N90-23062

Combining High Performance Simulation, Data Acquisition, and Graphics Display Computers

by

Dr. Robert J. Hickman
21 Brittany Ct.
Newport Beach CA, 92660

ABSTRACT

For a growing class of simulation problems, the generation of the motion and signal environment for testing hardware-in-loop requires high speed computing along with data transfer, mass storage, and graphic display rates sufficient to save and display the data generated. This typically requires a complex of specialized processors that are specifically selected for their processing tasks, along with a specialized communication computer system for fast inter-processor communication and data transfer. This computer complex can be employed in different roles as the test hardware-in-loop interface matures during an advanced development or full scale engineering development program. Early in such a program, all elements of the system to be studied (and their environments) are simulated. Then as in-loop elements are developed, they are inserted into the complex, and the simulation computers concentrate on increasing the fidelity of the test environment dynamics that the in-loop elements experience.

This paper discusses issues involved in the continuing development of an advanced simulation complex. This approach provides the capability to perform the majority of tests on advanced systems, non-destructively. The controlled test environments can be replicated to examine the response of the systems under test to alternative treatments of the system control design, or test the function and qualification of specific hardware. Field tests verify that the elements simulated in the laboratories are sufficient.

The digital computer complex is hosted by a Digital Equipment Corp. MicroVAX computer with an Aptec Computer Systems Model 24 I/O computer performing the communication function. An Applied Dynamics International AD100 performs the high-speed simulation computing and an Evans & Sutherland PS350 performs on-line graphics display. A Scientific Computer Systems SCS40 acts as a high-performance Fortran program processor to support the complex, by generating numerous large files from programs coded in Fortran that are required for the real-time processing.

Four programming languages are involved in the process, FORTRAN, ADSIM, ADRIO, and STAPLE. FORTRAN is employed on the MicroVAX host to initialize and terminate the simulation runs on the system. The generation of the data files on the SCS40 also is performed with FORTRAN programs. ADSIM and ADRIO are used to program the processing elements of the AD100 and its IOCP processor. STAPLE is used to program the Aptec DIP and DIA processors.

INTRODUCTION

As developers of complex systems that include sensors, computers and actuators we must continually examine the need to maintain and improve our capability to design and test such systems. Advances in technology have encouraged our customers to seek more advanced systems that involve increasingly complex on-board control. While laboratory tests do not totally replace field testing, the laboratory can provide a controlled test environment wherein semi-physical testing of a number of alternative systems and components can be replicated in complex computer generated environments. This is difficult to achieve in field tests as many elements are not under the control of the experimenter and flight tests, for non-recoverable systems, typically result in destruction of the test hardware. The modern concept is to perform the majority of tests in non-destructive testing in simulation laboratories and utilize field tests to verify that the elements simulated in the laboratories are sufficient. The reduction in the number and duration of field tests leads to a direct reduction in costs and an improved competitive position in the high-technology development business. This paper discusses an approach for the phased introduction of target sensor and signal processor elements into a hardware-in-the-loop simulation that is designed to provide the capability to test and evaluate such systems.

During the initial stages of development the entire system is simulated by computer. As sensor and signal processing hardware elements are developed they are inserted into the control loop, producing a hybrid hardware-computer system. This hardware-in-loop simulation allows significantly improved non-destructive evaluation of design performance. Typically breadboard flight controllers or processors, with their embedded software algorithms and special purpose hardware, computational accuracy and latency can be tested early in a development program to evaluate the adequacy of the candidate approaches for control. As long-lead items become available (e.g. sensors, signal processors) they are integrated into the loop and their simulated characteristics are replaced by actual physical performance leading to improved confidence in the results.

A generic block diagram for the simulation of a guided vehicle system employing a target sensor, flight motion sensors and a control servos for aerodynamic fin deflection is presented in Figure 1. These elements appear in the double lined boxes and have the same interface in the simulation facility as they have in flight tests. The target sensor interfaces with the target signal spectrum in the isolation chamber and the flight motion sensors interface with inertia from the motion simulator. The aerodynamic and kinematic models of the vehicle and the target kinematic model are handled by the simulation computer with the target signal model generated in a computer complex that is re-structured as the sensor hardware is inserted into the loop. The 3-way switch under the target signal model illustrates the following options (from right to left):

1. Simulate the entire target sensing and signal processing function. The goal is to provide realistic inputs to the control processor so that performance trades can be evaluated in the conceptual design phase of an advanced development program.

