NASA Conference Publication 2452

Joint University Program for Air Transportation Research–1984

Proceedings of a conference held at
Langley Research Center
Hampton, Virginia
January 18, 1985
Joint University Program for Air Transportation Research–1984

Compiled by
Frederick R. Morrell
Langley Research Center
Hampton, Virginia

Proceedings of a conference sponsored by
the National Aeronautics and Space Administration
and the Federal Aviation Administration and held at
Langley Research Center
Hampton, Virginia
January 18, 1985

NASA
National Aeronautics
and Space Administration
Scientific and Technical
Information Branch

1987
PREFACE

The Joint University Program for Air Transportation Research is a coordinated set of three grants jointly sponsored by NASA Langley Research Center, and the Federal Aviation Administration, one each with Massachusetts Institute of Technology (NGL-22-009-640), Ohio University (NGR-36-009-017), and Princeton University (NGL-31-001-252), to support the training of students for the air transportation system. These grants, initiated in 1971, encourage the development of innovative curriculums and support the establishment of graduate and undergraduate research assistantships and internships.

An important feature of this program is the quarterly review, one held at each of the schools and a fourth at a NASA or FAA facility. This latter review for 1984 was held at the NASA Langley Research Center, Hampton, Virginia, January 18, 1985. At these reviews the program participants, both graduate and undergraduate, have an opportunity to present their research activities to their peers, professors, and invited guests from government and industry.

This conference publication represents the fifth in a series of yearly summaries of the activities of the program (the 1983 summary appears in NASA CP-2451). The majority of the material is the effort of the students supported by the grants.

Three types of contributions are included. Completed works are represented by full technical papers. Research reported in the open literature (for example, theses or journal articles) is presented in an annotated bibliography. Status reports of ongoing research are represented by copies of viewgraphs augmented with a brief descriptive text.

Use of trade names or manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration or the Federal Aviation Administration.

Frederick R. Morrell
NASA Langley Research Center
CONTENTS

PREFACE ........................................................................ iii

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

INVESTIGATION OF AIR TRANSPORTATION TECHNOLOGY AT MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1984 ................................................ 3
Robert W. Simpson

AIRCRAFT APPROACH GUIDANCE USING RELATIVE LORAN-C NAVIGATION ................................. 9
Antonio L. Elias

PROBABILISTIC MODELING OF LORAN-C FOR NONPRECISION APPROACHES .............................. 17
John K. Einhorn

MEASUREMENT OF ICE ACCRETION USING ULTRASONIC PULSE ECHO TECHNIQUES ...................... 25
R. John Hansman, Jr. and Mark S. Kirby

OHIO UNIVERSITY

INVESTIGATION OF AIR TRANSPORTATION TECHNOLOGY AT OHIO UNIVERSITY, 1984 .................. 45
Richard H. McFarland

DIGITAL AUTONOMOUS TERMINAL ACCESS COMMUNICATION SYSTEM (DATAC) ......................... 49
Stanley M. Novacki, III

FIBER OPTIC DATA TRANSMISSION ....................................................................................................... 57
Steven T. Shreve

LORAN-C APPROACH CONSIDERATIONS .............................................................................................. 61
Robert W. Lilley

SATELLITE SURVEYING FOR A LORAN-C NONPRECISION APPROACH ...................................................... 65
Daryl L. McCall

ADVANCED MONITORING CONCEPTS .......................................................................................... 69
Robert W. Lilley

GPS ANTENNA DESIGNS .................................................................................................................. 73
Samuel J. P. Laube

PROCESSOR-CONTROLLED TIMING MODULE FOR LORAN-C RECEIVER ............................................. 77
Robert W. Lilley

A MICRO-COMPUTER-BASED SYSTEM TO COMPUTE MAGNETIC VARIATION ................................ 87
Rajan Kaul

A PROGRAM DOWNLOADER AND OTHER UTILITY SOFTWARE FOR THE DATAC BUS MONITOR UNIT ........................................................................................................................................ 113
Stanley M. Novacki, III
INVESTIGATION OF AIR TRANSPORTATION TECHNOLOGY
AT
MASSACHUSETTS INSTITUTE OF TECHNOLOGY,
1984

Robert W. Simpson
Flight Transportation Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts
SUMMARY OF RESEARCH

There have been three projects sponsored by the Joint University Program at MIT during 1984. Two projects are focused on the potential application of Loran-C in flying nonprecision approaches to general-aviation runways, and the third project involves basic research on aircraft icing. Four faculty members were responsible for supervising student research in the program: Robert Simpson, Antonio Elias, R. J. Hansman and Walter Hollister.

Aircraft Approach Guidance Using Relative Loran-C Navigation

This project was aimed at a flight test demonstration of an approach guidance system based on the relative navigation concept of Professor Antonio Elias. It uses the difference in TD's from those of the touchdown point to simplify and speed navigation computer processing, and to take advantage of the short-term accuracy of less than 100 feet for Loran-C. The project led to three sub-project areas concerning methods of displaying guidance information to the pilot, investigation of dynamic response of the Loran-C receiver, and possible smoothing with rate gyro data, and the need to study the micro-distortion of the Loran-C field discovered in surface measurements over open terrain.

The system being constructed for flight testing by Norry Dogan is based on a Micrologic M3000 Loran-C receiver especially modified to output TD's, and TD's in digital form every 12 GRI, and whose tracking filter constants can be set by the experimenter. A flight test pallet has been constructed to carry the receiver, the memory storage system, power supply, auxiliary display, etc. Three different display devices have been constructed. The original display was an LED bar-graph display of cross track distance and velocity, and an LED numeric digital display of range and desired altitude. This system will now display these data to the experimenter from the flight test rack. For the pilot, a CRT cross-pointer display has been constructed, driven by software resident in an Apple II computer. This has the advantage of flexible formats and ease in programming software changes for the pilot's display. The third display was developed by another student, Lee Marzke, using a Sharp LM-24003G Graphics Display. This is an 128x240 pixel display, which uses liquid crystal technology, and is roughly 4x8 inches in size. A special, programmable controller using the MCG-8085 processor was built to accept data over an RS-232 line and convert it to a cross-pointer display, with TO-FROM, altitude, and airspeed. It was decided to use the CRT display for flight test and to retain this display for other applications in future research. Flight testing of the system was planned for early in 1985, at a few selected sites in the 9960 Loran-C chain.

In the course of constructing the system, ground surveys were conducted using the Loran-C receiver and Apple Computer. This discovered that there were unexplainable micro-distortions of the Loran-C signals over a grid of roughly 50 meters. The averaged position deviation from the grid was repeatable and of the order of 10 meters. This would appear as noise to the receiver if it exists above the Earth's surface in the approach path to the runway. This has triggered investigations of methods to survey the approach airspace using the flight test pallet mounted on a tethered balloon or kitoon. The size of these micro-distortions will affect the performance of the Loran-C guidance system in estimating cross track velocities and thus its dynamic response in displaying cross track deviations to the pilot. Further research is planned by Professor Antonio Elias to gather more data on these microdistortions.
Probabilistic Modelling of Loran-C Errors for Nonprecision Approaches

The goals of this project are to develop a mathematical model which will predict the probability that an approach flown to a runway with a particular Loran-C receiver will fall within a given standard (such as FAA AC90-45A). Flight tests will be used to validate the model performance. During 1984, the model was developed and programmed to handle both short-term update (5 minutes) and long-term update (6 weeks) cases. It accounts for runway heading, the chain and slaves used and their geometry and distance relative to the airport, the current SNR (signal-to-noise ratio), and the seasonal history of Loran-C deviations at nearby monitoring stations. Several models for predicting long-term drift have been examined. The model produces a bivariate normal error ellipse for the position of the aircraft on approach to the runway. The percentage of this ellipse which falls within any other geometric standard can be computed.

This research is being performed by John Einhorn, with supervision by Walter Hollister.

Aircraft Icing Research

This research is a continuation of work done originally by R. J. Hansman as a graduate student in the Joint University Program. Experimental work in 1984 has focussed on two main areas:

1) Measurement of droplet trajectories and droplet impingement/runback characteristics

2) Measurement of real-time ice accretion using ultrasonic pulse-echo techniques

DROPLET RESEARCH

In order to experimentally measure the trajectories of droplets, a controlled method for producing droplets of the known, repeatable initial conditions to be "injected" into the free-stream flow at any desired location has been successfully developed. The system utilizes a syringe reservoir connected to an extremely fine bore length of tubing, described in the next section.

The syringe reservoir supplies water to a 0.005 in. (ID) tube approximately 2 in. long. The tube is extended into the free-stream flow at a 30 degree angle. Droplets bead at the end of the tube until they are drawn off by the free-stream drag. In order to minimize the droplet size at which this occurs, the tube tip is coated with wax to increase the droplet/tube contact angle. Droplets can be produced at a rate of several per second using this technique, although the current trajectory measurement scheme does not require such a "stream" of droplets. The approach being taken to measure the droplet trajectories is to use a double strobe light pulse to illuminate the droplet at two points along its trajectory. Since the time between these points is fixed by the strobe pulse interval, the velocity of the droplet between the points can be obtained by measuring the distance travelled in this interval from a 35-mm photograph. By moving the injection point across the entire injection plane upstream of the body, the velocity field around the body can be completely mapped. One of the distinct advantages of the developed injection technique is that the initial droplet conditions can be accurately measured and thus used as inputs to existing trajectory codes to aid verification and/or development of trajectory codes.
for complex bodies such as glaze ice shapes. Since the droplet is pulled off the injector tip, the initial velocity of the droplet is zero; furthermore the initial co-ordinates of the droplet are also those of the injector tip. The initial size of the droplet is obtained from a triggered photograph taken immediately after the droplet release occurs.

REAL-TIME ICE ACCRETION MEASUREMENT

Experiments to measure ice growth on a cylinder have been conducted in the Icing Research Tunnel (IRT) at the NASA Lewis facility in Cleveland, Ohio. A 4 in. (OD) cylinder approximately 12 in. long was suspended vertically from the roof of the IRT and exposed to a variety of icing conditions at the velocities from 110 mph to 230 mph. The cylinder was instrumented with 4 ultrasonic transducers to measure the real-time ice accretion using the pulse-echo technique. The transducers were all of the longitudinal wave type, three being 0.5 in. diameter transducers and a 0.25 in. diameter delay-line transducer. The transducers are mounted flush with the cylinder surface and are thus "non-intrusive." For the series of tests performed at NASA Lewis three of the transducers were mounted on the cylinder stagnation line and the remaining (0.5 in.) transducer was mounted 25 degrees off-axis to allow simultaneous measurement of horn growth rates and the growth rate on the stagnation line. Data were collected by video-taping the oscilloscope output from the transducers and pulser/receivers. Although data reduction from the entire series of tests is still under way (over three hours of real-time ice accretion data were recorded in 38 tests in the IRT), initial results have shown the following:

1) The ultrasonic pulse-echo system is capable of measuring ice thickness in real time over a wide range of icing conditions. Ice thickness data were successfully recorded for all the tests conducted, although ice formations at warmer temperatures (+27°F) sometimes resulted in a cavity forming over the transducer at some point in the ice growth, after which the transducer can no longer measure further ice accretion (due to the reflection of the pulse signal from the air cavity). At the colder temperatures (+10°F to -10°F) the pulse-echo system performed extremely well, producing a discrete, large amplitude echo at the moving ice/air interface.

2) The optimum transducer frequency appears to be 5 MHz, although even higher frequencies have yet to be evaluated. The optimum transducer size for ice accretion measurement is driven by two conflicting requirements. A small diameter transducer is relatively insensitive to the irregular ice/air surfaces that are characteristic of ice accretion at warm temperature, and hence produces a "narrower" interface echo than a larger diameter transducer would under similar conditions. However under the same warm (glaze) icing conditions, air cavities and inclusions were observed which

1A temperature range of 37 degrees Fahrenheit was covered, with runs from -10 degrees Fahrenheit to +27 degrees Fahrenheit. Liquid water contents were varied from 0.6 g/m³ to 2.4 g/m³. Droplet diameter was also varied from 12 micrometers to 20 micrometers.

2The ice thickness resolution achievable is fundamentally limited by the ice thickness variation across the ultrasonic field produced by the transducer, and hence by the diameter of the transducer (since the ultrasonic field is a collimated beam of the same diameter as the transducer).
threaten to "blind" a small diameter transducer to further ice accretion and thus dictate either a transducer large enough to see around these cavities or else several smaller transducers operating in parallel.

3) In addition to providing information on the thickness and surface variation of the accreted ice formation, the ultrasonic system also clearly shows the presence of liquid water on top of the ice layer, and it is hoped that further measurements on the data recorded will allow the thickness of this layer to be determined. At the colder temperatures no such water layer was observed, from which it can be concluded that the droplets are indeed freezing "on impact," as in the classical rime ice model.

FURTHER WORK

Droplet Studies

Having developed and tested the droplet injection technique, work is now under way on the development of a laser triggered multiple strobe setup. By breaking a laser beam aimed across its path, the droplet triggers a multiple strobe arrangement that allows the desired point velocity measurement to be photographically recorded as described earlier. In addition the laser trigger will also be used to study droplet impingement on bodies and runback characteristics. This can be accomplished by aiming the laser beam on the desired impingement point to be studied, and triggering from the droplet's occlusion of the reflected beam signal.

Ultrasonic Pulse-Echo Developments

Following the successful series of tests at NASA Lewis, work was conducted to allow flight testing of the ultrasonic system aboard the NASA Lewis icing research Twin Otter aircraft. Signal processing algorithms to track the ice/air interface echo under all icing conditions are being studied with a view to building a real-time data acquisition system providing ice thickness and ice accretion rate output. Existing, commercially available pulse-echo thickness readers will also be investigated for this purpose. Data reduction from the IRT tests will continue and were presented at the 23rd Aerospace Conference (Hansman R. J., and Kirby M. S., "Measurement of Ice Accretion Using Ultrasonic Pulse-Echo Techniques," AIAA 85-0471). The real-time ice accretion data taken will also be compared with existing computer model predictions for "identical" icing conditions on a cylinder. Additional theoretical modelling of the ultrasonic wave propagation in different ice types will also be undertaken and analytic models developed for wave propagation and reflection characteristics. Further refinement of the speed of sound in ice will be made from the results of the Lewis tests. It is hoped that fundamental information on the elastic properties of different ice types may be obtained from this velocity data.

Bibliography

AIRCRAFT APPROACH GUIDANCE
USING
RELATIVE LORAN-C NAVIGATION

Antonio L. Elias
Flight Transportation Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

PRECEDING PAGE BLANK NOT FILMED
The experiments carried out during 1984 at MIT focused on two aspects of LORAN-C relative navigation that will impact system performance at the sub-microsecond level of accuracy: tracking loop bandwidth and localized field deformations. Figures 1 to 3 show the result of a basic experiment illustrating both these effects. A Micrologic Model 3000 receiver mounted on a vehicle is accelerated to a constant speed and then decelerated to a full stop between points A and B. After a 2-minute pause, the vehicle is reversed and the maneuver repeated in the opposite direction from points C to D. The first measured set of TD's in the sample was taken as the definition of point "A", and all subsequent positions plotted relative to this datum. Features of interest in Figure 1 include: a) the random noise in the plotted position is sufficiently low to discern the 10-m width of the road over which the test was performed; b) the navigated position overshoots the actual position during braking (and lags it during acceleration); and c) a reduction in signal-to-noise ratio (SNR), as occurred at the end of the run while the vehicle was standing at point D, substantially increases the navigated position random noise.

The effect of vehicle acceleration and deceleration is better seen in the plot of Figure 2; here, measured and position-derived TD's are plotted against time. As can be seen, the tracking loops can be very well approximated by second-order linear systems, in spite of the discrete digital implementation used in this receiver. The trade-off between random noise and lag/overshoot in the navigated position implied by selection of the tracking loop bandwidth is illustrated in Figure 3, where the test of Figure 1 is repeated with a tracking loop filter bandwidth four times larger than that used in Figure 1.

If the tracking loops can be modeled as simple second-order linear systems with constant damping ratio (0 to 5 was assumed after Figure 2), then it is possible to determine analytically what the root mean square of the random component of time difference will be as a function of filter bandwidth. Figure 4 shows this relationship, where the filter is characterized by its time constant and acceleration gain \(\text{-3dB bandwidth } = \frac{1}{\pi} \sqrt{\frac{K}{A}}, \quad \tau = \frac{1}{\sqrt{\frac{K}{A}}}, \quad \frac{1}{\pi \omega}\). Also plotted in Figure 4 are some experimental points obtained with the Micrologic 3000 receiver in a single, stationary location with various levels of SNR and two tracking loop bandwidths. As it can be seen, there is general agreement between experimental results and the simple linear model.

Such simple linear models, if assumed valid, can then be used to optimize the tradeoff between acceleration-induced errors, which increase with reduced tracking loop bandwidth, and the random component of TD, which increases with increased tracking loop bandwidth. Figure 5 shows the 2-\(\sigma\) TD error versus SNR, resulting from choosing a bandwidth such that the acceleration-induced bias (at the level of acceleration indicated by the curve label) equals the 2-\(\sigma\) TD random noise at each SNR value. Marked along the curves are the optimal loop gains in each case. As can be seen, the resulting system performance is significantly lower than that which would be expected from a cursory look at Figure 4 (0.1 \(\mu\)sec is approximately 60 feet or 20 meters).

Performance at the sub-microsecond level of accuracy, as seems to be feasible in view of the results of Figure 5, requires that the Loran-C signal field itself be consistent to that accuracy in the area around the datum point in which operations will be carried out. A survey of measured TD's in the vicinity of Hanscom AFB, MA, indicates that there may be local distortions in the stationary Loran-C electromagnetic field much larger than 1 microsecond. Figure 6 shows vector differences between TD-derived and map-derived antenna positions around the base; the reference
TD's were selected so as to zero the sum of these vector differences, and thus can be interpreted as an "area average" ASF correction. Figure 7 summarizes those vector differences in a single plot; of interest is that the dispersion seems largest in the direction of the gradient of the second pair of stations, which happen to have the largest SNR in the area. As it can be seen, the 2-σ error is about 1 microsecond.

It was postulated that these large stationary differences might have been caused by large terrain features, metallic structures (e.g. hangars) and CW interference (e.g. power lines) in the area. Thus a second set of experiments, depicted in Figure 8, was carried out over a flat, obstacle-free area. Two hundred and eighty data points were taken at each location in a grid 300 ft by 200 ft in side, at 100 feet intervals. The resulting sampling error of estimate of the mean of each sample is less than 1 m. As can be seen, the differences between actual and (averaged) measured position are of the order of 10 to 20 meters (the point with zero difference was taken as the "anchor point" and the average TD's used as the reference TD's). We have labeled this repeatable distortion field "microdeformation" and intend to further explore this effect, both at ground level and at altitude, using NASA-supplied kitoons (kite-balloons).
Figure 1. Vehicle-mounted relative Loran-C test (loop time constant was 18 seconds).

Figure 2. Measured and theoretical time differences and measured SNR's for the test of Figure 1.
Figure 3. Vehicle-mounted test (loop time constant was 4 seconds).

Figure 4. Theoretical and measured time difference variances as a function of SNR for various loop time constants.
Figure 5. The 2-σ error as a function of SNR for balanced-design loop time constants.
Figure 6. Differences between actual and relative Loran-C derived positions around Hanscom AFB, MA.
Figure 7. Distribution of the position errors of Figure 6.

\[ \sigma_x = 84.6 \text{ m} \]
\[ \sigma_y = 100.0 \text{ m} \]

Figure 8. Actual and relative Loran-C derived position over open terrain.
PROBABILISTIC MODELING OF LORAN-C
FOR NONPRECISION APPROACHES

N87-22607

John K. Einhorn
Massachusetts Institute of Technology
Cambridge, Massachusetts
Abstract

The overall idea of this work is to predict the errors to be encountered during an approach using available data from the U.S. Coast Guard and standard normal distribution probability analysis for a number of airports in the North East CONUS.
The work consists of two parts: an analytical model that will predict the probability of an approach falling within a given standard (AGS, a given standard), not necessarily AC90-45A), and a series of flight tests designed to test the validity of the model.

**GOALS**

- Develop a mathematical model that will predict the probability that an approach using LORAN-C will fall within a given standard (AGS).

- Conduct flight tests to determine the validity of the model.
This figure outlines the mathematical model. A computer program takes as input geometrical considerations such as receiver and transmitter location, and propagation factors such as update frequency (these factors to be explained in the next figure). From these, a position error ellipse is created. The probability of falling within the ellipse is known and can be adjusted by varying its size. Once along- and cross-track error standards are chosen, such as those in AC90-45A, these will be compared with the error ellipse, and a probability of falling within the standard will be computed.

MATHEMATICAL MODEL
This figure shows the completed work. The program that creates the position error ellipse has been completed and is in FORTRAN. It takes into account runway heading, touchdown point, and chain and slaves, all of which are self-explanatory. The program also needs expected standard deviations of the TD coordinates. These are broken down into long-term and short-term $\sigma$'s. The short-term $\sigma$ is on the order of 1 to 5 minutes, and the long-term $\sigma$ is on the order of weeks and months. The long-term $\sigma$ is a function of what is called "update frequency." This assumes that the TD's of the touchdown point are updated on certain intervals, such as every 8 weeks, with the publication of approach plates. Using two slaves and their respective $\sigma$'s, a position error ellipse is generated.

**COMPLETED WORK**

Program that computes ellipse semi-diameters.

- Airport name
- Runway number
- Runway heading
- MAP longitude
- MAP latitude
- Chain
- Slaves
- Update frequency
- Long term STD
- Short term STD

- Assumes
- Gaussian distribution

- Position Error Ellipse
A way must be found in the future to accurately predict the long term $\sigma$. Two routes to be investigated are 1) looking at seasonal variations in the Loran signal, and 2) looking at the U.S. Coast Guard "Double-Range difference (DRD)" model. In addition, the short-term must be predicted. Possibilities are:

1) Picking a $\sigma$ needed to achieve AGS and calculating necessary SNR to achieve that $\sigma$. In order to make the approach, the SNR must be at least that high.

