NASA Technical Memorandum 86017

Ultrareliable, Fault-Tolerant, Control Systems

A Conceptual Description

Larry D. Webster, Roger A. Slykhouse,
Lawrence A. Booth, Jr., Thomas M. Carson,
Gloria D. Davis, and James C. Howard

JULY 1985

NASA

NASA Technical Memorandum 86017

Ultrareliable, Fault-Tolerant, Control Systems

A Conceptual Description

Larry D. Webster, Roger A. Slykhouse,
Lawrence A. Booth, Jr., Thomas M. Carson,
Gloria D. Davis, and James C. Howard
Ames Research Center
Moffett Field, California

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1985

TABLE OF CONTENTS

SUMMARY

An Ultrareliable, Fault-tolerant, Control-System (UFTCS) concept is described using a systems design philosophy which allows development of system structures containing virtually no common elements. Common elements limit achievable system reliability and can cause catastrophic loss of fault-tolerant system function The UFTCS concept provides the means for removing common system elements by permitting the elements of the system to operate as independent, uncoupled entities Multiple versions of the application program are run on dissimilar hardware Fault tolerance is achieved through the use of static redundancy management

INTRODUCTION

The use of redundant concepts enables fault-tolerant systems to be developed which are significantly more reliable than their simplex counterparts Unfortunately, no system can be developed which can be shown to be entirely free from design error Latent design faults which exist in system functions which are held in common between redundant elements will ultimately cause system failure The advent of a system failure caused by a common latent design fault represents an unpredicted event in the life of any fault-tolerant system The first attempt to launch the Space Shuttle Columbia was aborted as a result of a latent design fault in a common system feature, namely, the computer synchronization scheme Fortunately, the failure was not catastrophic However, it was costly in terms of financial loss, schedule delay, and prestige

Fault-tolerant systems which do not contain features common to redundant elements are inherently more reliable than system designs which do contain common features The effect of common elements on fault-tolerant performance generally are not considered when the reliability of a given system is calculated The difference between concepts containing common elements and those not containing common elements will become apparent when the reliability of the operational system is assessed

Common features are not required in systems where the redundant elements operate independently These features proliferate in fault-tolerant systems concepts which are popular today and can be generally categorized as (1) use of global redundancy management schemes, (2) use of redundancy synchronization, (3) cross-channel communication between redundant elements, (4) replication of identical hardware, and (5) replication of identical application programs

PROJECT HISTORY

The postulate that UFTCS systems could be developed from completely independent elements came from research begun in the mid-1970s at NASA Ames Research Center Dunn and Meyer recognized that significant increases in the computational power of microelectronics, coupled with similar decreases in their cost, size, and weight would permit the development of ultrareliable fault-tolerant system designs (ref 1) They developed a system structure which uses an efficient blend of hardware and software to achieve fault tolerance based upon independent redundant elements The resulting UFTCS system concept consisted of asymptotically stable independent control elements in a parallel, cross-strapped system environment Fault tolerance was achieved through the use of static redundancy management (ref 2)

Static-redundancy management is modeled after biological systems which compensate for continuous failure without any global fault-handling mechanism This method of accommodating the failures experienced in individual redundant elements was previously impractical because of the enormous penalties of cost, size, weight, and power incurred in N-module (replicated) redundant systems Because of this, the SIFT (ref 3) and FTMP (ref 4) fault-tolerant programs concentrated on hardware-conservation techniques based on dynamic redundancy management Both trade off hardware size for software and general system complexity The current computational power available in today's microelectronic technology makes possible the use of N-module, static redundancy in fault-tolerant system concepts

Using the original work as a basis, the authors have developed laboratory facilities, simulators, and an operating experimental test bed capable of examining the fundamentals of the theoretical work The laboratory has the capacity to synthesize and analyze ultra-reliable fault-tolerant systems concepts The experimental test bed provides a real-time implementation of a UH-1H helicopter control system and is used to examine the characteristics of independently operated fault-tolerant elements It consists of autonomous elements configured to operate independently, in parallel, and can be arranged in quadruple, triple, or dual-redundant configurations The UH-1H control algorithm implemented is asymptotically stable (refs 5 and 6)

An ultrareliable fault-tolerant control system (UFTCS) is one which, possessing virtually no common elements, will deliver its theoretical reliability This report describes the development of a design philosophy from which ultrareliable fault-tolerant control systems (UFTCS) can be generated The basis for the concept employs the use of independent, uncoupled redundant elements to eliminate the need for common elements in the system structure A properly configured UFTCS system will have the following characteristics and attributes (1) a structure that is relatively free of common elements, (2) probability of failure less than 10^{-13}/hr in the helicopter environment, (3) survivability from externally motivated physical damage caused by the dispersion capacity of the redundant elements, (4) independent operation of control tasks and redundant elements, (5) autonomous, nonidentical elements, (6) static, nonreconfiguring local fault management, (7) task replication and voting processes which are transparent to the application designer

The theoretical and experimental objective of our work is to provide proof that fault-tolerant systems based upon independent operation of redundant elements are a practical alternative for fault-tolerant system designs To this end, laboratory simulations were developed which allowed investigation of the responses of various control-system structures in the fault-tolerant environment, and the structuring and analysis of redundant data-management (voting) schemes A typical example of a UFTCS was constructed, using the architecture proposed by Dunn and Meyer (ref. 1), to allow investigations into the theoretical and practical bases of the concept in conjunction with the operational hardware and software The control system implemented in this experimental test bed was that of a UH-1H helicopter Some results of the experimental research are presented which establish an empirical basis indicating that ultrareliable systems which employ independent, uncoupled-redundant elements can be produced

ULTRARELIABLE CONCEPTS

Ultrareliable Fault Tolerance – A Definition

Ultrareliability can be achieved in a fault-tolerant system by eliminating the potential of system failure caused by latent design errors which exist in functions held "in common" among the redundant elements As no system design can be shown to be completely free of latent design error, these latent faults can be assumed to exist in fault-tolerant systems which contain common features Reliability analyses of fault-tolerant systems do not account for the presence of the latent design error (some general work has, however, been accomplished in this area (ref 7)) Therefore, the demonstrable reliability of a fault-tolerant system which contains common failure modes will be less than the calculated reliability because of the presence of latent design faults which exist in any common features However, fault-tolerant systems which contain virtually no common elements can be structured These systems are predicated on the concept that the redundant elements must exist as independent, autonomous entities Ultrareliable fault tolerance can be defined as "a concept which produces ultrareliable systems that have the capability of delivering expected reliability " Because of the simplicity of the designs which evolve from the concept, these systems are extremely reliable with probabilities of failure less than 10^{-13}/hr of operation in environments as adverse as the helicopter environment