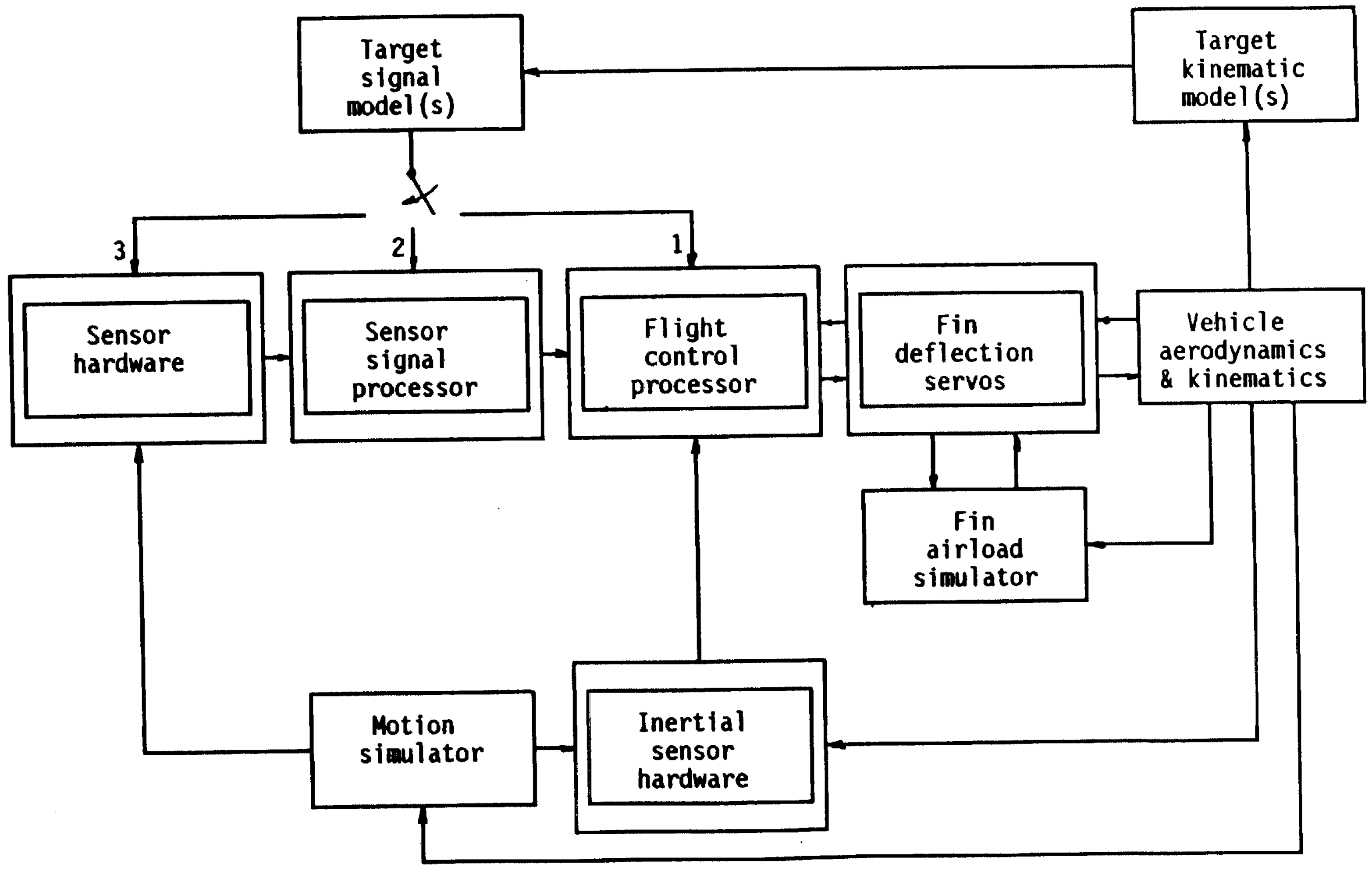

Figure 1, Hardware-in-Loop Block Diagram

2. Simulate the collection of energy from the target and the conversion of this energy into detector or pre-amplifier output signals that are processed by the signal processor. Here the concern is to exercise elements of the signal processing associated with the target sensor. This can involve automatic gain control for the amplifiers, signal thresholding for false alarm control, and target scene element state estimation for the flight control processor.

3. Encode the simulated target scene into an energy distribution of the scene and transmit this in the proper spectrum to the sensor, or in an analogous spectrum for surrogate detectors that are designed to provide "equivalent" response. This option is intended to exercise sensor pointing, energy collection, detectors and amplifiers as well as the elements discussed above.

The third option represents the goal that system developers may wish to achieve with their non-destructive semi-physical testing, however the cost of reaching the level of fidelity in target scene generation that can exercise the resolution and dynamic range of the sensor may inhibit that approach and lead to a compromise on the second option. While this may seem like a reasonable concession the fact that many unsuccessful attempts to develop target sensing systems have been associated with problems in the "front end" of the system involving energy collection, and conversion to signals for processing. Simulating the sensor and not including it as hardware-in-the-loop may really be whistling past the graveyard for the development program as the real need is to identify the anomalous behavior in the target sensor in dynamic non-destructive testing and develop signal processing and flight control "work arounds" until future versions of the sensor, that address fixes for the problems encountered, can be available for test.

REQUIREMENTS

Figure 2 presents a summary of desired features for the specialized computing functions necessary to implement the evolution of sensor hardware-in-the-loop testing. The major blocks in this figure are identified as specialized computing "servers", and may by themselves be a complex of computers serving to produce the specialized function in the overall simulation system. The requirements listed in the blocks are based on attaining high performance, mutual compatibility, and reasonable costs for peripheral devices and components with this distributed processing approach.

