DIGITAL SMOOTHING TECHNIQUES APPLICABLE TO SIMULATOR/TRAINER INTERFACES

F. O. Simons, Jr.
Assoc. Professor
Florida Technological University
Orlando, Florida

R. C. Harden
Professor
Florida Technological University
Orlando, Florida

B. L. Capehart
Assoc. Professor
University of Florida
Gainesville, Florida

INTRODUCTION

The concept of large-scale weapon system simulators for the purpose of training the required highly skilled military operating personnel has been influenced considerably by the advent of the modern economical digital minicomputer which not only lends itself to conveniently handling all the centralized bookkeeping operations required for large-scale simulations, but it can also function to generate dynamic variables for simulated subsystem components. In fact, in almost all cases the latter function will represent a large majority of the digital processing required to service large-scale simulators. Since interfacing techniques can influence the digital processor load for this latter function by several hundred percent, it is of prime importance to exercise careful consideration for the manner in which all dynamic subsystem variables are interfaced. To this end, the purpose of this presentation is to present analog hardware, digital hardware, and digital software interfacing techniques which can be traded off with digital processing requirements to realize economical savings in a proposed simulator/trainer system.

ANALOG HARDWARE INTERFACING TECHNIQUES

The term "analog hardware interface" is used here in the modern context; namely, the hardware may contain logic devices (gates, electronic switches, etc.) but these serve only to operate on analog signals. Usually the analog interface hardware can be considered to be driven by a DAC (digital to analog converter) analog voltage subject to discontinuities produced by digital update inputs. From this discontinuous* analog signal, the function of the analog interface hardware will be to provide a smoothed signal suitable for driving the appropriate input to a given training device.

The "brute force" means of data smoothing is to simply specify an update rate fast enough that DAC output changes will not exceed a required signal resolution. For example, let it

* In actuality, the voltage slopes between levels will exhibit velocity-limiting which is primarily a function of DAC output saturation current levels and DAC output loading.

be required to provide an analog output with a resolution of 0.1% of maximum value. If this variable is derived from a digital system simulation whose maximum eigenvalue, λm, is 10 (1/sec.), then the required update period, T_{ud}, is given by