2) Examine models for $\sigma$ as a function of distance from transmitters.

3) Use the "DRD" model to predict $\sigma$.

Finally, the probability that an approach will be inside AGS by looking at how much probability inside the ellipse is within the overlayed standard (most likely through numerical integration) must be completed.

**FUTURE WORK**

- Long term STD modeling
  - Seasonal changes
    - USCG "DRD" model

- Short term STD modeling
  - Pick a short term STD, minimum SNR to achieve that accuracy
  - Models for STD as a function of distance from transmitters
  - "DRD" model to predict STD

- Compute what % of ellipse is inside AGS
This is a simple "definition" of the Coast Guard "Double-Range difference" model, the key to which is the so-called "Double Range difference" (DRD).

\[
DRD = \{ R(s-p) - R(m-p) - [ R(s-sam) - R(m-sam) ] \}
\]
This figure outlines the basic idea for the flight tests.

1) Choose airports that have both good and poor probabilities of falling within AGS

2) Fly approaches to those airports using recorded data from the ILS localizer as a "control" reference

3) Eliminate error due to pilotage, using ILS data

4) Plot path that Loran says was flown.

The equipment needed to go up in the tests (for which a pallet is being constructed) includes:

1) Apple II Plus with disc drive and monitor

2) Loran-C (Micrologic ML-3000)

3) Power inverter

4) Gel-cell batteries to power equipment

FLIGHT TESTS

• Pick several airports with both "good" and "bad" probabilities

• Compare LORAN to ILS localizer error for X-track and use visual references for along track error.

Work in progress

• Building a pallet to carry equipment
MEASUREMENT OF ICE ACCRETION USING ULTRASONIC PULSE ECHO TECHNIQUES

R. John Hansman, Jr. and Mark S. Kirby
Dept. of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts
This figure shows the need for ice accretion measurement.*

Operational (rotary and fixed wing)
Avoidance & Escape (requires icing rate)
C & W
Management of Ice Protection Systems
Boots
Electrothermal

Research
Support Ice Modeling Efforts
Time Dependence
Document Meteorological Conditions
Certification

*Original figures in this paper not available at time of publication.
This figure depicts the basic concept of ultrasonic pulse-echo measurement. The thickness of the ice is related to the pulse-echo time $T_{P-E}$ by the speed of sound $C$.

\[
D = \frac{C_{ICE} T_{P-E}}{2}
\]
This figure shows a typical pulse-echo signal, showing multiple echoes from the ice-air interface.
The results of static calibration tests on a variety of NASA and MIT ice samples are shown in this figure. The speed of sound appears to be insensitive to ice type, with a value of approximately 3.8 mm/sec.
This figure is a schematic of the setup for ice accretion tests in the Icing Research Tunnel (IRT). Several transducers were mounted on a 4-in. diameter cylinder in the IRT. Echo patterns were videotaped during icing exposures under a variety of conditions.
The location of the test cylinder in the IRT is shown in this figure. It should be noted that the cylinder is not in the calibrated region of the tunnel.
This figure shows typical echo patterns for rime and glaze ice. The echo is broadened in the glaze case due to the irregularity of the air-ice interface.
This figure shows typical echo patterns for thin ice samples in which the pulse-echo time is shorter than the recovery time of the amplifier. The echo signal can be delayed by use of a delay line. For normal transducers, the pulse-echo time can be inferred from the time between multiple echoes.
A typical echo pattern for glaze ice with surface water present is shown in this figure. Surface waves in the water cause fluctuations in the echo pattern. This technique should allow accurate determination of the conditions under which runback occurs.
This figure shows ice thickness measured by the ultrasonic technique versus icing time for heavy icing conditions.
Shown below is the ice thickness measured by the ultrasonic technique versus icing time for light icing conditions.

4" DIAMETER CYLINDER
V=200 KTS
LWC=0.6 G/M³
MVD=12 MICROMETERS

ICING TIME (MIN)

ICING TIME (MIN)
This table summarizes the icing rates measured in the IRT, as well as the presence or absence of surface water.

**ICING RATES FOR A 4” CYLINDER IN THE IRT**

\( v=200 \text{ KTS (CYLINDER WAS NOT IN CALIBRATED REGION)} \)

<table>
<thead>
<tr>
<th>TEMPERATURE</th>
<th>+27°F</th>
<th>+10°F</th>
<th>-10°F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HEAVY ICING CONDITIONS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LWC=1.2 g/m³</td>
<td>0.75 MM/MIN</td>
<td>2.10 MM/MIN</td>
<td>3.15 MM/MIN</td>
</tr>
<tr>
<td>MVD=20 μM</td>
<td>GLAZE</td>
<td>GLAZE</td>
<td>RIME</td>
</tr>
<tr>
<td></td>
<td>Wet</td>
<td>Wet</td>
<td>Dry</td>
</tr>
<tr>
<td><strong>LIGHT ICING CONDITIONS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LWC=0.6 g/m³</td>
<td>0.55 MM/MIN</td>
<td>1.05 MM/MIN</td>
<td>1.05 MM/MIN</td>
</tr>
<tr>
<td>MVD=12 μM</td>
<td>GLAZE</td>
<td>Mixed</td>
<td>RIME</td>
</tr>
<tr>
<td></td>
<td>Wet</td>
<td>Dry</td>
<td>Dry</td>
</tr>
</tbody>
</table>
This figure shows the ice growth and echo patterns for a glaze ice run. The top trace is the delay line echo. The bottom trace is the normal transducer echo.
This figure shows the ice growth and echo patterns for a rime ice run. The top trace is the delay line echo. The bottom trace is the normal transducer echo.
This figure is an example of ice thickness versus time for a horn ice growth.

T=+27°F  LWC=1.2 G/M³  V=200 KTS

ICING TIME (MIN)
This figure shows ice thickness versus icing time for the multiple-echo resulting from an ice ridge growing over the transducer in glaze conditions.

V = 200 KTS  LWC = 0.6 g/m^3  MVD = 12 MICROMETERS
T = +27°F

RIDGE HEIGHT ★
MEASURED AFTER
3 MIN. EXPOSURE

ICE THICKNESS (MM)

ICING TIME (MIN)
INVESTIGATION OF AIR TRANSPORTATION TECHNOLOGY
AT OHIO UNIVERSITY, 1984

Richard H. McFarland
Ohio University
Athens, Ohio
INTRODUCTORY REMARKS

The year 1984 was one of transition, change, and growth for the Joint University Program for Air Transportation at Ohio University. The Loran-C work changed direction from that of receiver hardware development to the operational development of Loran-C for enroute navigation and non-precision approaches. Mr. Stanley M. Novacki III returned from his intern residency at NASA Langley Research Center and continued his work on the DATAC data bus monitor. During the summer, participation in a promotional effort resulted in the Federal Aviation Administration joining NASA Langley Research Center in supporting this program. This partnership allowed for increased funding to the JUP universities and, at Ohio University, allowed for the formation of three new research areas: global positioning system test-bed receiver development; fiber optic data bus applications in general-aviation aircraft; and advanced remote monitoring techniques.

One of the last hardware upgrades to the Ohio University Loran-C receiver was performed under the direction of Dr. Robert W. Lilley. The Memory Aided Phase Locked-Loop (MAPLL) developed for the O.U. Loran-C receiver was modified to reduce the number of devices required to control the MAPLL and to improve the efficiency and speed of the 6502 MAPLL device driver software. These modifications have been documented by Lilley in Ohio University NASA TM 90 (ref. 1).

One of the operational problems with using the O.U. Loran-C receiver for enroute navigation studies was due to the calculation of heading and distance to waypoints with respect to true north as opposed to magnetic north. Rajan Kaul, a JUP graduate intern, implemented a magnetic variation model for the continental United States, in FORTRAN IV, on Ohio University's IBM 370 mainframe computer and then in the 6502 machine language for use in the O.U. Loran-C receiver. This work is discussed in further detail by Kaul in Ohio University NASA TM 91 (ref. 2).

Rajan Kaul made a second contribution to the study of the operational enroute capabilities of Loran-C in his master's thesis, "Comparison of Great Circle and Rhumb Line Flight Paths in the Continental United States Using Simulation and Test Flights," also published in Ohio University NASA TM 93 (ref. 3). This paper discusses the discrepancies encountered between the two route structures, particularly over large distances, and the possible conflicts which may arise with their simultaneous use in the national airspace.

The work on the DATAC data bus monitor was initiated by the second NASA Langley Research Center/Ohio University Internship and continued through the work of that intern appointee, Mr. Stanley M. Novacki, III.
Novacki developed a program downloader, plus other utility software, for the DATAC bus monitor unit (documented by Novacki in Ohio University NASA TM 92, ref. 4). His later work involved the development of necessary tools, which with the aforementioned utility programs, provided the working base to complete the DATAC data bus monitor unit.

One of the new areas of research involved the siting of the Ohio University Gordon K. Bush Airport for Loran-C non-precision (NP) approaches. Dr. Robert W. Lilley, Associate Project Director, and Mr. Daryl L. McCall, JUP Project Engineer, have performed most of the required surveying tasks. The results of the survey data will be used in the calculation of the final approach fix (FAF) and approach path for a Loran-C NP approach. It was planned that Lilley, McCall, and an intern would perform flight tests for the purpose of evaluating the defined NP approach path.

Another new area of work involved the development of a reconfigurable GPS receiver test-bed. Work had already begun through the efforts of JUP graduate intern Mr. Samuel J. P. Laube who implemented and tested some antenna designs for use at GPS frequencies. Also, Ohio University planned to receive, as an equipment grant from the FAA Technical Center, Atlantic City, NJ, the dual channel GPS receiver hardware developed by Stanford Telecommunications, Inc. Upon receipt of this hardware, work was to begin to make the unit airworthy as soon as possible. Flight tests of the dual channel receiver were planned to examine the enroute capabilities of GPS.

The new effort to develop a fiber optic data bus was begun through work by JUP undergraduate intern Mr. Steve Shreve. This work was enhanced by corporate gifts from Aero Mechanisms, Chatsworth, CA and Electrosonics, Columbus, OH. The supporting gift items consisted of two blind encoding altimeters, two digital altitude displays, and an encoding transponder. This equipment, which normally requires a twelve conductor data bus, was to be serially interconnected by a fiber optic cable. Once the fiber optic hardware was developed and laboratory tested, several flight tests/experiments were to be considered.

The third area of new work involved the application of image processing techniques to monitor facilities remotely, such as antenna arrays, whose operation may be sensitive to various unexpected obstructions (e.g. fallen trees, ground slippage, vehicles, etc.). This work was to involve the use of Ohio University's COMTAL Vision One image processor with either the University's IBM/370 or the Department of Electrical and Computer Engineering's VAX 11/750. Digitized photographs, and later, actual data from navaids sites, were to be used to develop and demonstrate processing and data transmission techniques for successful monitoring.
REFERENCES


Hardware documentation is provided for the modified Loran-C timing module, which uses direct software control in determining loop sample times. Computer loading is reduced by eliminating polled operation of the timing loop.


A microcomputer-based implementation of a magnetic variation model for the continental United States is presented. The implementation computes magnetic variation as a function of latitude and longitude for general aviation receivers such as Loran-C.


A set of programs designed to facilitate software testing on the DATAC Bus Monitor is described.
DIGITAL AUTONOMOUS TERMINAL
ACCESS COMMUNICATION SYSTEM
(DATAAC)

Stanley M. Novacki, III
Ohio University
Athens, Ohio
In order to accommodate the increasing number of computerized subsystems aboard today's more fuel-efficient aircraft, the Boeing Company has developed the DATAC (Digital Autonomous Terminal Access Control) bus to minimize the need for point-to-point wiring to interconnect these various systems, thereby reducing total aircraft weight and maintaining an economical flight configuration. The DATAC bus is essentially a local area network providing interconnections for any of the flight management and control systems aboard the aircraft. The bus itself is passive: a twisted-pair, coax, or fiber-optic line with stub connectors for each DATAC terminal, the device which interfaces the various subsystems with the bus. DATAC uses a line-topology and CSMA (Carrier Sense, Multiple Access) protocol, much like the commercial Ethernet local-area network proposed by Intel and Xerox, which is in use in many large offices today. The main purpose of the Bus Monitor Unit is to receive all data which appear on the bus, reconstitute it into "real-word" parameters (e.g., altitude, heading, etc.), and display it in an easily interpreted form. This provides an indication of the overall operation of the bus without the liability of a total system failure caused by a central controller-type device, which could be disastrous at 35,000 feet.

The task of developing a Bus Monitor Unit can be broken down into four subtasks: 1) Providing a hardware interface between the DATAC bus and the Z8000-based microcomputer system to be used as the Bus Monitor; 2) Establishing a communications link between the Z8000 system and a CP/M-based computer system, which will be used as the Bus Monitor Unit console; 3) Generation of data reduction and display software to output data to the console device; and 4) Development of a DATAC Terminal Simulator to facilitate testing of the hardware and software which transfer data between the DATAC bus and the operator's console in a near real-time environment.

A. DATAC Terminal Interface Card:

The subsystem interface used on the DATAC bus terminals used the ZILOG Zbus system interconnect specification. However, the Bus Monitor test system uses the IEEE-696 bus. An interface card has been developed that will translate the two sets of signals, interchangeably, to allow each system to transfer data to or from the other in the same manner that a DATAC terminal will communicate with a conventional bus subsystem.

B. Software Communications Driver for Z8000 to VT-180:

A program will be written in Z8000 assembly language to detect new data placed in the Bus Monitor memory by the DATAC terminal, reduce it to a format which allows for some kind of error detection (for example, hex-to-ASCII conversion with parity) and transmit it via an RS-232C line to the Digital Equipment Corporation VT-180 microcomputer serving as the BusMon system console.

C. DATAC Bus Data Analysis and Display:

The actual reduction of data from the DATAC bus initially will be performed by a VT-180 using the "C" language. There are several reasons for
this approach: the Z8000 microcomputer will eventually perform most of the operations planned for the Bus Monitor Unit. At this time, however, few development tools can be used on this configuration of a Z8000 system. Further, the Bus Monitor processor system does not currently support any type of mass storage devices, printers, or other peripherals useful for prototype work. The intent is, then, to develop the Bus Monitor software using the standard "C" language under the CP/M operating system for which there are a variety of software development tools. Once the software is completed and tested, any functions that can be supported by the Z8000 system hardware can be transported to that system via a "C" cross-assembler targeted for that system. Also, the ease of writing involved algorithms in a higher-level language facilitates the programming of data analysis tasks, ordinarily cumbersome to write in assembly language. Additionally, a higher level language is portable; the Bus Monitor functions can be transported to a system based on another processor with little additional programmer time should Boeing and NASA decide that a different hardware configuration is needed. Thus, the investment in software is guarded against obsolescence in the event of a system hardware change or eventual upgrade.

D. DATAC Terminal Simulation:

A DATAC terminal simulator is needed because, at present, the only operational model of the DATAC subsystem interface exists on an active DATAC system. A programmable terminal simulator offers the flexibility needed to test and properly debug the Bus Monitor Unit hardware and software under a variety of conditions, something not easily done using an actual DATAC system.

CURRENTLY ENVISIONED BUS MONITOR FUNCTIONS

In order for the BusMon unit to provide a useable picture of the operation of the DATAC bus, it must be able to evaluate the data which appears on the bus and present it in an operator recognizable format. Since bus protocol violations and communications errors (e.g., parity errors, missing words, etc.) are detected by the DATAC terminal hardware and are never seen by any bus subsystem, the Bus Monitor software will address itself to problems determined by the data content of messages from other DATAC terminals:

1. Echo check of Terminal/System Interfaces: BusMon will verify proper operation of other subsystem hardware by means of a transmit block after receive block directive from the BusMon to the system in question. This assures integrity of RAM buffers and other terminal interface devices.

2. Major/Minor Frame Usage: The BusMon will be equipped with the bus system transmission schedule. If a system fails to transmit in its allotted time frame(s), the monitor can signal a device failure.

3. Data Stream Interpretation: The format of the data stream allows the BusMon to identify the source and information content
of the serial message. A useful feature of this function is to provide a check on the reasonability of data being placed on the bus by the various other bus subsystems. Reconstitution of bus data by an independent system provides another check on the accuracy of other devices operating on the bus. For example, if a display unit shows unreasonable data items, the problem could lie in the data source or in the display unit. By monitoring the data intended for the display, BusMon can help locate potential hardware problems.

4. Bus Status: The most passive mode of operation simply displays hexadecimal data extracted from the bus, formats and displays it on the BusMon console to provide transmitter ID, word count, and other raw data items.
WHAT IS DATAC?

- Serial data bus for use aboard aircraft

- Various A/C systems communicate via a single twisted-pair, coaxial, or fiber optic line

- No central controller (As used by MIL STD 1553B, system relatively free from central failures)

- Uses Ethernet-like CSMA/CD access protocol

- Line topology of bus allows for easy addition/deletion of systems for maintenance or expansion

SUMMARY OF DATAC ADVANTAGES

*WEIGHT REDUCTION- a single bus could serve a minimum-configuration system

*REDUCED INSTALLATION COSTS AND COMPLEXITY- 1-5 bus lines as opposed to one bus line per system inter-connect

*IMPROVED DIAGNOSTICS AND MAINTENANCE- subsystems could be programmed to provide status information to the bus monitor

*SIMPLE TO MODIFY- line topology of bus allows for easy addition and removal of subsystems

*POWERFUL LOCAL PROCESSING CAPABILITY- a subsystem could employ a specialized processor especially suited for a particular task

*MINIMUM REPLACEMENT TIMES- line bus configuration means plug-in replacement possible

*IMPROVED FLEXIBILITY- system configuration is independent of the bus itself

*NO BUS INTERFACE SOFTWARE- handled by bus terminal hardware
BASIC LAYOUT OF DATAC DATA BUS

Bus Monitor Unit

- VT-180 CP/M Computer
- Ithaca Intersystems Z8000 Processor
- Z-Bus 7/EIEE696 Interface

BusMon Console Device

BusMon Processor System: Z 8002, 64K, 6 Serial Ports

BusMon-to-DATAC Terminal Interface

Standard Bus Interface Device

Proposed Bus Monitor Test Configuration
Bus Monitor Software Development System
CONSOLE PORT
MODEM PORT
PRINTER PORT
M68ECB 68000 BASED SINGLE BOARD COMPUTER
ZBUS INTERFACE
Z8000 SYSTEM
DATAC TERMINAL EMULATOR
CONSOLE SYSTEM
BUS MONITOR UNIT

BUS MONITOR SOFTWARE TEST SYSTEM
Fiber optics (the transmission of light through glass fibers along the inside of plastic tubes) is a relatively new technology in avionics engineering. Present day avionics require sending ever increasing amounts of data through smaller spaces with less interference. Fiber optics appears to be the answer to this demand due to numerous advantages over traditional copper wire communications.

One of the most important advantages of fiber optics to the avionics industry is the weight savings when compared to copper cable. Glass weighs much less than copper, and considering the small size of individual fibers, the weight savings can be significant.

Optical fibers have lower attenuation (the loss of signal intensity as it travels the length of the fiber) than do coaxial cables. The fibers also do not emit energy, thus eliminating the need for shielding.

Greater bandwidth is an important advantage of fiber optic communications. The transmission frequencies of light are several orders of magnitude above those in common cables, allowing greater transfer of information.

With the above facts in mind, the Ohio University Avionics Engineering Center is currently developing a fiber optic data bus transmission and reception system that could eventually replace copper cable connections in airplanes. The original form of the system will transmit information from an encoder to a transponder via a fiber optic cable. An altimeter and an altitude display are connected to a fiber optic transmitter by copper cable (bottom figure). The transmitter converts the altimetry data from nine bit parallel to serial form and sends these data through a fiber optic cable to a receiver. The receiver converts the data using a cable similar to that used between the altimeter and display. The transmitting and receiving ends also include a display readout. After completion and ground testing of the data bus, the system will be tested in an airborne environment.

Thus far, the project has encountered no major problems. The fiber optic link was built from a kit produced by Advanced Fiber Optic Corporation. It has been tested and performed well. Three connectors have been chosen to be used in the circuit. The data modem is near completion and will be tested with the fiber optic link, connectors, and individual components when complete.
Past Joint University Program efforts have resulted in Loran-C receiver design and testing, followed by enhancements for full RNAV operation with CRT pilot display. The use of Loran-C in the airspace and the number of commercial receivers available have exploded in recent years; pressure is mounting for approval of Loran-C use during approaches to landing.

The Avionics Engineering Center, in work for the State of Ohio Department of Development, has evaluated such approach applications at Galion, Ohio Municipal Airport and at Mansfield, Ohio Lahm airport. Loran-C data were referenced to ground-tracker data to determine that the Loran-C approach path was straight, flyable, and parallel to the runway centerline. The 0.2 nm offset, while within FAA requirements for RNAV approaches without correction, can easily be corrected to an on-centerline approach path.

There is little argument that Loran-C signals offer the basic accuracy required to design approaches in the coverage regions of existing chains. Seasonal variation in the Loran-C grid and the permanent grid warp caused by overland propagation can be corrected by publication of offsets, either in Loran-C or geodetic terms, to the WGS-72 coordinates defining FAF and MAP. The remaining concerns, then, center on system integrity. One important integrity factor is the airborne receiver and the means for determining its correct operation prior to initiating an approach.

It is possible that a Loran-C receiver may track a zero-crossing of the 100 kHz signal other than the desired third cycle. The result of a single cycle slip can be a position error of one nautical mile or more, depending upon the receiver's position relative to transmitter pair baselines. A cycle slip on the station being used as the master might cause considerable resultant error, by introducing TD errors in both hyperbolic LOPs being used. Such errors might be tolerable in enroute flight, but are certainly unacceptable in terminal-area operations.

