AIAA 81-4053

Design of Guidance and Control Digital Autopilots

Willy Albanes*
Computer Sciences Corp., Huntsville, Ala.

The miniaturization, availability, and low cost of digital hardware has made possible a digital autopilot package for antitank homing missiles comparable in size and cost to existing analog controllers. This paper discusses the design, implementation, and validation of a digital guidance and control autopilot based upon current sampled-data techniques and methodologies. This controller is considered state-of-the-art technology, performing as well as or better than analog versions, and having been actually flown in small homing antitank missiles. It performs the classical flight control autopilot functions of attitude stabilization and gyro filtering, and in addition provides actuator control, seeker signal filtering, and guidance shaping. The digital guidance and control autopilot uses a laser seeker to automatically pursue and impact its target without operator guidance intervention.

Nomenclature

A,B,C,D	= matrices of system parameters
$F(w)$	= actuation compensator
G	= missile dynamics matrix
$H(s)$	= zero-order hold dynamics
J_t	= scheduled gain
K	= transfer function coefficient vector
$P(s)$	= actuator dynamics
$R(w)$	= attitude compensator
s,w,z	= complex variables
T	= sampling period, s
t	= time since launch
u	= input
x	= state vector
y	= output vector
z	= z-transform
Φ	= state transition matrix
γ	= input weighting matrix
ω	= real frequency
θ	= pitch angle
θ_c	= commanded pitch angle
φ	= roll angle
φ_c	= commanded roll angle
ψ	= yaw angle
ψ_c	= commanded yaw angle

Introduction

A DEFINITE need has existed for smaller, more accurate antiarmor missiles. Digital technology promises the capability of enhanced missile guidance and control, and thus increased accuracy.

Digital autopilot (DAP) design techniques are essentially of two forms: digitized analog and discrete domain. Since each method has its own advantages and disadvantages, two preliminary DAP's were designed and then evaluated via digital simulations. Both DAP's performed as well or better than the analog version. However, experience in developing the T6 missile autopilot has demonstrated that the w-plane (discrete domain) method of design has distinct advantages over the digitized analog method when a microprocessor is used as the autopilot controller.

Prior to extensive hardware development, a robust basic DAP was designed, and its characteristics investigated to define baseline system performance. The final DAP obtained was a true w-plane frozen-time point sampled-data control theory discrete domain design (D^3). It uses complex-plane transformations as an inherent part of the design process. The design was evaluated using six degree-of-freedom (6-DOF) and hardware-in-the-loop (HWIL) simulations, and validated through a limited number of test flights.

System Definition

The T6 missile[1] was picked to be the testbed for the DAP design effort. It was chosen due to its basic capabilities, known aerodynamic properties, modular arrangement, and low cost.

The T6 missile airframe is a modular test vehicle designed to accept a variety of seekers and autopilots. The 6-in. diameter, 70-lb body of the T6 yields performance comparable with that expected in future weapon systems. The missile can be either ground or helicopter launched. A typical direct fire (lock-on to target prior to launch) scenario is depicted in Fig. 1. Once fired, the missile automatically homes in on its target without operator guidance. The missile has a range of less than 6 miles.

The autopilot performs the classical flight control autopilot functions of attitude stabilization and gyro filtering, and, in addition, provides actuator control, seeker signal filtering, and guidance. The DAP controls four fins so as to successfully reach the desired state, or impact the designated target. The laser seeker generates pitch/yaw line-of-sight (LOS) rate signals that are used by the DAP in proportional navigation guidance (PNG). Two degree-of-freedom attitude gyros provide signals for pitch, yaw, and roll stabilization. Fin position potentiometers are used for the actuator servo loops. The complete autopilot then consists of a triple nest of multivariable loops, as shown in Fig. 2. Analog-to-digital (A/D) converters are used as DAP inputs, whereas pulse width modulator (PWM) controlled fins respond to DAP outputs.

Prior to the DAP design, an accurate system math model was developed. This model resulted in a highly nonlinear, time-varying, complex set of differential equations. By assuming small angles and frozen-time or operating-point dynamics, linear perturbation equations were obtained which significantly reduce the autopilot design effort. Figure 3 shows the simplified sampled attitude control block diagram. Linearized missile transfer functions used in Fig. 3 were obtained by 24 frozen-time points along projected trajectories. These are shown in Table 1.

Received Aug. 12, 1980; revision received Nov. 19, 1980. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Principal Engineer, Defense Systems Division; currently with the U.S. Army Missile Command, Army Missile Laboratory.