$$T_{ud} = \frac{0.001}{\lambda m} = 10^{-4} \text{ (sec.)} \qquad (1)$$

whereas the sample period, T, used in the simulation could be approximately

$$T = 0.1 \frac{1}{\lambda m} \qquad (2)$$

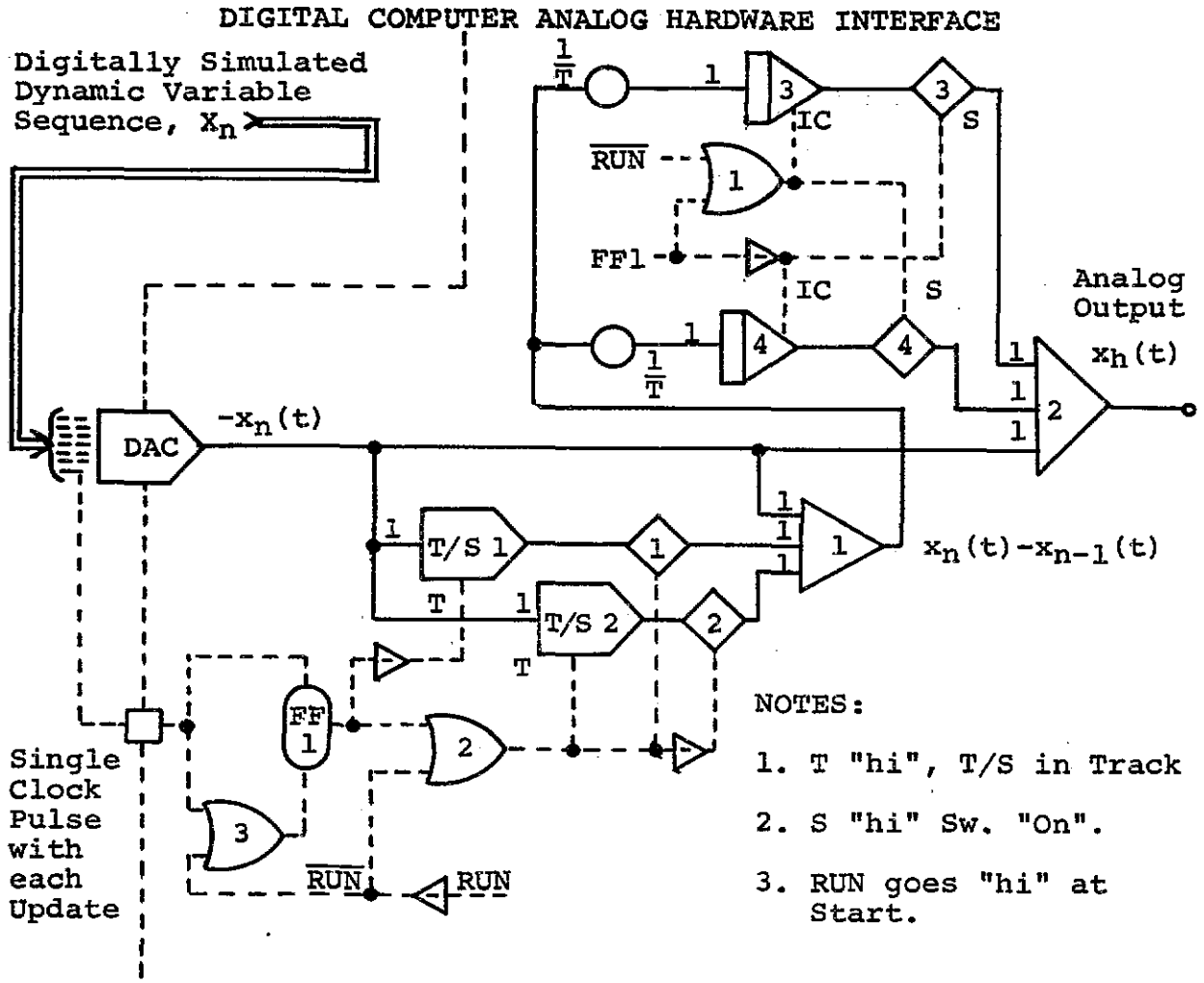
as obtained by invoking the Lomax Conjecture.[1,2*] From (1) and (2), it is implied that the digital simulation needs only to update the computational algorithm every 100th of a second but the DAC output needs to be updated every 100 microseconds. If analog hardware is used for data smoothing, a 100-to-1 processor time savings could be realized. To illustrate, consider two alternate approaches.

The first approach, which is to use standard sample and hold devices (zero-order hold, first-order hold, etc.),[3,4] should be used if delays from the "input" to the output of the analog interface device cannot be tolerated. The basis for designing such an analog interface device is depicted in the analog/hybrid program of Figure 1(a) which was taken from earlier work.[5,6,7] To illustrate how the program functions, an exaggerated form of the input-output functions are depicted in Figure 1(b). In effect, for first-order hold operation, the program (and any device designed on the basis of the program) attempts to predict the variable between sample instances based on the last two samples, and when a sample period (a DAC update) occurs, the output is updated to the new corrected value. The operation of the program in Figure 1 is to generate

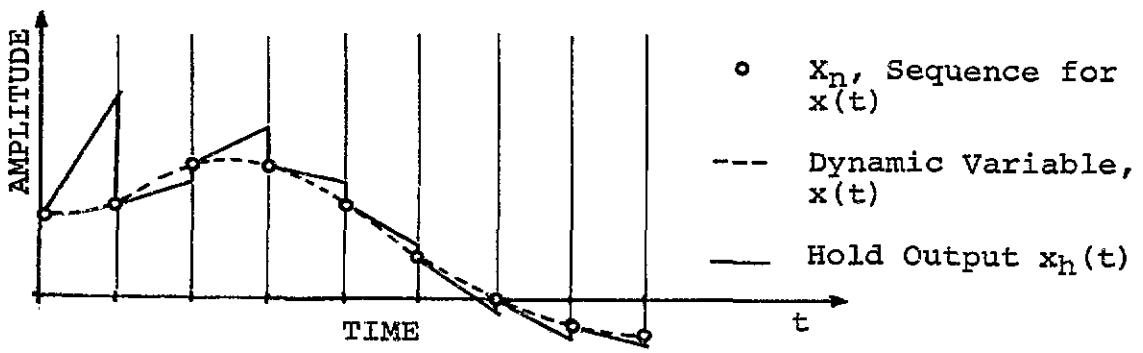
$$x_h(t) = x_n(t) + \frac{x_n(t) - x_{n-1}(t)}{T} t \qquad nT \geq t \geq (n+1)T \qquad (3)$$