The pilot needs a reliable method for receiver checking before an approach, while still in the enroute phase of flight, when primary (VOR) navigation aids are available. Overflight of a VOR station, comparison of Loran-C with a VOR/DME or VOR-RNAV fix, or perhaps a simple angular comparison when within some established distance from a VOR, all offer candidate cross-checking methods.

Effects of various combinations of cycle slip events upon the approach path flown need to be demonstrated and understood, also.

These Loran-C operational issues will be investigated using the following program steps:

1. A Loran-C approach will be designed for the Ohio University Airport:
a. WGS-72 coordinates will be determined for runway thresholds, using Navy TRANSIT data, ground surveys and compass readings of actual runway bearing.

b. Final approach fixes will be computed, at five miles from each threshold, for straight-in approach paths.

c. TERPS criteria will be consulted to determine compliance with approach obstruction limits.

d. Each approach will be test-flown, using theodolite/ranger ground-truth systems to provide referenced data giving uncorrected Loran-C path position and structure.

e. TD and geodetic correction values will be developed, and the approaches will be re-flown with corrections applied. Video-tape data will be collected to verify the quality of the resulting approaches.

f. The approaches will be re-flown with various combinations of cycle slip, and the resulting path location documented.

g. The offsets observed in item f., above, will give guidance as to available options for receiver integrity checks prior to approach.

Specific test plans are in preparation for each step, and results will be reported as data become available.
SATELLITE SURVEYING
FOR A LORAN-C
NONPRECISION APPROACH

Daryl L. McCall
Ohio University
Athens, Ohio

PRECEDING PAGE BLANK NOT FILMED
Work is continuing to site-in a Loran-C nonprecision approach at the Ohio University Gordon K. Bush Airport located in Albany, Ohio. A point has been surveyed just west of the Avionics Engineering Center's airport laboratory, in WGS-72 coordinates to within approximately 2 meters of latitude and longitude and 1.5 meters in altitude.

This survey was performed using the Motorola Mini-Ranger Satellite Surveying System, which uses the Navy's TRANSIT satellites. This position was obtained using the point-position method only, that is the position was calculated from sequential Doppler measurements as the TRANSIT satellite's passed within view of the receiver's antenna. Another method, called translocation, can use differential techniques which provide better results. However, this method requires two receiver's and a precisely known location. This method was not used earlier for lack of the second receiver.

The accuracies obtained, using the single receiver, are sufficient to site-in a Loran-C nonprecision approach. The end-points of the pavement of runways seven and two-five will be surveyed by extrapolating from the known point adjacent to the airport laboratory. Using a calibrated compass, theodolite, and laser ranger, the necessary angles and ranges were obtained to be used in the calculation of the location of the endpoints and final approach fix points of both runways.

Ohio University may be using the Motorola system again, using two receivers to use the translocation method to site-in the localizer antenna array at the airport to meet FAA certification requirements. This may result in a new benchmark located on the field at the airport, known to sub-meter accuracy. In this case, this accuracy will be transferred to the runway endpoints and the Loran-C FAF.
OHIO UNIVERSITY AIRPORT SURVEY POINTS
FOR LORAN-C NONPRECISION APPROACHES
Robert W. Lilley
Ohio University
Athens, Ohio
The emphasis in initial work will be on application of image processing techniques in the site monitoring of an ILS glide slope transmitting array. The glide slope requires a clear area in front of the antennas for use as a reflecting zone. Reflecting or scattering obstacles in this area can cause incorrect path formation and system outages.

Direct video monitoring is impractical due to the narrow-band nature of communications links from the ILS site to the regional maintenance location. Slow-scan video is possible, but even this technique may require transmission of large amounts of data, tying up remote monitoring lines.

Minimization of data transmitted will be investigated in several steps:

1. Video change detection: subtraction, pixel by pixel, of the current image from a baseline image and transmission of only the changed portion.

2. Edge detection of the changed image, providing a high-contrast outline drawing of the changed portion.

3. Calibration of obstruction shape and size, using known camera tilt angle and ground-plane location; investigation of the need for targets or gnomons in specific locations for image calibration.

4. Investigation of remote or pre-programmed pan/tilt/zoom operations for greater areal coverage or improved characterization of the obstacle.

5. 'Treat analysis' of obstruction versus monitor output versus effect on the glide slope signal in space.

6. Recommendations for interoperation with existing monitor/restart procedures and practices, or for changes in monitor philosophy.

Steps 1 and 2 will be investigated using existing COMTAL Vision One video digitizing and display facilities, with the addition of a data link to the IBM 4341 and 4381 computers, also available. Video digitization at 256x256x8 is implemented; color photographs of actual ILS ground planes, with and without obstructions, will be digitized initially, and used to drive change detection and edge detection algorithms. Required data volume will be measured.

Step 3 will be studied mathematically, using known camera optics and measured ground plane locations.

A prototype installation will be fabricated and tested, providing a test bed for remaining studies. Specific hardware will be selected after the computer studies demonstrate feasibility and necessary sensitivity to potential threats.
ADVANCED MONITORING CONCEPTS

IMAGE PROCESSING

- VIDEO CHANGE DETECTION

- EDGE DETECTION

- OBSTRUCTION CHARACTERIZATION
  CALIBRATION Gnomons?

- REMOTE PAN/TLT/ZOOM

- THREAT ANALYSIS -- VIDEO VS. REAL WORLD

- RECOMMENDATIONS
Application of the current GPS NAVSTAR system to civilian service requires that a right-hand, circularly polarized, -160 dBW spread spectrum signal be received from an orbiting satellite, where the antenna environment is also moving. This presents a design challenge when inexpensive antennas are desired. The current trend of the industry is a quadrifilar helix design first perfected by C.C. Kilgus [1], which has become commercially available in a variant form. This is not the only antenna which is usable, however.

A hybrid Archimedian spiral antenna by R. Milne [2] provides signal pattern advantages over the quadrifilar helix, but sacrifices small size and ease of construction. Other types of antennas, such as conical spirals or turnstiles, have not been considered as of yet. The intent of this survey is to provide information on the antennas mentioned above and to construct and test prototypes to determine whether the choice made by industry (of the quadrifilar helix) is the best.

The quadrifilar helix antenna is currently the low cost standard for GPS. It provides cardioid or omnidirectional coverage and right-hand circular polarization. The small size of the antenna at 1575 MHz minimizes wind loading on aircraft. With care in construction, a 20 MHz bandwidth is also achievable at this frequency. A modified version of this antenna provides ruggedness and simplicity.

Prototype versions, based on information from an article by C.C. Kilgus, were constructed using 12-gauge wire and subminiature coaxial hardline. The antenna can be matched using two 90 degree sections of different impedances. The half-turn half-wave helix provides an impedance of 10 to 100 ohms. SMA connectors were used and the copper wire was silverplated after assembly. The equations used to design the antenna and the final working dimensions are shown in figure 1.

The 90 degree balun can be eliminated by increasing one set of element lengths by 45 degrees and decreasing the other set by 45 degrees. This allows feeding the antenna from the top of the helix by using one of the elements as the feeder. Since this is a half-wave configuration, all elements are grounded at the base of the antenna. The helix must have a chosen radius, which determines the pattern of the antenna. The active area of the antenna is limited to the vertical elements. The radial parts of the elements have no radiation. Radial choice provides the cardioid pattern, front-to-back ratio and 3 dB beamwidth. Kilgus provides several graphs and experimental data in his article, and one only has to choose what parameters need to be satisfied for desired GPS antenna characteristics.

The constructed antennas were tested using a signal generator and a reference turnstile. A spectrum analyzer was used to measure the level of the received signal. Measurements were conducted in a large hall at Clippinger labs. The reflections in the hall were found to interfere, so future tests will take place in an open field. The 1575 MHz
signal is of relatively short wavelength, so reflection occurrence was more prominent than expected. Testing was done, as shown in figure 2.

The modified Archimedian spiral was developed by R. Milne. The antenna is basically an Archimedian spiral reflector elements and ground plane radials. The spiral is excited by a cavity which can be mounted inside the aircraft skin. The pattern is a basic cardioid with lobes towards the horizon. The lobes on the horizon are due to the external reflector elements, which provide gain in the direction where it is most needed. The reflector elements are mounted in the radome that covers the spiral. The bandwidth of this antenna is broad enough to cover both the 1575 and the 1227 MHz GPS signal format. Complexity of the antenna makes construction difficult.

Turnstiles and conical spirals offer some usefulness, but lack good cardioid patterns. Using reflectors and directors can provide useful antenna patterns, but sacrifice size.

Industry's choice in using the quadrifilar helix is a valid one when it comes to low cost and ease of construction. The narrow bandwidth limits use of this antenna for C/A GPS work only. The helical spiral offers a better approach when P code reception is desired.

References


HALF-TURN HALF-WAVE QUADRIFILAR HELIX

Direction of propagation

Where:

- 0.5A inches
- 0.27A inches
- 0.18A inches

Simplified match

Matching detail

Turn elements one-half turns

Figure 1.

GPS ANTENNA COMPARISON

Figure 2.
PROCESSOR-CONTROLLED TIMING MODULE FOR LORAN-C RECEIVER*

N87-22616

Robert W. Lilley
Avionics Engineering Center
Department of Electrical and Computer Engineering
Ohio University
Athens, Ohio

*Published as Ohio University Technical Memorandum 90, February 1984.
Hardware documentation is provided for the modified Loran-C timing module which uses direct software control in determining loop sample times. Computer loading is reduced by eliminating polled operation of the timing loop.

I. SUMMARY

The original design for the Ohio University Loran-C receiver featured a software/hardware locked-loop signal processor, based upon the Mostek 50395 timing integrated circuit (IC) [1,2]. This IC provides a six-digit binary-coded decimal (BCD) counter and a BCD register compared to generate an output pulse when the register and counter contents are equal. Operated at 1 MHz, this timing circuit permits microcomputer-selected sample times to be set precisely within a one-second counter interval, with a resolution of one microsecond.

In order to accomplish the data load for the IC register, the computer must detect that the IC digit scan oscillator has selected the appropriate 4-bit BCD digit, and then strobe the new digit data into the register. The IC design requires a scan oscillator frequency of no higher than 20 kHz, which places a lower limit on the time required to load all six digits. In practice, the complete register load required 500-800 microseconds (μs).

The original receiver design makes all six IC digit strobe signals available to the microcomputer, which then strobes the new digit data into the register after detecting the presence of the appropriate strobe signal. This operation must take place in order to preset the next loop sample time, triggered by the EQUAL pulse from the Mostek IC. Therefore, the register load must be performed between each Loran-C pulse. The technique of polling the digit strobe lines to detect the next digit to be loaded requires full attention from the microcomputer, delaying background processing.

The modified circuit described in this technical memorandum ignores digit strobes, by bringing the SCAN input under control of the processor. Register loading becomes a synchronous process, with no polling loop. When a digit datum is ready for loading, the scan counter is updated immediately by the processor, with no lost time. See figure 1 for a block diagram.

This modification is required to permit the single processor (a MOS Technology 6502) to perform all required computations for the entire Loran-C process. Expansion of the receiver from a three-station tracker to full five-station operation would cause over 20 percent of processor power to be lost to the strobe polling operation, causing a reduction in navigation data output rates, and a reduction in the number of pilot-oriented features which could be added using the single processor. This modified circuit saves some 350 to 650 μs per Loran pulse.

Full hardware documentation is provided in this paper for the circuit card implementing the Loran-C timing loop, and the receiver event-mark and re-track functions. This documentation is to be combined with overall receiver drawings to form the as-built record for this device. Computer software to support this module is integrated with the remainder of the receiver software, in the LORPROM program.
II. CIRCUIT DESCRIPTION

Figure 2 shows the complete logic diagram for the Loran-C timing module. To the far left are signal descriptors for the system computer, an MAI SuperJolt based upon the MOS Technology 6502 with 6520 peripheral interface adapter (PIA). All connections, except for CLOCK and IRQ, are made through the 6520 PIA. Figure 3 gives a summary of PIA pin assignments, useful in software design and coding. The Mostek 50395 chip description is given as figure 4, and pinouts are shown in figure 5.

Referring to figure 2, note that seven lines provide data and control signals to the Mostek IC (U3). All these signals are output by the computer as TTL-compatible signals, and must be changed to the 12-volt MOS specification required by U3. This conversion is performed in open-collector drivers U1 and U2, pulled up to 12v through 1.1 K-ohm resistors. These lines carry the four data bits for register digits Ra, Rb, P, the LR (Load Register) strobe and SET (Set digit counter to most-significant digit) signals.

The timing chip U3 is wired for free-running counter, counting up driven by the CLOCK, which is a buffered version of the main microcomputer clock (a temperature-compensated crystal oscillator). The SCAN input which increments the digit counter to indicate the load window for each register digit is driven by the computer via the SCAN line.

The computer is programmed to select (SET) the high order digit first, and to set LR to enable the register load digit at a time. Each digit is then placed on RA through RD, and loaded by toggling SCAN.

Once the register is fully loaded, the U5 Loran-C interrupt and data latches are enabled by bringing CLRPF high. When the free running counter in U3 reaches the register value just loaded, U3 issues an EQUAL pulse for one clock period (one μs) which clocks the U5 latches. The IRQ low since the Q output of the U5 interrupt latch always goes low upon clocking. An interrupt is signaled at IRQ to the computer. LDAT, latched by the U5 Loran-C latch, assumes the instantaneous value of the Loran-C digital waveform received at LRIN from the receiver front-end module. Note that LRIN is processed by U4 to set a pulse width of approximately 70 μs before it is sampled. This pulse width is necessary to provide a guard time after the leading pulse edge to permit successful pulse tracking, and to minimize initial search time. Since the various front-end processors designed to date have presented various pulse widths, this U4 mono-stable multivibrator has been provided to equalize the waveform before sampling.

The remainder of the circuit, U12, deals with receiver features included for evaluation. The event latch is driven by a front-panel pushbutton to place on the receiver output tape a unique mark so output data may be correlated with flight events. The retrack latch, also operated by pushbutton, signals the computer that the operator wishes to restart the Loran-C search process. To minimize contact bounce, these latches are configured to operate on the pushbutton release cycle.
Once the computer program has serviced the Loran-C sample interrupt thus generated, the U5 latches are disabled by a low at CLRP. Another register load sequence begins.

In this manner, successive samples may be taken of the Loran-C input waveform at times which are precisely controlled by the microcomputer. The programmer may now select algorithms for detecting received Loran-C chains and stations by varying the sample time and observing the result at LDAT.

The module pictorial appears in figure 6, giving placement of ICs and other major components.

III. REFERENCES


Figure 1. Block diagram, Loran-C timing module.
Figure 2. Revised timing board schematic.
Figure 3. Super Jolt to timing board summary of PIA assignments.
Figure 4. MOSTEK 50395 integrated circuit.
Figure 5. MOSTEK 50395 pinouts.

Figure 6. Pictorial and signal glossary.
A MICRO-COMPUTER-BASED SYSTEM
TO COMPUTE MAGNETIC VARIATION*

Rajan Kaul
Avionics Engineering Center
Department of Electrical and Computer Engineering
Ohio University
Athens, Ohio

*Published as Ohio University Technical Memorandum 91, March 1984.
A micro-computer-based implementation of a magnetic variation model for the continental United States is presented. The implementation computes magnetic variation as a function of latitude and longitude for general aviation receivers such as Loran-C.

I. INTRODUCTION

A mathematical model of magnetic variation in the continental United States (COT48) has been implemented in the Ohio University Loran-C receiver. The model is based on a least squares fit of a polynomial function. The implementation on the micro-processor based Loran-C receiver is possible with the help of an Am 9511 math chip, manufactured by Advanced Micro Devices, which performs 32 bit floating point mathematical operations. A Peripheral Interface Adapter (M6520) is used to communicate between the 6502 based micro-computer and the 9511 math chip. The implementation provides magnetic variation data to the pilot as a function of latitude and longitude. This report briefly describes the model and the real time implementation in the receiver.

II. THE MATHEMATICAL MODEL

The model was developed at the United States Geological Survey (USGS) by Fabiano et al. [1], by performing least squares analysis on more than 34,000 data measurements taken between 1900 and 1974. The analysis provides an analytical model of the magnetic field which is used to compute the magnetic variation.

For the actual magnetic variation calculation the COT48 region is partitioned into five, 12 degree longitudinal bands from 66 degrees West to 126 degrees West. A set of coefficients for each of the five bands is determined by the analytical model. A seventh order polynomial function of the analytical model is applied to compute the magnetic variation. The secular change is calculated in a similar way, the only difference being that a sixth order polynomial function is used. Also, the secular change case is not partitioned into bands and therefore the same set of coefficients is used for the entire COT48 region.

The polynomial function adapted for the procedure is

\[ \sum_{i=0}^{n} \sum_{j=0}^{i} a_{ij} (\theta_c)^{i-j} (\lambda_c)^j \]

- \( a_{ij} \) - co-efficients
- \( \theta_c \) - normalized latitude = \( \theta - 52 \)
- \( \lambda_c \) - normalized longitude
- \( \gamma \) - Table 1 - east longitude normalizing factor
- \( \Theta \) - co-latitude = 90°-latitude
- \( \lambda \) - east longitude = 360°-longitude

The limits on each band and other constants are specified in Table 1 [2].
For the magnetic variation calculation n=7, and thus 36 coefficients are specified for each band in the COT48 region, while in the secular change calculation n=6, therefore only 28 coefficients are required for the whole COT48 region. All the coefficients are given in Appendix A.

The model was simulated in FORTRAN on an IBM 370 computer at Ohio University and a contour plot was made of the magnetic variation in the COT48 region (figure 1). The FORTRAN program listing is included in Appendix B. A copy of an actual magnetic variation chart published by the Defense Mapping Agency (DMA) is shown in figure 2. Comparisons between actual published values of the magnetic variation and values calculated by the model were made and are described later in this report.

III. MICRO-COMPUTER IMPLEMENTATION

The magnetic variation model was implemented on a 6502 based Super-Jolt micro-computer. The 6502 microprocessor has only an 8-bit data bus, so the processor needs a large amount of memory and rapid access. The calculations in the implementation of the model require complex floating point operations of exponents. It is therefore desirable to use an external hardware device to support the microprocessor in these calculations.

The Am9511 was chosen to be implemented with the Super-Jolt system. It is a peripheral math processor which performs the necessary floating point mathematical operations. The Am9511 is designed to be used in conjunction with microprocessor systems that have an 8-bit data bus. The stack oriented processor can handle 16 and 32-bit floating point operands and performs arithmetic and trigonometric functions. An instruction set of the Am9511 is included in Appendix C [3].

Additional hardware is necessary to allow the microprocessor to communicate with the math processor. An M6520 peripheral Interface Adapter (PIA) is used for handshaking with the microcomputer. The PIA consists of two 8-bit ports and several control registers for interface with external support devices. The overall design of the microcomputer, which is a part of the Ohio University Loran-C project is shown in figure 3.

IV. INTERFACING SOFTWARE

Special software is needed to allow the hardware components to interact with one another. The four subroutines 'PINT', 'PUSH', 'POP' and 'CMND' were written by Fischer [4] with this particular goal in mind. 'PINT' initializes the Am9511 and the PIA and, also, the scratchpad RAM locations. 'PUSH' is used to copy a four byte number from RAM to the stack of the Am9511. 'POP' does exactly the opposite by copying a four byte floating point number from the stack of the Am9511 to scratchpad RAM. 'CMND' sends an instruction byte to the Am9511 to perform the desired operation. It also checks the status register of the math processor to determine the outcome of the operation. Flow charts of the above subroutines are given in figure 4.

The actual magnetic variation program 'MAGVAR', occupies about 800 bytes of memory including scratchpad RAM locations. The coefficients 36
for each of the five bands, occupy 900 bytes of memory. Each of the coefficients is converted into a 32-bit floating point format compatible with the Am9511 representing four bytes. The secular variation calculation is not included in the real time implementation for reasons which shall be addressed later in this report. The complete program listing is given in Appendix D at the end of this report. The 'MAGVAR' program takes about 1.5 seconds in execution time. However, since the magnetic variation does not change rapidly in a small geographic region, it does not need to be computed every time navigation position information is updated, when included in the actual navigation receiver such as the Ohio University Loran-C. For example a small change in the software can allow computation of the magnetic variation every 30 miles, or a one degree change in geographic position or any other interval desired.

V. RESULTS AND CONCLUSIONS

Initially, the values for the magnetic variation were computed by the FORTRAN simulation and compared to values published by National Geophysical Data Center.* The results obtained were accurate to a large degree. Table 2 summarizes two points in each band, of which comparisons were made in the COT48 region. The reason for the discrepancy in the values could arise from the differences between the data and the model. Fabiano and others [1] evaluated the model and compared it to surveyed data for 1,450 points. From these measurements an overall root mean squared deviation of 0.5 degrees was found in the magnetic variation in the COT48 region. Also, a probable cause for the larger discrepancy in the region of bands 2 and 3 could indicate magnetic variation anomalies in the Great Lakes region.

In general, the results were found to be satisfactory and the decision was made to implement the model on the Ohio University Loran-C receiver. The results computed by the microcomputer were within 0.1 degrees of the values computed by the FORTRAN simulation. As indicated earlier, the secular change was not implemented on the receiver. The magnetic variation in the COT48 region changes less than 11 minutes of arc annually at its worst case. This translates to a change of less than one degree over a period of five years at its worst case. Since the Ohio University Loran-C receiver is a research tool, not implementing the secular change function would not have a crucial impact on the outcome of future research. The coefficients for the model are derived every five years by the USGS, and can be updated very easily to keep the model current.

The overall performance of the implementation proves to be satisfactory. The major advantages are that the magnetic variation is available all the time to the pilot to allow accurate determination of the compass heading. It is computed automatically and is one less adjustment or source of error during a flight, thus also reducing the chances of pilot error.