Ultrareliable Fault Tolerance – The Criteria

Historically, the development of fault-tolerant systems has been motivated by the persistent need of the user for ever-increasing system reliability Numerous fault-tolerant system concepts have been developed to satisfy this need Each fault-tolerant structure is constructed attempting to satisfy two basic criteria

(1) the redundant system elements must be independent,
(2) the redundantly generated outputs must be unambiguous

It is the relative compliance with these two criteria which determines the inherent reliability of a given fault-tolerant structure In the past, it has not been possible to comply completely with both criteria for reasons which are discussed later Usually, a satisfactory compromise between compliance and system structure will be reached which satisfies the second criterion, but only partially satisfies the first

On initial consideration, it appears that these two criteria are mutually exclusive The first condition establishes that the redundant elements must be independent and totally autonomous with no common features But the second criterion requires that the outputs produced by the independent redundant elements cannot be ambiguous For functioning elements, the output values which are generated must be close enough in value throughout time so that logical comparisons can be made These comparisons are performed by a portion of the system, usually known as the "voter," which is responsible for determining the failure status of the redundant elements

Independent Elements – A Historical Perspective

It would seem that independent elements executing, for example, a flight-control algorithm would not produce control values which could be logically voted for any length of time Under certain conditions, this has been experimentally shown to be the case The original system architecture of the AFTI-F16 triplex, fault-tolerant system was structured such that the elements were independent However, tests of the system, as described by Mackall (ref 8) disclosed significant differences between redundantly generated control values The magnitude and unpredictability of the discrepancies made channel selection and fault detection virtually impossible In attempting to comply with the first fault-tolerant criterion, it became impossible to fulfill the second Steps to correct the problem were implemented A second case, described by Osder (ref 9) disclosed the results obtained from experimental hardware-in-the-loop simulations of tri/quadraplex fault-tolerant systems employing independent elements As with the AFTI-F16 control-system experience, significant differences existing between independently generated redundant data were recorded and a method was devised to correct for the differences It is interesting to note that both systems solved the divergence problem by implementing a form of cross-channel communication to force the control values generated to conform to each other The first used equalization to balance the integration terms directly in each control law processor thus bringing the generated redundant values into accord The second system used equalization to obviate differences in the values generated by each control law processor Regardless of the technique employed, the intent of cross-channel communication between the redundant elements is the same Osder (ref 9) properly describes the need "to correct static or long-term differences (between the control law processors) so that the channels track "

However, the use of cross-channel communications to force the generation of votable, redundant, output sets violates the first premise upon which ultrareliable concepts are based The system elements must be independent Cross-channel communications are needed only if static or long-term differences between redundant elements exist Both requirements for the development of ultrareliable fault-tolerant systems are met by a systems concept which contains independent elements whose outputs do not produce long-term differences In the next section, the mechanism through which these differences are produced is demonstrated Several simple techniques which may be employed to cause long-term differences between redundant element outputs to effectively disappear without resort to cross-channel communication is then displayed in a later section

Sources of Common Elements

The generation of fault-tolerant system concepts based upon independent elements requires the implementation of certain philosophies which maintain the independent relationship By specifying that the redundant elements must be independent, systems are produced which are common-element free Common elements can reside not only in the hardware, but also in the application programs that are embedded in the hardware and in the philosophies involving redundancy management All three areas of the system structure contain the human element The sources of commonality in fault-tolerant systems are (1) use of global redundancy management schemes, (2) use of redundancy synchronization, (3) cross-channel communication between redundant elements, (4) replication of identical hardware, and (5) replication of identical application programs

Common-Element-Free Concepts

Unsynchronized operation– The need for tight synchronization of the redundant elements is predicated on the requirement that the output values generated by these elements be votable Synchronization, in conjunction with cross-communication of the input state variables, is used to force the redundant elements to march in lockstep, transducing input state variables at the same time and providing computed outputs for voting simultaneously Synchronization is used because it forces the redundant channel outputs to track, thereby making the voting process simple Thus, synchronization fulfills the criterion that the generated outputs must be unambiguous However, a synchronized fault-tolerant system violates the criterion that the redundant elements must be independent Because of this, the synchronized system cannot fulfill projected reliability as the synchronization scheme (hardware, software, and human element) represent an uncalculated common element in which a latent fault will cause total loss of system function Further, synchronized systems tend to contain other sources of common elements such as the use of cross-channel communications, identical hardware, and identical application programs To make the resulting system more reliable, unsynchronized redundant elements should be used

Multiple-version redundant elements– The same reasoning can be invoked to discuss the use of (1) identical hardware, (2) identical application programs, and (3) cross-channel communication All three methods of implementing redundancy violate the independence criterion The human element is involved with the development of each They can, therefore, contain latent faults embedded into the design as a result of human error which, when disclosed, could lead to catastrophic loss of system function The use of cross-channel communication is unnecessary if the outputs produced are unambiguous The use of identical hardware and application programs should be supplanted with the use of dissimilar redundancy for each item The voting techniques implemented need not be identical Different voting philosophies may be used

Static redundancy management– Independent operation of system elements obviates the use of the global or dynamic forms of redundancy management The philosophy of static redundancy management is employed instead Static redundancy management eliminates the interelement coupling found in fault-tolerant systems which employ dynamic or analytic redundancy management The static method of redundancy management may best be described as management where Nothing of system wide significance occurs to the system structure on the advent and detection of elemental failure

Static redundancy management eliminates the need for global redundancy managers performing dynamic reconfiguration of the system Statically managed systems do not have the common element represented by the global management scheme and are therefore more reliable Moreover, static redundancy management does not require system-wide knowledge of how the system can fail, global knowledge of the current fault status of the system, or development of any system-wide reconfiguration strategy The static-fault managers ("voters") are autonomous entities which neither share fault status nor cause any external activity when failure of system elements are detected Basically, it is the function of the voter to control the flow of redundantly generated data from "Task A" to "Task B" as shown in figure 1 Each redundant task member has its own autonomous voter Each voter receives independently generated, redundant data from each member of the previous task Here the incoming data sets are correlated and only uncorrupted data reach the task.