Table 1 Missile transfer functions

	Launch
$P(s)=26937.9$	$\dfrac{1}{(s+0.07360)(s+108.8)}$
$G_{11}=1.22691$	$\dfrac{(s+250.0)(s-0.06909\pm j4.7806)}{(s+0.72891)(s+0.16726)(s-0.33579\pm j4.6286)}$
$G_{31}=89.6201$	$\dfrac{(s-6.4686)}{s(s+0.72891)(s+0.16726)(s-0.33579\pm j4.6286)}$
$G_{22}=35.0873$	$\dfrac{(s+0.34769)}{s(s+5.9060)(s-5.4373)}$
$G_{13}=-16.6103$	$\dfrac{(s+290.84)}{(s+0.72891)(s+0.16726)(s-0.33579\pm j4.6286)}$
$G_{33}=35.0873$	$\dfrac{(s+2.4975)(s-1.2918\pm j1.6938)}{s(s+0.72891)(s+0.16726)(s-0.33579\pm j4.6286)}$
	Max-Q
$P(s)=26937.9$	$\dfrac{1}{(s+4.116)(s+188.3)}$
$G_{11}=27.5052$	$\dfrac{(s+250.0)(s+17.622)(s-14.065)}{(s+18.0264)(s+1.8855)(s+0.03006)(s-14.481)}$
$G_{31}=35668.5$	$\dfrac{(s+1.0851)}{s(s+18.0264)(s+1.8855)(s+0.03006)(s-14.481)}$
$G_{22}=532.707$	$\dfrac{(s+1.9486)}{s(s+22.103)(s-19.50)}$
$G_{13}=301.001$	$\dfrac{(s+1120.8)}{(s+18.0264)(s+1.8855)(s+0.03006)(s-14.481)}$
$G_{33}=532.707$	$\dfrac{(s+5.5990)(s+1.1861)(s-2.1409)}{s(s+18.0264)(s+1.8855)(s+0.03006)(s-14.481)}$
	Terminal
$P(s)=26937.9$	$\dfrac{1}{(s+0.5740)(s+114.6)}$
$G_{11}=6.10580$	$\dfrac{(s+250.0)(s+6.6153)(s-5.0640)}{(s+8.4111)(s+1.0418)(s+0.25060)(s-7.0874)}$
$G_{31}=2445.79$	$\dfrac{(s+0.96586)}{s(s+8.4111)(s+1.0418)(s+0.25060)(s-7.0874)}$
$G_{22}=139.031$	$\dfrac{1}{s(s+0.62474)}$
$G_{13}=145.710$	$\dfrac{(s+524.07)}{(s+8.4111)(s+1.0418)(s+0.25060)(s-7.0874)}$
$G_{33}=139.031$	$\dfrac{(s+5.6529)(s+0.96596)(s-4.5623)}{s(s+8.4111)(s+1.0418)(s+0.25060)(s-7.0874)}$

Autopilot Design Concepts

The task of designing a missile DAP could be accomplished by using either frequency or time-domain tools. Linear-design tools in both frequency and time domains are available to most designers. However, due to the very real problem of system nonlinearities and the formulation of nonlinear constraints, nonlinear analysis and design tools are required. Dominant nonlinearities, such as actuator deadzone relay, are easily treated with describing function techniques in the frequency domain. Input aliasing, folding, and noise-generation effects are easily defined and analyzed with Bode plots in the frequency domain. Thus the DAP analysis is easier to approach using frequency-domain techniques, and this methodology was used here. As a check, results were verified with 6-DOF and HWIL comprehensive time domain simulation.

Two algorithms, representing two different DAP design approaches, were developed for the T6 missile. This airframe has flown numerous test flights using an analog autopilot. The two approaches are: 1) a direct digitization of analog filters, and 2) a discrete domain design using sampled-data control theory. The digitization approach in general requires a fast sampling rate in order to match analog filters response, and uses all of the available microcomputer memory. The D^3 approach, however, permits a relatively slow sampling rate, and results in a more compact program. These two synthesis exercizes will now be summarized as follows.

Digitized Analog Approach

A digitization of an analog autopilot[2] is accomplished by converting each analog filter (see Table 2) to a digital filter in state variable form. The frequency response matching criterion holds degradation of phase and gain margins at loop crossover frequency and below to 5 deg and 1 dB, respectively.

The state variable digitization approach used here was composed of the following actions:

1) Expand each analog filter transfer function into partial fractions.

2) Break the resulting terms into a system of simple integrators, gains, and summing points.

3) Write down the system state vector differential equation (again, assuming constant coefficient frozen-time points) in the standard form

$$\dot{x}=Ax+Bu \qquad y=C^Tx+Du \tag{1}$$

4) Assuming a hold, obtain the corresponding difference equations

$$x_{n+1}=\Phi(T)x_n+\gamma u_n \tag{2}$$

where $\Phi(T)$ and γ are approximated by truncating the series

$$\Phi(T)=\sum_{i=0}^{\infty}\frac{A^iT^i}{i!} \tag{3}$$

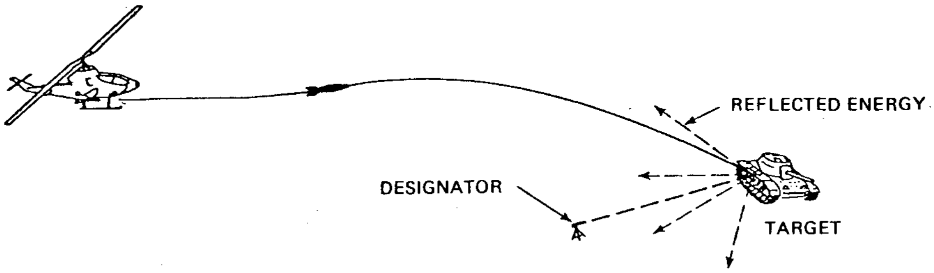

Fig. 1 Typical scenario.