VI. ACKNOWLEDGMENTS

The work presented in this technical memorandum has been supported by the National Aeronautics and Space Administration at Langley Research Center under grant number NGR 36-009-017. It was performed at Ohio University's Avionics Engineering Center.

* National Geophysical Data Center, Boulder, Colorado.
The author would like to acknowledge the help of Dr. Robert Lilley, Associate Director and Mr. James Nickum, project engineer at Avionics Engineering center, whose suggestions proved invaluable in all stages of this research. Special thanks is also due to Dr. Hugh Bloemer, Department of Geography, whose advice was indispensable.

VII. REFERENCES


[2] Ibid., Fabiano, Jones and Peddle.


VIII. APPENDICES

A. Co-efficients for the 5-band and secular change in COT48.

B. FORTRAN Program Listing of "MAGVAR".

C. Instruction Set for the Am9511.

D. 6502 Assembly Language Program Listing of the Magnetic Variation Implementation on the Ohio University Loran-C Receiver.
Table 1. Limits on Five Bands and the East Longitude Normalizing Factor.

<table>
<thead>
<tr>
<th>Band</th>
<th>Partition °W longitude</th>
<th>λ(degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66-77</td>
<td>289</td>
</tr>
<tr>
<td>2</td>
<td>78-89</td>
<td>277</td>
</tr>
<tr>
<td>3</td>
<td>90-101</td>
<td>265</td>
</tr>
<tr>
<td>4</td>
<td>102-113</td>
<td>253</td>
</tr>
<tr>
<td>5</td>
<td>114-125</td>
<td>241</td>
</tr>
</tbody>
</table>
Table 2. Comparisons Between Actual and Computed Values of Magnetic Variation.

<table>
<thead>
<tr>
<th>Band</th>
<th>Latitude Deg. N</th>
<th>Longitude Deg. W</th>
<th>Magnetic Variation Actual</th>
<th>Magnetic Variation Computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36</td>
<td>77</td>
<td>7.72°W</td>
<td>7.40°W</td>
</tr>
<tr>
<td>66°W - 78°W</td>
<td>40</td>
<td>73</td>
<td>13.05°W</td>
<td>12.71°W</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>81</td>
<td>3.98°W</td>
<td>2.78°W</td>
</tr>
<tr>
<td>78°W - 90°W</td>
<td>40</td>
<td>87</td>
<td>1.12°W</td>
<td>0.51°W</td>
</tr>
<tr>
<td>3</td>
<td>34</td>
<td>93</td>
<td>4.37°E</td>
<td>5.90°E</td>
</tr>
<tr>
<td>90°W - 102°W</td>
<td>38</td>
<td>101</td>
<td>9.05°E</td>
<td>10.84°E</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>103</td>
<td>9.93°E</td>
<td>10.91°E</td>
</tr>
<tr>
<td>102°W - 114°W</td>
<td>40</td>
<td>107</td>
<td>12.62°E</td>
<td>13.54°E</td>
</tr>
<tr>
<td>5</td>
<td>38</td>
<td>123</td>
<td>16.18°E</td>
<td>16.59°E</td>
</tr>
<tr>
<td>114°W - 126°W</td>
<td>32</td>
<td>113</td>
<td>12.58°E</td>
<td>13.25°E</td>
</tr>
</tbody>
</table>
Figure 1. Continental U.S. magnetic variation.
Figure 2. Magnetic variation in the United States from world declination chart (source Defense Mapping Agency).
Figure 3. Block diagram for the microcomputer system.
Figure 4. Logic flow diagrams illustrating steps control program executes to communicate with 9511.
Figure 4. Concluded.
Appendix A. Co-efficients For the 5-band and Secular Change in COT48.

The coefficients \((a_{ij})\) for the magnetic variation in the conterminous United States (5 bands).

<table>
<thead>
<tr>
<th>Band 1</th>
<th>Band 2</th>
<th>Band 3</th>
<th>Band 4</th>
<th>Band 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_{00})</td>
<td>(-0.12544E\ 02)</td>
<td>(-0.26754E\ 01)</td>
<td>(0.66671E\ 01)</td>
<td>(0.13031E\ 02)</td>
</tr>
<tr>
<td>(a_{10})</td>
<td>(0.47404E\ 00)</td>
<td>(0.30498E\ 00)</td>
<td>(0.73577E\ 01)</td>
<td>(-0.24981E\ 00)</td>
</tr>
<tr>
<td>(a_{20})</td>
<td>(-0.79262E\ 01)</td>
<td>(-0.85329E\ 01)</td>
<td>(-0.62856E\ 00)</td>
<td>(-0.36471E\ 00)</td>
</tr>
<tr>
<td>(a_{30})</td>
<td>(-0.14935E\ 01)</td>
<td>(-0.11933E\ 01)</td>
<td>(-0.75537E\ 02)</td>
<td>(-0.24981E\ 00)</td>
</tr>
<tr>
<td>(a_{40})</td>
<td>(-0.25381E\ 04)</td>
<td>(0.31737E\ 04)</td>
<td>(0.54348E\ 04)</td>
<td>(0.13665E\ 04)</td>
</tr>
<tr>
<td>(a_{50})</td>
<td>(-0.18112E\ 04)</td>
<td>(-0.51048E\ 04)</td>
<td>(0.77275E\ 04)</td>
<td>(0.10816E\ 03)</td>
</tr>
<tr>
<td>(a_{60})</td>
<td>(0.55077E\ 04)</td>
<td>(0.72203E\ 04)</td>
<td>(0.14306E\ 03)</td>
<td>(-0.50373E\ 04)</td>
</tr>
<tr>
<td>(a_{70})</td>
<td>(0.62795E\ 04)</td>
<td>(0.60910E\ 03)</td>
<td>(0.55063E\ 04)</td>
<td>(0.31817E\ 04)</td>
</tr>
<tr>
<td>(a_{80})</td>
<td>(-0.15257E\ 05)</td>
<td>(0.19467E\ 05)</td>
<td>(0.40952E\ 05)</td>
<td>(-0.82114E\ 06)</td>
</tr>
<tr>
<td>(a_{90})</td>
<td>(-0.14304E\ 05)</td>
<td>(0.59918E\ 05)</td>
<td>(0.88037E\ 05)</td>
<td>(-0.89843E\ 05)</td>
</tr>
<tr>
<td>(a_{100})</td>
<td>(-0.53360E\ 05)</td>
<td>(0.38753E\ 04)</td>
<td>(-0.14001E\ 04)</td>
<td>(0.58428E\ 05)</td>
</tr>
<tr>
<td>(a_{110})</td>
<td>(-0.76365E\ 05)</td>
<td>(-0.51048E\ 04)</td>
<td>(0.77275E\ 04)</td>
<td>(0.10816E\ 03)</td>
</tr>
<tr>
<td>(a_{120})</td>
<td>(-0.62672E\ 07)</td>
<td>(-0.76459E\ 07)</td>
<td>(0.12533E\ 06)</td>
<td>(-0.20612E\ 07)</td>
</tr>
<tr>
<td>(a_{130})</td>
<td>(0.79694E\ 07)</td>
<td>(-0.62486E\ 07)</td>
<td>(0.17499E\ 07)</td>
<td>(-0.41744E\ 09)</td>
</tr>
<tr>
<td>(a_{140})</td>
<td>(0.15540E\ 06)</td>
<td>(0.17782E\ 06)</td>
<td>(0.39251E\ 06)</td>
<td>(-0.10090E\ 06)</td>
</tr>
<tr>
<td>(a_{150})</td>
<td>(0.78517E\ 07)</td>
<td>(-0.30012E\ 07)</td>
<td>(-0.27612E\ 07)</td>
<td>(0.57044E\ 07)</td>
</tr>
<tr>
<td>(a_{160})</td>
<td>(0.11556E\ 06)</td>
<td>(0.24909E\ 06)</td>
<td>(-0.15699E\ 06)</td>
<td>(-0.84960E\ 06)</td>
</tr>
<tr>
<td>(a_{170})</td>
<td>(-0.76126E\ 06)</td>
<td>(0.74355E\ 06)</td>
<td>(-0.18538E\ 06)</td>
<td>(0.62532E\ 06)</td>
</tr>
<tr>
<td>(a_{180})</td>
<td>(0.23144E\ 07)</td>
<td>(0.48778E\ 05)</td>
<td>(-0.38992E\ 06)</td>
<td>(-0.45290E\ 07)</td>
</tr>
<tr>
<td>(a_{190})</td>
<td>(-0.16860E\ 09)</td>
<td>(0.24723E\ 08)</td>
<td>(0.84712E\ 08)</td>
<td>(0.10184E\ 08)</td>
</tr>
<tr>
<td>(a_{200})</td>
<td>(0.43030E\ 09)</td>
<td>(0.36370E\ 09)</td>
<td>(0.44108E\ 08)</td>
<td>(-0.11497E\ 07)</td>
</tr>
<tr>
<td>(a_{210})</td>
<td>(-0.69117E\ 08)</td>
<td>(0.19785E\ 07)</td>
<td>(0.77909E\ 08)</td>
<td>(0.16786E\ 07)</td>
</tr>
<tr>
<td>(a_{220})</td>
<td>(0.33027E\ 08)</td>
<td>(-0.76731E\ 08)</td>
<td>(-0.37480E\ 09)</td>
<td>(0.64112E\ 09)</td>
</tr>
<tr>
<td>(a_{230})</td>
<td>(0.93928E\ 08)</td>
<td>(-0.40026E\ 07)</td>
<td>(0.88535E\ 07)</td>
<td>(0.27692E\ 07)</td>
</tr>
<tr>
<td>(a_{240})</td>
<td>(0.91628E\ 08)</td>
<td>(0.52668E\ 07)</td>
<td>(0.12561E\ 06)</td>
<td>(-0.93332E\ 07)</td>
</tr>
<tr>
<td>(a_{250})</td>
<td>(0.22806E\ 07)</td>
<td>(-0.21183E\ 06)</td>
<td>(0.14212E\ 06)</td>
<td>(0.52297E\ 08)</td>
</tr>
<tr>
<td>(a_{260})</td>
<td>(0.56746E\ 08)</td>
<td>(-0.29999E\ 06)</td>
<td>(-0.15083E\ 07)</td>
<td>(-0.54997E\ 07)</td>
</tr>
</tbody>
</table>
Appendix A Concluded.

The coefficients \((a_{ij})\) for the secular change in the cor48 region.

\[
\begin{align*}
a_{00} & = -0.95533E\ 01 \\
a_{10} & = 0.11582E\ 00 \\
a_{11} & = -0.93474E\ -01 \\
a_{20} & = 0.13750E\ -01 \\
a_{21} & = -0.22416E\ -01 \\
a_{22} & = 0.12437E\ -01 \\
a_{30} & = 0.27558E\ -05 \\
a_{31} & = 0.53560E\ -03 \\
a_{32} & = -0.70816E\ -03 \\
a_{33} & = 0.32333E\ -03 \\
a_{40} & = -0.49965E\ -04 \\
a_{41} & = 0.18579E\ -04 \\
a_{42} & = 0.38544E\ -05 \\
a_{43} & = 0.11451E\ -04 \\
a_{44} & = -0.63005E\ -05 \\
a_{50} & = -0.23992E\ -06 \\
a_{51} & = 0.35195E\ -06 \\
a_{52} & = -0.26724E\ -06 \\
a_{53} & = -0.53815E\ -06 \\
a_{54} & = 0.44979E\ -06 \\
a_{55} & = -0.12145E\ -06 \\
a_{60} & = 0.85465E\ -07 \\
a_{61} & = -0.70500E\ -07 \\
a_{62} & = 0.26013E\ -07 \\
a_{63} & = 0.30705E\ -09 \\
a_{64} & = -0.12121E\ -07 \\
a_{65} & = -0.15529E\ -08 \\
a_{66} & = 0.23462E\ -08
\end{align*}
\]
Appendix B. FORTRAN Program Listing of "MAGVAR".

THIS PROGRAM COMPUTES THE MAGNETIC VARIATION AS A FUNCTION OF LATITUDE AND LONGITUDE. THE SECULAR CHANGE IS ALSO CALCULATED. IT IS BASED ON THE USD 80 POLYNOMIAL MODEL DEVELOPED BY FABIANO AND OTHERS AT THE UNITED STATES GEOLOGICAL SURVEY IN DENVER CO. PLEASE CONSULT USGS CIRCULAR NO. 810 FOR DETAILS.

RAJAN KAUL - 3/84

THE INPUT VARIABLES ARE ALAT, ALON AND YEAR REPRESENTING LATITUDE, LONGITUDE AND YEAR. VARIABLES A AND A1 ARE THE COEFFICIENTS TO BE READ

DIMENSION A(8,8), A1(8,8)
DATA EAST/'EAST'/*WEST'/*WEST'/

READ LATITUDE AND LONGITUDE
WRITE(6,9)
FORMAT(IX,F6.2,1X,F6.2)' TYPE LAT. AND LONG. AS NNN.NN NNN.NN (F6.2,1X,F6.2')
READ(7,8) ALAT, ALON
FORMAT(F6.2,1X,F6.2)

DETERMINE WHICH BAND THE POINT IS IN TO LOAD CORRECT SET OF COEFFICIENTS CORRESPONDING TO PARTICULAR BAND.

IF(ALON.GE.66.0.AND.ALON.LT.78.0) K=11
IF(ALON.GE.78.0.AND.ALON.LT.90.0) K=12
IF(ALON.GE.90.0.AND.ALON.LT.102.0) K=13
IF(ALON.GE.102.0.AND.ALON.LT.114.0) K=14
IF(ALON.GE.114.0.AND.ALON.LT.126.0) K=15

READ NORMALIZED LONGITUDE FOR PARTICULAR BAND AND THE COEFFICIENTS.
READ(K,7) DLON
FORMAT(F6.2)
DO 5 NN=1,8
DO 6 II=1,NN
READ(K,3,END=13) A(NN,II)
FORMAT(E12.5)
CONTINUE
CONTINUE

DEFINE COLATITUDE AND NORMALIZED EAST LONGITUDE
DLA=90.0-ALAT
DLO=360.0-ALON

INITIALIZE MAGNETIC VARIATION AND PERFORM CALCULATION
AK=0.0
DO 1 N=1,8
DO 2 I=1,N
KK=IABS(N-I)
JJ=IABS(I-I)
DL=DLO-DLON
IF(DL.EQ.0.0) DL=360.0
AK=AK+(A(N,II)*((DLA-52.01)**KK)*((DL)**JJ))
CONTINUE
CONTINUE

READ COEFFICIENTS FOR SECULAR CHANGE CALCULATION
DO 15 NN=1,7
DO 16 I=1,NN
READ(16,23,END=14) A1(NN,II)
FORMAT(E12.5)
CONTINUE
CONTINUE
DEFINE COLATITUDE AND NORMALIZED EAST LONGITUDE

DLA=90.0-ALAT
DLO=360.0-ALON

INITIALIZE SECULAR CHANGE AND PERFORM CALCULATION

SV=0.0
DO 11 N=1,7
DO 12 I=1,N
KK=IABS(N-I)
JJ=IABS(I-1)
DL=DLO-DLON
IF(DL.EQ.0.0) DL=360.0
SV=SV+(A(N,I)*((DLA-52.01)*_KK)*((DL)**JJ))
11 CONTINUE
12 CONTINUE

READ YEAR

WRITE(6,17)
17 FORMAT(5X,'TYPE YEAR AS NN.N (F4.1) E.G. JUN 84 = 84.5')
READ(7,18) YEAR
18 FORMAT(F4.1)

COMPUTE SECULAR VARIATION ANNUAL AND TO PRESENT DATE.
ALSO COMPUTE MAGNETIC VARIATION

SECVAR=SV*(YEAR-85.0)/60.0
SS=SV/60.0
VAR=AK+SECVAR
IF(VAR.LT.0.0) DIR=WEST
IF(VAR.GT.0.0) DIR=EAST
V=ABS(VAR)
WRITE(6,4) ALAT,ALON,V,DIR,SS
4 FORMAT(5X,'LATITUDE = ',F6.2/,5X,'LONGITUDE = ',F6.2,5X,'MAGNETIC
 & VARIATION = ',F6.2,5X,'SECULAR CHANGE (ANNUAL) = ',F6.2)
STOP
END
Appendix C. Instruction Set for the AM9511.

<table>
<thead>
<tr>
<th>Command Code</th>
<th>Command Mnemonic</th>
<th>Command Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 1 1 0 0</td>
<td>SADD</td>
<td>Add TOS to NOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 1 0 1 1 0 1</td>
<td>SSDU</td>
<td>Subtract TOS from NOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 1 0 1 1 1 0</td>
<td>SMULT</td>
<td>Multiply NOS by TOS Lower half of result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 1 0 1 1 1 1</td>
<td>SGIV</td>
<td>Divide NOS by TOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 0 1 1 1 0 0</td>
<td>DADD</td>
<td>Add TOS to NOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 0 1 1 1 0 1</td>
<td>DSUB</td>
<td>Subtract TOS from NOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 0 1 1 1 1 0</td>
<td>DMUL</td>
<td>Multiply NOS by TOS Upper half of result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 1 0 1 1 1 1 1</td>
<td>DQIV</td>
<td>Divide NOS by TOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>FADD</td>
<td>Add TOS to NOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 0 0 1 0 0 0 0</td>
<td>FSUB</td>
<td>Subtract TOS from NOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 0 0 1 0 0 0 1</td>
<td>FMUL</td>
<td>Multiply NOS by TOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 0 0 1 0 0 1 1</td>
<td>FDIV</td>
<td>Divide NOS by TOS Result to NOS Pop Stack</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>SQRT</td>
<td>Square root of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>SIN</td>
<td>Sine of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>COS</td>
<td>Cosine of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>ATAN</td>
<td>Inverse tangent of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>ASIN</td>
<td>Inverse sine of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>ACOS</td>
<td>Inverse cosine of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>LOG</td>
<td>Common logarithm (base 10) of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>LN</td>
<td>Natural logarithm (base e) of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>EXP</td>
<td>Exponential (e^x) of TOS Result in TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>PWR</td>
<td>NOS raised to the power in TOS Result in NOS Pop Stack</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>NOP</td>
<td>No Operation</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>FIXS</td>
<td>Convert TOS from 16-bit fixed point to 32-bit fixed point</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>FIXD</td>
<td>Convert TOS from 32-bit fixed point to 16-bit fixed point</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>FLS</td>
<td>Convert TOS from 16-bit fixed point to floating point</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>FLD</td>
<td>Convert TOS from 32-bit fixed point to floating point</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>CGHS</td>
<td>Change sign of 16-bit fixed point operand on TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>CGSU</td>
<td>Change sign of 32-bit fixed point operand on TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>CHSF</td>
<td>Change sign of floating point operand on TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>CHSD</td>
<td>Change sign of floating point operand on TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>PTOS</td>
<td>Push 16-bit fixed point operand on TOs to NOS (Copy)</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>PTOU</td>
<td>Push 16-bit fixed point operand on TOs to NOS (Copy)</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>POPM</td>
<td>Pop 16-bit fixed point operand from TOs NOS becomes TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>POPD</td>
<td>Pop 32-bit fixed point operand from TOs NOS becomes TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>POPL</td>
<td>Pop floating point operand from TOs NOS becomes TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>POPL</td>
<td>Pop floating point operand from TOs NOS becomes TOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>XCHB</td>
<td>Exchange 16-bit fixed point operands TOS and NOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>XCHD</td>
<td>Exchange 32-bit fixed point operands TOS and NOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>XCHF</td>
<td>Exchange floating point operands TOS and NOS</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>PUP</td>
<td>Push floating point constant in same TOs Previous TOs becomes NOS</td>
</tr>
</tbody>
</table>

**NOTES**

1. TOS means Top of Stack NOS means Next on Stack.
2. AMD Application Brief: Algorithm Details for the AM9511A.
3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived function may destroy several stack locations. See Application Brief for details.
4. The incommensurate functions handle angles in radians, not degrees.
5. No remainder is available for the floating point divide functions.
6. Results will be undefined for any combination of commutative coding bits not specified in this table.

* Original not available at time of publication.
Appendix D. 6502 Assembly Language Program Listing of the Magnetic Variation Implementation on the Ohio University Loran-C Receiver.

ORG $A8
BASE BSS 2
ORG $55
CTRN BSS 1
CTRI BSS 1
COFCTR BSS I
* USED IN LEAST SQUARES ALGORITHM
COFTA BSS 2
ADDRESS OF COEFFICIENT TABLE
MTEMPI BSS 1
TEMPORARY LOCATION USED BY MAGVAR CALCULATION.
MTEMPI BSS 1
USED BY MAGVAR

* EQUATES TO SUBROUTINE CALLS

PUSH EQU $28AC SUBROUTINE TO PUSH NUMBER ON TO 9511 STACK
POP EQU $28DC SUBROUTINE TO POP NUMBER FROM 9511 STACK
CMND EQU $290C SUBROUTINE TO ISSUE COMMAND TO 9511 TO PERFORM
* OPERATION

* EQUATES TO VARIABLE ADDRESSES USED IN RNAV

PHGS EQU $ED0 LATITUDE OF RECEIVER
TGS EQU $DC0 LONGITUDE OF RECEIVER
PIE EQU $2C0 180.0/PI
PA12 EQU $300 2*PI
F90 EQU $240 PI/2
* AM9511A COMMANDS.