Development Server – The development server is the host computer for the system and acts as such for the other computers in the system. This leads to the requirement for open bus peripherals (to reduce costs) and configuration management software tools on this element as the majority of the software for the other computers will be written, cross-compiled, stored in libraries, linked, and down loaded at execution time by the host. The configuration management software is quite important as the complexity created by the inter-dependence of software across the system can become formidable. The VMS operating system from Digital Equipment Corp. (DEC) is an excellent choice for this element of the system as it provides the framework for achieving these requirements. This operating system executes on a DEC VAX computer and most potential users already have considerable experience with this combination. However it is probably not a good idea to use a VAX heavily loaded with unrelated use as the host for this system when a dedicated MicroVAX can handle this work-load for a very reasonable cost.

Development Server
- open bus peripherals
- configuration management
- VMS operating system

Communication Server
- high-speed communication
- transparent mode option
- high-capacity data logging
- modular organization
- VMS compatible

Simulation Server	Fortran Server	Graphics Server	Telemetry Server
- high-speed	- high-speed	- high-resolution	- high-speed
- 64 bit arithm.	- 64 bit arithm.	- high-speed	- on-line gather, smooth & tag
- high-level code	- open bus I/O	- high-level graphics	- modular organiz.
- real-time I/O	- on-line control	- local pan/zoom	- on-line control
- on-line control	- VMS compatible	- VMS compatible	- VMS compatible
- VMS compatible			

Figure 2, Desired Processing Features

Communication Server -- The communication server provides high-speed communication among the specialized servers of the system and as a result requires modular organization to accomodate the stepwise growth as this simulation system is developed. An Aptec Computer Systems I/O computer performs the communication function as it provides a high-speed data bus architecture with tightly coupled parallel controllers that supports simultaneous data transfers at the full input/output data rate of the attached processors, thus permitting real-time capture of the simulated environment and test data for storage on high-speed high-capacity disk drives and rapid interpretation of results through on-line graphic displays. An important feature of this computer is its "transparent mode" in which it mimics the DEC Unibus interface. This mode is very useful in reducing the software impact of connecting a set of different computers and/or devices, especially when these elements have been designed for use as attached processors to a DEC VAX host computer. Typically, in distributed systems, the majority of the communication software is involved with non-time intensive initializing and terminating the elements of the distributed complex as it is applied to a test problem. The real-time portion of the communication software may comprise only 10 to 20 percent of the total, so that the major portion of the software can be high-level language calls (e.g. Fortran) to libraries provided by the suppliers of these elements for execution on the VAX host processor for their processor or device. The real-time communication involves the "program mode" where a sub-set of the routines are required to be programmed for the Aptec controllers for high throughput. While this involves re-programming in many cases the form of the code is the same as that provided for the host VAX computer.

Simulation Server -- An Applied Dynamics AD100 simulation computer is employed to perform the table interpolation for aerodynamic coefficients and mass properties as well as the numerical integration of the set of non-linear ordinary differential equations that describe the motion of the flight vehicles and the sightlines. The performance of the AD100 on simulation problems has been compared to a number of general purpose computers, and it takes the capability of super-computers to match its results (Applied Dynamics, 86). The AD100 is a tightly coupled set of high-speed parallel pipelined processors that provide not only high-speed and high-precision but also real-time control of I/O to the attached hardware-in-the-loop, and on-line interactivity through the VAX host. Support software for the AD100, the ADSIM and ADRIO compilers and the INTERACT run-time program significantly reduce the programming and checkout burden and run-time inflexibility that has been associated with special purpose computers.

Sensor "front end" simulation can be performed using an Applied Dynamics AD10 computer to simulate sensor scene scanning and detector/pre-amplifier response. This response is decomposed into table interpolation of simulated detector/pre-amplifier response to the principal elements of the target scene and combined as a composite signal from this simulated portion of the sensor. The scene elements are defined by: 1) Response to the target as a function of 2 relative sightline angles, target aspect angle, and relative range; 2) Response to countermeasures as a function of one relative sightline angle and relative range; 3) Response to background as a function of 2 inertial sightline angles; 4) Atmospheric transmission coefficient as a function of relative range and altitude. The entries for the interpolation are determined from flight vehicle, sensor, and target motion variables that are determined in the AD100 and sensor scanning and signal processing variables simulated in the AD10. The AD10 and AD100 communicate at high rates through dual-ported memories on a frame to frame basis to provide the inputs for very high speed sensor response interpolation. The AD10 design was specifically optimized for table interpolation an its performance on this function is without equal. The AD10 is a set of tightly coupled high-speed parallel pipelined processors that operate on 16 bit data. This data format is more in keeping with the target scene data as the intensity data from sensors is quantified much less than this.

