MICROCOMPUTER BASE FOR CONTROL LOADING

DR. F. GERRY ALBERS

School of Engineering
University of Dayton

INTRODUCTION

Simulator designers have been faced with two familiar problems throughout the digital age of simulation; namely, framing time crunch and discrete system anamolies associated with models of analog systems. This project has been oriented to impact both of these problem areas as well as to produce a piece of gear suitable for general simulator usage. The primary rationale for conducting this research has been to distribute the intelligence of the simulator to points where it is needed and thus relegate the host computer to the role of a system manager. The control loading task was selected because of its suitability for distributed processing and because of its need for frame rates higher than the nominal 15 to 30 frames-per-second simulator rate. The U.S. Air Force[1] specified that the results of this effort must contain data from which it can write specifications and select future simulator configurations.

THE CONTROL LOADING TASK

Pilots have always and, no doubt, will always complain that the simulator doesn't quite "feel" like the aircraft. This tendency for precise control on the part of the pilots is often offset by the astute engineer who taps on the side of a control cabinet and then says "How's that feel now?"

The "feel" which the pilot experiences in his control system is usually composed of six independent forces: spring, breakout, travel limit, viscous friction (damping), Coulomb friction, and velocity limit. The first three forces are functions of displacement whereas the later three are velocity functions. Occasionally inertia, which is an acceleration term, is also required for complete control synthesis. Another indispensible parameter is the deadband which the pilot experiences around a variable trim setting. In addition, the airframe may very well influence this "feel" through actuator, bobweight and/or control surface force feedback.

The selected hardware configuration for this project includes a force-type control loader equipped with position and velocity transducers as shown in Figure 1. A microcomputer is employed to calculate the various forces mentioned above based on inputs from the control loader's transducers and selected parameters from the "host" computer, a Honeywell 316. The control loader selected for the task was the McFadden Electronics 392A[2] 3-axis control loader and the microcomputer was an Intel System 80/20. In order to provide the project with direction and credibility, the A-10 was selected as the study aircraft. This selection was made primarily on the basis of the availability of suitable control loading data.

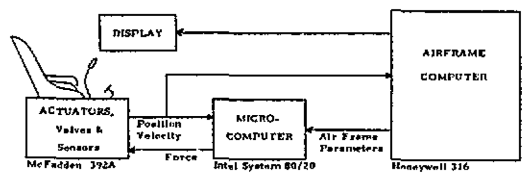

Figure 1. General Configuration Diagram

PARAMETRIC ANALYSIS

Each of the individual forces in some way models an element of the aircraft control system. These forces are, in general, additive as shown in Figure 2, even though they may be the result of nonlinear processes. However, the total solution requires that these forces be logically connected to accurately represent the fully integrated control system.

---

[1] Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio.

[2] Control Loader was loaned at no-cost to the University of Dayton or the U.S. Government.

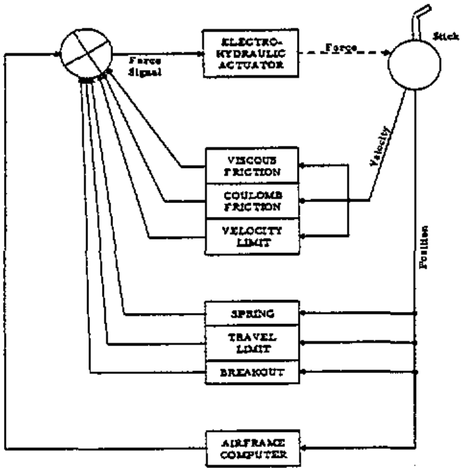

Figure 2. Force Component Diagram - Single Axis

For example, consider the forces generated for a single axis (e.g. pitch control) which has a deadband due to rigging slack at the control stick. The pilot would therefore experience nearly zero force within the deadband which is described as follows:

$$F_{TOT} = [F_{SP} + F_{VI} + F_{VO} + F_{BR}]\,U(DB) + F_{TR} + F_{VEL} + F_{AC} \qquad (1)$$

where the above terms stand for total, spring, viscous, coulomb, breakout, travel limit, velocity limit and aircraft related forces, respectively. The unit operator "U" indicates that the quantities within the brackets are nil in the deadband.