PWR EQU SOB
FADD EQU $10
FSUB EQU $11
FMUL EQU $12
FDIV EQU $13
SQRT EQU $14
CHSF EQU $15

* CONSTANTS AND VARIABLES
BAND EQU $0 ADDR FOR NORMALIZED LONGITUDE FOR PARTICULAR BAND
* FOLLOWED BY 36 COEFFICIENTS FOR EACH BAND AT LOCATION
* $C800 TO $C3FF - ONE PAGE FOR EACH OF 5 BANDS.
* CONSTANTS FOR DIVISION IN LEAST SQUARES ALGORITHM FOR MAGVAR.
MZERO EQU $0 -0.0
MONE EQU MZERO+4 - 1.0
MTWO EQU MONE+4 -2.0
MTHREE EQU MTWO+4 -3.0
MFOUR EQU MTHREE+4 -4.0
MFIVE EQU MFOUR+4 -5.0
MSIX EQU MFIVE+4 -6.0
MSEVEN EQU MSIX+4 -7.0

* CONSTANTS THAT DEFINE LIMITS IN BANDS OF COT48 TO DETERMINE
WHICH SET OF COEFFICIENTS NEED TO BE USED IN THE ALGORITHM
TO DETERMINE MAGNETIC VARIATION.
A78 EQU MSEVEN+4
A90 EQU A78+4
A102 EQU A90+4
A114 EQU A102+4
A5201 EQU A114+4
F180 EQU A5201+4
P1/180
ATEMP EQU F180+4 TEMPORARY LOCATION USED WHILE DETERMINING THE PA
NDL EQU ATEMP+4 NORMALIZED LATITUDE = 52.01*PI/180 RADIANS
MDLA EQU NDL+4 MDL0-NORMALIZED LONGITUDE FOR PARTICULAR BAND.
MDLA52 EQU MDLA+4 MDLA-AS201
MAGVAR EQU MDLA52+4 CUMULATIVE MAGNETIC VARIATION
RLTEMP EQU MAGVAR+4 TEMPORARY LOCATION
MAGVD EQU RLTEMP+4 MAGNETIC VARIATION

ORG $C000

104
MOVE CONSTANT NUMBER TABLE IN SCRATCH SPACE

LDA #1
STA BASE+1   BASE = $0100
LDA =0
LDY =0
MO
LDA TABLE1+,Y
STA (BASE)+,Y
INY
CPY #56
BNE MO

MAGNETIC VARIATION CALCULATION

LDA =0
STA CTRN
STA CTRI
INITIALIZE COUNTERS

CALCULATE MDLA

INC BASE+1
INC BASE+1   BASE = $0300
LDY =F90
JSR PUSH
DEC BASE+1
LDY =PHGS
JSR PUSH
LDA =FSUB
JSR CMDN
INC BASE+1
LDY =P18
JSR PUSH
LDA =FMUL
JSR CMDN
DEC BASE+1
DEC BASE+1   BASE = $0100
LDY =MDLA
MDLA = 90-PHGS
JSR POP

CALCULATE MDLO

INC BASE+1
INC BASE+1   BASE = $0300
LDY =A12
JSR PUSH
DEC BASE+1
LDY =THGS
JSR PUSH
LDA =FSUB
JSR CMDN
INC BASE+1
LDY =P18
JSR PUSH
LDA =FMUL
JSR CMDN
DEC BASE+1
DEC BASE+1   BASE = $0100
LDY =MDLO
MDLO = 360-THGS
JSR POP

CALCULATE MDLA52

LDY =MDLA
JSR PUSH
LDY =A5201
JSR PUSH
LDA =FSUB
JSR CMDN
JSR CMDN
LDY =MDLA52
MDLA52 = MDLA-52
JSR POP
** INC BASE+1   BASE = $0200
LDY =THGS
JSR PUSH
DEC BASE+1   BASE = $0100
LDY =RLTEMP
JSR POP   PUT THGS IN RLTEMP FOR COMPARISON PURPOSES IN THE NEXT SEGMENT TO DETERMINE WHICH BAND TO USE
**
** DETERMINE WHICH BAND IT IS TO CALCULATE MAGVAR.
**
LDY =A78
JSR PUSH
LDY =RLTEMP
JSR PUSH
LDA =FSUB
JSR CMND
LDY =ATEMP
JSR POP
ATEMP = 78*PI/180 - THGS
LDY =ATEMP
LDA (BASE),Y
BPL M1   IF ATEMP IS +VE -- BAND 1, IF NOT TRY FOR BAND 2
LDY =A90
JSR PUSH
LDY =RLTEMP
JSR PUSH
LDA =FSUB
JSR CMND
LDY =ATEMP
JSR POP
ATEMP = 90*PI/180 - THGS
LDY =ATEMP
LDA (BASE),Y
BPL M2   IF ATEMP IS +VE -- BAND 2, IF NOT TRY FOR BAND 3
LDY =A102
JSR PUSH
LDY =RLTEMP
JSR PUSH
LDA =FSUB
JSR CMND
LDY =ATEMP
JSR POP
ATEMP = 102*PI/180 - THGS
LDY =ATEMP
LDA (BASE),Y
BPL M3   IF ATEMP IS +VE -- BAND 3, IF NOT TRY BAND 4
LDY =A114
JSR PUSH
LDY =RLTEMP
JSR PUSH
LDA =FSUB
JSR CMND
LDY =ATEMP
JSR POP
ATEMP = 114*PI/180 - THGS
LDY =ATEMP
LDA (BASE),Y
BPL M4   IF ATEMP IS +VE -- BAND 4
JMP M5   MUST BE BAND 5
**
** SET COEFFICIENT TABLE ADDRESS TO CORRESPOND WITH PARTICULAR BAND
**
M1   LDY =MDLO
JSR PUSH
LDA =$C8
STA BASE+1
STA COFTAB+1   BAND 1
JMP M6
M2   LDY =MDLO
JSR PUSH
LDA =$C9
STA BASE+1
STA COFTAB+1
JMP M6

M3
LDY #MDLO
JSR PUSH
LDA #$CA
STA BASE+1
STA COFTAB+1
JMP M6

M4
LDY #MDLO
JSR PUSH
LDA #$CB
STA BASE+1
STA COFTAB+1
JMP M6

M5
LDY #MDLO
JSR PUSH
LDA #$CC
STA BASE+1
STA COFTAB+1
JMP M6

M6
LDA =0
STA COFTAB
LDY =BAND
JSR PUSH
LDA =FSUB
JSR CMND
LDA =1
STA BASE+1
LDY =NDL
JSR POP
STA COFCTR
CLC
LDA CNTR
LOAD OUTER LOOP COUNTER
ROL A
ROL A
POINTER TO LOCATION FOR EXPONENTS FOR LEAST SQUARE

C2
LDA=MTEMP
LDY=MTEMP
JSR PUSH
LDA =FSUB
JSR CMND
N-1
LDY =RLTEMP
JSR POP
LDY =MDLA52
JSR PUSH
LDY =MDLA52
LDA (BASE),Y
BPL C6
LDA =1
STA MTEMP1
LDA =CHSF
JSR CMND
JMP C9

C6
LDA =0
STA MTEMP1
LDY =RLTEMP
JSR PUSH
LDA =PWR
JSR CMND
MDLA52**(N-1)
CLC
LDA CNTR
SBC CNTR
AND =1
EXPONENT EVEN ?
BNE C4 YES, LOOP OUT
LDA MTEMP1 NO, IS NEGATIVE FLAG SET?
BEQ C4 NO, FLAG NOT SET --- LOOP OUT
LDA =CHSF EXPONENT WAS ODD AND NEGATIVE FLAG
JSR CMND WAS SET --- THEREFORE CHANGE SIGN AGAIN

C4

LDY =ATEMP
JSR POP
CLC
LDA CTRI LOAD INNER LOOP COUNTER FOR LEAST SQUARES PROCED
ROL A
ROL A POINT TO LOCATION FOR EXPONENTS

LDA MTEMP
LDY MTEMP
JSR PUSH
LOY =RLTEMP
JSR POP
LDY =NDL
JSR PUSH
LDY =NDL
LDA (BASE), Y
BPL C7 LOAO INNER LOOP COUNTER FOR LEAST SQUARES PROCED

BNE C5

LDA =0
STA MTEMP1
LDA =CHSF
JSR CMND
JMP C8

C5

LDY =ATEMP
JSR CMND
LDA =FMUL
JSR CMND

MDLA52**(N-I)
LDA COFTAB+1 PUT CO-EFFICIENT TABLE ADDRESS IN BASE
STA BASE+1 POINT TO CO-EFFICIENT COUNTER
JSR PUSH
LDA =FMUL
JSR CMND A(N1)*NDL**(1)*MDLA52**(N-I)
LDA =1
STA BASE+1 BASE = $0100
LDY =MAGVAR
JSR PUSH
LDA =FADD
JSR CMND ADD TO ACCUMULATE THE VALUE OF MAGVAR
LDA =FMUL MAGVAR=MAGVAR+A(N1)*NDL**(1)*MDLA52**(N-I)
JSR POP
INC CTRI
INC CTRI
INC COFCTR
INC COFCTR
INC COFCTR
INC COFCTR POINT TO NEXT SET OF CO-EFFICIENTS
LDA CTRN
CMP CTRI IF THEY ARE EQUAL INNER LOOP DONE, CHECK IF OUTE
BEQ C1
INC CTRI
JMP C2 IF NOT, GO BACK AND COMPLETE OUTER LOOP

C1

LDA CTRN CHECK TO SEE IF OUTER LOOP COMPLETE
CMP =7
BEQ C3 OUTER LOOP ALSO DONE, BRANCH OUT
TABLE 1

<table>
<thead>
<tr>
<th>BAND1</th>
<th>BAND2</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC CTRN</td>
<td>INC CTRN</td>
</tr>
<tr>
<td>LDA :0</td>
<td>LDA :0</td>
</tr>
<tr>
<td>STA CTRI</td>
<td>STA CTRI</td>
</tr>
<tr>
<td>JMP C2</td>
<td>JMP C2</td>
</tr>
<tr>
<td>RTS</td>
<td>RTS</td>
</tr>
</tbody>
</table>

* TABLE OF CONSTANTS USED BY MAGNETIC VARIATION STORED
* STARTING AT $0100 IN SCRATCHPAD RAM LOCATION.

TABLE 1

<table>
<thead>
<tr>
<th>hexadecimal</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX 00,00,00,00</td>
<td>0.0 (MZERO)</td>
</tr>
<tr>
<td>HEX 01,80,00,00</td>
<td>1.0 (MONE)</td>
</tr>
<tr>
<td>HEX 02,80,00,00</td>
<td>2.0 (MTWO)</td>
</tr>
<tr>
<td>HEX 03,80,00,00</td>
<td>4.0 (MFOUR)</td>
</tr>
<tr>
<td>HEX 03,00,00,00</td>
<td>5.0 (MFIVE)</td>
</tr>
<tr>
<td>HEX 03,00,00,00</td>
<td>6.0 (MSIX)</td>
</tr>
<tr>
<td>HEX 03,00,00,00</td>
<td>7.0 (MSEVEN)</td>
</tr>
<tr>
<td>HEX 01,AE,40,F1</td>
<td>78*PI/180 (A78)</td>
</tr>
<tr>
<td>HEX 01,09,06,DA</td>
<td>90*PI/180 (A90)</td>
</tr>
<tr>
<td>HEX 01,0E,DE,C4</td>
<td>102*PI/180 (A102)</td>
</tr>
<tr>
<td>HEX 01,FE,AD,AE</td>
<td>114*PI/180 (A114)</td>
</tr>
<tr>
<td>HEX 06,0D,0A,30</td>
<td>52.01*PI/180 (A5201)</td>
</tr>
<tr>
<td>HEX 78,8E,FA,35</td>
<td>P1/180.0 (F180)</td>
</tr>
</tbody>
</table>

* CO-EFFICIENTS FOR THE FIVE BANDS OF THE COT48
* THE DATA LABELED BAND(N) IS THE NORMALIZED
* LONGITUDE FOR EACH BAND. THE CO-EFFICIENTS
* ARE STORED IN ONE PAGE CHUNKS STARTING AT
* $3800 TO $3CFF.

ORG $C800

<table>
<thead>
<tr>
<th>hexadecimal</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX 09,90,81,48</td>
<td>289.01</td>
</tr>
<tr>
<td>HEX 84,08,02,96</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 7F,F2,96,07</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 80,CA,E9,68</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX FA,F4,B3,9C</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 79,9B,AC,04</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 79,EF,5E,07</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 75,97,EB,10</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 75,FI,C0,9A</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX FF,C5,A5,66</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 77,A2,06,A6</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 71,DA,E9,7F</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX FF,F8,7E,18</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX FF,97,FO,15</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 72,EF,02,28</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 73,B3,00,EB</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 6A,F8,66,41</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX ED,C0,F2,4A</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 6D,BF,FC,83</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX EF,B2,50,90</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 6F,B9,C1,F7</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX F0,80,1E,7A</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 69,86,96,3E</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 69,AB,24,13</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 6A,AB,DO,20</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 69,02,22,6E</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 69,F8,29,08</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX EC,C0,59,10</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 67,C6,0D,89</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX E0,B9,62,17</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 61,EC,8F,0F</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX E3,ED,7C,38</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 64,E2,F5,2F</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 66,AC,A5,31</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 66,9D,6A,60</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 67,C3,E7,CE</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 55,C2,FA,4F</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

ORG $C900

<table>
<thead>
<tr>
<th>hexadecimal</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX 09,98,81,48</td>
<td>277.01</td>
</tr>
<tr>
<td>HEX 82,AB,39,97</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 7F,9C,26,00</td>
<td>0.0000</td>
</tr>
<tr>
<td>HEX 80,DA,4A,06</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

109
HEX FA,C3,82,A1
HEX 7B,96,1F,46
HEX 7B,A0,F5,0D
HEX 7A,A0,0D,F8
HEX F6,C0,C6,EB
HEX F7,05,7C,74
HEX F7,8F,48,07
HEX 72,85,1D,21
HEX 72,53,BC,09
HEX 72,06,1C,5F
HEX 70,97,6B,8F
HEX F5,0D,AC,25
HEX EE,82,A3,89
HEX 6E,82,1F,27
HEX EF,00,00,58
HEX EF,C3,7E,79
HEX 72,82,8A,81
HEX 72,89,90,7C
HEX 84,41,8A
HEX 86,30,51
HEX 6A,BE,EF,E4
HEX 88,09,E6,8B
HEX 6B,83,BA,65
HEX 6C,71,98,81
HEX 6F,83,AC,3A
HEX 64,09,81,7F
HEX 65,83,06,58
HEX 67,9F,41,1D
HEX 86,83,02,0F
HEX 88,88,88,48
HEX 68,82,83,51
HEX 8A,83,72,90
HEX 8B,81,00,7C
ORG $C000

BAND3
HEX 09,B4,81,A8 - 265.01
HEX 03,05,38,CD
HEX 79,F1,19,1D
HEX 80,0D,89,70
HEX 69,F7,84,81
HEX 74,81,81,09
HEX 6A,82,53,EB
HEX 76,97,8B,AF
HEX 65,9A,03,6F
HEX 76,87,82,A7
HEX 72,85,70,94
HEX 72,85,83,4B
HEX 70,80,0F,1E
HEX 73,02,3E,7C
HEX 74,96,03,2F
HEX 72,6E,F3,3C
HEX EF,69,69,28
HEX EE,88,01,DA
HEX F0,93,83,7A
HEX F0,89,5E,A9
HEX F0,80,65,68
HEX EF,71,99,6C
HEX EA,83,91,11
HEX 67,96,50,EE
HEX 68,8B,95,5F
HEX E7,ED,2E,80
HEX ED,0D,85,5A
HEX ED,80,00,15
HEX 6B,11,56,79
HEX 66,91,88,02
HEX 85,97,80,FC
HEX 66,85,08,91
HEX E1,CE,0C,45
HEX 69,8E,20,A2
HEX 6A,86,DF,80
HEX 6A,98,99,19
HEX E7,81,8F,63
ORG $C800

110
BAND4
HEX 08,FD,02,8F - 253.01
HEX 04,00,80,90
HEX FE,FF,02,74
HEX FF,8A,80,C8
HEX 76,0C,64,B4
HEX 78,0C,F5,DA
HEX FB,80,07,85
HEX 74,98,19,1A
HEX F7,0C,38,C3
HEX 75,8B,CE,22
HEX F7,03,84,25
HEX 70,E5,42,BA
HEX EF,06,86,8B
HEX 73,02,02,2E
HEX F2,03,47,EB
HEX 72,05,73,91
HEX EC,0C,6C,A8
HEX 6F,09,0D,4E
HEX F0,9E,38,67
HEX 6F,0A,90,74
HEX 6F,04,00,19
HEX 71,A2,4F,72
HEX E7,01,DE,70
HEX E1,05,70,20
HEX E9,0B,AD,43
HEX 6B,05,00,52
HEX EC,04,10,69
HEX 6C,07,0C,02
HEX E8,02,84,06
HEX 63,88,F8,78
HEX E6,05,86,8A
HEX 67,90,31,E7
HEX 62,80,38,29
HEX ED,DE,8E,81
HEX E9,08,6D,A9
HEX 65,83,B1,5E
HEX E8,EC,36,4B
ORG $C000

BAND5
HEX 08,F1,02,8F - 241.01
HEX 04,FF,F4,F1
HEX FF,CE,82,29
HEX FE,00,F4,93
HEX 7A,89,50,CC
HEX 79,98,89,C9
HEX F9,05,2C,6D
HEX 75,8C,F5,25
HEX ED,0E,2A,44
HEX 74,86,5B,73
HEX 78,96,8C,74
HEX F0,F1,E6,21
HEX F2,AA,AC,0F
HEX 73,89,FF,23
HEX 71,9C,AD,69
HEX F2,D9,31,E5
HEX EE,07,CO,F0
HEX 6E,92,95,4D
HEX EA,83,02,45
HEX 0E,91,1C,DA
HEX 71,8A,7C,2A
HEX F2,9E,31,E4
HEX ED,02,85,5A
HEX ED,F8,00,15
HEX E8,01,56,79
HEX 66,91,88,02
HEX 65,97,8D,FC
HEX 66,85,0B,91
HEX E1,CE,0C,45
HEX 69,8E,20,A2
HEX 6A,86,DF,BO
HEX 6A,98,99,19
HEX E7,81,8F,35
ORG $C800
BAND4
HEX 08,FD,02,8F - 253.01
HEX 04,DD,80,90
HEX FF,FE,CE,74
HEX 76,DC,64,84
HEX 78,9C,35,0A
HEX FB,80,07,85
HEX 74,98,19,1A
HEX F7,BC,5B,03
HEX 75,BC,CE,22
HEX F7,A3,34,25
HEX 70,EC,42,8A
HEX EF,DD,86,BB
HEX 73,EC,02,22
HEX F2,03,47,EB
HEX 72,85,73,91
HEX EC,DC,5C,68
HEX 6F,0A,0D,4E
HEX F0,96,8B,67
HEX 6F,DA,9C,74
HEX 6F,C4,00,19
HEX 71,A2,4F,72
HEX E7,B1,0E,70
HEX E1,E5,7D,20
HEX E9,08,AD,43
HEX 68,F5,00,52
HEX EC,E4,10,59
HEX 6C,A7,DC,02
HEX E8,C2,84,06
HEX 63,8B,F8,7B
HEX 65,CA,86,8E
HEX 67,90,31,E7
HEX 62,B0,3B,29
HEX 67,ED,0E,31
HEX 69,C8,6D,A5
HEX 63,B3,B1,5E
HEX 88,EC,36,48
ORG $C000

BAND5
HEX 08,F1,02,8F - 241.01
HEX 04,FF,84,F1
HEX FF,CE,B2,29
HEX FE,DD,F4,95
HEX 7A,89,90,CC
HEX 79,98,99,C9
HEX F9,D5,2C,6D
HEX 75,BC,F5,25
HEX ED,00,20,AA
HEX 74,96,8B,73
HEX 78,96,8C,74
HEX F0,F1,66,21
HEX F2,AA,AC,0F
HEX 73,89,FF,23
HEX 71,9C,AD,69
HEX F2,99,31,43
HEX EE,D7,CO,F0
HEX 6E,92,95,AD
HEX EA,03,02,45
HEX F0,E9,1C,DA
HEX 71,8A,7C,2A
HEX F2,9E,31,E4
HEX 67,DA,5D,00
HEX 6A,8F,5A,F3
HEX EB,ED,06,25
HEX 6D,98,6D,AB
HEX ED,00,43,5B
HEX EC,E1,2A,FE
HEX 6D,92,4C,1D
HEX 65,DF,30,34
HEX 6E,82,9F,74
HEX 6E,7F,39,BF
HEX 69,B5,F0,07
HEX 69,95,10,C7
HEX 67,FE,63,22
HEX E9,88,09,80
HEX 6A,CF,F8,04
END
A PROGRAM DOWNLOADER AND OTHER UTILITY SOFTWARE
FOR THE DATAC BUS MONITOR UNIT*

Stanley M. Novacki, III
Avionics Engineering Center
Department of Electrical and Computer Engineering
Ohio University
Athens, Ohio

*Published as Ohio University Technical Memorandum 92, July 1984.
A set of programs designed to facilitate software testing on the DATAC Bus Monitor is described.

I. INTRODUCTION

The DATAC Bus Monitor Unit (BusMon) is a Z8000-based microcomputer system designed to receive, interpret, and display selected data items appearing on a DATAC Digital Data Bus. Software for the Bus Monitor Unit is developed on a Tektronix 8550 Microprocessor Development System (MDS). Once a program is written and compiled to object code, it may be tested using the in-circuit emulation and memory-partitioning capabilities of the 8550. The in-circuit emulator allows the MDS to imitate the Z8000 processor, giving the operator extensive control of the test system, while memory partitioning allows the prototype system to utilize memory in the 8550 as though it were part of the target system's memory. This is a great help in lab-testing of the prototype system because of the simplicity of loading and running the test software.

Because of the size of the Tektronix hardware, it is somewhat cumbersome to transport the entire MDS and the prototype system to a field installation simply to test programs in situ. To make on-site testing easier, a series of programs was developed to allow the Z8000 system, running in a standalone mode, to receive program code via its RS232C ports and ports on the host system, which stores the test program in a disk file. Once the program design is finalized, another utility program allows the Z8000 system to send the test software in ASCII form to a ProLog PROM programmer, eliminating the need for an integral PROM programmer on the MDS. These software tools are intended to simplify the development and testing of the data acquisition, reduction, and display routines planned for the DATAC Bus Monitor Unit.