which is precisely the first-order hold extrapolation of x_n, the digital number sequence which represents the dynamic variable x(t). RUN goes "hi" when the simulation starts and FF1 (flip-flop) toggles at each update period thereafter. Since FF1 is "lo" initially, TS 1 & 2 (track/store) are in track to insure not drifting to saturation while integrators 3 and 4 are in IC for the same reason. Subsequently, the T/S (integrators) alternately initialize and track (operate) to produce $x_h(t)$ as set forth in (3). A design based on this program would include the ultimate refinements in that speed of operation is only limited by the amplifier velocity-limits. For

* Superscripts refer to references in LIST OF REFERENCES.



(a)



(b)

Figure 1 -- (a) An Analog/Hybrid Program Which Could Be Used To Design a First-Order Hold Analog Hardware Interface, (b) with a Typical Variable Display.

slower variables, the amount of hardware could be greatly reduced.

Obviously, the first approach would result in substantial savings in the digital processing required for the digital simulation variable that was discussed earlier - T_{ud} and T given in (1) and (2). However, one should not be deceived, the full 100-to-1 saving cannot be realized because of the accumulative sample, track, and store errors. Furthermore, the basic process of prediction is differentiation, a noise producing process. With care, a 10-to-1 savings in update rates could be realized which is more than adequate for justifying trade-off considerations between analog hardware interface units with digital processor requirements. It should be noted that such trade-offs should consider analog interface designs based on other extrapolator (sample-hold) forms; e.g., second-order hold, exponential-hold, etc.

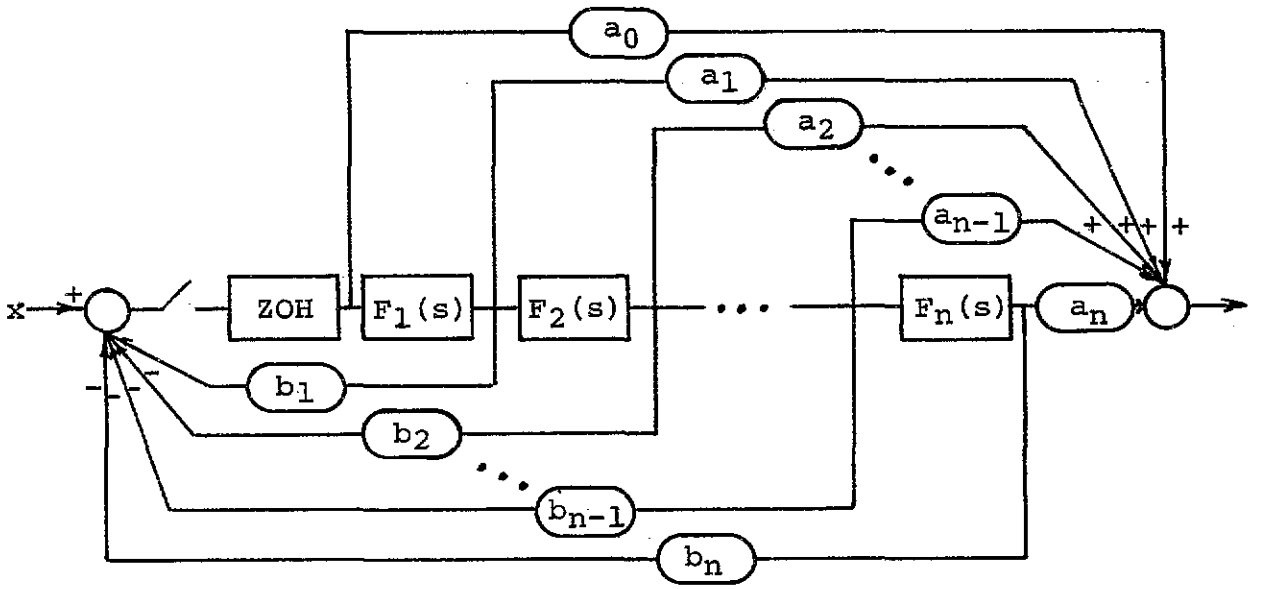
The second approach to analog hardware interface design represents the antithesis to the first in that the hardware functions basically as an integration scheme to perform interpolation. The design of such hardware could be based on the data reconstruction model[8,9] of Figure 2(a) which depicts a completely general interpolation scheme. The type of interpolation is determined by the selection of $F_k(s)$ whereas n determines the order. The model of Figure 2(b), a simple form of 2(a), could serve as a basis for designing a first-order linear data smoothing interpolator with the corresponding variables displayed in Figure 2(c) which is self-explanatory. The form for the first-order extrapolator output of Figure 2(c) is given by