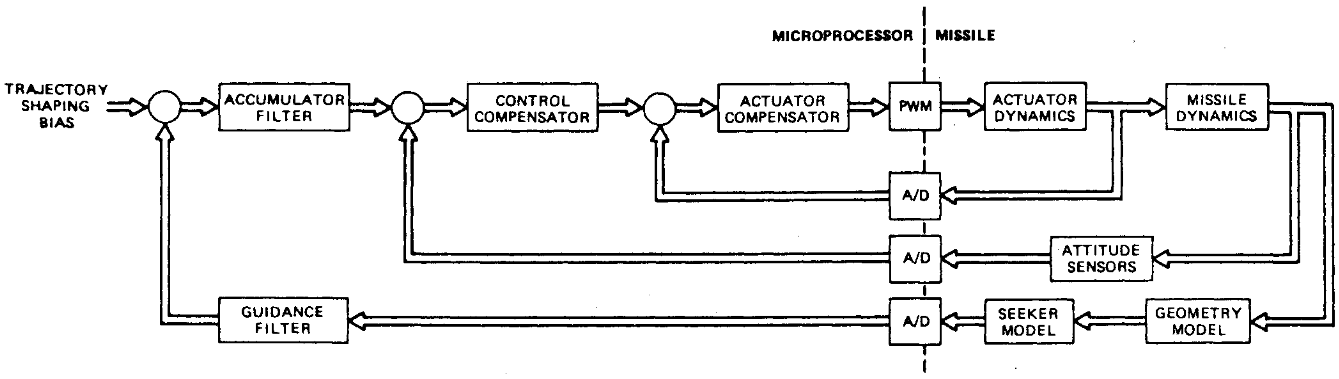

Fig. 2 Basic autopilot block diagram.

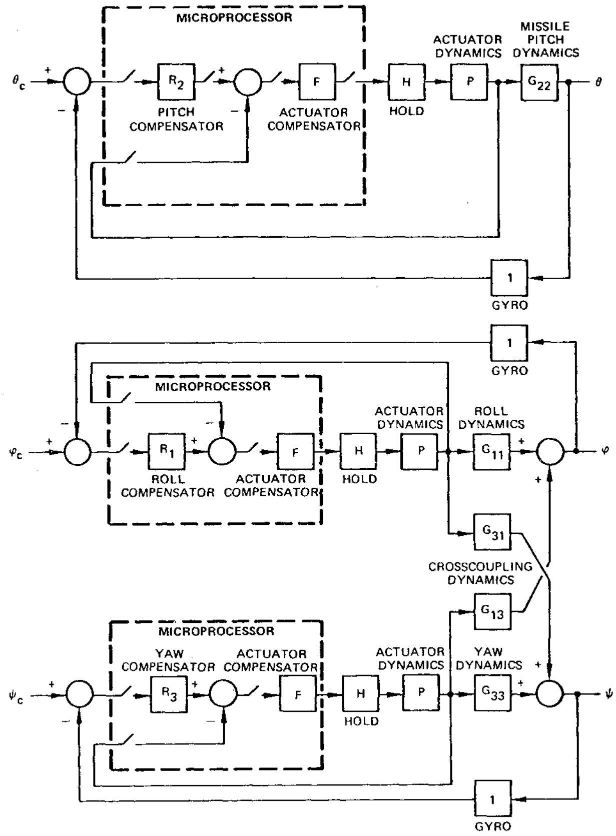

Fig. 3 Sampled-attitude loop block diagram.

$$\gamma = \sum_{i=0}^{\infty} \frac{A^i T^{i+1}}{(i+1)!} \cdot B \tag{4}$$

5) Take the z-transform of Eq. (1)

$$x^*(z) \triangleq z[x] \qquad u^*(z) \triangleq z[u] \tag{5}$$

$$X(z) = x^*(z) = [zI - \Phi(T)]^{-1} \gamma u^*(z) \tag{6}$$

and

$$y^*(z) = C^T x^*(z) + Du^*(z) \tag{7}$$

6) Since this derivation is being performed for the frequency domain, the $s = \sigma + j\omega = j\omega$, and Eqs. (6) and (7) are evaluated using

$$z = e^{j\omega t} = \cos\omega T + j\sin\omega T \tag{8}$$

Table 2 Analog autopilot filters

Actuator	$\dfrac{(s/125+1)^2}{(s/65+1)(s/1500+1)}$
Pitch and yaw	$\dfrac{1.6(s/25+1)(s/125+1)}{[(s/150)^2+(0.6s/150)+1]\,[(s/200)^2+(0.8s/200)+1]}$
Roll	$\dfrac{0.32(s/30+1)(s/250+1)^2}{[(s/300)^2+(0.6s/300)+1]\,[(s/500)^2+(0.8s/500)+1]}$

Since matching is always better at lower frequencies and for smallest T (sampling period in seconds), the largest value of T still satisfying the frequency response criterion would normally be selected for use in Eq. (8).

The result of the state variable conversion technique is equivalent to taking the z-transform of the analog filter combined with a zero-order hold. This zero-order hold introduces phase lag which requires compensation. Other lag is created by computational delay. Compensation of these lags is normally accomplished by adding a predictor to the digital filter output. An alternate approach would be to use the value of x_{n+1} to compute y. This is equivalent to introducing a phase lead which experience has shown adequately compensates for the lags due to the zero order hold and the computation time delay.