II. IMPLEMENTATION

On the Tektronix 8550 MDS:

Once a program for the Z8000 system has been written and reduced to machine code, it can be transferred to a DOS/50 disk file. DOS/50 is the operating system currently in use on the MDS. The file format consists of lines of ASCII characters in a format called Standard TEKHEX (figs. 1, 2). There are two types of records in a TEKHEX file: data records and the "null" or terminator record. The format for a data record begins with the slash character "/" which denotes the start of a valid record. The slash is followed by 4 hex digits which specify the absolute loading address for the data contained in this record. Next are two hex digits which specify the number of bytes of data contained in the record. The following two digits form a nybble checksum of the load address and the datum count; that is, each digit of the load address and byte count are added together. This number, modulo 256, provides the first checksum. Following the checksum comes the data bytes representing the actual machine code of the program. After the data is the data nybble checksum. As with the first checksum, this is the sum of the individual hex digits of the data, modulo 256. Each record is terminated by an ASCII CR (0D hex). The last record in a TEKHEX file is the "null" record, that is, one with a datum count of zero. An address/byte-count checksum is still generated, usually with a zero value.
A file in this format can be sent to a slave system via RS232C communications ports on the slave and the MDS. The host system will read a record from the TEKHEX file, send it to the prototype system, and wait for a single ASCII token to indicate a good (ACK) or bad (NAK) reception. The 8550 uses the digits "0" as the ACK token and "7" as the NAK symbol. If the prototype system replies with an ACK, the MDS will send the next record, wait for the prompt for that record and so on until the entire file is sent. If the prototype system fails to reconstitute the same checksums sent in the TEKHEX record, it will reply with the NAK token. The 8550 will recognize this as a failed transmission and re-send the same record. The 8550 will continue to send the flagged record until the slave system elects to abort the load operation with an abort message, which appears on the 8550 console and halts the load operation, or the number of retries exceeds a limit set by the host system operator. After all data records are sent, followed by the null record, the 8550 exits from the load routines and resumes terminal emulation. From this point, the MDS may simply be used as a console device to the prototype and the program is run on the prototype.

On The Bus Monitor Unit:

The loader program for the Z8000-based system (fig. 3; listing 1) is designed to accept serial ASCII data TEKHEX format, convert it to machine code, and store it in the prototype system memory. The processor monitor software for the Bus Monitor Unit provides serial I/O routines which allow it to transmit and receive blocks of ASCII data via serial port A, the default console port, by using the Z8000 System Call instruction, SC #0. The Z8000 loader program begins by sending the ACK token to the host system to indicate that it is ready to receive characters. The input operation of SC #0 returns a string in memory terminated by a carriage return. Once a string has been read, the loader routine scans the input buffer to find the "/" character to define the beginning of the record. If the slash does not occur in the first 80 bytes, it is assumed that part of the record was lost; TEKHEX records do not usually exceed 73 characters including the terminating carriage return. The loader routine sends a NAK token to request a re-send and waits for the next transmission.

Once a record has been received and the slash found, the load address and byte count are converted from ASCII representations to their actual hexadecimal values. This is done by shifting the seven-bit-code for the most-significant-digit of a data byte (i.e, a single ASCII character) to the left by 4 bits, producing a datum of the form "x0" from "zx" in hex. The next character ("zy"), the least-significant digit of the datum being reconstituted, is logically ANDed with 0F hex to zero the high order bits, leaving a "0y" pattern in hex. The loader then ORs the two patterns together, giving a byte of the form "xy". If the character being converted is a numeric, the binary-coded decimal (BCD) representation of the number and the least significant nybble match exactly and the conversion process may proceed. If the hex character is an alphabetic, A-F, some adjustment is needed because the 4 low-order bits of the ASCII characters A through F do not correspond to the hexadecimal values A through F (10 to 15 decimal). In fact, the low-order nybble of ASCII characters A-F has the values 1-6
in BCD. Because of the sequential value, we may correct these characters' codes to correspond to their actual value by adding 09 hex to the character code before the masking process. This addition bumps the low-order bits to a pattern corresponding to the binary representation of their namesakes. With this correction, the characters A–F can then be processed like the numerics 0–9. The alphabetic character adjustment is handled by subroutine TSTNUM and the ASCII-to-hexadecimal conversion is performed by ASCHEX.

Once the load address and byte count are reconstituted, the first checksum is generated. If the computed and transmitted checksums do not agree, a NAK token is sent and the Bus Monitor waits for a new transmission. Otherwise, the program reconstitutes the data stream using ASCHEX, stores it using the load address it generated earlier, and maintains a running checksum. After all data have been stored in the prototype's RAM, the data checksum is reconstituted from the string buffer and compared with the calculated value. If a mismatch occurs, a NAK token is sent and the Bus Monitor waits for the the same record to be retransmitted from the host. Otherwise, it issues an ACK, waits for the next record, and continues the load-and-store process until the entire file has been sent. In the event 5 successive checksum errors occur, the Bus Monitor will abort the load operation by sending an "Abort Load" record, whose message is displayed on the system console (line 198 of listing 1). When the null record is received, the Z8000 returns to the resident monitor via SC #3. No integrity check is performed on the checksum, since a transmission error at this point doesn't affect any data that has been stored.

On the CP/M-based Bus Monitor Console System:

In field experiments, a DEC VT-180 will be used as the host for the program down-loading in addition to being a data display/command input device. The file down-loader (listing 2) is written in the "C" language for the CP/M environment by Manx Software Systems. This loader contains two deviations from the 8550 down-load procedure: one is that the VT-180 itself counts errors and exits on 5 successive errors; the other is that on completion of file transmission, the loader is exited and the VT-180 returns to the CP/M command processor rather than to terminal mode.

Prolog PROM Programmer Support:

This utility can be thought of as a complement to the downloader program for the Z8000. The program (listing 3) sends machine code from the Bus Monitor Unit to a Prolog PROM Programmer equipped with an RS-232C serial port. Two factors complicate this seemingly simple task: one is that the serial port drivers for the PROM programmer expect to see only ASCII data. The other is that the memory for a Z8000 system is organized as 16-bit words. As yet, there are no 16-bit-wide memory devices being manufactured. The designers of these microcomputer systems routinely solve the latter problem by using 2 byte-wide RAMs or ROMs in parallel, one device located at an even byte address, the other at the subsequent odd address. The first "trick" is that we must read alternating memory locations (all even or all odd) addresses when sending data to the programmer.
We will solve the former problem by a procedure which complements the ASCHEX subroutine described earlier. The program produces two ASCII characters from one hexadecimal byte by splitting the byte into high and low-order nybbles and then shifting the high order nybble to the right 4 bit places. For example, byte "xy" becomes two bytes "0x" and "0y". For the hexadecimal digits 0-9, we simply add 30 hex to each byte and we have the ASCII character corresponding to the BCD digit. The hex digits A-F again pose another problem: the ASCII collating sequence has specified that the low-order nybbles of the codes for the characters A-F are 1-6 decimal, not A-F hex. Further, the high order nybble of those letter digits is a hex 4, not a 3, as is the case for the numeric characters. To handle this case, the program tests the nybble being converted to see if it lies in the range of A-F. If so, an adjuster of 07 hex is added to the nybble first. This corrects the least significant digit to the proper value and puts a 1 in the most significant digit. For example, to turn 0C hex to 43 hex (the ASCII code for the letter "C") the following happens: add 07 to OC giving 13 hex, then add 30 hex giving 43 hex, giving the desired character code.

The PROLOG utility is usually used with the 8550 running in processor emulation mode in the Bus Monitor system. A data rate of 2400 baud between the test system and the PROM programmer is assumed. The programmer support routine normally resides at address 4000 hex. If this conflicts with the intended load address of the program being sent to the PROM programmer, the support routine can be moved to another memory location. This is possible because the utility program uses only relative addresses, excepting the I/O port addresses which present no relocatability problems. Once the utility program and the application program have been loaded into Bus Monitor memory, the PROM programmer is set to receive the first block (even or odd) of data. Using the 8550 emulator or the Resident Monitor, the following CPU registers are initialized: R10 contains the address of the first byte if the program being sent to the programmer, R11 contains the address of the last byte to be programmed, and R12 contains a 0 if even-numbered bytes are being ROMmed, and a 1 if odd-numbered bytes are being sent to the programmer. Execution begins at the label GO; the "B" serial port on the serial I/O card is used to send data to the PROM programmer, R9 points to the machine code being processed. A pass is complete when R9 is greater than R11, the stop address. For convenience, a breakpoint can be set at GO + 4C hex, so that R12 can be toggled to send the second block of data bytes without having to reset R10 and R11. With R12 readied for the next series of data and the programmer fitted with a new chip, execution may be resumed with a "GO" command, completing the programming process.
III. SUMMARY

The software described in this paper will facilitate the design and testing of software for the DATAC Bus Monitor Unit. By providing a means to simplify program loading, firmware generation, and subsequent testing of programs, we can reduce the overhead involved in software evaluation and use that time more productively in performance, analysis and improvement of current software.

IV. ACKNOWLEDGMENTS

I would like to thank Mr. Kim Constantikes of Carnegie-Mellon University, Mr. John Simmons of Tektronix, Inc., and Mr. Jim Ramsay and Mr. Bill Lynn, both of Kentron International, for their support and patience during the development of these programs.

V. BIBLIOGRAPHY


Data Record
/aaaaabbabcd...dddc  \(\text{<CR>}\)

LOAD ADDRESS  BYTE COUNT CHKSUM  DATA BYTES  2nd CHKSUM  RECORD TERMINATOR

Terminator Record
/xxxxxooac

LOAD ADDRESS  ZERO- CHKSUM

Abort Record
//Abort message text

Figure 1. TEKHEX-format records used by BusMon loader program.
Figure 2. Sample TEKHEX file.

Figure 3. Z8000 loader outline.
APPENDIX A

LISTING 1

ASM Z8001/Z8002
V01.01-01 (8550)

; DATA BUS MONITOR:
; LOADER FOR Z8000 PROCESSOR INTERFACE TO DATAC SYSTEM
; AUTHOR: S.M. NOVACKI 2 SEPT 83
; REV 22 NOV 83: INCLUDES ERROR HANDLER- EXITS TO MONITOR
; MACRO DEFINITIONS HERE:
MACRO NYBSUM

LDB RL2,RH2 ; TRANSPOSE HEX DIGITS
SRAB RL2,#04 ; MAKE H0D THE LOD
AND R2,#0FOFH ; MASK OFF HO BITS
ADDB RH2,RL2 ; ADD NYBBLES W/O CARRY

RH2 HOLDS NYBBLE CHECKSUM, TRANSFERRED TO RH7
ENDM

; THIS MACRO PERFORMS A TEST FOR CHECKSUM ERRORS, IF >5 THE LOAD IS ABORTED
MACRO ERRMSG

INC R13 ; COUNT NEW ERROR OCCURRENCE
CP R13,#5 ; REACH MAX# OF ERRORS?
JR UGT,ABRTLD ; TOO MANY ERRORS- RETURN TO MONITOR
SET R12,#01 ; SET 'OLD STRING, REPEAT' FLAG
LDB ACKBUF,#NAK ; READY BAD TX MSG
JR NEWSTR
ENDM

; I/O STRING BUFFER DEFINITIONS, MUST BE ORG'D IN RAM

ORG OFEOM

0000FE00 4 ACKBUF BLOCK 4 ; THREE BYTE BUFFER TO HANDSHAKE WITH 8550 DURING FILE TX
0000FE04 50 INBUF BLOCK 80 ; 80 BYTE BUFFER FOR RECEIVING TEKHEX FILES
0000FE54 8 TKHIN BLOCK 8 ; I/O STRING
0000FE5C 8 TKINAK BLOCK 8 ; I/O FC BLOK (WORKSPACE)

; BEGINNING OF LOADER ROUTINE;
; CONSULT ZMON.DASSY AND .DLI_FILE TO DETERMINE ACTUAL ADDRESSES
; BEGINNING OF RCOMABLE ROUTINES, ALL JUMPS RELATIVE, ONLY
; RAM REFERENCES ARE ABSOLUTE FOR DURATION OF LOADER OPERATION

FMMSG ASCII 'STD TEKHEX LOADER ' ; NOTE# OF BYTES IN STRING MUST BE EVEN

INCOM CLR R12
CLR AOKBUF ; ONLY 3 OF 4 BYTES USED
ACKBUF+1,#RECEND ; READY STRING FOR RCOMABLE
AOKBUF+2,#PRCI_T ; Tek Handshake
CLR R12
CALR SET IO
OUTMSG
LD RI0,#80
LD R9,#INBUF
ZAPWRD
LD R9(RIO),RL2
DEC RI0
JR NZ,Z#PWRO;
OUTMSG
LD R1,#TKINAK
SC #0
; OUTPUT PROMPT VIA MONITOR ROUTINE

121
; HOPEFULLY WITH A SERIAL LINE DEDICATED TO THE Z8K-TEK INTERFACE
; THERE WON'T BE ANY JUNK BEFORE THE PROMPT AND THE FIRST HEX RECORD.
; UNTIL THAT SERIAL LINE IS ESTABLISHED, WE'LL SHARE THE ONE WITH
; Z8K CONSOLE DEVICE AND PROVIDE FOR GETTING RID OF ANY BAD DATA
; WE MAY HAPPEN TO READ. ONCE A SEPARATE SERIAL LINE IS AVAILABLE, WE CAN
; DISCARD THE 'FIND START-OF-RECORD' ROUTINE

; IDLE 8550 BEGINS TO TX AFTER THE PROMPT SENT BY OUTF MSG

; AT THIS POINT, WE SHOULD HAVE ONE COMPLETE TEKHEX RECORD FOR PROCESSING
; REGISTER ASSIGNMENTS FOR REDUCING THE ASCII STRING
; R1: TRANSIENT AREA FOR CONSOLE I/O
; R2: WORK AREAS FOR CHECKSUM COMPUTATION
; R3,R4: WORK AREAS FOR ASCII HEX CONVERSION
; R5: WORKSPACE FOR FINDING INCOMING ASCII STRING
; R6: CONTAINS THE LOAD ADDRESS OF THE DATA
; R7: CONTAINS THE N YB BLES CHECKSUMS
; R8: POINTER INTO ASCII STRING FOR HEX GENERATION
; R13: CONTAINS CHECKSUM ERROR COUNT
; FIRST WE'LL SCAN FOR JUNK THAT THE Z8K MAY HAVE READ BEFORE
; THE 8550 STARTED TX OF THE HEX FILE; THIS SECTION CAN BE
; DELETED IF WE DEDICATE A SERIAL PORT FOR 8550/ZBK COMMUNICATION
; 'SLASH' CHAR DELIMITS START OF DATA

; AT THE END OF THE INPUT BUFFER?
; IF SO, THE WHOLE RECORD WAS JUNK, GET ANOTHER
; SCAN TO THE NEXT CHAR
; HEADER NOT FOUND, TRY AGAIN
; BAD TX, ASK FOR REPEAT OF STRING

; END OF SOM-SCANNER ROUTINE
; WE'LL ASSUME THAT A VALID RECORD HAS BEEN READ
; WE'LL ASSUME THAT A VALID RECORD HAS BEEN READ

; IS THIS NEW OR OLD DATA?
; DON'T RESET ERROR ACCUM IF THIS IS A REPEAT
; ZERO OUT CKSUM ERROR ACCUMULATOR
; OLDSTR CALL CHKTRM ; SEE IF THE RECORD IS THE ZERO-LENGTH TERMINATOR
; IF TERM RECORD IS FOUND, RETURN TO MONITOR
; MOVE POINT PAST HEADER TO FIRST ASCII CHARACTER
; (R8) ADDRESS OF FIRST CHAR IN STRING
; GET FIRST BYTE OF ADDRESS
; ADD NYBBLES W/O CARRY
; LOAD ADDRESS IS NOW COMPLETE

; TRANSPOSE HEX DIGITS
; MAKE MOD THE LOD
; MASK OFF HO BITS
; ADD NYBBLES W/O CARRY
; TO CHECKSUM ACCUMULATOR
; HOBYTE OF ADDRESS TO R6
; LOAD ADDRESS TO R6; LOAD ADDRESS IS NOW COMPLETE

00000850 2101FE54 R
GETSTR LD R1,#THXIN ; SELECT HEX RECORD READ-MODE
SC #0 ; GET HEX RECORD AND SAVE IT AT INBUF
LDA R8,INBUF ; SET BASE ADDRESS OF HEX STRING
LDA R8,INBUF+80
JR EQ,STREQ
INC R8
JR NE,SEEK
STREQ LD AC,<BUF,#NAK ; AT THE END OF THE INPUT BUFFER?

0000085A 0808FE.54 R
STREQ LD AC,<BUF,#NAK ; AT THE END OF THE INPUT BUFFER?

0000085E 6605
JR EQ,STREQ ; IF SO, THE WHOLE RECORD WAS JUNK, GET ANOTHER

00000860 0C812F2F
CPB @R8,#RECMRK
JR EQ,TSTSTR ; FOUND IT!

00000864 E606
JR EQ,TSTSTR
INC R8
ON TO THE NEXT CHAR

00000866 A980
JR NE,SEEK
HEADER NOT FOUND, TRY AGAIN
BAD TX, ASK FOR REPEAT OF STRING

00000868 EEF8
STREQ LD ADBUF,#NAK ; BAD TX, ASK FOR REPEAT OF STRING

0000086A 4D05EF00 R
STREQ LD ADBUF,#NAK

00000870 EBE3
JR NEWSTR ; DO THE ASKIN'

00000872 8DC4
STSTR TEST R12
IS THIS NEW OR OLD DATA?
DON'T RESET ERROR ACCUM IF THIS IS A REPEAT
ZERO OUT CKSUM ERROR ACCUMULATOR
OLDSTR CALL CHKTRM ; SEE IF THE RECORD IS THE ZERO-LENGTH TERMINATOR
IF TERM RECORD IS FOUND, RETURN TO MONITOR
MOV PAST HEADER TO FIRST ASCII CHARACTER