$$x_e(t) = x_{n-1}(t) + \frac{x_n(t)-x_{n-1}(t)}{T}t \qquad nT \leq t \leq (n+1)T \qquad (4)$$

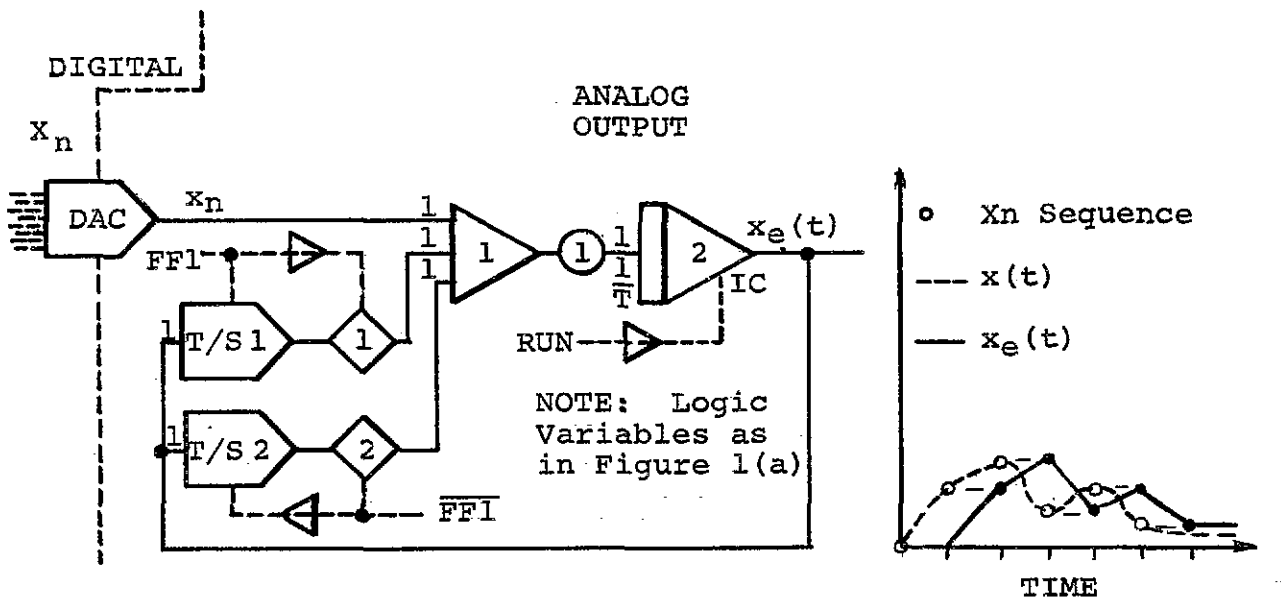
This second approach to analog hardware interface design has the inherent disadvantage that a one update period, T_{ud}, delay cannot be avoided. However, the output experiences no discontinuities such as those present in data extrapolators--compare Figures 1(b) and 2(c). In addition, it may be possible to modify the digital simulation to account for this T_{ud} delay depending on the purpose and origin of the signal. If this correction is possible or if it doesn't matter, then the full savings discussed relative to equations (1) and (2) could be realized; namely, a digital processing saving of 100-to-1 could be realized for servicing a signal with characteristics described by (1) and (2).

