

Microcomputer Software for the Unmanned Free Swimming Submersible

D. E. Shirley

Naval Research Laboratory
Washington, DC 20375

ABSTRACT

A microcomputer guidance and control system for the Unmanned Free Swimming Submersible (UFSS) has been designed, fabricated and tested.

The UFSS is a laminar flow submersible developed to demonstrate long range, autonomous operation in open water. The operation of the vehicle is centered in the guidance/control computer. The central computer is aided by separate computers for navigation and acoustic telemetry. This paper will describe the software associated with the guidance/control computer.

The major responsibilities of the guidance/control computer software are: (1) mission sequencing and guidance and (2) depth and heading control. These areas are subdivided into several functional blocks described herein.

During shakedown tests in the summer of 1979, the guidance/control computer performed satisfactorily in all phases of the test runs.

1. INTRODUCTION

Under the sponsorship of the Naval Sea Systems Command, the Naval Research Laboratory has undertaken a program to design and develop an autonomous submersible to perform relatively long range ocean data gathering missions. The Unmanned Free Swimming Submersible (UFSS) has been designed and partially tested [1], [2].

The UFSS vehicle features an axisymmetric low-drag hull form that minimizes drag for a given volume at the designed operating speed. The overall length is approximately 20 ft, and the maximum diameter is 4 feet. The design speed is 5 knots. Maximum operating depth is 1500 feet.

The vehicle is propelled by a single 3 bladed propeller driven by a $\frac{1}{2}$ hp, three-phase induction motor. The vehicle is stabilized and maneuvered by aft-mounted cruciform fins with movable control surfaces. Variable ballast and trim systems provide additional control capabilities.

During a long mission the vehicle "pops up" near the surface to receive underwater OMEGA navigation signals to establish its position. The

vehicle returns to cruise depth to avoid surface turbulence that has a deleterious effect on the drag.

Autonomous operation of the submersible is centered in the Guidance/Control Computer. This subsystem is described briefly in the next section. A Digital Acoustic Command System (DACS) provides a limited number of commands to be sent from a surface support ship. This system is designed to function in a relatively severe multipath situation. The DACS initiates and terminates untethered operation of the vehicle. The DACS also provides range and bearing of the submersible relative to the surface ship. A second acoustic system provides further command capability to the vehicle, and returns data from the vehicle. This telemetry system requires a favorable submersible-surface ship geometry to aid in multipath protection. This system is called the Command and Data Acquisition Subsystem (CODAS).

A hardwired Failsafe subsystem acts both in conjunction with and independent of the Guidance/Control Computer to ensure vehicle safety in the event of malfunctions. The Failsafe subsystem monitors electrical power bus voltages, subsystem temperatures, pressure cylinder leaks, elapsed time, pressure-depth, and the Guidance/Control Computer.

2. COMPUTER AND SUPPORT EQUIPMENT

The UFSS vehicle has three computer units: the Guidance/Control Computer (G/CC), the OMEGA Navigation Equipment processor, and the Command and Data Acquisition Subsystem microcomputer unit. This paper will describe the software in G/CC unit.

The key element of the G/CC is an 8-bit, 8080 microprocessor. The G/CC has a total of 24 K bytes of Erasable-Programmable Read-Only Memory and 5 K bytes of Read-Write Memory (RAM). The G/CC has a 16 input, multiplexed analog-to-digital converter (ADC) and an eight channel digital-to-analog-converter. The digital input/output is configured as seven 8-bit parallel input ports and five 8-bit parallel output ports. There are two serial bidirectional (input-output) ports. One of these is configured for a 20 mA. current loop for connection to a teletypewriter. The second serial port is configured for an RS232 interface to the

Command and Data Acquisition Subsystem. A crystal-oscillator derived timer provides one second interrupts to establish real time and to supply uniform timing for the vehicle control loops.

The on-board G/CC is supported by both surface-craft-mounted and/or on-shore equipments. A teletypewriter can be connected to the vehicle by a ninety foot long oil filled umbilical cable. The vehicle end of the cable plugs into an underwater-mateable connector that is exposed by removing a cover plate in the afterbody. A Microcomputer Development System is used to make modifications to the software in the field. The components of this system are: (1) microcomputer, (2) high speed paper tape reader, (3) PROM programmer, (4) UV-EPROM eraser and (5) video terminal. A dual floppy disk unit has been added to this system since the in-water trials in the summer of 1979.