As an alternate example, consider a case where the rigging slack is remote from the stick. The Coulomb friction due to pulley drag may then be placed outside the influence of the deadband with the resultant force equation

$$F_{TOT} = [F_{SP} + F_{VI} + F_{BR}]\,U(DB) + F_{CO} + F_{TR} + F_{VEL} + F_{AC} \qquad (2)$$

These two examples represent wiring differences in an analog system but only software differences in a digital system, an important consideration in a research or development environment. For the A-10 model, the deadband is very small but follows the format of Equation 1. The complete pitch force equation then becomes

$$F_{TOT} = -\{\underbrace{K_S X_D}_{\text{spring}} + \underbrace{2.5\,\text{sign}\,(X_D)}_{\text{breakout}} + \underbrace{1.5\,\text{sign}\,(\dot{X})}_{\text{coulomb}} + \underbrace{64\,\dot{X}}_{\text{viscous}}]\,\underbrace{U(DB)}_{\text{deadband}} - \underbrace{K_T X_T}_{\text{travel}} - \underbrace{(3\,N_Z + .026\dot{q})}_{\text{bobweight}} \qquad (3)$$

where X_T is the displacement (in inches) past a travel limit, X_D is the displacement past a deadband limit and $\dot{X}$ is the velocity in inches-per-second. N_Z and $\dot{q}$ are the normal and angular accelerations from the airframe computer which produce the control system bobweight effects. K_S represents five nonlinear spring coefficients (breakpoints) which characterize the A-10. Similarly, K_T represents a large gain coefficient which produces the large feedback forces for small amounts of travel (X_T) past the limit. The force is calculated in pounds and, of course, requires a sign change to produce the required opposing force. Equation 3 is obviously stylized and requires special handling due to the nonlinearities involved and must be properly scaled. The nonlinearities are discussed below and the microcomputer processing, which was performed entirely in integer format, is discussed in the software section.

In order to handle the nonlinearities of such a system, the frequency response of the control loader must be considered. The nominal 30 frames-per-second for digital simulators is usually sufficient to make the pilot believe that he is flying in a parallel, analog world as evidenced by his visual displays. However, the pilot's tactile mechanism is capable of much higher frequency response. Empirical studies conducted on the McFadden control loader revealed frequency components of 1000 Hz and higher. As one might expect, these components are experienced at the breakout and travel limits where forces suddenly change. To satisfy the ground rules of information theory put forth by Shannon, the microcomputer should ideally be framing at a 5000 Hz rate or better. However, this project demonstrated that quite acceptable results can be had at 120 frames-per-second by judiciously choosing compensation schemes. That is, by optimizing lead, lag, and gain coefficients in the individual component force calculations relatively sharp breakouts and stops were obtained.

Various techniques were attempted to achieve the proper compensation, including a Tustin recursion method to approximate the desired transfer function. However, the final product was a result of an educated cut-and-try effort. For example, consider the travel limit force component which can be expressed mathematically as a recursion relationship

$$F_{TR} = K_T[X_T(N) + X_T(N-1)] + K_L\dot{X} \qquad (4)$$

In the digital implementation, the needed lag term is obtained by employing both the present frame value for travel limit displacement $X_T(N)$ and the previous frame value $X_T(N-1)$. The gain is controlled by the value of K_T and the needed lead term is obtained from the available velocity signal $\dot{X}$ which is modified by the constant K_L. The unscaled magnitudes of K_T and K_L and the number of terms in the recursion portion of the lag component were determined empirically with the final values set at $K_T = 6$ and $K_L = 2$. The breakout term of Equation 3 required a slight lag compensation augmented by a lead term ($K_B = 0.3$) and was empirically structured as

$$F_{BR} = 2X_D + K_B\dot{X} \quad \text{for} \quad 2X_D < 2.5 \qquad (5a)$$

$$= 2.5\ \text{sign}\ (X_D) + K_B\dot{X} \quad \text{for} \quad 2X_D \geq 2.5 \qquad (5b)$$

Similarily, the Coulomb friction term required compensation to offset a tendency to "dither" because of its dependency on the sign of the velocity. In this case a simple lag was implemented by employing the velocity value as follows:

$$F_{CO} = \dot{X} \qquad \text{for} \quad |\dot{X}| < 1.5 \qquad (6a)$$

$$= 1.5\ \text{sign}\ (\dot{X}) \quad \text{for} \quad |\dot{X}| \geq 1.5 \qquad (6b)$$

THE HARDWARE

One of the important driving factors in this study was that the hardware must be composed of off-the-shelf components to the greatest extent possible. A preliminary analysis showed that four primary functions must be supplied by the hardware; namely, central processing, analog input, analog output, and high-speed mathematics. Figure 3 illustrates the hardware configuration which employs the System 80/20 components from Intel Corporation. Each of

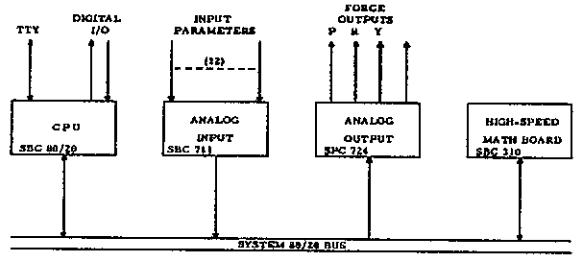

Figure 3. Hardware Configuration

the four blocks represents a bus compatible circuit board which resides in the System 80/20 chassis. The only hardware modifications performed were wiring options which are incorporated on the individual boards for customizing purposes.