Software development and validation in fault-tolerant systems which are statically managed is highly modular The control process is distributed into several tasks Each task may be built as a separate entity and linked into the whole through proper definition of the data flow between tasks As data arriving at any task have been evaluated for correctness by the voter, the applications programmer does not have to deal with the technicalities of redundant-data manipulation The task application programs are developed separately from the programs involving the voting process in a statically managed system

In conclusion, fault-tolerant systems employing independently redundant elements controlled through static redundancy management and whose system elements perform different versions of the application programs (both process and redundancy management) in dissimilar hardware are inherently more reliable than those which do not

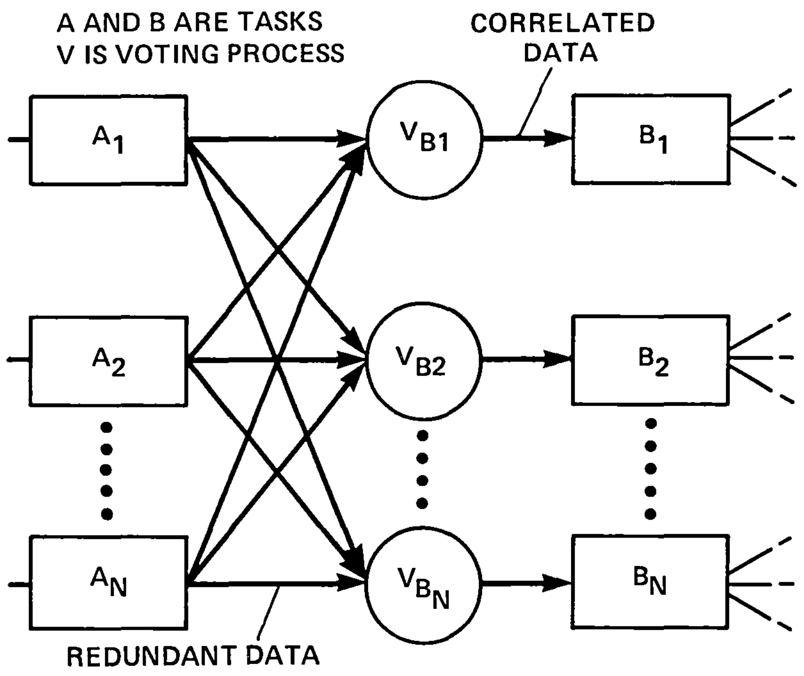

Figure 1 – Static redundancy management using independent elements

Sources of Channel Divergence

Ultrareliable systems can only be generated if the elements which perform redundant processes are functionally independent and the outputs they produce are unambiguous However, the system design practices which have currently attained industry-wide acceptance do not allow for independent operation of the redundant elements The need to produce unambiguous results has led industry to develop methods which obviate the sources of "channel divergence " These methods also cause the redundant elements to become highly coupled and dependent It is clear that other methods need to be found which allow for both unambiguous output and independent elements But it is first necessary to fully understand the origins of channel divergence and the effect that each source has upon the magnitude of the divergence between channels One important variation in philosophy will be introduced which takes the criterion of unambiguous results literally In the fault-tolerant system, it is not necessary that the agreement between results, as determined by the system voting elements, be perfect It is only necessary that the results not be ambiguous It is acceptable to relax the voting criteria to one which, for example, will tolerate long-term (10+ sec) channel differences of 1% full scale, and short-term differences (300 ms) of 3% or 4% of full scale This relaxation has little or no effect on field reliability

Variation between personal clocks– As shown in figure 2, each redundant computer contains a source of "personal time " The base of this time is usually a crystal-controlled oscillator within each computer It is a practical impossibility that the relative passage of personal time noted by Computer 1 will be identical to that noted by Computer 2 or Computer N at any instant With all other factors equal, results of computed functions which are based on personal time will vary from computer to computer Digital integration is a process affected by the computer's perception of time Time differences between computers appear as if the constant of integration is varied between the N computers For processes which are cyclic (roll or pitch control, etc), the divergence between results peaks when the cycle peaks and returns to zero when the cycle returns to its null point. Channel divergence does not accumulate, and can be made small through proper specification of the time accuracy between redundant computers For integration processes which are not cyclic, such as are found in navigation, the perceived differences in time acculülate Given sufficient time, channel divergence (caused by various channels wanting to be in different places at the same instant of real time) will grow until channel failure is declared by the voter If mission time is short compared to the intercomputer time drift rate, this effect can be ignored If mission times are long, such as in autonavigation spaceflight, this variation must be taken into account

Personal clock inaccuracies– The personal clocks shown in figure 2 not only vary relative to each other, but drift relative to the value of real time If the application program presumes a knowledge of real time (e g, a Euler first-order integration used for navigation purposes) and the real time differs from presumed time, output variations from proper outputs (those generated if the presumed time were equal to real time) occur from the point of view of the system As above, integration errors caused by cyclic operation of the system return to zero when the system returns to its initial condition, and noncyclic errors accumulate

Bias, scale, and nonlinear factor variations– The sampling interfaces shown in figure 2 are replicated and independent As physical devices, it is not possible to produce N samplers with identical properties Therefore, even if the state variable set was transduced at exactly the same moment in real time, it can be assumed that each independent computer will subsequently not contain exactly identical digital values for each of the variables The differences appear as bias, scale, and nonlinear variations in the sampled input data sets from computer to computer When the application program operates on different data, different results will be recorded by the voter When the control law contains free integrators, the effect is pronounced as the effect is summed from cycle to cycle

Sampling skew and sampling rate– These two related parameters have the most pronounced effect on the channel divergence State variable dynamics, sampling rates (the data are processed between samples), and varying sample skews will combine to produce situations where the channel divergence varies in what would seem a random manner However, if all of the system variables were known, the process is actually deterministic A simple timing diagram, figure 3, displays a possible relationship of the N independent computers shown in figure 2 Note that in independently operated redundant elements, both the sample skew and sample rate may vary with time

If redundant processes do not sample dynamic-state variables at the same instant of real time, variations between redundantly computed outputs will be recorded by the voter Sampling skew is the difference in time between input state variable samples taken by one redundant computer relative to the sample time of any other The sample skew relationship can be fixed or can vary as a function of the loop time of the application program Presume, for example, that Computer 1 and Computer N have exactly the same real time sampling rate (say 20 ms) and are identical in all other ways except that Computer N always samples data 9 3 ms after Computer 1 As the value of the input data is changing from instant to instant, Computer N will "see" different values for sampled variables than will Computer 1 If a particular parameter is increasing in value, then Computer N