DIGITAL HARDWARE INTERFACING TECHNIQUES

Digital hardware interfacing techniques will be defined as those techniques that involve data smoothing hardware with



(a)



(b) (c)

Figure 2 -- (a) The Generalized Data Reconstructor Model, (b) with the First-Order Data Extrapolator Realization and (c) Its Associated Variables.

both digital input and output. To illustrate, consider the hybrid servo-lens position drive system of Figure 3.[10] Assume it is desirable to drive it with the signal whose update parameters are once again described in equations (1) and (2). Such an assumption is reasonable since typically these systems are characterized by high resolution and slow tracking speeds. In the present form the proposed hybrid servo of Figure 3 would drive the lens to track very small digital step input increments. However, the digital hardware in the servo control logic could be easily designed to--according to the assumptions in (1) and (2)--accept digital steps 100 times as large with a 100-to-1 reduction in update rate and simply synchronously apply one one-hundredth of the total step input every 100 microseconds. Such design is practically trivial with modern available IC logic components. To synchronize digital hardware interfacing, one could utilize the interrupt functions in modern digitals or simply exercise care in the timing of service from the digital.

DIGITAL SOFTWARE INTERFACING TECHNIQUES

Digital software techniques as applied to interface considerations which could be used to reduce digital processor loads can be divided roughly into the following classes:

1. The inclusion of simple data-smoothing subroutines, such as linear extrapolation or interpolation, which function only to smooth data updates which of necessity are derived from complex time-consuming computational algorithms. In essence, truncation error in algorithms can be traded off with data smoothing subroutines.

and 2. Precalculated or external input data which are stored for table "look up" and "play-back" which again could make use of data smoothing subroutines.

In addition to the above, one should consider various program languages for implementing these.

SUMMARY AND CONCLUSIONS

In brief, the basis for designing analog hardware and considerations for digital hardware and software that could be applied to interfacing large-scale simulator/trainers, which use the centralized digital processor concept, have been presented. The scope of coverage was of necessity limited by space.

Even with the limited scope of coverage, it can be concluded that the digital smoothing techniques presented here