Target scene simulation is required for the third option of sensor-in-the-loop simulation. This requires the generation of a video bandwidth signal of the encoded energy distribution of the simulated target scene. A Pixar image computer provides the processing throughput and bus bandwidth to dynamically update a target scene from stored scene components as a function of inertial positions determined in the AD100. The scene components being generated by image processing off-line and transferred to the Pixar during initialization. A key decision in the selection of components to implement the third option for sensor simulation involving scene generation is the selection of the video format standard. The finite resolution and synchronous nature of the frame format presented to the sensor can lead to artifacts in the response of the sensor, especially if the sensor signal processing is in some sense differentiating elements of the observed scene. The conventional National Television Standards Code (NTSC) video format (also known as RS-170) callsfor a data frame of 480 lines of 525 element resolution being presented in an interlaced fashion at 60 fields a second. These interlaced fields contain either the 240 odd or even lines of a frame, so the total 480 line frame is presented 30 times a second in two successive interlaced fields. The RS-343 standard provides for increased lines per frame and more resolution elements per line (both in excess of 1000) however this limits the use of commercially available video equipment (cameras, recorders, monitors, control panels, mixing, special-effects, etc.). Many computer generated raster graphic displays exceed these standards by providing more lines, higher resolution, and higher frame rates without interlacing, however these displays are usually conventional cathode ray tubes driven by non-standard video interfaces. Use of these video displays is potentially acceptable as a medium for the presentation of a target scene to a sensor when the spectrum for the sensor is near the visible spectrum, very bright elements (countermeasures, sun glints, etc.) are not part of the scenario, and the combination of frame spatial resolution being high enough and tube phosphor time constant being slow enough is such that artifacts in the sensor response are not excited.

The techniques for theater projection of video signals offer potential to overcome these limitations by employing a video signal to modulate a deformable reflective surface for radiant energy produced by an intense source (over 4000 lumens). The xenon arc lamp is used as a source for visible or near infra-red energy, but the Nernst glower can generate intense radiant energy over the band of interest for a number of infra-red sensors. A key issue for further examination are the compatibility of the method of generating the deformable surface (scanning an oil film with an electron beam modulated by the video signal) with the longer wavelength energy from the glower and the spatial and temporal resolution necessary to avoid artifacts with the sensor and its signal processing.

Fortran Server -- This computer is provided to handle the portions of the simulation problem that are either difficult to implement on the specialized computers or have already been implemented in Fortran and represent an investment that either cannot be replaced (nobody really understands the "rats nest of old code" but the experts believe the results) or the cost in time to re-program the code is out of the question. These types of problems can involve real-time processing that would be best handled by the simulation server but due to the above mentioned reasons cannot be re-programmed, but they are more likely to impact the initialization phase rather than the real-time mode and typically are involved in the generation of aerodynamic tables, mass properties, atmospheric properties, fuse function, warhead effects, etc. A number of this class of computing problems have been coded for the Cray supercomputers, taking advantage of the high-speed and high-precision to numerically evaluate Lagrangian or Eulerian integration of non-linear partial differential equations over finite element grids. These codes will probably run on the host MicroVAX but the problem is that they are long running on a general purpose computer and become the principal delay in the use of a very high performance simulation system. The addition of a high-speed high-precision Fortran server can have a major impact in the turn-around time for setting-up runs on the system. The requirement is to include a computer that runs "Cray code" in a reasonable elapsed time (and a reasonable cost) has high-speed open bus I/O, on-line control, and is compatible with the VMS nature of the communication server. A significant feature of the CTSS operating system for Cray computers is a process recovery feature. This allows special-user processes to seize the processor for high-priority runs and then return control to numerous other users with no loss of data. An important aspect of the Cray architecture is the superposition and integration of a scalar (SISD) and an array (SIMD) processor on the same bus and memory structure. Several computer manufacturers offer systems that have been designed along these lines with some actually copying the Cray instruction set architecture but implementing the hardware on more affordable electronic components that can be operated in the laboratory environment. The SCS40 from Scientific Computer Systems meets these requirements as its architecture is a clone of the Cray XMP/24, it executes public domain software, and operates as an attached processor to a VAX host.

Graphics Server -- On-line high-resolution graphics displays of both simulated and hardware-in-loop variables are produced by an Evans & Sutherland PS350 graphics station which performs graphics computing in an attached processor to the VAX host. However with the Aptec communications server this graphics station is actually attached to an Aptec programmable controller that mimics the VAX interface in "transparent mode" and provides data at the limit of the E&S interface during "program mode". Like the other systems attached to the Aptec the PS350 is actually a complex of processors working together and dedicated to a particular type of processing. The task of generating on-line graphics and subsequent hard-copy of the display is simplified by E&S graphics support software. This allows the use of the graphics control processor during initialization and termination of a real-time run and bypasses this processor during the real-time portion to improve throughput. This graphics control processor is commanded by Fortran subroutines from a support software library that E&S provides for VAX host computers. Subsequent hard-copy of the on-line display or off-line post-run graphics hard-copy of data saved on the run log-file can be easily accessed by multiple users via network communications on the VAX host computer.

Telemetry Server -- Collecting data from the test hardware-in-the-loop requires telemetry processing. A Fairchild Weston EMR 8715 on-line telemetry processor provides a modular capability to gather selected variables from the telemetry stream, tag the data quality, smooth the data, and transform the values to engineering units on the fly. Thus the on-line data logging can include hardware based data combined with simulated data so that direct comparisons and causality can be examined. The EMR 8715 is a complex of modular processors than can be flexibly re-configured for a variety of formats and data rates, with parallel or pipelined processing. This system is supported by MicroVAX host resident software that allows the user to dynamically configure the hardware and on-line processing.