The CPU is an 8-bit processor with 2K of RAM (random access memory) and with 8K of ROM (read only memory) and has a clock cycle time of 490 nanoseconds. The analog boards convert analog signals with 12-bit (0.025%) accuracy and were configured to handle conventional analog signals ($\pm$10v) and offset binary (2's complement) digital values. The high-speed mathematics unit provides both integer, and floating point operations with appropriate 16 and 32-bit accuracies. Additionally, the CPU provides a teletype (TTY) interface for changing equation coefficients and digital parts for trim switch inputs and for frame-time monitor outputs.

The interfaces with the control loader and host computer (as shown in Figure 1) were entirely analog due to contractual requirements for generality. An ideally flexible system would have a digital host/microcomputer interface for parameter passing and software downloading. Each of the three axes required four input parameters to satisfy its force equation (e.g. Equation 3), stick displacement and velocity, airframe normal and rotational accelerations. The analog input board, which has a capacity for 16 analog inputs, converted the required 12 analog values in approximately 800 microseconds. Three of the four available analog outputs were employed to drive the respective control loader force inputs for each axis, pitch, roll, and yaw.

THE SOFTWARE

The entire control loader software was written in the Intel high-order microcomputer language called PLM, with the exception of the analog-to-digital conversion routine which was performed in assembly language. The programs were structured into a utility module and a simulator module. As the name implies the utility module provided system support in the form of calibration and test routines, system interface procedures, and system initialization. The simulator module contains the frame rate generator, simulator initialization routines, the analog scan/convert routine, and the simulator equations themselves. The executable machine code was placed in PROM (programmable read only memory).

Perhaps the most unique feature of the software structure is that the calculations were carried out entirely in integer format resulting in a very significant reduction in processing time required for each frame. As seen in the above equations, the only mathematical processes required are addition, subtraction and multiplication. Since both the analog input and analog output boards operate with 12-bit precision, the microcomputer was designed to employ 16-bit precision (two 8-bit words). The PLM language and the high-speed math board both support 16-bit operations of this nature. All parameters were scaled to keep the values within these limits. The equation coefficients were scaled to be either integer values or fractional values less than 1. The fractional coefficients were rescaled upward by multiplying them by 2^{16} (65536). The result of an integer multiplication (a 32-bit value in the high-speed math board) was scaled back down by 65536 by taking only the most significant 16 bits from the result register. The only drawback to this method of processing was the unsigned nature of the multiplication, thus requiring procedures for sign conversion of negative quantities.

CONCLUSIONS

A purely objective evaluation of an engineering system usually involves graphs, strip charts and computer printouts. Thus, the usual static and dynamic tests were carried out with excellent success. However, the last word on a system involving "feel" must be subjective. Those who were able to experience the "feel" of the control loader employing the microcomputer base considered it to be of "good" quality compared to the same system employing the analog base supplied with the McFadden system (which is easily judged as excellent). The microcomputer, of course, was performing additional airframe related calculations which the standard analog system does not. The only difference in quality seemed to be at the stops and breakouts. Due to the limited frequency response of the microcomputer system, the respective gains were held relatively low for stability reasons. This practice resulted in the discontinuities being less "sharp" than the analog based version. On the other hand, the system was judged by most to be suitable for direct application in existing simulators.

RECOMMENDATIONS

This study has clearly demonstrated the feasibility of microcomputer control. The only consequentional recommendation as a result of this study would be to increase processing frame rate by a factor of five to six times. This improvement is within the capabilities of present technology but not exactly "off-the-shelf." The most immediate thought would be to use a 16-bit processor instead of the 8-bit processor but this would result in only a 50% to 60% speed improvement without any other changes. The most significant improvement would be to use a multiprocessor architecture with a processor devoted to each axis. Another dramatic speed improvement could be obtained if a hardware multiply unit was employed which performed signed (as opposed to the present unsigned) integer multiplication. If these three changes were coupled with improvements in microcomputer technology (a daily occurrence), frame rates up to 1000 per second are within reach.

ABOUT THE AUTHOR

DR. GERRY ALBERS is an Associate Professor in Electrical Engineering at the University of Dayton. He is currently performing research in the area of simulator design and technology. He is a former T-38 Instructor Pilot and his professional experience also includes digital system design, system simulation and modeling, engineering education, and private consultation. His most recent research programs include the design of a microcomputer base for control loaders and the system design and math modeling for the Boom Operator Part Task Trainer, an Aeronautical Systems Division simulator design for SAC.