After all the filters were converted, it was estimated that all of the baselined 1K words processor memory would be used by the digitized DAP algorithm. In addition, the time required to multiplex the analog data through the A/D and to evaluate the difference equations was estimated as 3.11 ms, thus the lowest available sampling period, 4 ms (250 Hz), was chosen. The digital filter coefficients were computed and the frequency responses of these digital filters compared favorably with those of the corresponding analog filters. DAP implementation will soon be discussed.

Discrete Domain Approach

Discrete domain design is accomplished[3] by means of the proper transformation of the sampled s-plane block diagram through the z-plane to the w-plane. The necessary standard transformations are

$$w = \frac{z-1}{z+1} \qquad z = e^{sT} \tag{9}$$

where T is the sampling period in seconds. The above transformations map the left half of the s-plane primary strip into the left half of the w-plane. Familiar s-plane design techniques (Bode, Nyquist, Root Locus, Routh-Hurwitz, etc.) can be used while working in the w-plane.

All plant dynamics equations were computer-generated at a set of frozen-time points along a typical aerodynamic trajectory. These were transformed into *w*-plane transfer functions and then plotted as Bode, polar and Nichols plots. After using conventional compensator design techniques at several operating or frozen points, compensated system stability margins are obtained.

The autopilot attitude channels are typically a mixture of servo and regulator systems. A zero roll angle is desired whereas pitch and yaw should execute the guidance commands with acceptable response characteristics while maintaining proper attitude stability. Figure 3 illustrates the simplified attitude stabilization system structure. Using the sampled-data control theory techniques presented in Ref. 4, the block diagram of Fig. 3 was converted to the *w*-plane diagram of Fig. 4. The proper design procedure then was to: 1) obtain $P(s)$ actuator dynamics (from Table 1), multiply by $H(s)$ zero-order hold, and transform the result to *w*-plane to obtain $HP(w)$; 2) using known Bode-Nyquist classical techniques, design $F(w)$ the actuator compensator, in the inner loop; 3) obtain $G_{22}(s)$ missile pitch and uncoupled dynamics (from Table 1), multiply by $H(s)$ and $P(s)$, and transform the result to the *w*-plane to obtain $HPG(w)$; and 4) design $R_2(w)$ the pitch filter compensator.

The actual design procedure was of necessity a more complicated and iterative one, especially for the roll and yaw channels which are coupled. Reference 3 carries a detailed explanation. The final D^3 DAP filters are shown in Table 3. The *w*-plane D^3 DAP stability margins are shown in Table 4.

It was estimated that the D^3 DAP uses less than one-half of the baselined 1K word processor memory. The time required to convert the analog inputs to digital form and to evaluate all the difference equations was estimated as 1.65 ms, running twice as fast as the digitized version. Further, due to the type of programming form selected, computational delay time was significantly reduced to 1.0 ms by separation of the filter output calculation from the update of its intermediate variables.

Sample Frequency Selection

In the digitized DAP, sample frequency selection is very straightforward: since it took 3.11 ms to process all data, then the next highest available sampling period (4 ms) was chosen. The object here is to sample as fast as possible. This fixes its sampling frequency at 250 Hz. It was verified that the DAP still barely met the 1 dB and 5 deg degradation rules. Although the DAP met the degradation rules, it was felt that a faster sampling frequency (e.g. 500 Hz) would have produced better matching and thus better stability margins. This higher sampling frequency, however, was not possible due to hardware speed limitations.

Four factors are considered in the selection of 125 Hz as the D^3 DAP sample frequency: 1) Shannon's sampling theorem,[4] 2) nonsynchronization with laser pulse rate, 3) insensitivity to future missile or guidance scheme changes, and 4) synchronization with PWM actuators.

One of the major advantages in a discrete domain design is the lower sample rate generally achieved. This lower rate is required solely for stabilization purposes. In addition to the stability requirements, a digitization approach must meet the signal reconstruction requirements imposed by Shannon's sampling theorem. This theorem basically states that to reconstruct a continuous signal, the sampling rate must be at least twice as fast as the highest frequency contained in the signal. If an analog compensator is designed and then digitized, this sampling requirement must be rigidly met.

Table 3 *w*-plane final DAP compensators[a]

Actuator		$\dfrac{(w+1)}{(w/10+1)^2}$
Pitch and yaw	$1.6\ J_t$	$\dfrac{(w/0.06+1)}{(w+1)}$
Roll	$0.32\ J_t$	$\dfrac{(w/0.15+1)}{(w/2+1)(w/5+1)}$

[a] $J_t = \max(5-5t, 1)$, where t is time since launch.