IMPLEMENTATION

Figure 3 presents an element interconnect block diagram of the simulation system. The upper right hand portion of the diagram illustrates the MicroVAX host computer and its associated open bus peripherals. Note that the AD100, Aptec, and SCS40 all have parallel interfaces. The AD100 interface connects the host to the supervisor processor for initialization and on-line control of the AD100. The SCS40 interface is a DR11 emulator used for concentrated terminal I/O and file transfers to the host. The Aptec host I/O controller (HIA) is a Unibus interface that operates through a Unibus to Q-bus converter in a Unibus BA-11 module. The Aptec computer consists of the row of I/O controllers (DIA's, DIP's), connected to both the Unibus and the Aptec data interchange bus. The Aptec high-speed scatter/gather access memory (MEM) appears to the VAX host as an RMS disk device and data is stored here during initialization by the host using VAX Fortran statements for file output. The STAPLE code device drivers and procedures are executed by an interpreter in the local memory of the I/O controllers during the real-time mode of operation. The I/O controllers also have micro-coded procedures stored in their local memory that are called by the STAPLE procedures to perform the time intensive functions, so that high throughput can be achieved.

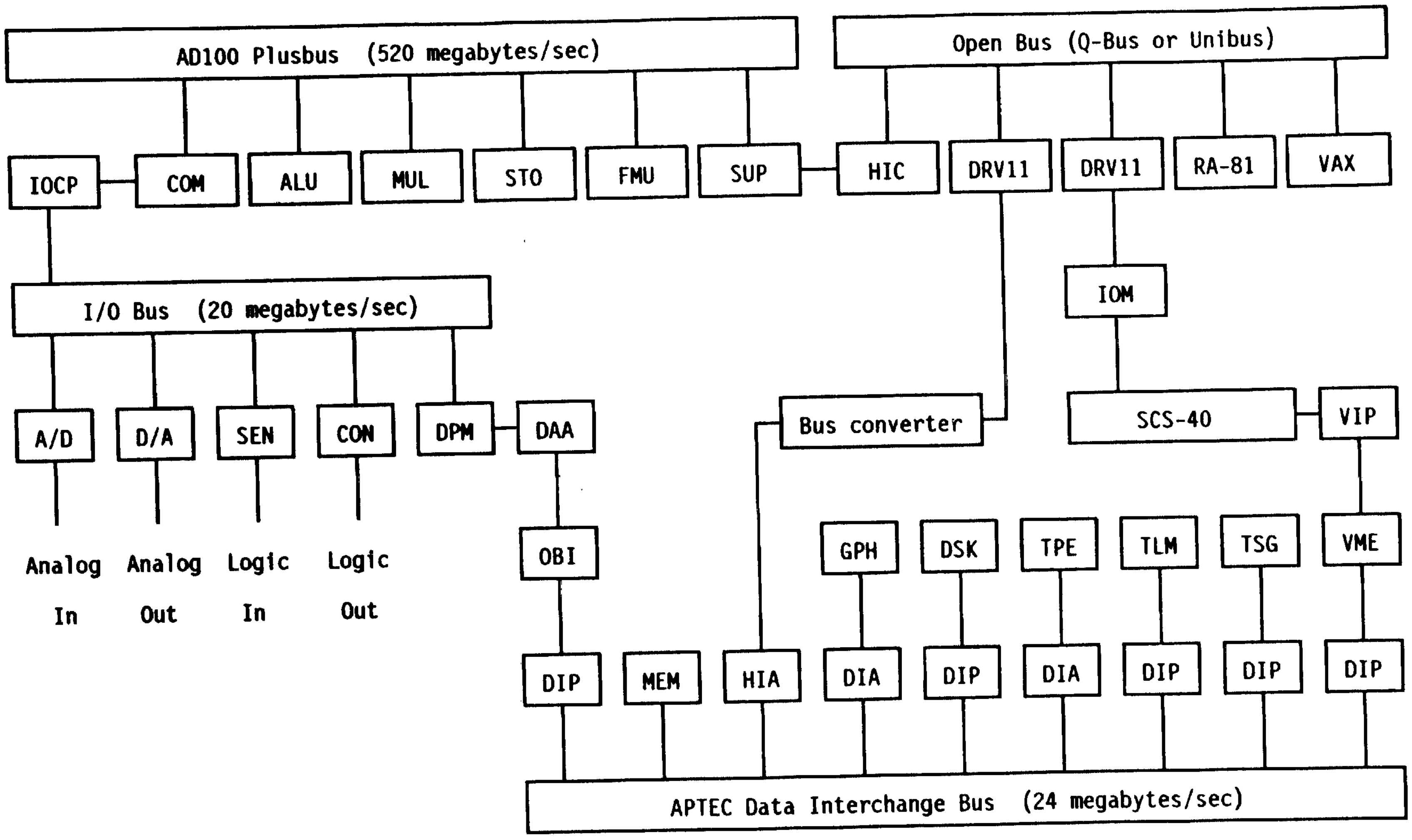

Figure 3. Simulation Interconnect Block Diagram

The I/O controllers connect to the specialized elements of the simulation system through either a DEC Unibus interface or an Aptec private bus interface. The private bus interface can be driven at nearly four times the data rate of the Unibus, but the throughput depends on the total chain of devices in the path, so the Unibus interface may be adequate if the port controllers on the specialized elements are not able to support the higher data rate. The IOCP and DPM on the AD100 and the Ibis disk controller can both support private bus rates, whereas the GPIO controller on the PS350 supports Unibus rates. The IOM controller on the SCS-40 supports Unibus rates whereas its VIP processor supports VME bus I/O and private bus rates. The EMR 8715 (TLM) and the VLDS (TPE) will support either Unibus or VME bus interfaces, but the data rates are limited to that of the Unibus.