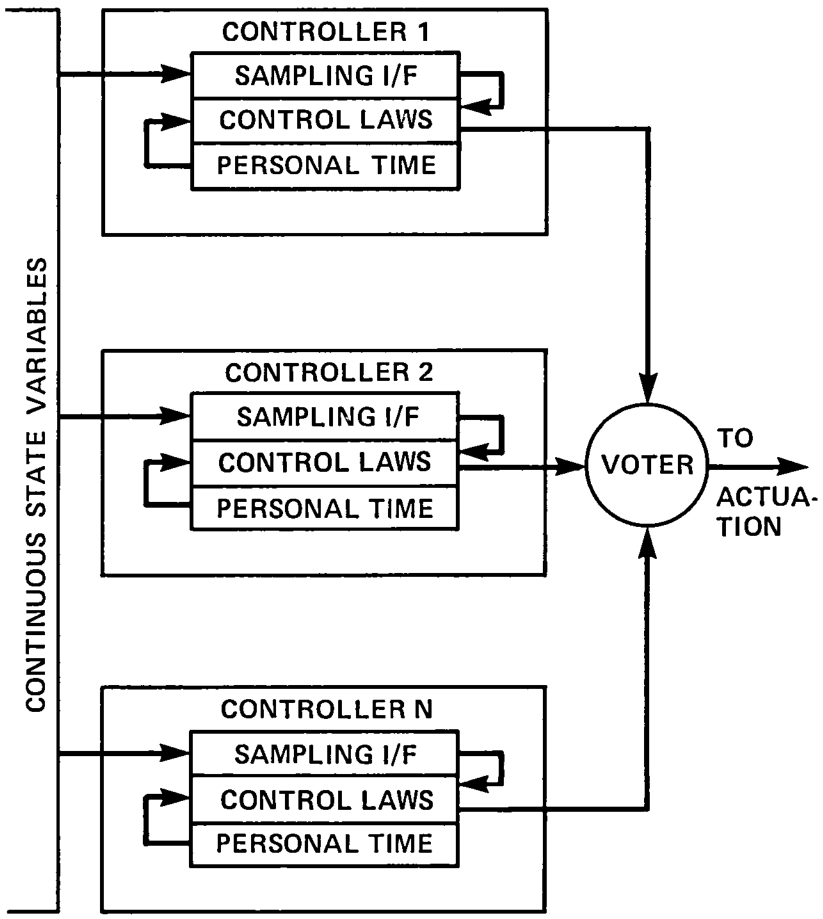

Figure 2 – Independent redundant controllers

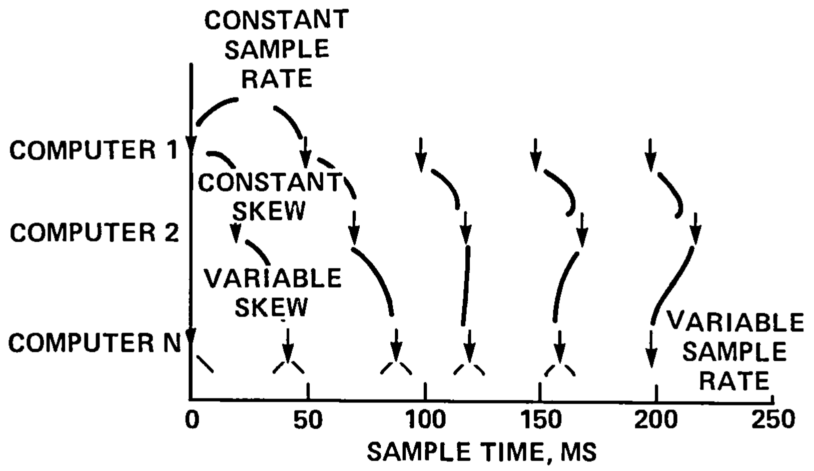

Figure 3 – Sample rate and sample skew

will see a larger value than will Computer 1 As the output of the computational process is based upon the value of the input parameters, the redundant computers must produce output values which are different The instantaneous amount of channel divergence recorded at the voter is dependent upon the absolute difference of the variable value between the time it is evaluated by Computer 1 and Computer N (how fast the parameter is moving), and how the parameter is used by the control law

Two factors dealing with sampling rates affect channel divergence the absolute real-time sample rate and the relative sampling rate of one computer to another The first factor, in conjunction with sampling skew, determines the absolute difference between the value sampled by Computer 1 and the value sampled by Computer N for state variables which are dynamic For a given variable dynamics and sampling skew, increasing the sampling rate for all of the redundent computers will cause the absolute value difference between samples to decrease. Regardless of inter-computer skew, the amount of channel divergence decreases as sampling rate increases

If the redundant channels have different or dynamic (e g , data driven programs) sample rates, the sampling skew of Computer 1 to Computer N will vary with time The closer the times are to each other, the slower the skew varies This can affect channel divergence by causing the relationship of the redundant channels to change with increasing time Depending on the dynamics of the sampled data, this can either increase or decrease channel divergence

Use of free integrators– "Integral Control" is an extremely useful tool to the control systems designer Its main use is to control long-term variations in or compensate for incomplete definition of the state of the plant being controlled It provides zero-error, steady-state control of the plant Integrators also contain the sum of all past events occurring at the input to the integration process If the control laws processed by the redundant computers of figure 2 contain free integrators, then the sampling, skew and personal time differences will cause the output values produced by the integrators to vary effectively in a random fashion For instantaneous input differences, the same may be said for lead/lag or constant-gain circuits The difference with the pure integrators is that when the input difference goes to zero, the difference between the integrator outputs will not The integrator will "remember" the past history and maintain the difference indefinitely Given that the process of channel divergence owing to integration is effectively a random one, scenarios can be easily constructed wherein the integrators cause the channel divergence to increase to indefinitely large values Lead/lag and constant-gain elements do not contribute to long-term or steady state channel divergence

NULLIFYING CHANNEL DIVERGENCE – A METHODOLOGY

The factors which cause channel divergence are a function of the practical constraints in producing redundant (replicated) elements, and the skew/sampling relationship between them These involve the method of input data collection of each redundant member In cycling through the transfer function, which must be identical from redundant computer to redundant computer, output values are produced which are predicated on the sampled value of the input At any given instant, the redundant values produced by independently operated redundancy will be different To produce unambiguous results, the tracking error must be kept within acceptable limits Unambiguous means, for example, that the channels track to within 1% of full scale on average and the instantaneous variations cannot exceed 4% of full scale Also the time to declare a fault can be extended from one or two frames to several seconds of disagreement Nothing is lost through relaxation of the failure-detection parameters The fault-tolerant structure will not propagate improper or faulty data What is gained is a relaxed failure criterion which permits the fault-tolerant system designer to employ design techniques that do not require common elements in the system structure As the results do not have to agree exactly, different application programs, different voting routines, dissimilar hardware, and independent redundant elements may be employed Common latent design faults will not then exist if the concept of static redundancy management is also used

Use of Asymptotically Stable Controllers

The differences which are inherent between independently operated redundant computers cannot be allowed to cause channel divergence which exceeds a reasonably small time/magnitude threshold at the voter The threshold need not be static, but can be varied as a function of the dynamics of the voted variables Instantaneous differences caused by the mechanics of data sampling and time generation can be minimized by specifying sufficiently tight tolerance on the hardware that constitutes these functions This specification is the one item over which the designer has good control Free, pure integration terms cannot be allowed in the control system design However, the integration terms can be approximated by lead/lag elements with long time constants What is lost is the steady state zero tracking error feature of control loops which employ free integrators What is gained is that the lead/lag element will "forget" the short-term differences between redundant channels Instantaneous differences will still exist, but their effects on channel output will be reduced to zero as time passes

A laboratory simulator has been developed which provides insight into the mechanics of the independently operated fault-tolerant processes It is a computer model of a dual-redundant, fault-tolerant system Redundant-element cycle times, initial skews, control-system parameters, and more may be varied by the operator The simulator outputs as an x-y plot the current state of the input (Input) and plant (System Output), the values produced by the redundant control elements (Output A, Output B) and the difference between the outputs of the two elements (Error A-B) versus time into each run Currently, only two simple voting algorithms have been modeled (1) Voter Output equals Channel A Output, and (2) Voter Output equals the average of the current values of Channel A and Channel B outputs.