3. SOFTWARE ORGANIZATION

The UFSS guidance and control software is a real time control program written for a 8 bit microcomputer. The program is mainly written in PL/M, a high order language. A few subroutines are coded in assembly language to implement processing that is cumbersome or slow executing in PL/M, but are relatively simple or fast executing in assembly language.

The UFSS software is partitioned into nine functional parts (Fig. 1): Executive, Guidance, Control, Input-Output, Navigation Service, Failsafe-Confidence, Checkout, Utility and Initialization. In general, each of these parts is subdivided into smaller modules. Each of the modules has well defined inputs and outputs with narrow functional responsibility. The modularity is implemented by both untyped and typed PL/M procedures.

Fig. 2 shows the major flow of information in the UFSS software. Shown in dotted boxes are the major program databases. The major subprograms having to do with vehicle management are shown in solid boxes. The diagram shows that the Guidance subprogram gets information from the Guidance Instruction List, Navigation Variables, and the Raw Data Table. From these sources it computes commands and stores them in the Command Table. The Control subprogram accesses the Command Table and Raw Data Table, performs control calculations, and stores the results in the Control Table. The Input/Output subprogram gets information from the Control Table and sends it to the output devices. It also samples the input device signals and stores them in the Raw Data Table.

The Executive Subprogram serves to keep data flowing with results being obtained in a timely manner. At each one second interrupt the Executive invokes first the Input/Output, then the Control, and then the Guidance subprograms.

4. MAJOR SOFTWARE FUNCTIONS

The details of the software will be described in this section by detailing the function of the component subprograms.

Executive

The purpose of the Executive is to manage the computational resources in the G/CC. The executive routines handle interrupts, schedule and run other subprograms, keep track of real time, and provide some system diagnostics.

When interrupted by the one hertz timer, the real time clock interrupt handler updates two timekeeping clocks. The first is "executive time", and it is kept in seconds using three bytes of memory. This clock is used to control the scheduling of periodic software functions. The second clock keeps time in seconds, minutes, hours, and Julian days. It is initialized to Greenwich Mean Time by the system operator at the time that the software is started up. Greenwich Mean Time and Julian date are required by the Navigation subsystem for computing propagation corrections.

The Executive-controlled subprograms are referred to as tasks. The tasks can be called to run periodically or they can be called to run "as required".

The scheduling of tasks to be run is done on a priority basis, with high priority tasks being run before low priority tasks. The task scheduling information is maintained in a data area called the Task Control Block. Within the Task Control Block are five arrays. These arrays define the priority, iteration interval, next execute time, enable status, and suspend status of each task. At present up to 16 individual tasks can be handled.

The priority of each task can be assigned any of 8 levels. The iteration interval defines how often the task will execute. The next execute time is compared against the current executive time to see if a task is ready to run. In order to be ready to be run a task must have its enable status flag set. Each time a task runs the iteration interval is added to the current value for next execute time to find the next time that the task will be ready to run. Tasks can set the enable flag thus invoking another task on an "as required" basis.

Each time a task begins executing a suspend status flag is set in the Task Control Block. If an interrupt strikes before a running task is completed the task is marked as suspended. If there are higher priority tasks ready they will be run before continuing the suspended task. Upon completion of a task its suspend status flag is reset.

Guidance

The purpose of the Guidance subprograms is to direct the operation of the UFSS vehicle by

DACS, Failsafe subsystem, and the Command and Data Acquisition Subsystem. The Guidance routines determine the vehicle's position in earth coordinates by dead reckoning and call for OMEGA navigation position fixes when required. When the Guidance subprogram executes the results are stored in the Command Table. The Guidance software supervises the vehicle resources in contrast to the executive which manages computer resources.

The guidance software primary function is to provide a preprogrammed sequence of commands that accomplishes the desired mission. In addition it has to provide command sequences in response to externally occurring events. The external stimuli can come from the DACS, the Failsafe subsystem, or the CODAS. Since these inputs have different levels of importance, the guidance software incorporates a fixed priority scheme to honor the request with the greatest importance. The resolution and decoding of the inputs determines one of six mutually exclusive states of the vehicle. These states are: Disabled, Timed, Circle, Surface, Emergency Stop, and Abort.

The Disabled state is the initial state for any mission, and it is terminal state for a mission without emergencies. The vehicle is largely inactive (in a low power drain configuration) in this state.

The Timed state begins when the vehicle receives an acoustic "start" command from the DACS. In the Timed state the vehicle is guided by a preprogrammed mission sequence. More details on the Timed state are given below.