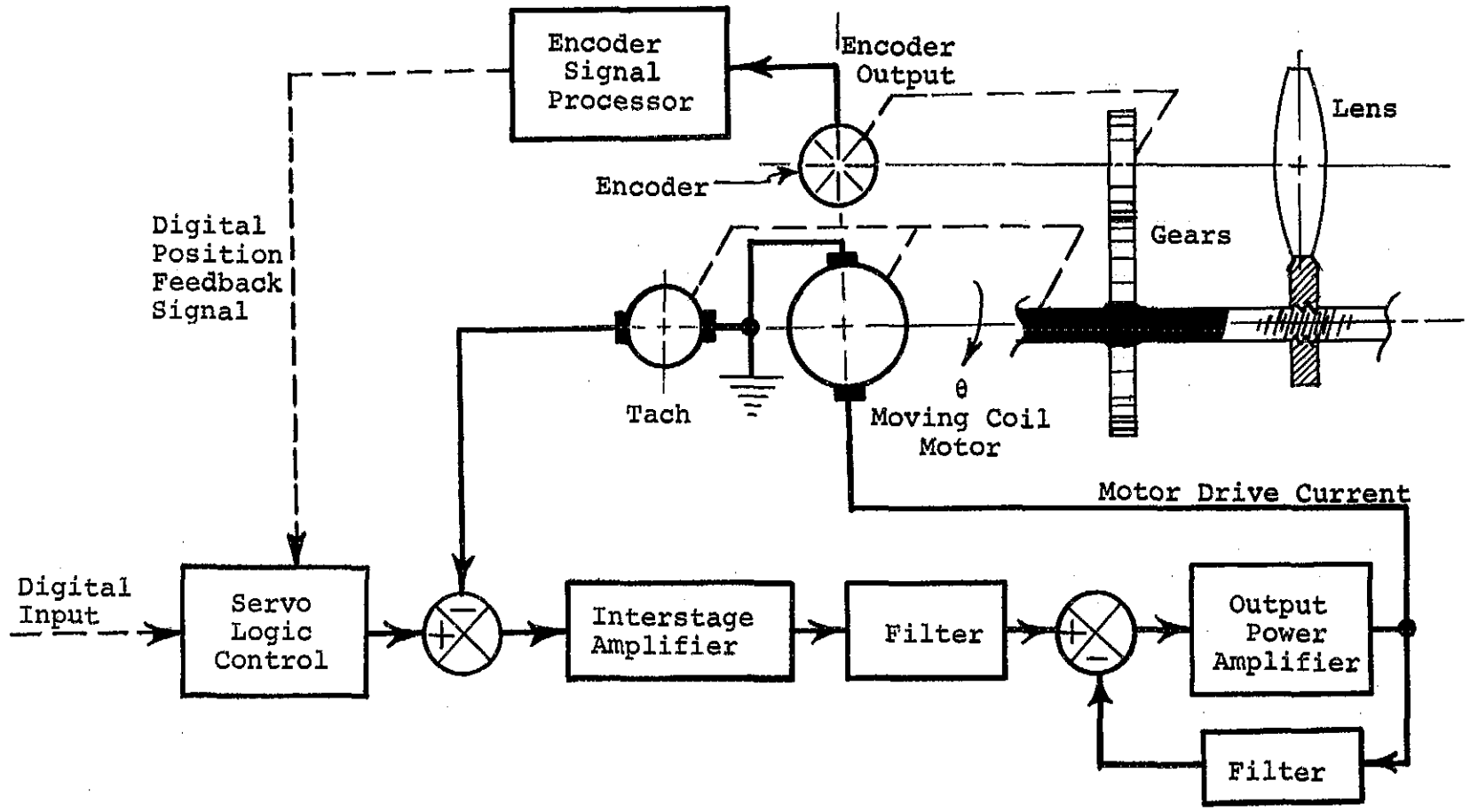


Figure 3 - The Block Diagram Form for a Proposed Typical Hybrid Servo for Device 21A38/2, a Periscope View Simulator

should be considered in detail for any and all large-scale simulator/trainers because:

1. Substantial economic savings are realizable

and 2. Judicious applications of techniques as set forth would permit expanding the capability and scope of present trainers way beyond the realms that are realizable when very little consideration is given to interfacing techniques.

In closing, it is conceded that there must be even other considerations for interfacing--an area characterized by neglect--that would lead to further advantages.

FURTHER RESEARCH

There is very little known or available for the optimization of adaptive processes for MIMO (multiple-input, multiple-output) sampled-data systems which are serviced by a centralized digital processor. The authors have made some inroads in this area and one proposal for research is at present pending. The results of such a study would be directly applicable to the problem addressed by this presentation.

LIST OF REFERENCES

1. Lomax, Harvard, "An Operational Unification of Finite Difference Methods for the Numerical Integration of Ordinary Differential Equations", NASA Technical Report R-262 (May 1967).

2. Capehart, B. L., and Simons, F. O., Jr., "Analog Computer Technique for Estimating Sample Times for Digital Simulations", Proceedings of the 1972 IEEE Region III Conference, University of Tennessee, April 10-12, 1972.

3. Kuo, B. C., Analysis and Synthesis of Sampled-Data Control Systems, Prentice Hall, 1963.

4. Tou, J. T., Digital and Sampled-Data Control Systems, McGraw-Hill Book Co., 1959.

5. Simons, F. O., Jr., Harden, R. C., and Walter, W. A., "The Continuous vs the Discrete Simulation of Sampled-Data Systems", Proceedings of the Second Annual Pittsburgh Conference on Modeling and Simulation, University of Pittsburgh, March 1971.

6. Simons, F. O., Jr., Harden, R. C., and Walter, W. A., "The Analog Simulation of Sampled-Data Systems with Ideal Samplers", Proceeding of the Southeastern Symposium on Systems Theory, Georgia Institute of Technology, April 1971.

7. Simons, F. O., Jr., Harden, R. C., and Monte, S. J., "Perfected Analog/Hybrid Simulations of All Classes of Sampled-Data Systems", Proceedings of the Fourth Annual Pittsburgh Conference in Modeling and Simulations, University of Pittsburgh, March 1973.