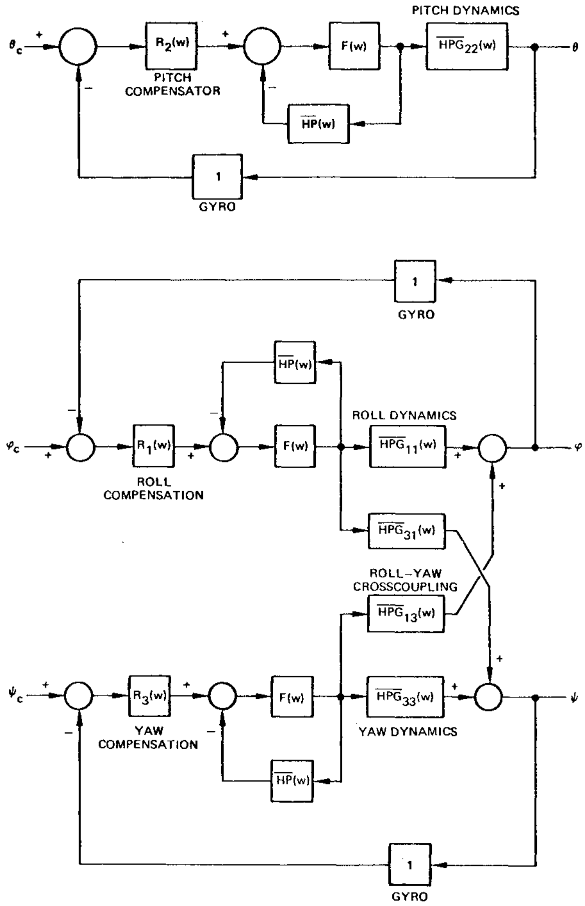

Fig. 4 Attitude loop block diagram in the *w*-plane.

Table 4 *w*-plane DAP stability

	Pitch			Roll			Yaw		
Flight region	Crossover frequency, rad/s	Phase margin, deg	Gain margin, dB	Crossover frequency, rad/s	Phase margin, deg	Gain margin, dB	Crossover frequency, rad/s	Phase margin, deg	Gain margin, dB
Launch	0.07	45	13	0.09	27	15	0.08	45	14
Max-*Q*	0.21	41	5.6	0.30	38	4.9	0.22	40	5.5
Terminal	0.07	45	15	0.10	31	13	0.08	52	15

Actuator loop frequency content dictates a sample frequency above 50 Hz. Seeker loop pulse rate frequency either forces synchronization or dictates sample frequency on the order of 80 Hz (asynchronously) and above.

Application of advanced guidance techniques on the basic test vehicle would certainly improve performance; however, this would also change operating points and stability margins. By picking a high sampling rate, the compensated system stability has reduced sensitivity to such changes.

The choice of T for the D^3 DAP was arrived at through a selection study which looked at the w-plane pole-zero travel diagram of the G_{xx} plant dynamic equations in a way similar to that used for root-locus plots. Frequencies above the 80 Hz required by the seeker and below the digitized DAP's 250 Hz were investigated. Harmonics and subharmonics of 60 cycle standard 110 Vac line voltage were rejected due to expected problems. Results indicated[3] that sample frequencies between 80 and 125 Hz could have been used.

The D^3 DAP sampling frequency ($1/T$) was chosen as 125 Hz, as was also the frequency of the PWM carrier. Synchronization to the PWM loop reduces system complexity and enables a relatively easy single-rate design.

Implementation

Once the designer has arrived at a satisfactory control system design, the filters and compensators, along with any other nonlinear controls, must be converted into an implementable format. The control element chosen here is a microcomputer, a very small digital computer. Thus, instead of the classical operational-amplifier-based set of analog-circuit electronics, a set of digital chips will comprise the electronics. The program stored in the microprocessor's memory is the logical or controlling element, rather than op-amp interconnections. The stored program, referred to as the control software, must be obtained from the control system design.

Programming Form

The compensators are now transformed to the z-plane where microprocessor difference equations are obtained through choice of the rectangular programming form. For example, a second-order transfer in the z-plane

$$\frac{y}{u}=\frac{K_1z^2+K_2z+K_3}{z^2+K_4z+K_5} \tag{10}$$

taken through the rectangular programming form results in the FORTRAN equations

$$\begin{aligned} &Y=K1*U+X1\\ &X1=K2*U+X2-K4*Y\\ &X2=K3*U \qquad -K5*Y \end{aligned} \tag{11}$$

These algebraic difference equations can now be programmed for each filter to execute on the microprocessor. The choice of a rectangular programming form was made due to its minimum delay time, explicit coefficient storage, and inherent simplicity. Note that the results obtained through the use of different programming forms differ only if real-world scaling and quantization are present. In an infinite-word-length situation, most programming forms will produce similar behavior.

Quantization and Scaling

Implementation of the difference equations in the microprocessor requires the quantization of all data and parameters within the processor to a finite set of allowable values due to a finite word length. Quantization errors consist of errors due to the A/D conversion of the input, errors due to coefficient quantization, and errors due to arithmetic roundoff.

In general, the input signal is continuous and A/D conversion is necessary prior to digital processing. The error in A/D conversion is a random sequence, and the quantization noise power at the output of a digital filter can be computed. If measured signals are input to the digital processor through a $\pm A$ volt, m-bit A/D converter, then the resolution is $2A/2^m-1$ mV/bit.

Addition is an exact operation except possibly for overflow, which is avoided by proper choices of word length and quantization interval. Multiplication is performed computationally using bit expansion and, therefore, must be followed by truncation. Thus, many arithmetic errors are due to truncation after multiplication.

It was found[3] that a 16-bit word length with a 12-bit A/D was necessary to keep quantization levels below significance.