The maximum I/O rates are required for data transfer from the AD100 and EMR 8715 to the Ibis disk in the real-time mode, as this is the data logging path with the objective of saving as much data as possible. Data transfer to the PS350 (GPH) and Pixar (TSG) are much lower with the PS350 requiring at most 24 data sets (each involving five 16 bit transfers) every nth frame of the real-time run, where n depends on Nyquist sampling considerations for the particular simulation. The Pixar, when used for on-line real-time target scene simulation requires the following slow moving parameters from the AD100 (they are functions of the inertial position of the scene elements): 1) relative range, target aspect angle, and two inertial reference angles from the target; 2) two inertial reference angles from the background; 3) relative range with one inertial reference angle from each countermeasure that can affect the sensor; 4) atmospheric transmission coefficients for each of these elements.

Figure 3 also presents a block diagram of the ADI interface to the Aptec. The ADI high speed adapter (DAA) interfaces the dual-ported memories (DPM) in the I/O rack to an Aptec open bus interface (OBI) which in turn interfaces to an Aptec high-speed DIP programmable controller. This path can support the high speed private bus data rates. Figure 4 displays a block diagram of the modules of FORTRAN, ADSIM, ADRIO, and STAPLE code that control the execution of the combined system and the flow of data. Frame data is buffered in the COM memory by ADSIM region sync5 code and flags control the transfer of the frame buffers to larger buffers in the DPM by ADRIO main-loop code executing on the I/O control processor (IOCP). Initialization of the communication between the AD100 and IOCP is performed in ADSIM region sync3 and ADRIO pre-run modules while termination occurs in ADSIM region terminal module which signals ADRIO main-loop to terminate. FORTRAN sections start and finish control the execution of the ADSIM code and also performs graphics initialization and termination for the PS350 in the transparent mode.

Figure 5 presents a software block diagram of the Aptec STAPLE code for the I/O controllers (IOC's). The Aptec code is initialized by the FORTRAN routine Aptec-init which is called from the FORTRAN section start. Aptec-init initializes the STAPLE programs for the DIA's controlling the DPM and PS350 interfaces as well as the DIP controlling the Ibis disk interface. Data is transferred from ADSIM region sync5 to the STAPLE main-loop code where it is buffered in the Aptec ram memory. The PS350 DIA gather reads specified data from the buffer for display on the screen and the Ibis disk DIP transfers the entire buffer to the disk for long-term storage.