8. Bullock, T. E., and Durling, A. E., "Analog Computer Simulation of Discrete Integrating Operators (or the Modeling of a Digital Computer on an Analog Computer)", System Simulation Conference, Princeton University, 1969.

9. Bullock, T. E., and Durling, A. E., "A Unified Method for the Reconstruction of Sampled-Data", Submitted to IEEE Transactions on Computers, 1972.

10. Simons, F. O., Jr., "High Performance Hybrid Servo Positional Systems Study", Technical Report: NAVTRAEQUIPCEN 72-C-0129-1, Orlando, Florida, October 1972.

ABOUT THE AUTHORS

DR. FRED O. SIMONS, JR. (S'59-M'65) was born in Heidelberg, Mississippi on July 7, 1937. He received his B.S.E.E. degree in August 1960 from Mississippi State University, and his M.S. and PhD degrees in Electrical Engineering from the University of Florida, Gainesville, Florida in 1962 and 1965, respectively.

From August 1960 until June 1961 he was an instructor with the Department of Electrical Engineering at Mississippi State University. He later joined NASA at Huntsville, Alabama for 3 months. From 1965 to 1971 he was with the University of Florida's Graduate Engineering Education System (GENESYS), Orlando, Florida as an Associate Professor of Electrical Engineering. From September 1971 to September 1972 he was an Associate Professor of Electrical Engineering at the University of Florida's Eglin AFB Center. Since September 1972 he has been an Associate Professor of Engineering with Florida Technological University.

Dr. Simons is a member of Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, and Institute of Electric and Electronic Engineering.

DR. RICHARD C. HARDEN (S'56-M'61) was born in Miami, Florida on August 11, 1920. He received a B.M.E. (with honors) degree in 1944, a B.E.E. (with honors) degree in 1956 and his M.S. and PhD degrees in Electrical Engineering in 1957 and 1961, respectively, all from the University of Florida, Gainesville, Florida.

From 1944 to 1945 he was a U.S. Naval Reserve officer on active duty. From 1945 to 1950 he was a rocket and guided missile range engineer at the U.S. Naval Ordnance Test Station, China Lake, California. From 1950 to 1955 he served in the capacities of Chief Engineer at Cape Canaveral launching facility, Deputy Chief of the Optics-Mechanical Department of Technical Systems Laboratory, and as Manager of the Optical Development Lab. and Shops, RCA Missile Test Project, Patrick AFB, Florida. He was a Professor of Electrical Engineering at the University of Missouri at Rolla, Rolla, Missouri from 1961 to 1967. From 1967 to July 1972 he was Professor of Electrical Engineering, and Resident Director of the University of Florida's Graduate Engineering Education System (GENESYS) at Orlando, Florida. Since July 1972 he has been Professor of Engineering with FTU and Director of Florida Technological University's South Orlando Resident Center.

Dr. Harden is a member of Eta Kappa Nu, Phi Kappa Phi, Sigma Tau, Sigma Xi, Tau Beta Pi, IEEE, and ASEE.

DR. BARNEY L. CAPEHART was born in Galena, Kansas August 20, 1940. He received his B.S.E.E. degree in June 1961, his M.E.E. in July 1962, and his PhD in Industrial and Systems Engineering in June 1967, all from the University of Oklahoma, Norman, Oklahoma.

From June 1963 to September 1965 he was a 1st Lt. in the U.S. Air Force and served as Systems Engineer on the 416M Air Defense Computer System.

From June 1961 to August 1961 he was an Electrical Engineer with the Link Aircraft Company of Binghamton, New York. From September 1965 to June 1967 he was an instructor in Electrical Engineering at the University of Oklahoma. From June 1967 to June 1968 he was a member of the technical staff of Aerospace Corporation, San Bernardino, California.

From June 1968 to September 1972 he was an Assistant Professor of Industrial and Systems Engineering, University of Florida, GENESYS-Orlando. From September 1972 to June 1973 he was an Associate Professor of Industrial and Systems Engineering at the University of Tennessee at Knoxville, Tennessee. Since September 1973 he has been an Associate Professor of Industrial and Systems Engineering with the University of Florida.

Dr. Capehart is a member of Sigma Tau, Sigma Xi, and the Institute of Electrical and Electronic Engineers.