Implicit in any discussion on arithmetic quantization is magnitude scaling. Magnitude scaling attempts to utilize the full dynamic range of the quantized signals which maximizes the signal/noise ratio for a given processor-word length.

Scaling and quantization were approached by the implication of a binary point within the 16-bit word. Thus pitch attitude maxed at ±40 deg and truncated every 0.02 deg. This approach proved sufficiently accurate. Future DAP's will employ a floating point processor (such as the tightly coupled 8086/8087), thus eliminating scaling problems and significantly reducing quantization problems.

Computational Delay

Digital effects such as aliasing and sampler noise can be eliminated through proper hardware/software design techniques (pre- and post-sampler filtering). Other effects like quantization and scaling can be significantly reduced by either using available compensation techniques or by implementing arithmetic floating point processor chips in the microcomputer design. Computational delay then remains the single most significant and uncorrectable digital effect.

Computational delay is that time between A/D read and D/A (or PWM) output. It can be reduced by choice of a good programming form. Some compensation can be provided by the use of modified z-transforms when handling the plant transfer functions. Some delay will still remain after the use of these techniques. It was verified that a 1 ms delay does not significantly degrade the D^3 DAP stability margins (of Table 4). It is due to stability degradation through computational lag that the microprocessor's speed specification is a very significant one.

Code Comparisons

Table 5 shows relative code efficiencies measured for the two digital autopilots as implemented in a baseline microcomputer 6-DOF emulator. It is clear to see that the D^3 DAP code was more compact, and thus more efficiently executes the same function. Subsequent implementation into a DAP wirewrapped breadboard verified the emulator code numbers.

Results also show that a baselined 1K words (2024 bytes) are sufficient for DAP program storage.

Microcomputer Hardware

The currently available DAP hardware design is based on the first commercially available four-bit slice bipolar

Table 5 Digitized and discrete domain DAP actual requirements

Measured DAP requirements	Digitized design DAP	Discrete domain DAP
Memory required, bytes	1732	906
Time required, ms	3.2	1.7
I/O delay time, ms	3.2	1.0

microprocessor chip (the 6701). Although the hardware was state-of-the-art for high-speed devices at the time of the decision to build, the rapidly changing technology in chip design currently offers several tempting choices. Due to the required computational speeds (low delay) and power necessary, and because of quantization and estimated accuracy requirements, it was deemed necessary to use a microcomputer capable of fast 16-bit arithmetic.

The computer was configured for 16-bit arithmetic operations with 1024 words of main memory (256 RAM, 768 ROM) and 1024×50 bit words of microprogram memory. The main design objectives were to minimize hardware and maximize processing power. Minimum hardware was achieved by utilizing a design which did not have microsequencers, pipelining registers, look-ahead carry generators, or program control and branch units. Instead, a six-phase clock was logically gated with certain control bits in the microprogram memory which determined bus priority (only one 16-bit bus for address and data), sequencing, and branching. Even with minimum hardware, the bipolar bit-slice design provided ample amounts of processing power with a microinstruction execution time of 750 ns.

The DAP accommodates analog input signals with a 16-channel analog multiplexer through a 12-bit A/D converter, and drives the fin actuators with four digital pulse-width modulators. Internal data (such as digital filter outputs) is monitored during flight with a bit-serial pulse code modulation telemetry output circuit.

Initial software support for the DAP system consisted of an emulator[5] developed concurrently with the hardware design. The hand coding of a microprogrammable computer proved to be tedious, error prone, and time consuming. Therefore, a user-definable (meta) cross-assembler was designed[6] to process both micro- and macrocode. To further enhance programming, since the largest portion of DAP processing is equation solving, a high-order language (HOL) compiler was developed. The HOL is a subset of FORTRAN with an editor/optimizer which optimizes the partitioning of micro/macrocode in a post-translation effort. The use of the HOL greatly reduces the turnaround time for application program changes, but did not hinder execution time or memory resources due to the care taken in its design.

Design Evaluation

Prior to flying a prototype missile containing the DAP, performance analysis through extensive 6-DOF and HWIL simulation was necessary. In simulation, expensive development of hardware prototypes are not necessary, thus reducing the project's cost. Further, the use of Monte Carlo techniques permit checkout of unknown random parameters, such as in the seeker, evaluation of the effects of random but real-world disturbances on the system, and investigation of the sensitivities of the missile to various initial condition scenarios. The use of simulation also permits checkout of all system nonlinearities which cannot be easily analyzed otherwise.

Digital Simulation (6-DOF)

The 6-DOF time domain Monte Carlo Terminal Homing Simulation Program provides the capability of simulating flights of the T6 missile under varying conditions. Both Monte Carlo statistical run sets and deterministic flights are possible.

This all-digital simulation incorporates validated representations of all system elements (except DAP) sufficient for determination of overall prototype weapon-system performance. The simulation techniques employed have been validated in conjunction with the firing of over 60 laser semiactive-guided missiles. There is, therefore, a sound basis for confidence in the basic approach to the missile-system simulation.

It must be noted that system fine tuning was accomplished by interaction with the deterministic 6-DOF simulation to accommodate any significant nonlinearities. Once a properly tuned design has been obtained, statistical evaluation is the only way to properly measure guided performance. Seeker uncertainties in the simulation model (e.g., spot jitter, etc.) rule out deterministic evaluation. The 6-DOF was the main DAP verification tool.