0000087C DF97
CALR ASCHEX
GET FIRST BYTE OF ADDRESS

0000087E A042
LDB RH2,RH4
1ST BYTE TO CKSUM ACCUMULATOR

00000880 A02A M
LDB RH2,RH4
TRANSPOSE HEX DIGITS

00000882 B2A9FCFC M
AND R2,R0FH
MASK OFF HO BITS

00000886 07020FOF M
ADD RH2,RH2
ADD NYBBLES W/O CARRY

0000088A 80A2 M
LDB RH2,RH4
TO CHECKSUM ACCUMULATOR

0000088C A980
LDB RH6,RH4
HOBYTE OF ADDRESS TO R6

0000088E A046
INC R8
NEXT DIGIT

00000890 A980
INC R8

00000892 CFA2
CALR ASCHEX ; GET SECOND BYTE OF LOAD ADDRESS

00000894 A042
LDB RH2,RH4
2ND BYTE TO CKSUM ACCUMULATOR

00000896 A02A M
LDB RH2,RH4
TRANSPOSE HEX DIGITS

00000898 B2A9FCFC M
AND R2,R0FH
MASK OFF HO BITS

0000089C 07020FOF M
ADD RH2,RH2
ADD NYBBLES W/O CARRY

000008A0 80A2 M
LDB RH4,RH4
LDB RH7
LDB RH2,RH4
ADD IT TO ACCUM

000008A2 8027
ADD RH2,RH2
ADD IT TO ACCUM

000008A4 A04E
LDB RL6,RH4
LDB RL7
LDS TO R6; LOAD ADDRESS IS NOW COMPLETE
01-DEC-83/08:48:48

136 000008A6 A980 INC R8 ;ON TO THE BYTE COUNT
137 000008A8 DFAO CLAR ASCHEX ;GET # OF BYTES IN MSG
138 000008AA A042 LDB RH2,RH4 ;ADD IT TO CHKSUM
140 M NYBSUM
141 000008AC A02A M LDB RL2,RH2 ;TRANSPOSE HEX DIGITS
142 000008AE 82AFCFCC M SRAB RL2,#04 ;MAKE HOB THE LOW
143 000008B2 07020F0F M AND R2,#0F0FH ;MASK OFF HO BITS
144 000008B6 80A2 M ADDB RH2,RL2 ;ADD NYBBLES W/O CARRY
145 000008BA 8027 M ADDB RH7,RH2 ;ADD RUNNING NYBBLE CHECKSUM
148 000008BB A04F M LDB RL7,RH4 ;SAVE # OF DATA BYTES IN HEX FOR RAM LOAD
150 000008BE DFAA M JR RB ;SET CHAR CNT FROM STRING
151 000008C0 E609 M CALR CHKSUM ;TEST 1ST BYTE-CHECKSUM
152 M ERRMSG
153 000008C2 A900 M INC R13 ;COUNT NEW ERROR OCCURRENCE
154 000008C4 08000005 M CP R13,#5 ;REACH MAX # OF ERRORS?
155 000008C8 EB25 M JR UGT,ABRTLD ;TOO MANY ERRORS- RETURN TO MONITOR
156 000008CA A5C1 M SET R12,#01 ;SET 'OLD STRING, REPEAT' FLAG
157 000008CC 4C05FEOO MR LDB ACKBUF,#NAK ;READY BAD TX MSG
158 M 3737
159 000008D2 E8B2 M JR NEWSTR ;REQUEST REPEAT OF MSG AND CLEAR INBUF
160 000008D4 8C78 M SUMOK CLRB RH7 ;RESET ACCUMULATOR FOR FOR SECOND CHECKSUM
161 000008D6 A980 M HXLOAD INC R8 ;NEXT CHR
162 000008DF DFC5 M CALR ASCHEX ;FORM DATA BYTE
163 000008DA A042 M LDB RH2,RH4 ;SEND TO CHKSUM ACCUM
164 M NYBSUM
165 000008DC A02A M LDB RL2,RH2 ;TRANSPOSE HEX DIGITS
166 000008DE B2AFCFCC M SRAB RL2,#04 ;MAKE HOB THE LOW
167 000008E2 07020F0F M AND R2,#0F0FH ;MASK OFF HO BITS
168 000008E6 80A2 M ADDB RH2,RL2 ;ADD NYBBLES W/O CARRY
169 000008E8 8027 M ADDB RH7,RH2 ;ADD RUNNING NYBBLE CHECKSUM
170 000008EA 2E64 M LDB #R6,RH4 ;STORE MACHINE CODE
171 000008EC A960 M INC R6 ;NEXT RAM LOCATION...
172 000008EE A9F0 M DCSB RL7 ;ONE LESS BYTE TO STORE
173 000008F0 EEP2 M HXLOAD ;UNTIL (RL7)=0, STORE THEM BYTES!
174 M 3737
175 000008F2 A980 M INC R8 ;RECORD LOAD COMPLETE
176 000008F4 DFC5 M CALR CHKSUM ;PRODUCE AND COMPARE SECOND BYTE-CHECKSUM
177 000008F6 E609 M JR EQ,GOODRX ;NO ERRORS
178 M ERRMSG
179 000008F8 A900 M INC R13 ;COUNT NEW ERROR OCCURRENCE
180 M NYBSUM
181 000008F8 A900 M INC R13 ;COUNT NEW ERROR OCCURRENCE
182 000008FA 08000005 M CP R13,#5 ;REACH MAX # OF ERRORS?
183 000008FE EB25 M JR UGT,ABRTLD ;TOO MANY ERRORS- RETURN TO MONITOR
184 00000900 A5C1 M SET R12,#01 ;SET 'OLD STRING, REPEAT' FLAG
185 00000902 4C05FEOO MR LDB ACKBUF,#NAK ;READY BAD TX MSG
186 M 3737
187 00000908 E897 M JR NEWSTR ;REQUEST REPEAT OF MSG AND CLEAR INBUF
188 0000090A 4C05FEOO R GOODRX LDB ACKBUF,#ACK ;SET ACKNOWLEDGE TOKEN
189 M 3030
190 00000910 80C8 M CLR R12 ;CLEAR FLAG FOR A NEW STRING
191 00000912 82A9CFCC M JR NEWSTR ;SEND IT TO THE 8550
192 00000914 2101091CR M ABRTLD LD R1,#MSGBLK ;READY ERROR MSG FOR TX TO TX CONSOLE
193 00000918 7F00 M SC #0 ;SEND IT OUT
194 0000091A 7F03 M RETO RETURN TO 8500 MONITOR
195 0000091C 0200 M MSGBLK WORD 0200H ;TX MODE FOR SC40
196 0000091E 0000 M WORD 0000H ;NOT USED
197 00000920 9240 M WORD ENDMSG ;ADDRESS OF ERROR MSG
198 00000922 0228 R WORD LSTCHR-ENDMSG ;# OF CHARMS IN STRING TO BE TX'D
199 00000924 24922020 M ENDMSC ASCII 00H ;ERROR LIMIT EXCEEDED, LOAD IS ABORTED' ;SELF-EXPLANATORY
200 00000940 000A CLRF BYTE 0DH,0AH
201 0000094F 00 LSTCHR BYTE 0
;END OF MAIN ROUTINE; HERE ARE THE SUBROUTINES...

;ASCHEX: THE ASCII CHARACTERS WHOSE ADDRESSES ARE (R8) AND (R8)+1 ARE
;CONSOLIDATED TO FORM ONE HEXADECIMAL BYTE. R3 AND R4 ARE THE WORK SPACE WITH
;THE FORMED HEX BYTE LEFT IN RH4.

;ASCHEX
LD8 RL4,#808 ;GET 1ST ASCII CHARACTER
CALR TSTNUM ;ADJUST ASCII IF CHR IS A-F
ANDB RL4,#0FH ;MASK OFF ZONE BITS
SLAB RL4,#04 ;LSBITS BECOME MSBITS
LDB RH4,RL4 ;READY FOR NEXT DIGIT
INC R8 ;NEXT DIGIT
LDB RL4,#808 ;GET IT
CALR TSTNUM ;ADJUST ASCII IF CHR IS A-F
ANDB RL4,#04 ;FORM COMPLETE BYTE OF DATA
RET ;GO HOME

;CHKSUM: COMPARE THE COMPUTED CHECKSUM WITH THE VALUE CONTAINED IN THE
;STRING TRANSMITTED FORM THE 8550. RUNNING CHECKSUM IS MAINTAINED IN
;RH7. THIS ROUTINE CALLS ASCHEX TO READ THE ASCII STRING AND GEN THE
;TX CHECKSUM.

;CHKSUM
CALR ASCHEX ;GET 1ST BYTE-CHECKSUM
CPB RH7,RH4 ;COMPARE CALCULATED AND GIVEN CHECKSUMS
EXIT RET ;REQUEST ANOTHER TX OF THE STRING IF NEEDED

;CHKTRM: SCANS THE INPUT BUFFER FOR A BYTE COUNT OF ZERO. USES ASCHEX
;TRANSLATE THE TWO ASCII CHARACTERS OF THE DATA COUNT TO HEX. IF THE
;BYTE COUNT IS ZERO, THE LOAD IS CONCLUDED WITHOUT A CHECKSUM SCAN AND CONTROL
;IS RETURNED TO THE MONITOR

;ENTER WITH (R8)= LOCATION OF 1ST CHAR IN LOAD ADDRESS

;CHKTRM
LD R00,R8 ;SAVE CURRENT POSITION IN STRING
IN R8,#5 ;AIM AT 1ST CHR OF BYTE COUNT
CALR ASCHEX ;FORM BYTE COUNT
LD R8,R00 ;RECOVER ORIGINAL POINTER
TESIB RIP, ;IS DATA STRING LENGTH ZERO?
RET NE ;NO, GO BACK AND FINISH PROCESSING

;AT THIS POINT, WHO CARES ABOUT A BIT-ERROR?
LD ACKBUF,#ACK ;SIGNAL THE END
LD R1,#'PTINAK ;READY THE MSG
SC _0 ;SIGNAL TRANSFER END TO HOST COMPUTER
SC #3 ;LOAD COMPLETED, RETURN TO MONITOR

;SET IO: USED TO RESET FC8 FOR SC#0

;SET IO
LD R00,#IKHXIN ;DEST FOR MOVE
LD R01,#ICBLK ;SOURCE FOR MOVE
LD R9,#08H ;_ OF WORDS TO MOVE
WMOVE LDIR @R00,@R01,R9 ;DO IT!
RET ;GO HOME...

;TSTNUM: CORRECTS ASCII CHARACTERS FROM 0 TO F TO ALLOW
;FOR SIMPLE MANIPULATION TO HEX FOPJ4

;TSTNUM
CPB RL4,#39H ;IF 0-9, NO CORRECTION NEEDED
JR LE, ISNUM ;ELSE ADD OFFSET OF 9 TO PRODUCE USEABLE LO NYBBLE
ADDW RL4,#9 ;ELSE ADD OFFSET OF 9 TO PRODUCE USEABLE LO NYBBLE

;MANIPULATION TO HEX FORM

;MANIPULATION TO HEX FORM

;TSTNUM OPB RL4,#59H
JR LE,SNUM ;IF 0-9, NO CORRECTION NEEDED
ADDW RL4,#9 ;ELSE ADD OFFSET OF 9 TO PRODUCE USEABLE LO NYBBLE
SNUM RET ;BACK TO ASCHEX

;end of loader and subroutines

;INTCOM; PROGRAM START ADDRESS FOR ASSEMBLER
### Scalars

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>00000030</td>
</tr>
<tr>
<td>NAK</td>
<td>00000037</td>
</tr>
<tr>
<td>PROMPT</td>
<td>00000038</td>
</tr>
<tr>
<td>RECEND</td>
<td>0000003D</td>
</tr>
</tbody>
</table>

### Strings & Macros

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERMMSG</td>
<td>M</td>
</tr>
<tr>
<td>NYBSUM</td>
<td>M</td>
</tr>
</tbody>
</table>

### Section = $BLLOAD, Inpage Relocatable, Size = 0000FE64

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABRTLD</td>
<td>00000914</td>
</tr>
<tr>
<td>CHKSTR</td>
<td>00000972</td>
</tr>
<tr>
<td>GETSTR</td>
<td>00000850</td>
</tr>
<tr>
<td>INTCOM</td>
<td>00000820</td>
</tr>
<tr>
<td>MSGBLK</td>
<td>0000091C</td>
</tr>
<tr>
<td>SEEK</td>
<td>0000085A</td>
</tr>
<tr>
<td>TKHIN</td>
<td>0000FE54</td>
</tr>
<tr>
<td>TSTSTR</td>
<td>00000872</td>
</tr>
</tbody>
</table>

230 Lines Read
268 Lines Processed
0 Errors
APPENDIX B
LISTING 2

1: /*
2: -
3: - BUSLODR.C: 8550 DOWNLOAD EMULATOR FOR DEC VT-180
4: - WRITTEN IN AZTEC C FOR THE CP/M ENVIRONMENT
5: -
6: - AUTHOR: S. NOVACKI
7: - CREATED: JULY, 1984
8: -
9: */
10: 
11: #include "b:stdio.h" /* standard I/O used for file handling */
12: #define ACK '0' /* definitions of: the ACK token */
13: #define NAK '7' /* the NAK token */
14: #define CR 13 /* end-of-line flag */
15: #define TX_RDY 0x01 /* UART transmitter ready flag bit */
16: #define RX_RDY 0x02 /* receiver ready bit */
17: #define COMM_DATA 0x58 /* UART data register port number */
18: #define COMM_STAT 0x59 /* status register port number */
19: 
20: /*
21: infile:
22: pointer for source file (from disk)
23: numchar:
24: subscript for reading characters from disk file into buffer vector
25: outptr:
26: subscript for sending buffer characters to UART
27: argc:
28: command line argument count, used by "C" console processor
29: errcount:
30: number of consecutive reception errors
31: ilinebuffer:
32: array used in moving characters from disk file using standard
33: I/O to UART using system-specific hardware
34: reply:
35: token read from BusMon system to indicate quality of message
36: tx_stat, rx_stat:
37: UART register statuses used during character-send procedure
38: */
39: 
40: FILE *infile,*fopen();
41: int numchar,outptr,argc,errcount = 0;
42: char ilinebuffer[80],reply,tx_stat,rx_stat;
43: 
44: /*************************************************************/
45: 
46: main(argc,argv)
47: char *argv[];
48: { 
49: 
50: 
51: 
52: /*
open disk file to be sent to the BUSMON system

if a NULL is returned, OPEN has failed, exit to CP/M

if ((infile = fopen(*++argv, "r")) == NULL) {
    printf("open failure on file %s\n", *argv); exit(99);
}

while () { /* a DO-ALWAYS loop, a la BASIC */
    get_reply(); /* get first ACK to commence file transmission */
    get_line(); /* read a line from the TEKHEX disk file */
}

#ASM
/* after reading a line from the disk file, kill IRQs for */
DI /* polled serial I/O for both the record output */
/* and the REPLY input */
#ENDASM

1
/* send record to waiting BusMon unit */
get_reply();
errcount == 0; /* zero error count for each record being sent */

while (reply != ACK) { /* if NAK is received: */
    retrans_record();
    get_reply();
}

#ASM
/* bring back IRQs for BDOS/BIOS disk I/O routines */
ENDASM

get_line(); /* function to read <=80 character from the TEKHEX disk file */

for (numchar =1; numchar <= 80; ++numchar) { /* for numchar = 1 to 80 */
    iolinebuffer[numchar] = getc(infile); /* read from infile to */
    the line buffer */
    if (iolinebuffer[numchar] == EOF) { /*have we reached the end? */
        fclose(infile); /* if so, close the disk file */
        exit(0); /* and back to CP/M... */
    }
    if (iolinebuffer[numchar] == CR) break; /* if a CR, exit from the read */
}

get_line(); /* function to read <=80 character from the TEKHEX disk file */
105: tx_line()
106: /* function to send a character at a time to the 8251A UART */
107: {
108: /* send all the chars in the line buffer to the 8251A */
109: for (outptr = 1; outptr <= numchar; ++outptr) {
110: /* idle until UART transmitter is ready */
111: while (((tx_stat = in(COMM_STAT)) & TX_RDY) != TX_RDY) {}  
112: out(COMM_DATA, iolinebuffer[outptr]); /* send out the character */
113: }
114: }
115: /****************************************************************************/
116: /* get_reply() */
117: /* receives reply token from the BusMon unit after tx_line is performed */
118: {
119: while (((rx_stat = in(COMM_STAT)) & RX_RDY) != RX_RDY) {}  
120: /* idle until UART receiver is ready */
121: reply = in(COMM_DATA); /* get ACK/NAK token */
122: if (reply != ACK) {
123: if (++errcount > 5) load_error(); /* if too many errors, exit */
124: }
125: }
126: /****************************************************************************/
127: /* retrans_record() */
128: /* tx_line by another name, done for improved legibility */
129: /* since numchar is not destroyed by tx_line, this offers a very convenient */
130: /* way to retransmit the same line of characters */
131: {
132: tx_line();
133: }
134: /****************************************************************************/
135: /* load_error() */
136: /* only if five successive load errors are reported by the BusMon */
137: {
138: /* EI */ /* restore IRQS for standard I/O functions */
139: /* printf("error limit exceeded, load operation aborted\n"); */
140: fclose(infile); /* close the disk file */
141: exit(88); /* return to CP/M with error code 88 */
142: /****************************************************************************/
APPENDIX C

LISTING 3

ASM Z8001/Z8002
V01.01-01 (8550) 30-NOV-83/12:00:49

<table>
<thead>
<tr>
<th>1</th>
<th>4000 R</th>
<th>ORG 4000H</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>00004000 21007A3A</td>
<td>GO LD R0,#7A3AH ;SET UP UART FOR 2400 BAUD,</td>
</tr>
<tr>
<td>3</td>
<td>00004004 3A060006</td>
<td>OUTB 0006H,RH0 ;EVEN PARITY, 1 STOP BIT</td>
</tr>
<tr>
<td>4</td>
<td>00004008 3A860006</td>
<td>OUTB 0006H,RL0 ;7 DATA BITS ON 6510</td>
</tr>
<tr>
<td>5</td>
<td>0000400C 2B27</td>
<td>LDB RL0,#27H ;'B' SERIAL PORT TO DUMP</td>
</tr>
<tr>
<td>6</td>
<td>0000400E 3A860007</td>
<td>OUTB 0007H,RL0 ;BYTES TO THE PROLOG</td>
</tr>
</tbody>
</table>

; R10: START ADDRESS (BYTE BOUNDARY) OF PROGRAM TO BE SENT TO PROLOG |
; R11: END ADDRESS (BYTE BOUNDARY) OF PROGRAM |
; R12: 0=FOR EVEN NUMBERED BYTES, 1 FOR ODD NUMBERED BYTES |
; NOTE: PLEASE RECALL THAT THE EVEN BYTES ARE LOW ORDER ADDRESSES BUT |
; ARE ACTUALLY THE HIGH ORDER DATA BYTE. PLEASE REMEMBER THIS WHEN |
; YOU USE THE NOTATION 'HIGH ORDER BYTE' WHEN DETERMINING WHICH |
; PROM YOU ARE PROGRAMMING |

INIT LD R9,R10 ;USE R9 AS WORKSPACE, SAVE R10 FOR NEXT LOAD |
ADD R9,R12 ;SET EVEN/ODD ADDRESSES TO BE DUMPED |
MOVE LDB RL3,#69 ;GET DATUM |
AND R3,#FF00FH ;ISOLATE EACH NYBBLE |
SRLB RH3,#4 ;REDUCE NO DIGIT TO HEX DIGIT |
CPB RH3,#9 ;IS DIGIT DECIMAL OR HEX?? |
LDB RH3,#7 ;IF DECIMAL, NO OFFSET NEEDED |
ADDDB RH3,#7 ;IF HEX, ADD 7 TO PUSH ASCII CODE TO ALPHA |
ADDB RH3,#30H ;IN ANY EVENT, ADD ZONE BITS TO MAKE ASCII CHAR |
LDB RL3,#30H ;SAME ZONE BITS |
ADDB RL3,#7 ;SAME ZONE BITS |
CALR PUTCHR ;MOVE FOR OUTPUT TO PROLOG |
LDB RL4,RL3 ;SAME AS ABOVE |
JR ULE,NOTHX2 ;THIS TIME FOR LO NYBBLE |
ADDDB RL3,#30H ;SAME ZONE BITS |
LDB RL4,RL3 ;SAME AS ABOVE |
CALR PUTCHR ;PUT LETTER IN THE MAILBOX |
CALR PUTCHR ;HERE COMES THE POSTMAN |
INC R9,#2 ;MOVE TO NEXT BYTE OF THE PROGRAM |
CP R9,R11 ;AT THE END OF THE PROGRAM? |
JR ULE,MOVE ;IF NOT, GET ANOTHER BYTE!! |
JR INIT ;BREAKPOINT SET TO STALL HERE, THEN |
LD R9,#2 ;GO TO INIT FOR NEXT PROM |
PUSH R6,0005H ;GET STATUS BITS |
OUTB RL6,0005H ;IS UART STILL BUSY? |
JR INIT ;NO? THEN START OVER |
OUTB RL4,RL3 ;SEND DATA TO THE B-PORT |
RET ;BACK TO MAIN PROG |
END GO ;THAT'S ALL FOLKS!!!
INVESTIGATION OF AIR TRANSPORTATION TECHNOLOGY
AT PRINCETON UNIVERSITY, 1984

Robert F. Stengel
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey

PRECEDEDING PAGE BLANK NOT FILMED
SUMMARY OF RESEARCH

The Air Transportation Technology Program at Princeton University, a program emphasizing graduate and undergraduate student research, proceeded along four avenues during 1984:

- Guidance and Control Strategies for Penetration of Microbursts and Wind Shear
- Application of Artificial Intelligence in Flight Control Systems
- Effects of Control Saturation on Closed-Loop Stability and Response of Open-Loop-Unstable Aircraft

Areas of investigation relate to guidance and control of commercial transports as well as general aviation aircraft. Interaction between the flight crew and automatic systems is a subject of principal concern.

Recently, it has become apparent that severe downdrafts and resulting high velocity outflows present a significant hazard to aircraft on takeoff and final approach. This condition is called a microburst, and while it often is associated with thunderstorm activity, it also can occur in the vicinity of dissipating convective clouds that produce no rainfall at ground level. Microburst encounter is a rare but extremely dangerous phenomenon that accounts for one or two air carrier accidents and numerous general aviation accidents each year (on average). Conditions are such that an aircraft's performance envelope may be inadequate for safe penetration unless optimal control strategies are known and applied. While a number of simulation studies have been directed at the problem, there are varied opinions in the flying community regarding the best piloting procedures, and optimal control strategies remain to be defined.

Graduate student Mark Psiaki has undertaken a study of guidance and control strategies for penetration of microbursts when encounter is unavoidable. In work completed in 1984 [1], it was shown that simple control laws could greatly reduce an aircraft's response to wind shear. Although the response mechanism is the same, jet transport and general aviation aircraft behave somewhat differently in microbursts; the larger, heavier aircraft are more adversely affected by variations in the horizontal wind, while the smaller, lighter aircraft have greater difficulty with the downdraft.

The work in 1984 focused on the determination of optimal control strategies for the microburst encounter. The study began with the computation of optimal control histories using steepest-descent and second-order gradient algorithms. Once an envelope of safe flight has been determined for a typical jet transport, attention will be directed at a general aviation type, and optimal closed-loop control laws will be investigated. During 1984, a survey paper on the subject was presented at the NASA Workshop on Wind Shear/Turbulence Inputs to Flight Simulation and Systems Certification [2].

Undetected system failures and/or inadequately defined recovery procedures have contributed to numerous air carrier incidents and accidents. The infamous DC-10 accident at Chicago's O'Hare Airport, in which loss of an engine pod, subsequent loss of subsystems, and asymmetric wing stall led to disaster, provides a prototype for the kind of tragedy that could be averted by intelligent flight control systems. (An intelligent control system is one that uses artificial intelligence concepts, e.g., an expert systems program, to improve performance and fault tolerance.) Although
many methods of modern control theory are applicable, the scope of the problem is such that none of the existing theories provides a complete and practical solution to the problem. At the same time, heuristic logic may be applicable, but it has yet to be stated in satisfactory format.

Graduate student David Handelman has begun to develop a knowledge-based reconfigurable flight control system that will be implemented with the Pascal programming language using parallel microprocessors. This expert system could be considered a prototype for a fault-tolerant control system that could be constructed using existing hardware. In a parallel effort, graduate student Chien Huang used the LISP programming language to investigate the utility of a string-oriented, recursive logical system in the same role. A principal distinction between this and the previous approach is that flight control code will be modified in response to control system failures.

One of the virtues of highly reliable electronic flight control systems is that an aircraft's stability and response, i.e., its closed-loop flying qualities, can be tailored to the pilot's needs. For reasons of performance and maneuverability, it may be desirable to design the aircraft so that its natural (unaugmented) modes of motion are unstable, with the understanding that the flight control system will provide the necessary stability by deflecting control surfaces to counter potentially divergent motions. Because control surfaces have limitations on their displacements and rates of travel, stability can be restored only within a bounded region about the trim point. If the aircraft's motions exceed the boundaries, the available control forces and moments will not be sufficient to prevent divergence.

Graduate student Prakash Shrivastava has been developing methodologies for determining the stability boundaries and control response for systems containing control saturation[3,4]. Analysis has been carried out using phase-plane plots, in which saturation and stability boundaries are represented by straight lines, stable trajectories approach equilibrium points, and unstable trajectories diverge to infinity. The analysis pertains to systems containing unequal saturation boundaries, as well as those with multiple saturating controls.

