hardwired ID codes that are read upon startup; the software can branch as desired.

The ARCS control software implements a classic ‘producer/consumer’ model, in which fresh messages from IMUs/GPSs, etc., are of special interest at the beginning of a real-time task frame and are solicited at that time. Analog inputs such as current control surface positions are also pulled in. Discrete inputs are likewise learned at the beginning of a frame as these are often used to affect control operations. For example, on BLB-70, a keyswitch has been used to select between different control algorithms, or to enable or disable bow thrusters. Once all the needed serial-, analog- and discrete-inputs

have been gathered by the Core, or by PPs who transmit their refined and validated inputs to the Core, the control algorithm is executed, and new control signals are issued to the servo-valves.

Currently, the Pentacon board uses five identical microcontrollers. Microcontrollers have enough program code space and RAM to implement complex software builds if a software engineer is careful, particularly since there is no operating system that must share this space. We have used the Core’s nonvolatile EEPROM to store analog calibration data for the board itself and for the overall system; e.g., conversion factors from input rudder voltage to rudder angle. Because calibrations are needed whenever a physical change is made to the boat, we have developed an automated software procedure for such calibrations involving a laptop-hosted interface that is not compiled into the final flight control build.

A Pentacon has numerous features for testing and debugging code, including 30 LEDs to indicate various aspects of subsystem status and an independent watchdog timer that must be strobed by a cycling (non-frozen) processor in order to permit control functionality.

Based on our experience with ride control systems and examination of data collected from the ARCS on BLB-70, we have concluded that the version of the Pentacon

board used on BLB-70 has more than adequate computing throughput for that application. An upgraded PentaCon board, currently under development, replaces the existing 8-bit microcontrollers with more advanced 16-bit microcontrollers. The net result is approximately three times the floating-point throughput, which creates the opportunity for more advanced filtering techniques to be used in our control algorithms and kilohertz-band control. The upgraded PentaCon board has been designed to be appropriate for safe, real-time control of boats traveling at speeds of up to 100 knots.

3.2 ARCS Development History

The PentaCon board is the culmination of a deliberate research and development effort over the past ten years to produce a robust, reliable ride control system for lifting body ship applications. The following discussion presents a brief history of that development.

a) *First Generation ARCS*. Our first ARCS implementation was developed for the fully-flying hydrofoil *Westfoil* (Figure 5), a 28 m, 40-knot boat with twin high-performance fully-submerged hydrofoils each with six trailing-edge flaps. The control system employed six independently functioning PC104-stack COTS x86-based control computers running MS-DOS driving the 12 trailing edge flaps, which were actuated by electric servomotors. Custom hardware included analog filtering cards and a highly elaborate CPU interconnection wiring harness comprising hundreds of wires. Redundancy—and therefore safety—was provided by hydrodynamic voting, an arrangement in which an errant control surface’s authority is countered by the overcompensation of other control surfaces.

A simplified version of this control system was implemented on *Midfoil*, a 19 m, 25-knot boat incorporating a 41 t lifting body amidships and a T-foil forward (Figure 5). A single PC104 x86-based computer, identical to that used on *Westfoil*, and a combination of serial and analog input/output cards was sufficient to control pitch via the T-foil to enhance ride quality.

b) *Second Generation ARCS*. The second generation of our ARCS development was for *Sea Flyer*, a 59 m, 32-knot vessel with a 163 t lifting body located amidships and a hydrofoil located aft (Figure 5). Both the lifting body and hydrofoil incorporated trailing-edge flap control surfaces. *Sea Flyer* was designed to fly her catamaran hulls out of the water, so the ARCS had to provide full flight-control capability. For this application, we focused on redundancy in the system architecture to provide the required reliability. Three independent PC104-stack COTS x86-based control computers were connected in a ring configuration communicating via fiber optics. Each control computer received data from all of the feedback and environmental sensors, which included pairs of IMU, GPS, and height sensors that were considered critical. Each control computer independently evaluated the control law and issued commands to end-effector

computers called ‘voters’ that in turn resolved the correct command to follow using a mid-value voting scheme.



Figure 5: Three generations of ARCS development.