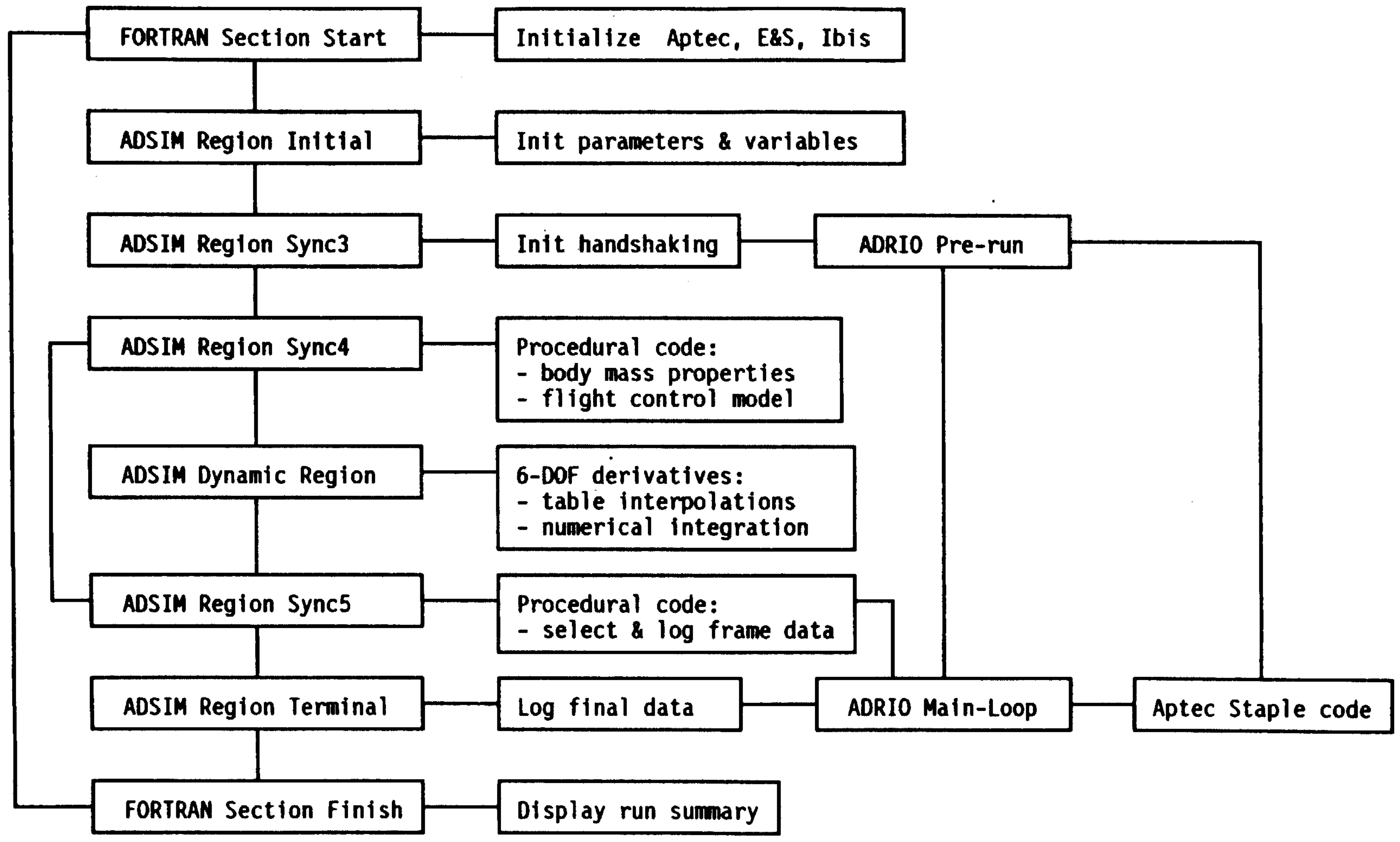

Figure 4, Run Control for the Combined System

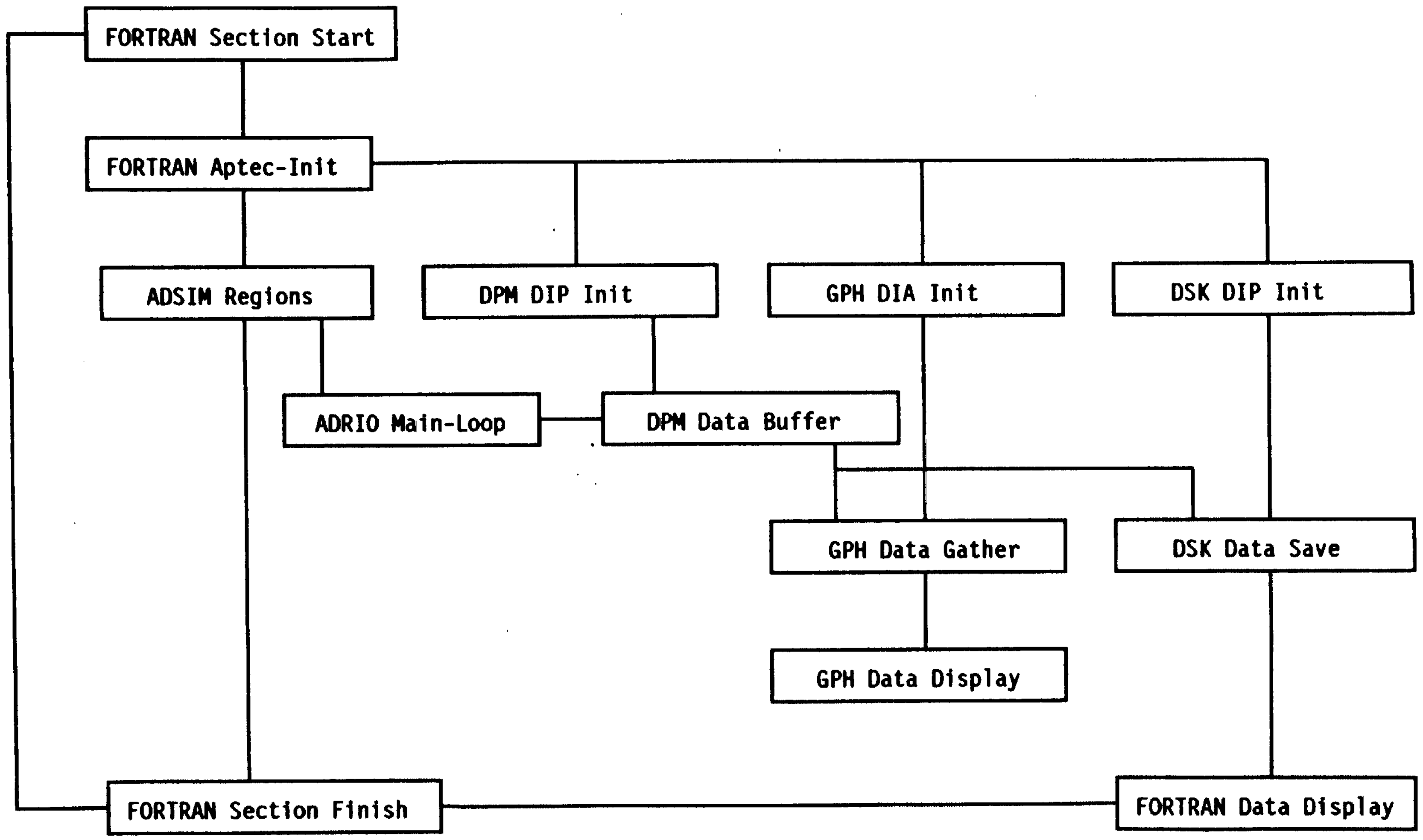

Figure 5. Aptec Software Block Diagram

OPERATIONAL USE

Use of this simulation facility in a typical hardware-in-loop test involves five phases; pre-run, initialization, real-time, termination, and post-run. These phases are described here for illustrative purposes.