Two models of a simple redundant control system run on the simulator are displayed in figure 4 The plant being controlled in this example is represented by a 1/S+1 term A variety of waveforms may be input to the system. The control system is built as a unity-gain feedback system with two redundant controllers and one voter Each controller consists of two control elements in the forward loop – the first a simple constant gain term The second may be a free-integration term (1/S) or a lag term (1/S+e) as determined by the operator Plots of system responses to varying parameters of loop times, skews, voting algorithms, and input wave shapes were made The performance of the control elements using free integrators is compared to the performance of the control elements which use lag elements (note that this controller is asymptotically stable) In the asymptotically stable control system, channel divergence which results from the redundant channels responding to different input data will disappear with the passage of time However, a price must be paid for the improvement in performance. The steady state error is not zero Typical values for *K*0, *K*1, and ϵ are, respectively, 2 1, 0 2, and 0 01 or 2 8, 4 0, and 0 1 This will leave the system with a 0 5% to 2 0% steady-state error This steady-state error is sufficiently small that it is within the overall accuracy of most flight systems Additionally, new control system designs can be structured to compensate for the error

Plots are presented to show the response of the two types of control systems to various inputs These plots demonstrate the effect which sample skew and sample rate, and bias, scale, and nonlinearities of the sampling interface have on channel divergence The response of the integral control system to a step input is shown in figure 5 The voting philosophy adopted here is "choose Channel A " Channel A sees the step first and produces an output for the plant The state value for the plant, which has started to move, decreases the magnitude of the error signal seen by Channel B when it samples the variable 20 ms later Because Channel B always samples 20 ms after Channel A (their clocks are highly accurate and are running the same application program), Channel B consistently sees smaller values for the error signal during most of the transient response time When the system has achieved its steady state, a constant channel divergence of about 15% of the step size has been introduced into the system At 10 0123 sec, the value of the input is returned to zero The relationship of sample time to system dynamics is important. If the input in figure 5 had been returned to zero at exactly 10 000 sec, the steady state channel divergence generated by the step input would go to zero However, using the scenario of figure 6, bringing the input back to zero causes an increase in the steady-state channel divergence to 30% of the amplitude of the original step Notice that the system has returned to its original null state, but the output values of the redundant channels have become highly divergent The steady-state channel divergence is due solely to the presence of the integrator term (4 0/S) "remembering" the different sequence of input events as seen by each channel

Now, compare the response of figure 5 to that of figure 6 Exactly the same conditions and sequence of events were established for this simulation, except that the integrator has

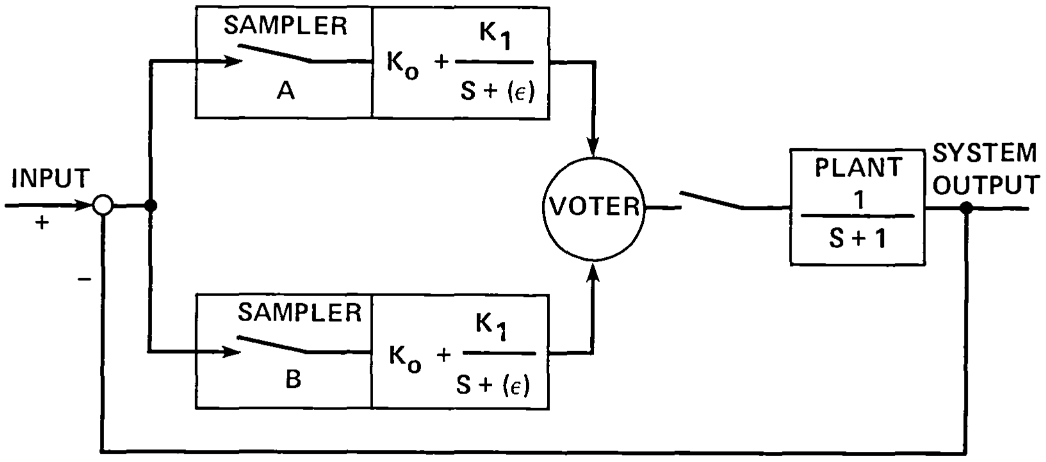
SAMPLER
A
$K_o + \frac{K_1}{S + (\epsilon)}$
INPUT
+
–
VOTER
PLANT
$\frac{1}{S+1}$
SYSTEM OUTPUT
SAMPLER
B
$K_o + \frac{K_1}{S + (\epsilon)}$

Figure 4.– Fault-tolerant system using integral control.