HWIL Simulation

The hybrid HWIL simulation consists of a hybrid-digital analog real-time computer simulation employing the guidance and control DAP HWIL. This closed loop operation in a realistic simulation offers the opportunity to study the dynamic stability of the hardware system at first hand, with the hope of optimizing performance while reducing the length and cost of the flight testing cycle.[7]

Problems Encountered

During the HWIL simulation verification, problems encountered were grouped into three categories: 1) noise problems, impacting stabilization filter design, 2) microcode errors, pointing out faulty logic, and 3) hardware errors, including layout problems. These problems were cleared up through software changes (filter modification, logic patches, overflow correction) and minor wiring changes.

Several standard simulation problems were encountered in the initial HWIL simulation. These typically included modeling problems, such as with the actuator analog model gains, aerodynamic hinge moments, and the like. They occurred due to the lack of adequate modeling and test procedures. Testing should include not only the closed-loop sinusoidal response, but also open-loop time and frequency responses.

The correction of some hardware problems by means of software patches points out the power and versatility of using microprocessors as digital-controller elements.

Solutions to the above problems were attained mostly by DAP design fine tuning and hardware layout changes. Design fine tuning by means of frequency breakpoint shifting (and the addition of another pole in pitch) was necessary for noise reduction. The examination of power supply noise pointed out the need for extreme isolation for all DAP power lines, from each other and from the digital mainframe.

Design Verification and Validation

Complete agreement between frequency domain tools, 6-DOF, and HWIL simulations was accomplished after several iterations.

Figure 5 shows the 6-DOF performance comparisons of the digitized and D^3 DAP's normalized against analog-autopilot performance. It shows the D^3 DAP to perform better than either analog or digitized DAP. Under HWIL simulation, the digitized, discrete domain and analog autopilots behaved similarly acceptable, and very much like the DAP emulator in the 6-DOF (see Fig. 6). DAP design methodology, logic and hardware were verified by means of the HWIL simulation.[8]

Both DAP designs were judged acceptable. The D^3 DAP was judged to be the winning design due to its much higher code operating efficiency, and to its robustness of stability. Both these factors are necessary in order to use the DAP as a module in a building-block approach to adding complex guidance schemes and seeker signal processing. Robustness, or the lack of sensitivity to parameter variations, is necessary due to the increase of lag times and decrease of guidance bandwidth expected when exchanging guidance heads or schemes.

Although comprehensive 6-DOF and HWIL simulation should be sufficient for concept verification, a limited number of test flights are necessary for complete validation and enhanced confidence in the DAP design. Further, early

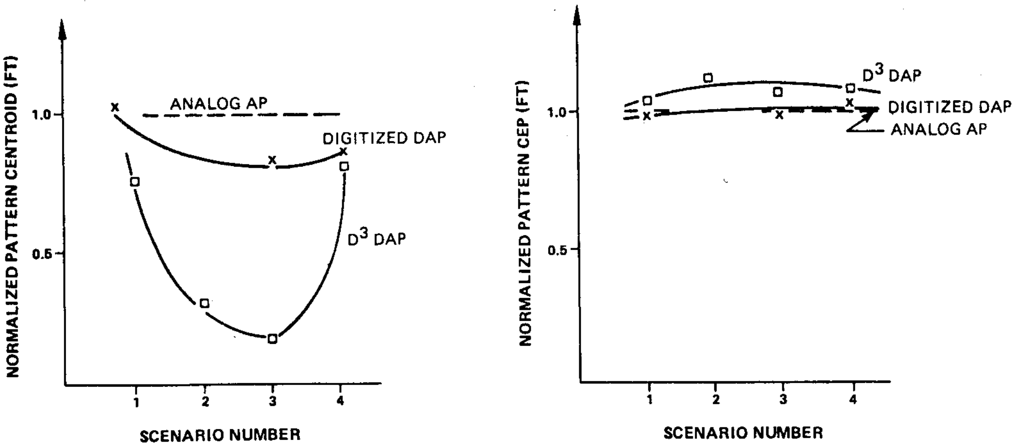

Fig. 5 Autopilots performance comparisons.

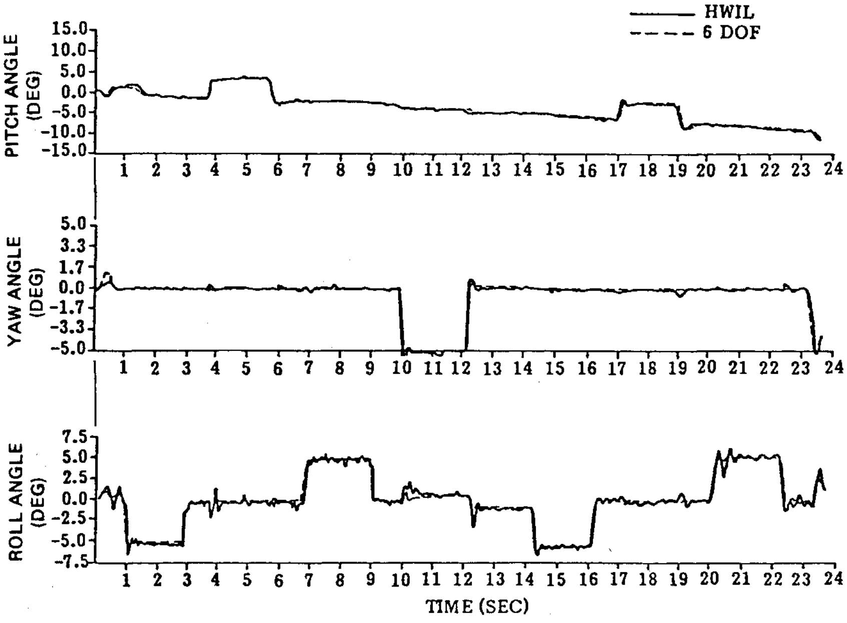

Fig. 6 6-DOF and HWIL verification.