The complete system comprised 29 COTS and custom CPUs. Unfortunately, due to the challenging marine environment present onboard this high-performance craft, COTS board failures were not uncommon among the system’s 100+ ARCS boards. Navatek analyzed the failures and decided to replace the most-failed COTS stack, the voter, with a highly reliable in-house developed board, which became the PentaCon board.

The next iteration of control system development, produced for HDV-100 (Figure 5), a 28.8 m 45-knot vessel employing lifting bodies aft and an incidence-controlled foil forward, was a variation of that used on *Sea Flyer*. An all-composite design, HDV-100 was the first high-performance boat to use the PentaCon as an end-effector controller.

c) *Third Generation ARCS*. The third generation of our ARCS development was for the 21.3 m, 30-knot BLB-70 (Figure 1, Figure 5). Based on its demonstrated reliability on HDV-100 and its multiple microcontrollers that could be used for reading data from multiple sensors, we built the main control computers for BLB-70 around the PentaCon board with the system architecture shown in Figure 4. The successful use of PentaCon aboard BLB-70 has led us to conclude that we can eliminate most COTS technology from our ARCS, except for its use in sensor peripherals such as the IMU, GPS, etc., and in non-mission-critical displays.

d) *Fourth Generation ARCS*. The latest generation in the evolution of ARCS is currently in development. Designed for larger, faster ships, the system will employ a similar architecture to that used on BLB-70 with a pair of

PentaCon boards working in tandem as control computers. This part of the system will be repeated in triplicate to provide a high degree of system-level redundancy. Control surfaces will not be driven directly by the control computers as with the BLB-70 system; instead, the end-effector controllers will be PentaCon boards as in the HDV-100 system and commands will come from the BLB-70 style control computers via fiber optics. All PentaCon boards will be the upgraded version.

3.3 PentaCon Future Possibilities

The PentaCon represents a significant advance in marine-capable ride control system hardware functionality, but several enhancements to this board, and thus our ARCS portfolio, are conceivable:

Navigation. Current ARCS implementations include no navigation capabilities such as waypoint following—to which a typical GPS receiver is accurate within three meters—yet including such capability using a PentaCon would require only software changes since GPS is already in use.

Throughput Enhancement. Dual-ported- or quad-ported-RAM could replace the internal UARTs or SPI buses, if desired. Although we have not found system throughput to be a problem, this hardware change would permit more throughput-intensive algorithms to be implemented, such as large-matrix equation solvers that could take advantage of the combined power of several processors. Parallel algorithms could be implemented in PPs, with the Core coordinating the overall effort.

Onboard non-volatile disk-drive support. Hosting a high-capacity flash memory card would be useful in the case of data-collection functionality; for example, when a PentaCon's function is not to implement real-time control but instead to gather data in some remote location, then deliver it to a laptop in 'burst mode' at a later point.

CAN bus. Like the microcontroller itself, the Controller Area Network (CAN) bus has been rapidly increasing in importance due to its use in automobile applications. One possible enhancement to our PentaCon system would be to add CAN support. In this way, we could directly interface our PentaCon to certain marine engines and gain the kind of information about fuel consumption, torque, etc., that is useful in evaluating the true performance of different hull designs. Other communications extensions, such as Ethernet or USB, could also be added if needed.

4 CONCLUSIONS

We have successfully developed a new microcontroller-based hardware platform for marine embedded control systems—the PentaCon board. Specifically designed for the rigors of the marine environment, this hardware is well-suited to the real-time data processing demands of control systems that provide stabilization for high-performance boats, and it offers significant advantages over microprocessor-based COTS technology. Using this versatile building block, we have successfully performed data acquisition and real-time embedded control on several vessels, most recently on the 21.3 m BLB-70 technology demonstrator. Future applications will include larger distributed control systems with the PentaCon board configured for multiple, different end uses.

ACKNOWLEDGMENTS

We wish to acknowledge the support of the Office of Naval Research (ONR) in Navatek's ongoing research and development efforts.

REFERENCES

- Loui, S., Keipper, T., Shimozone, G., & Yamashita, S., (2007). 'Bow Lifting Body.' U.S. Patent No. 7,191,725, B2.