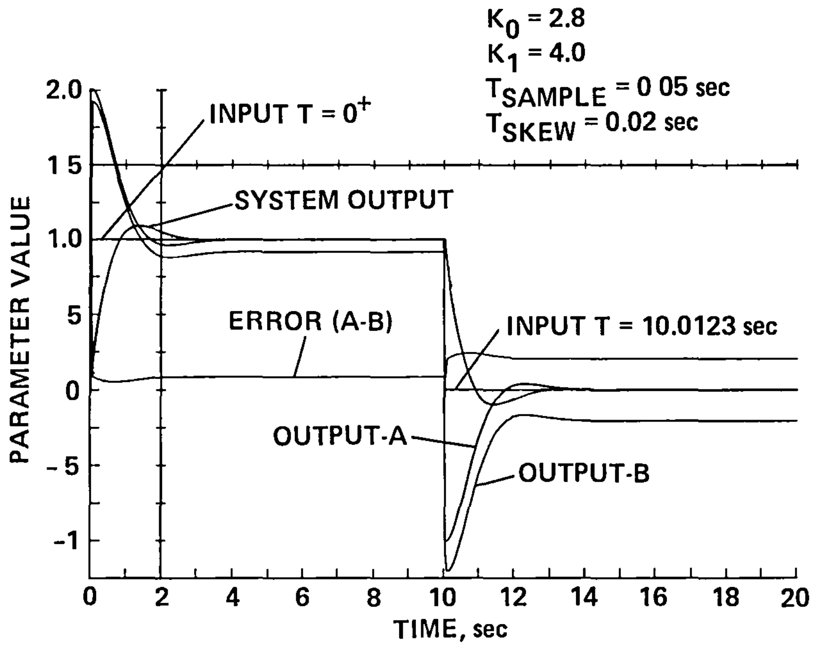

Figure 5 – Integral system (fig 4) step response

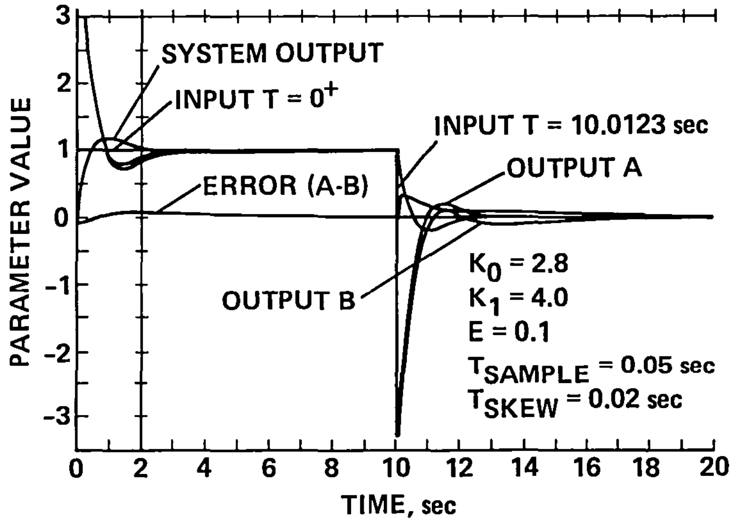

Figure 6 – Asymptotic controller step response (fig 5)

been replaced by a (4 0/S + 0 1) term The response of this term is slightly slower than that of the integrator, and the steady-state gain has decreased from infinity to 40 0 – but notice the significant effect this slight modification to the integration term has made The channel divergence generated by the asymptotically stable controller at Time = 0 0+ is much less than that generated by the integral control system Further, the channel divergence decays to zero as Channel B "forgets" the different sequence of input events and begins to operate solely on the currently available information The response of the plant (system output) to the two controllers (integral versus asymptotic) is virtually indistinguishable At 20 0 sec, the asymptotically stable redundant controllers have reduced the channel divergence to 0 0% of the value of the step (versus 30% for the integral control system). The voter must tolerate instantaneous channel divergence which may be very large (three units for the first 20 ms after the advent of the step in figure 6) The average channel divergence for asymptotically stable redundant controllers, however, can be made very small

Figure 7 shows the response of the asymptotic system to noise impulses of unit amplitude This amplitude could represent sensor noise with magnitude equal to one or two bits of digital resolution The sample times for the two channels was set at 50 ms with Channel B sampling 20 ms later than Channel A The simulation was run such that only Channel A sampled the noise impulses, which occurred every 500 ms The voting philosophy employed was "Choose A " The channel divergence, which was zero at the start of the simulation has reached a limit of about 0 8 unit, or less than the size of the noise impulses The channel divergence of the integral system (not shown) increases by 0 01 unit at each impulse The size of the integral channel divergence would eventually trip the failure threshold of even the most relaxed voter Figure 7 demonstrates that even if the impulse differences enter the asymptotically stable, fault-tolerant system faster

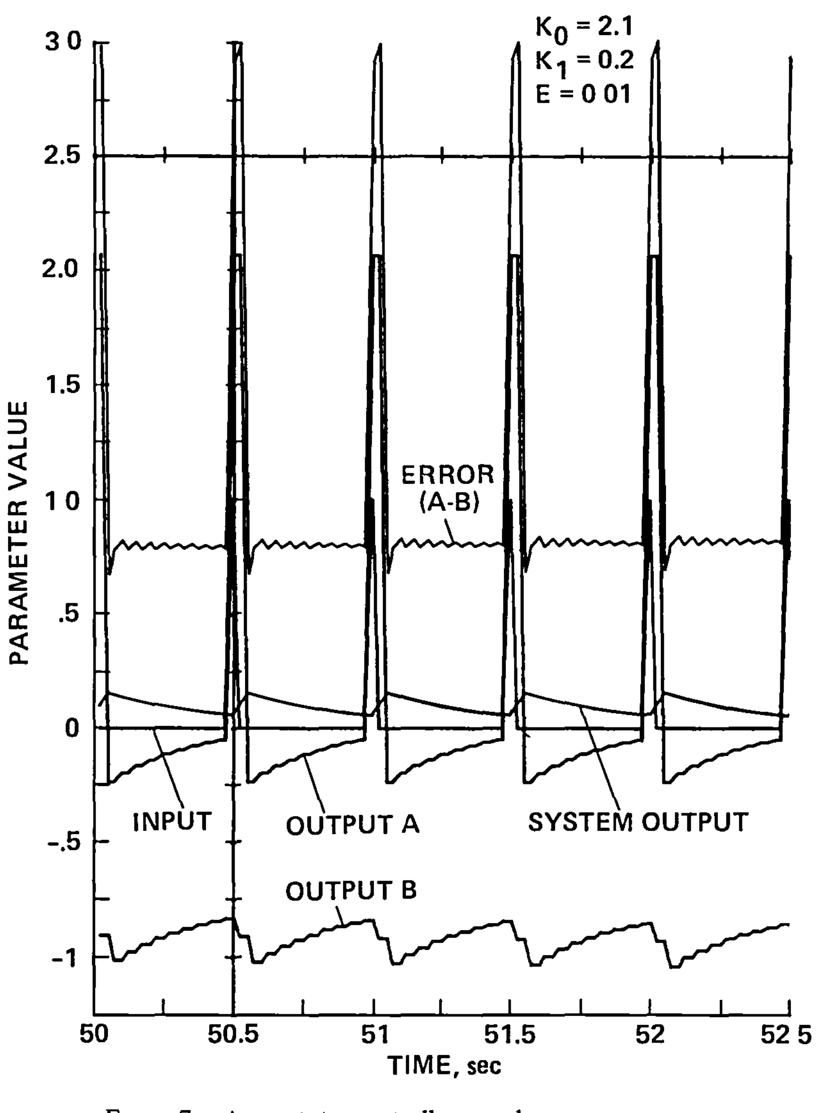

Figure 7 – Asymptotic controller-impulse-noise response

than the asymptotic stability can wash them out, the channel divergence will reach a limit and will not grow without bound Notice that an "intelligent" voter could successfully determine that neither channel has failed As the channel divergence is constant, the intelligent voter could deduce that the wave shape of the redundant channels must be identical and logically assume that neither channel is inoperative or failed Instantaneous change in the channel divergence is again excepted Note that even if the channels were totally synchronous, the channel divergence caused by the variation in sampling interfaces cannot be avoided without cross communication of the input data, which thus destroys channel independence