The Circle state commands the vehicle to circle until acoustic tracking signals are reacquired. If the signals are not received for one hour the vehicle surfaces, and the mission is aborted.

The Surface state causes the vehicle to climb to the surface using normal control procedures. This state results from a DACS command to surface or from the one hour circle timer.

The Emergency Stop state removes power from the propulsion motor and puts the control surfaces in their neutral position. It is in response to a DACS command input.

The Abort state uses all vehicle emergency measures to bring the vehicle to the surface. Variable ballast and auxillary buoyancy airbags are employed to provide lift.

While in the Timed state, commands from the CODAS can replace the current vehicle commands that have been issued by the preprogrammed sequence. These commands allow changing cruise depth, popup depth, propulsion speed and heading.

When the guidance subprogram is in the Timed state, a 3-byte guidance time clock is allowed to increment in one second steps. This clock is stopped if the Timed state is not valid. The Timed

mode processor reads and interprets a stored sequence in the Guidance Instruction List. The instruction repertoire includes 35 instructions which can: create wait intervals, close and open control loops, set heading and depth commands, invoke a navigation position fix, and several others. Each instruction has an operation code part and a single operand part. The operand part of an instruction varies from 0 to 3 bytes in length depending on the operation code part.

Control

The control routines implement the heading and depth control functions, set the propulsion motor speed, and control the variable ballast subsystem. Discrete flags in the Command Table open and close the depth or heading control loops. When the control loops are open the rudders or elevators are commanded directly from the Command Table.

Figure 3 shows a functional block diagram of the heading control mechanization. This controller features an outer heading to rudder feedback loop, and an inner yaw rate to rudder feedback loop that enhances the vehicle damping characteristics. Turn direction logic ensures that the vehicle turns through the smallest of complementary angles, and it obviates problems with the 0/360 degree ambiguity.

The heading control calculations are performed in double precision (16 bits). The input and output data are 12 bits. The update rate of 1 hertz was selected by a computer simulation of the control system, and it provides control that closely approaches continuous analog control.

Figure 4 shows a functional block diagram of the depth control implementation. This multi-feedback loop controller features an outer depth to elevator feedback loop, a pitch to elevator feedback loop, and a pitch rate to elevator feedback loop. Because the vehicle's pitch attitude is a close analog to depth rate, the pitch loop increases the depth damping of the vehicle. The pitch trim integrator produces an integral-type compensation that converts the depth control system from a type 0 (non-zero steady-state depth error) to a type 1 (zero steady-state depth error) control system. The integrator computes the pitch trim angle that is required to offset the vehicle's positive buoyancy. The depth switch logic selects an increased depth gain and an increased pitch command limit to generate greater downward lift to break the vehicle away from the surface. When the depth exceeds 10 feet the gain and limit are decreased to their nominal design values.

The depth control calculations are performed using double precision operands; the update rate is one hertz. The design of the heading and depth control systems is disclosed in detail in reference [3].

The variable ballast controller operates in two distinct modes. In the first mode, Direct, the controller selects the valves and turns on the pump motor to pump the onboard ballast to the desired position indicated in the Command Table. The second mode, called "Bootstrap", is used to aid in breaking away from the surface by decreasing the vehicle's net buoyancy. The Bootstrap mode makes the vehicle adaptive. It is useful when heavy seas make diving more difficult. The Bootstrap mode ballast control software monitors the vehicle's depth. If the vehicle has not achieved a 10 foot depth threshold in one minute, the ballast system decreases the net buoyancy until either the 10 foot threshold is achieved or until the buoyancy has been decreased by 10 pounds. In the case where the threshold was achieved the ballast controller then restores the original buoyancy condition. If the depth threshold was not achieved after decreasing the buoyancy by 10 pounds the ballast control routine sends a message to the Guidance software to abort the run.

Propulsion speed is set in an open loop manner from the value in the Command Table. The speed control software features command rate limiting to protect the motor controller/motor hardware from potentially damaging overload conditions.

Input/Output

The purpose of the Input/Output software is to access data outside of the G/CC and to provide data for use outside the computer. The data handled in these routines is handled by programmed I/O. The techniques employed are conventional and need not be elaborated.