Pre-run -- This phase of operation is concerned with the generation of data tables for the interpolation in the AD100 and AD10, and target scene elements for the scene generation in the Pixar. The SCS40 is used to generate the aerodynamic and mass property tables for the ADSIM program on the AD100. These tables are then transferred to the host VAX and processed by the FUNGEN utility provided by Applied Dynamics to generate interpolation tables in the proper format for the micro-coded routines on the AD100. When an AD10 is used to simulate the sensor "front end" the SCS40 is used to generate scene element tables that are also transferred to the host and processed by the INPBDA utility for the assembly coded routines on the AD10. These tables are loaded through the host interface during the initialization phase.

Initialization -- This phase of operation is controlled by the host processor with the I/O controllers for the AD100 and PS350 initially in the transparent mode. The host attaches and loads the AD100 through the Q-bus supervisor interface, whereas the PS350 is initialized with Fortran subroutines through its parallel interface. The final step of this phase is to open Files-11 format data files in Aptec memory and on the Ibis disk for subsequent data storage during the real-time phase, and write files in Aptec memory for the STAPLE procedures and control parameters (addresses, pointers, word counts, offsets, scaling factors, etc.) that the I/O controllers require in the real time mode. Control is then transfered to an ADRIO program in the IOCP of the AD100, which monitors the AD-I/O equipment for the real-time start signal that initiates the execution on the AD100. The I/O controllers then begin processing their real-time programs, which check for control flags to change value and indicate the availability of data to be handled.

Real-time -- The IOCP starts the real-time run of the AD100 when it receives the real-time start signal from an external switch attached to the AD-I/O. The AD100 processes a specified number of simulation frames and sets communication control flags with the IOCP so it can move data from COM memory to the DPM of the AD-I/O. When a buffer is ready for transfer to the Aptec, control flags are set for the I/O controller attached to the DPM. This controller can then move data from the DPM buffer to an Aptec memory buffer and then set flags to indicate that this buffer is ready for the controller attached to the PS350 GPIO to gather-read data from this buffer. The PS350 controller can then scale and offset the selected data, according to the parameters stored in Aptec memory, and transfer this data to the PS350 main memory for on-line display. Thus the AD100 controller moves data in response to ADRIO generated control flags and sets flags for the PS350 controller to select, format, and transfer graphics data, and also for the Ibis disk controller to transfer the entire buffer to the log-file on the disk.

Termination -- Termination results from either an external switch on the AD-I/O or the end-run condition being satisfied on the AD100. This results in the present simulation frame being the last frame to be generated on the AD100 and the execution of the "terminal" region of ADSIM code and the "finish" Fortran section on the host. The terminating phase places the last frame in the COM buffer and sets flags so that the IOCP can transfer the terminal parameters to the DPM and signal the Aptec AD100 controller that the another buffer is ready for transfer. The IOCP will then wait for the Aptec to signal it has completed the handling of this last buffer before it signals the Aptec to shut-down processing. This avoids a pre-mature shutdown and loss of the last buffer of data. The Aptec controllers then return to the transparent mode for the initialization of the next run.

Post-run -- This phase of operation is concerned with the analysis and interpretation of on-line graphic data and the generation of off-line additional graphic and tabular data from the log-file to document the test. The log-file may then be archived to the VLDS tape for long-term storage, thereby providing adequate space on the Ibis disk for the generation of large contiguous block files that are necessary for the high throughput real-time data capture.

SUMMARY

Phased implementation can be accomplished by initially integrating the AD100, AD10, and PS350 into the VAX host, then adding the Aptec controller, Ibis disk, motion simulator, and SCS40. Initial operation for application software development typically can proceed prior to obtaining real-time I/O capability with the Aptec. Then as test hardware availability dictates, the simulation of hardware function is replaced by physical operation and the simulation equipment concentrates on more detailed models of the environment that the hardware experiences.

The phased insertion of sensor hardware into the loop certainly depends on the nature of each particular sensor, its signal processing, and also on the mission scenarios, which dictate target scene generation. It may prove impractical to simulate each detector's response for sensors with large focal plane arrays of detectors, and the logical second step for this type of system may be target scene generation for the optics and focal plane. This may also prove to be the case for systems involving reticle modulation of the target scene. In the same sense, when surrogate detectors are not credible, low temperature background and target scenarios can prove difficult to present to sensors, so that scene generation for these scenarios may not be practical.

REFERENCES

Applied Dynamics International. Dec. 1986. "Applications Summary, Structural Dynamics Benchmark Simulation". 3800 Stone School Rd., Ann Arbor, MI 48104.