Increase Sampling Rate

As stated earlier, the channel divergence which can be expected from independently operated redundant elements is heavily dependent upon the dynamics of the input relative to the sampling rates of the redundant elements if the samples are not taken concurrently This is brought forth in figure 8 and figure 9 In the former, the sampling rate of both redundant elements is 50 ms with Channel B's sampling time skewed 20 ms after Channel A The voter algorithm is "Choose A " A sinusoidal input is injected into the system The dynamics of the input and plant are sufficient to cause the channel divergence to oscillate about zero with a peak magnitude of about 0 25 unit Notice that as the rate of change of the input decreases, the value of the channel divergence decreases thus clearly demonstrating the sensitivity of channel divergence to input dynamics The simulation shown in figure 9 is equivalent to that of figure 8 except the sampling period of the redundant elements has been decreased to 10 ms Notice that decreasing the sampling period to 10 ms decreased the channel divergence to 20% of its original value This demonstrates that the channel divergence is also sensitive to the sampling rate

Control Time

The time at which state variables are sampled is of utmost importance to the amount of channel divergence which will be experienced To a lesser extent, the relative variation in timekeeping between elements will also affect channel divergence Obviously, if the redundant channels can be made to sample at nearly the same time, channel divergence will be created only as a result of the variations in the sampling interfaces discussed above This can be accomplished without resort to cross communication between channels Each redundant element can be given access to a redundant clock whose time data enter the system in exactly the same way as any sensed variable If the redundant elements are programmed such that state variables are to be sampled at each 50-ms boundary and if the latency of the redundant time data is small, it can be shown that the sampling time of the redundant elements can be made to approximately converge The coincidence of the sampling would depend solely on the uncertainty in each element of the actual value of time Controlling time will eliminate the drift of time between computers for long-duration missions The application programs would return to being frame-based However, the redundant elements are still independent entities Channel divergence is significantly decreased and the use of asymptotically stable controllers is still required

EXPERIMENTAL RESULTS

A working example of a UFTCS system has been configured and tested at NASA Ames Research Center It is a microprocessor-based, quadraplex-redundant control system configured with independently operated redundant elements Sampling times were deliberately made highly asynchronous with cycle times of the application program being data driven and varying from 35 to 52 ms Total real-time control of all four axes of a UH-1H helicopter is provided both in simulation and flight test

Laboratory and manned simulation testing of the device have proven highly successful Several hundred simulated "flight" hours have been logged to date The four channels have demonstrated acceptable, votable levels of channel divergence Figure 10 shows an expanded view of the four redundant controls for the pedal axis during 18 sec of a 500-sec flight in the manned flight simulator The voting algorithm employed was a version of midvalue select The control theory employed was that described by Meyer and Cicolani in reference 6 As can be seen, the redundant control values generated remain well with 0 05° of each other Over the entire flight, and other flights, the redundantly generated control values for each of the four axes remain within 0 1° of being consistent Detailed information concerning the test system structure and operation is the subject of a following report

Calculated Reliability of the Test System

A detailed reliability analysis of the UFTCS structure was conducted by Curry/Frey of Search Technology and VanderVelde of MIT (ref 10) The hardware developed for the test system was used as a model, and the study considered potential configurations combining quadruplet, pentad, and hexad members Voting philosophies considered that simple voters would allow the system to fail to two operating, and "smart" voters could allow the system to fail to one remaining operative