Navigation Service

The Navigation Service routines are designed to send data and receive data from the Omega navigation Receiver/Processor Unit (RPU). The G/CC is interfaced to the RPU in such a manner as to appear as the Control/Display Unit (CDU) of this commercial equipment. The CDU has a rotary selector, a decimal keypad, and other pushbuttons to facilitate entering data into the RPU. The CDU displays data it receives from the RPU on 7-segment and 16-segment alphanumeric display readouts.

When a navigation position update is required, the RPU must be "initialized". This operation consists of entering the Julian day, Greenwich Mean Time, heading, true speed, estimated present latitude and longitude, and the destination latitude and longitude. The Navigation service software converts the binary stored information to decimal coded character strings that are sent serially to the RPU. In this manner the G/CC in effect rotates the selector switch and presses the pushbuttons to emulate a human entering the initialization data.

When the position has been computed the G/CC "reads" the "display windows" to find the current vehicle position, and bearing and distance to the destination. The Navigation Service routines

convert the segment-coded display information to decimal strings by table lookup.

Failsafe-Confidence

The purpose of the Failsafe-Confidence software is to indicate to the Failsafe subsystem that the G/CC is functional. This is accomplished by requiring that the G/CC periodically send a pulse to the Failsafe system. This task is given a low priority in the Executive. If the necessary processing which runs at higher priority is not being accomplished because of hardware or software failure, the pulses will cease and the Failsafe subsystem will attempt to recover from the casualty.

Checkout

The purpose of the Checkout software is to facilitate testing the UFSS vehicle prior to launch. The Checkout software provides the ability to: print sensor data, exercise the vehicle control surface actuators, and to print navigation information. The Checkout software also provides for exiting from the real-time mode to return to the monitor mode. By selectively setting the Task Control Block enable flags and entering commands into the Command Table, the operator can verify the operation of the vehicle control system software and hardware.

Utility

The Utility section of the UFSS software furnishes general utility functions such as: multiplication, division, square root, trigonometric and inverse trigonometric functions, and relational comparisons for signed double precision operands.

Initialize

The Initialize software presets variable data in the read/write memory. The general initialization is done in a two stage process. The first stage presets the data required by the Executive, Guidance, etc.. The second stage sets default initial conditions which the vehicle software operator can subsequently override if desired. Such items as control system gains, sensor bias errors, etc. can be adjusted before launch in this manner.

5. CONCLUSIONS

It has been found that the software based command and control system works very satisfactorily. The system flexibility afforded by having a software-based control system was demonstrated during the initial in-water tests. When difficulties were encountered in diving with moderate buoyancy margins and adverse weather conditions, the controllers were reprogrammed to produce additional diving force without adversely affecting dynamic stability while at depth.

The use of a high order language and structured programming principles has resulted in

6. REFERENCES

1. Johnson, H. A. "Design Features and Test Results of an Unmanned Free Swimming Submersible." OCEANS '80, IEEE Publ. No. 80CH1572-7, September 1980, p. 304.
2. Ozimina, C. D. , Shirley, D. E., and Carey, E. W., "A Microprocessor Application in an Unmanned Submersible", Proceedings of MIMI '80 Asilomar Conference, Jan. 30 - Feb. 1, 1980.
3. Ozimina, C. D. "Design of a Control System for an Unmanned Submersible", Proceedings of SOUTHEASTCON '79, Region 3 Conference, April 1-4, 1979.

1. Johnson, H. A. "Design Features and Test Results of an Unmanned Free Swimming Submersible." OCEANS '80, IEEE Publ. No. 80CH1572-7, September 1980, p. 304.
2. Ozimina, C. D., Shirley, D. E., and Carey, E. W., "A Microprocessor Application in an Unmanned Submersible", Proceedings of MIMI '80 Asilomar Conference, Jan. 30 - Feb. 1, 1980.
3. Ozimina, C. D. "Design of a Control System for an Unmanned Submersible", Proceedings of SOUTHEASTCON '79, Region 3 Conference, April 1-4, 1979.

1. Johnson, H. A. "Design Features and Test Results of an Unmanned Free Swimming Submersible." OCEANS '80, IEEE Publ. No. 80CH1572-7, September 1980, p. 304.
2. Ozimina, C. D., Shirley, D. E., and Carey, E. W., "A Microprocessor Application in an Unmanned Submersible", Proceedings of MIMI '80 Asilomar Conference, Jan. 30 - Feb. 1, 1980.
3. Ozimina, C. D. "Design of a Control System for an Unmanned Submersible", Proceedings of SOUTHEASTCON '79, Region 3 Conference, April 1-4, 1979.