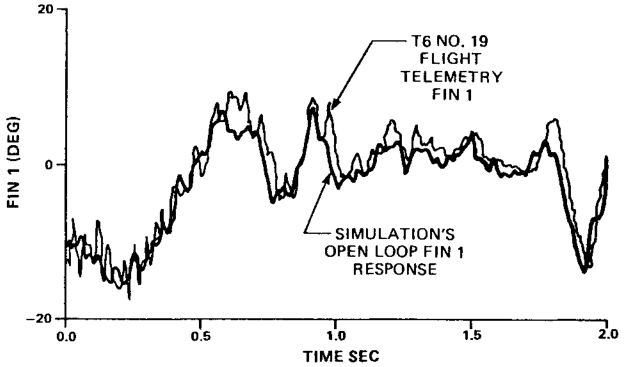

Fig. 7 Telemetry gyros input producing fin outputs.

correction of design problems through a combination of simulation and test flights has been shown to reduce the overall cost and startup time of military projects.

In Fig. 7, fin telemetry data obtained during the flight of a DAP-controlled T6 shot (the T6 #19 flight launched in Sept. 1979) are compared to results obtained from a 6-DOF with DAP emulator. Good matching is apparent in actual position, as well as in phase. Frequency contents for the two signals were similar. The data agree well, indicating complete model and design validation.

Conclusions

Several different autopilot-design concepts have evolved over the years. The most common are the digitized approach and the discrete-domain approach.

A digital autopilot (DAP) for a microprocessor-controlled T6 terminal homing missile has been designed that, based upon comparative performance, is equal to or better than current analog autopilots, but offers room for enhancements. This discrete-domain DAP is much more efficient than current digitized versions, using only half the memory, half the computational time, and having only a third the computational delay of the digitized DAP. This results in cost savings, as well as performance improvements. The DAP was fully checked out and verified by means of six-degree-of-freedom and hardware-in-the-loop simulation, and the concept validated through a limited number of test flights. The DAP has performed flawlessly.

Since digital hardware is becoming smaller, cheaper and faster, digitalization will replace analog logic in other missile subsystems. Complex guidance schemes, easily implemented with a DAP, promise to increase warhead penetration by trajectory shaping and optimum impact conditions. Future DAP's will incorporate full guidance and seeker signal processing functions, such that in the limit an all-digital missile may prove to be the next generation field weapon.

Acknowledgments

This autopilot effort has been performed under several contracts for the U.S. Army Missile Command, Missile Laboratory, Guidance and Control Directorate, under the Modular Guidance R&D program, with Dr. Robert E. Yates, K. Wayne Plunkett, and John B. Meadows as technical coordinators. The author wishes to thank Dr. Yates, Dr. Klaus D. Dannenberg, and Samuel L. O'Hanian for their long-term guidance, and Jerry T. Bosley for his support and many contributions.

References

[1] Aitken, B., Sliz, C., and Leonard, J., "T6 Missile Description," U.S. Army Missile Command TR-RG-73-18, Aug. 1973.

[2] Asquith, C.F., "T6 Digital Autopilot Data Processing Analysis and Specifications," U.S. Army Missile Command TR-RG-75-36, March 1975.

[3] Plunkett, K.W., Meadows, J.B., Albanes, W.V., and Bosley, J.T., "Design and Analysis of a Microprocessor-Based Digital Autopilot for Terminal Homing Missiles," U.S. Army Missile Research and Development Command TR-T-78-57, March 1978.

[4] Kuo, B.C., *Analysis and Synthesis of Sampled-data Control Systems,* Prentice-Hall, Englewood Cliffs, N.J., 1963, pp. 27-28.

[5] Brookshire, J.R., "Multipurpose Digital Microprocessor Emulator," U.S. Army Missile Research and Development Command TR-RG-76-62, May 1976.

[6] Baxter, W.F., "Microprocessor Controller/Debug System," U.S. Army Missile Research and Development Command TR-RG-76-61, May 1976.

[7] Pastrick, H.L., Will, C.M., Isom, L.S., Jolly, A.C., Hazel, L.H., and Vinson, R.J., "Hardware-In-The-Loop Simulation: A Guidance System Optimization Tool, AIAA Paper 74-929, 1974 AIAA Mechanics and Control of Flight Conference, Aug. 1974.

[8] Plunkett, K.W., Meadows, J.B., and Albanes, W.V., "Computer-Aided Digital Autopilot Design and Analysis: Methodology, Implementation, and Verification," U.S. Army Missile Command TR-RG-80-19, Nov. 1979.