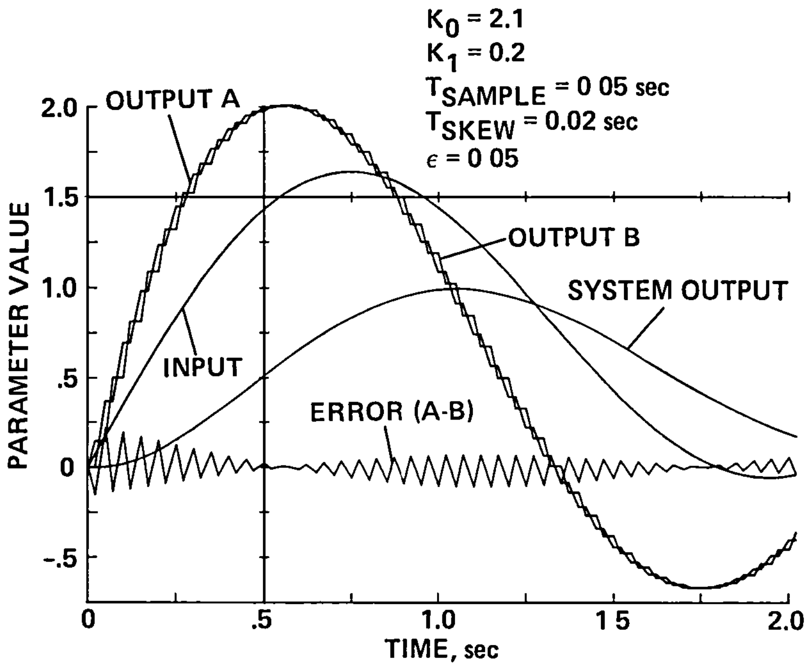

Figure 8 – Asymptotic controller-sine response with slow sampling rate

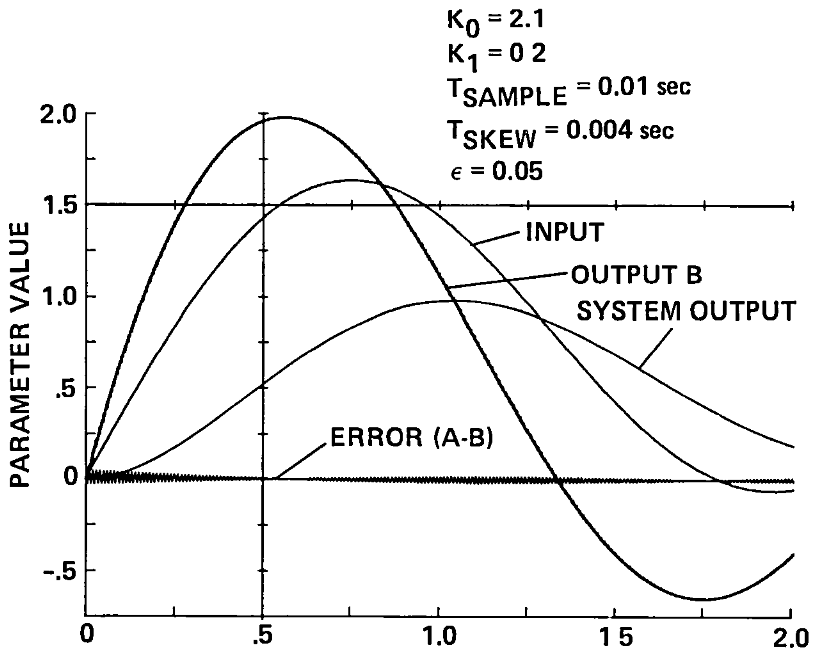

Figure 9.– Asymptotic controller-sine response with fast sampling rate

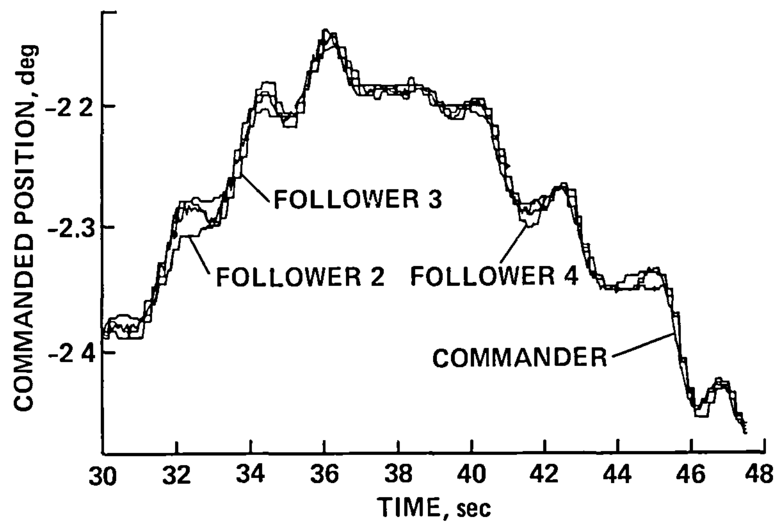

Figure 10 – UH-1H yaw control valves

The results of the analysis show that the probability of system failure for the first 10 hr of a helicopter flight using a pentad UFTCS (excluding the unreliability of sensors and actuation hydraulics) with simple voters is $8\ 6 \times 10^{-11}$ The system is based on microcomputer technology with the total volume of a pentad UFTCS controller being less than a small suitcase, with the elements dispersable throughout the controlled environment In the air-transport environment, under the same conditions, the probability of system failure is $2\ 9 \times 10^{-11}$ In space applications, such as the Space Station, the two week unattended probability of failure is $2\ 4 \times 10^{-7}$ As the system contains virtually no common elements and is not subject to latent common design failure, the delivered reliability of a UFTCS system should approach the calculated figures

CONCLUSIONS

Fault-tolerant control systems can be configured from independent, autonomous, redundant elements Made from dissimilar hardware, processing different versions of application programs, and controlling system-fault status through the use of static redundancy management, these systems assume the property of ultrareliability Systems so structured do not possess sources of latent common-design faults, and their demonstrated field reliability will approach predicted system reliability

Channel divergence normally encountered in fault-tolerant systems using independent redundant elements is caused by sampling and time differences between channels and the use of integral control theory Based on experimental evidence, channel divergence can be decreased to levels which readily voted through (1) the use of asymptotically stable control laws, (2) sampling rates which are adequately rapid relative to the frequency response of the total system, and (3) forcing sampling times to converge

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, California, 94035, August 22, 1984

REFERENCES

1 Dunn, W , Johnson, J , and Meyer, G A Fault Tolerant Distributed Microcomputer Structure for Aircraft Navigation and Control Fourteenth Asilomar Conference on Circuits, Systems, and Computers, Asilomar, Calif , IEEE Cat no 80CH1625-3, Nov 17-19, 1980

2 Yeh, E Applied Computation Theory, Analysis and Modeling Prentice Hall, 1976, pp 352-359

3 Wensley, J , Lamport, L , Goldburg, J , Green, M , Levitt, K , Melliar-Smith, P , Shostak, R , and Weinstock, C SIFT. Design and Analysis of a Fault Tolerant Computer for Aircraft Control Proceedings, IEEE, vol 66, no 10, Oct 1978, pp. 1240-1255

4 Hopkins, Jr , A , Smith III, T , and Lala, J FTMP – A Highly Reliable Fault Tolerant Multiprocessor for Aircraft Proceedings, IEEE, vol 66, no 10, Oct 1978, pp 1221-1239

5 Meyer, G , Hunt, L , and Su, R Design of a Helicopter Autopilot by Means of Linearizing Transforms Advances in Guidance and Control Systems, Advisory Group for Aerospace Research & Development AGARD-CP-321, 1982, pp. 4-1 to 4-11

6 Meyer, G., and Cicolani, L Application of Nonlinear System Inverses to Automatic Flight Control Design– System Concepts and Flight Evaluations, Theory and Applications of Optimal Control in Aerospace Systems, Advisory Group for Aerospace Research & Development AGARD-AG-251, 1980, pp 10-1 to 10-29

7 Luppold, R., Gai, E , and Walker, B Effects of Redundancy Management on Reliability Modeling Proceedings of the 1984 American Control Conference, June 6-8, 1984, pp. 1763-1770

8 Mackall, D AFTI-F16 Digital Flight Controls System Experience First Annual NASA Aircraft Controls Workshop, NASA Langley Research Center, Hampton, Va , Oct 25, 1983

9 Osder, S Generic Faults and Design Solutions for Flight-Critical Systems In Guidance and Control Conference, San Diego, Calif , Aug 9-11, 1982, AIAA Paper 82-38926 19018, Aug 1982, pp 509-518

10 Curry, R , VanderVelde, W , and Frey, P Reliability Analysis of an Ultra-reliable Fault Tolerant Control System NASA CR-166594, July 1984

1 Report No NASA TM-86017	2 Government Accession No	3 Recipient's Catalog No
4 Title and Subtitle ULTRARELIABLE, FAULT-TOLERANT, CONTROL-SYSTEMS – A CONCEPTUAL DESCRIPTION		5 Report Date July 1985
		6 Performing Organization Code
7 Author(s) Larry D Webster, Roger A Slykhouse, Lawrence A Booth, Jr , Thomas M Carson, Gloria D Davis, and James C Howard		8 Performing Organization Report No A-9876
9 Performing Organization Name and Address Ames Research Center Moffett Field, Calif 94035		10 Work Unit No
		11 Contract or Grant No
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13 Type of Report and Period Covered Technical Memorandum
		14 Sponsoring Agency Code 532-06-11

15 Supplementary Notes

Point of Contact Larry Webster, Ames Research Center, MS 210-5, Moffett Field, CA 94035
(415) 694-6526 or FTS 464-6526

16 Abstract

An Ultrareliable, Fault-Tolerant, Control-System (UFTCS) concept is described using a systems design philosophy which allows development of system structures containing virtually no common elements Common elements limit achievable system reliability and can cause catastrophic loss of fault-tolerant system function The UFTCS concept provides the means for removing common system elements by permitting the elements of the system to operate as independent, uncoupled entities Multiple versions of the application program are run on dissimilar hardware Fault tolerance is achieved through the use of static redundancy management

17 Key Words (Suggested by Author(s)) Fault-tolerant systems Reliability	18 Distribution Statement Unclassified - Unlimited Subject Category – 08

19 Security Classif (of this report) Unclassified	20 Security Classif (of this page) Unclassified	21 No of Pages 18	22 Price* A03

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document