Future control-system engineers will benefit from design procedures that are computer-intensive, and it is important to create computer programs that allow designers to describe and analyze complex systems interactively. Russell Nelson has been developing a control-system design program based on the LISP language mentioned above *. Design algorithms will be based on classical concepts of transfer-function analysis, and LISP will allow multiloop systems to be assembled and tested within the computer.

The NASA grant supporting student research in air transportation technology has inestimable value in helping educate a new generation of engineers for the aerospace industry, and it is producing research results that are relevant to the continued excellence of aeronautical development in this country.

* Nelson, R. F., "Computer Aided Control System Design - Progress Report", Princeton University Independent Work Report, Jan. 1985. The goal of this independent project is to create a program, using LISP, that will serve as a tool in the design of control systems. The program will be menu driven and graphic with menu prompting. The frequency response of control systems will be examined through Bode analysis techniques. The program will enable the user to make rapid, interactive changes to a control system and then reevaluate the system's performance in order to optimize the design. The project also will give indication of LISP's usefulness in the field of control system design.
REFERENCES AND ANNOTATED BIBLIOGRAPHY


Penetration of a microburst during takeoff or approach is an extreme hazard to aviation, but analysis has indicated that risks could be reduced by improved control strategies. Attenuation of flight path response to microburst inputs by elevator and throttle control was studied for a jet transport and a general aviation aircraft using longitudinal equations of motion, root locus analysis, Bode plots of altitude response to wind inputs, and nonlinear numerical simulation. Energy management relative to the air mass, a pitch-up response to decreasing airspeed, increased phugoid-mode damping, and decreased phugoid natural frequency were shown to improve microburst penetration characteristics. Aircraft stall and throttle saturation were found to be limiting factors in an aircraft's ability to maintain flight path during a microburst encounter.


Much remains to be learned about the hazards of low-altitude wind shear to aviation. New research should be conducted on the nature of the atmospheric environment, on aircraft performance, and on guidance-and-control aids. In conducting this research, it is important to distinguish between near-term and far-term objectives, between basic and applied research, and between uses of results for aircraft design or for real-time implementation. Advances in on-board electronics can be applied to assuring that aircraft of all classes have near-optimal protection against wind shear hazards.


Although unstable systems can be stabilized using feedback control, constraints on the rates and magnitudes of control variables limit the region of stability. Stability boundaries must be evaluated during control system design to assure satisfactory system performance. This analysis becomes more relevant where compromises between the degree of plant instability and upper limits on the rates and magnitudes of control variables have to be made. This paper presents stability boundaries for linear feedback control laws. The method of normal-mode decomposition using the phase-plane approach appears to be well-suited to the evaluation. It is concluded that although the saturation boundaries are dependent on feedback gain magnitudes, the stability boundaries principally depend on the open-loop system dynamics and the allowable control deflection.


Constraints on the magnitudes of control variable limit the region where unstable systems can be stabilized using feedback control. Stability boundaries must be evaluated to assure satisfactory system performance. A method is presented to determine the stability boundaries for linear second-order plants with
saturating control and several classes of open-loop instability. In the saddle-point case, the modal axis of the stable mode centered on the equilibrium points and the saturation boundaries establish the regions of stability. For unstable nodes and foci, the stability boundaries are represented by unstable limit cycles enclosing the stable origin. The stability regions vary with changes in feedback gain. These results have fundamental significance for determining the degree to which unstable plants can be controlled in practical application.
SAFE MICROBURST PENETRATION TECHNIQUES:
A DETERMINISTIC, NONLINEAR, OPTIMAL CONTROL APPROACH

Mark L. Psiaki
Mechanical and Aerospace Engineering Department
Princeton University
Princeton, NJ

PRECEDING PAGE BLANK NOT FILMED
Efforts in the first area involve the design of compensators for microburst disturbance rejection. Preliminary results indicate improvements when the microburst is modeled as a colored noise process. The second task is to model the unsteady aerodynamic effects upon an aircraft's longitudinal dynamics as related to the wind shear encounter problem. The third effort concentrates on the determination of optimal trajectories through microbursts. Standard techniques of deterministic nonlinear optimal control have been used. This is the main subject of this presentation.

- STOCHASTIC LINEAR OPTIMAL CONTROL WITH COLORED NOISE ASSUMPTIONS

- UNSTEADY AERODYNAMIC EFFECTS UPON LONGITUDINAL DYNAMICS

- DETERMINISTIC, NONLINEAR, OPTIMAL CONTROL THROUGH GIVEN MICROBURSTS
DETERMINISTIC, NONLINEAR, OPTIMAL CONTROL PROBLEM

To calculate an optimal trajectory, one must have some criterion by which a determination can be made of what is optimal. For the unconstrained, fixed-time, free-end-point problem, the criterion is that the scalar, positive definite cost function, $J$, be minimized. In this formulation $x$ is the state vector, $u$ is the control vector, and $x = f(x,u,t)$ defines the plant dynamics including disturbance inputs. $J$ is minimized by an appropriate choice of the control vector time history. Because the aircraft plant dynamics are represented by differential equations, the continuous-time formulation of this problem seems natural. Numerical solution techniques, on the other hand, are more appropriate to the discrete-time problem. Therefore, a zero-order-hold assumption is made for the control time history, and the continuous-time problem is transformed into a discrete-time problem. Now the optimal control problem is in the general form of a static, finite-dimensional, constrained optimization problem. Standard techniques may be applied. Two such techniques are the Steepest-Descent and Newton's Second Gradient methods. The Steepest-Descent method uses an initial guess for the control time history, then differentiates the cost with respect to it to determine an optimization step that will yield the largest decrease in the cost. Newton's method is merely a generalization of the Newton-Raphson method for determination of a root of a scalar equation. In this case it is applied to the set of simultaneous equations which comprise the necessary condition for optimality: $dJ/duk = 0.0$ for $k = 1...N-1$. A FORTRAN package was developed for the implementation of these solution techniques. It uses 4th order Runge-Kutta integration to transform the continuous time problem into the discrete time problem. The Steepest-Descent method is used for the initial improvements to the control time history because it is cheaper per optimization step and yields large changes in cost, $J$, per step when not in the neighborhood of the optimum. Newton's method is used to get to the final solution because it converges rapidly in the neighborhood of the solution.
**CONTINUOUS TIME**

Given $x(t_0) = x_0$, $\dot{x} = f(x, u, t)$

Find $u(t)$ for $t_0 < t < t_f$

To minimize $J = \int_{t_0}^{t_f} L(x, u, t) dt + V(x(t_f))$

**DISCRETE TIME**

Given $x_1$, $x_{k+1} = f(x_k, u_k, k)$

Find $u_k$ for $k = 1 \ldots N-1$

To minimize $J = \sum_{k=1}^{N-1} L(x_k, u_k, k) + V(x_N)$

**SOLUTION TECHNIQUES**

- Steepest Descent
- Second Gradient (Newton's Method)
An optimal trajectory was calculated for the Boeing-727 model and the range dependent, sinusoidal microburst used in previous studies. After each optimization step the cost function was evaluated. This figure contains a plot of the cost as a function of the number of optimization steps. It took 51 steps to reach the optimum to within reasonable accuracy. Thirty-four steps were Steepest-Descent steps, and seventeen were taken using Newton's method. Note that the Steepest-Descent portion of the optimization required only about a third as much CPU time as the Newton's method portion, despite the fact that there were twice as many Steepest-Descent steps. This optimization probably could have been done more efficiently by waiting longer to make the switch from Steepest Descent to Newton's method. The erratic pattern of the cost function decreases, though typical of numerical optimization techniques, makes automation of the switching process difficult.
The results of this trajectory optimization are compared with the results of two previous flights by the same model through the same microburst: an open-loop flight and a closed-loop flight. The control law used in the closed-loop flight was the best so far designed by the author using classical design techniques. On this plot the optimal trajectory is indistinguishable from the nominal trajectory, a -3 deg. glide slope. In fact, it deviated no more than 1.5 ft. from the nominal. The previously best trajectory (the closed-loop run) yielded a 180 ft. perturbation, while the open-loop perturbation was about 1000 ft.
THREE AIRSPEED AND THREE THROTTLE SETTING TIME HISTORIES

For the optimal case, it is interesting that the airspeed variation is approximately that of the microburst itself. The optimum solution expends little effort in trying to maintain airspeed. The airspeed for this case never comes near the 1-g stall speed, so no problem is encountered. The throttle activity is much lower for the optimum case than for the poorer performing closed-loop case. Its phase lead may partially explain this. Both the airspeed and the throttle time histories for the optimum case are perturbed from the nominal prior to the initial encounter of the microburst. This is due to the nature of the deterministic optimization; the algorithm "knows" ahead of time what is about to happen and acts accordingly. This behavior is also visible on the next figure. This fact precludes the implementation of this algorithm as a control law unless sensors can be developed which sense the wind ahead of the aircraft (a possibility which will be pursued at a later date). Also note that the headwind, downdraft and tailwind zones marked are approximate. The times of their encounters vary slightly from case to case.
THREE ANGLE-OF-ATTACK AND THREE PITCH-ANGLE TIME HISTORIES

The open-loop and closed-loop angle-of-attack time histories are very similar to each other. The optimal angle-of-attack time history is out of phase with the corresponding airspeed variation. This indicates that the optimal glide slope control is primarily by angle-of-attack variation to maintain lift in the presence of airspeed variations. The pitch-angle time history bears out this interpretation, taking into account the changes in the relationship between the two angles due to the wind variations. In the open-loop and closed-loop cases, the pitch-angle is not held low enough during the headwind zone, and it is not held high long enough during the tailwind zone.
CONCLUSIONS

A relatively large amount of computer time (64 minutes of CPU activity on an IBM 3081 computer) was used for the calculation of this optimal trajectory, but it is subject to reduction with moderate effort. The Deterministic, Nonlinear, Optimal Control algorithm yielded excellent aircraft performance in trajectory tracking for the given microburst. It did so by varying the angle of attack to counteract the lift effects of microburst-induced airspeed variations. Throttle saturation and aerodynamic stall limits were not a problem for the case considered, proving that the aircraft's performance capabilities were not violated by the given wind field. All closed-loop control laws previously considered performed very poorly in comparison, and therefore do not come near to taking full advantage of aircraft performance.

- DETERMINISTIC, NONLINEAR, OPTIMAL CONTROL, AN EFFECTIVE THOUGH EXPENSIVE NOMINAL SOLUTION

- SUFFICIENT AIRCRAFT PERFORMANCE FOR SAFE ENCOUNTER OF GIVEN MICROBURST

- INSUFFICIENCY OF PRACTICAL CONTROL LAWS STUDIED TO DATE
PLANNED FUTURE WORK

Effort will be made to reduce the CPU time per trajectory optimization by improving the efficiency of the algorithm, but the basic approach will remain the same. The microbursts used thus far have been idealized. Microburst data from the JAWS project will be used to get realistic wind fields. These will be checked to see if and how any of these exceed aircraft performance limits by doing trajectory optimizations through them. Optimal trajectory solutions are also greatly affected by variations of the cost functions, $L(x,u,t)$ and $V(x(tf))$. These effects will be considered as will the optimum for a general aviation aircraft. Once the performance capabilities are well understood, the goal will be to design practical control laws which come as close to these limits as possible. The use of lead information about the wind shear will be considered during this phase to determine what information would be useful to a closed-loop control law. Unsteady aerodynamics effects remain to be studied to determine their impact on the validity of the aircraft models used here.

- **DETERMINISTIC, NONLINEAR, OPTIMAL CONTROL (DNLOC) ALGORITHM IMPROVEMENTS**

- **FUTURE OPTIMIZATION RUNS**
  - JAWS MICROBURSTS
  - VARYING COST FUNCTIONS, $L(x,u,t)$, $V(x(tf))$
  - GENERAL AVIATION AIRCRAFT

- **PRACTICAL CONTROL STRATEGIES APPROACHING OPTIMUM PERFORMANCE**

- **UNSTEADY AERODYNAMIC EFFECTS**
INTELLIGENT FAULT-TOLERANT CONTROLLERS

Chien Y. Huang
Princeton University
Princeton, NJ
OVERVIEW

• ROLE OF (ARTIFICIAL) INTELLIGENCE IN FAULT-TOLERANT CONTROLS
• DESCRIPTION OF THE SYSTEM BEING DESIGNED
• AN EXAMPLE OF NEW CONTROL RECONFIGURATION APPROACHES
• SUMMARY AND FUTURE WORK
A system with fault-tolerant controls is one that can detect, isolate, and estimate failures and perform necessary control reconfiguration based on this new information. Artificial intelligence (AI) is concerned with semantic processing, and it has evolved to include the topics of expert systems and machine learning. Our research represents an attempt to apply AI to fault-tolerant controls, hence, the name intelligent fault-tolerant control (IFTC).

We are seeking generic solutions to problems, providing a system based on logic in addition to analytical tools, and offering machine learning capabilities. The advantages, to name a few, are that redundant system-specific algorithms are no longer needed, that "reasonableness" is used to quickly choose the correct control strategy (among many available ones), and that the system can adapt to new situations by learning about its effects on system dynamics.

INTELLIGENT FAULT-TOLERANT CONTROLS

- FAULT-TOLERANT CONTROL
  - FAULT DETECTION AND ISOLATION
  - FAULT ESTIMATION
  - CONTROL RECONFIGURATION

- ARTIFICIAL INTELLIGENCE (AI)
  - SEMANTIC PROCESSING
  - EXPERT SYSTEM (PROBLEM SOLVERS)
  - MACHINE LEARNING

- INTELLIGENT FAULT-TOLERANT CONTROLS (IFTC)
  - HYBRID OF ANALYTICAL METHODOLOGY AND AI
  - DIFFERENCES TO CONVENTIONAL APPROACH
  - ADVANTAGES
The Intelligent Fault-Tolerant Control (IFTC) System, which is currently being designed at the Laboratory of Control and Automation in Princeton University, consists of three main sub-systems: front-end processor, which interfaces the system with the users; knowledge base, which houses the data base IFTC needs; and a heuristic and computational engine, which has both logic and numeric processing capabilities.
The front-end processor is responsible for providing a friendly interface between the system and aircraft crew and designers of control strategies and heuristic rules. Through it the knowledge base can be updated or augmented. It will also guide a designer (usually an expert in some aspect) through the process of creating new rules and control strategies.

FRONT-END PROCESSOR

- Interfaces the system with aircraft crew/designer
- Updates and augments the knowledge base
- Guides user through the process of creating new rules and control strategies
The knowledge base stores the knowledge that the system has about itself and about the "WORLD" (i.e., the flight environment). It contains dynamic models of the aircraft in which the IFTC system is resident; control laws for different flight conditions (high, low Mach number, wind shear, etc.), the rules under which they apply, algorithms for failure detection, isolation and estimation, and an interconnection map (and status) of all devices present on the aircraft.

**KNOWLEDGE BASE**

**STORAGE FOR**

- DYNAMIC MODELS OF AIRCRAFT
- CONTROL LAWS FOR DIFFERENT FLIGHT CONDITIONS AS WELL AS RULES UNDER WHICH THEY APPLY
- FAULT DETECTION, ISOLATION, AND ESTIMATION ALGORITHMS
- INTERCONNECTION MAP FOR ONBOARD DEVICES
There are two distinct processors in the Heuristic and Computational Engine subsystem. The heuristic processor possesses the logic needed to prune branches in free-structure rule-space. It also has the power to make decisions when logic does not apply. The computational processor is endowed with sufficient numerical capability to find, for example, new gains for feedback and observer equations and to evaluate the effectiveness of a new control law by simulation.

**HEURISTIC AND COMPUTATIONAL ENGINE**

- **HEURISTIC PROCESSOR**
  - PRUNES BRANCHES IN TREE-STRUCTURE RULE-SPACE
  - PERFORMS HEURISTIC DECISIONS

- **COMPUTATIONAL PROCESSOR**
  - FINDS NEW CONTROL GAINS
  - EVALUATES EFFECTIVENESS OF THE CONTROL LAWS
The lateral dynamics of an F-8C aircraft provides an example. The objective is to control its lateral motions by model-following. Suppose that the system is warned about an aileron-failure after a maneuver (it became biased at 1°). The question is what to do now.

"INTELLIGENT" CONTROL RECONFIGURATION - AN EXAMPLE

• MODEL: F-8C LATERAL DYNAMICS

\[
x = Fx + Gu + w, \quad z = x, \quad u = -Cx + C_Mx_M
\]

\[
x = (p, r, \beta, \phi, \delta_A, \delta_R, \delta_{AC}, \delta_{RC}), \quad u = (\delta_{AC}, \delta_{RC})
\]

• FAILURE: AILERON BIASED AT 1° DEGREE

(FAULT DETECTION AND ESTIMATION ASSUMED)
A possible solution in conventional approaches is to set aside a value slot in software to represent the bias and add it to aileron commands. This slot normally holds a zero until a bias is detected at which time the slot is set to the estimated bias magnitude. Disadvantages of this method are that extra computation is necessary even when no bias is present and one such slot must be provided for every control on the airplane.

Faced with the same problem, the IFTC will instead change the dynamic model to reflect the new knowledge it acquired. This solution differs from the previous one in that the codes which control the aileron are being changed in real time. Therefore there is no need for reserved slots and no extra computation is carried out when there is no failure. Furthermore, the command is generic so that it will handle all cases with biases. We call this approach incremental learning because the IFTC uses its own tools to modify its "world" knowledge without outside intervention. It must also be pointed out that this example actually illustrates a restructurable control (as opposed to reconfigurable control) in the sense that the structure of the system has been changed.

Other failures that will be addressed in the future are losses of actuator or sensor. We will also consider critical flight conditions like wind shear penetration, where we will provide IFTC with knowledge to carry out such a task. In all these cases, we see clearly the need for intelligent and generic controls.
CONTROL RECONFIGURATION

• CONVENTIONAL FIX: ADD BIAS TO COMMAND IN A RESERVED SLOT

• NEW APPROACH: CHANGE DYNAMIC MODEL (USING A GENERIC COMMAND) TO REFLECT THE NEW KNOWLEDGE (INCREMENTAL LEARNING)

• OTHER "FAILURE" SITUATION: LOSS OF ACTUATOR OR SENSOR, WIND SHEAR - NEED OF INTELLIGENT GENERIC CONTROLS
A simulation was run with an aileron failure; (A) shows the normal roll response to a doublet aileron command, and (B) shows the failed case where at the end of the doublet command, the aileron became biased at 1°. (C) shows the control restructured a half second after the failure occurred; thus bringing the roll back to a desired state (at 0 deg/sec). (D) shows the aileron command, failure, and restructuring.
SUMMARY

WE HAVE

• OUTLINED AN INTELLIGENT FAULT-TOLERANT CONTROLLER
• ILLUSTRATED THE NEW APPROACH BY AN EXAMPLE

FUTURE WORK

• EXPAND ON THE SUBSYSTEMS
• EXPERIMENT WITH ANALYTIC FAULT DETECTION ALGORITHMS
• STUDY THE MACHINE LEARNING
STABILITY BOUNDARIES FOR
COMMAND AUGMENTATION SYSTEMS

P. C. Shrivastava
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, NJ
INTRODUCTION

For statically unstable aircraft, there is an increased need to understand the effects of control saturation. The longitudinal mode of unstable aircraft usually has a single unstable pole, whereas the lateral-directional mode may have two real poles or a pair of complex-conjugate poles in the right half plane. Reduced-order models are often used to determine the stability and performance of the aircraft approximately. This study examines the effects of control saturation using reduced-order models.

The stability boundaries for command augmentation systems are determined for three types of singularities: saddle-point, unstable nodes, and unstable foci. Control saturation imposes bounds on command vectors for which equilibrium can be attained. For the cases of saddle-point and unstable nodes, the region of stability reduces to zero for command vectors which demand a steady value of control exceeding the control saturation limits. In the case of unstable foci, the region of attraction does not gradually reduce in size, but at some point it breaks abruptly.

OVERVIEW

- INTRODUCTION
- PREVIOUS WORK
- COMMAND AUGMENTATION SYSTEMS
- CONCLUSIONS
STABILITY AUGMENTATION SYSTEM

A block diagram of the system under consideration is shown. It consists of a dynamic system to be controlled (the "plant"), a feedback controller, and a saturating element on one or more of the controls. Command inputs can be ignored in the stability analysis of this constant-coefficient system.

\[
\dot{x} = Fx + Gu
\]

\[
u = -Cx
\]

\[|u| < u_M\]
Specific inputs and outputs must be considered in the command augmentation system. Defining \( y \) as an output vector that is a linear combination of state vector components, \( y^* \) is the desired value of the output. The resulting closed-loop system is described by an ordinary differential equation, whose equilibrium state and control vectors, \( x^* \) and \( u^* \), can be related to the desired output.

The steady-state value of control is independent of feedback gain, which can be obtained either from open- or closed-loop dynamics. For open-loop unstable systems, feedback is mandatory to achieve stability and command equilibrium. The linear feedback regulator provides satisfactory transient response to meet performance specifications otherwise not obtainable. The state equilibrium depends only on the open-loop dynamics and control magnitude. A steady-state control \( u^* \) exists only for the "nonsingular" command vector.

\[
\begin{align*}
\dot{y} &= C y^* \\
\dot{x} &= F x + G u \\
u &= C_B x + C_F y^* \\
y^* &= H x \quad x^* = \text{equilibrium}
\end{align*}
\]

**CLOSED-LOOP**

\[
\begin{align*}
\dot{y} &= (F - G C_F) x^* + G C_F y^* \\
x^* &= -(F - G C_F)^{-1} G C_F y^* \\
u^* &= -C_B (F - G C_F)^{-1} G C_F y^* + C_F y^*
\end{align*}
\]

**EQUILIBRIUM**

\[
\begin{align*}
x^* &= -(F - G C_F)^{-1} G C_F y^* \\
u^* &= -C_B (F - G C_F)^{-1} G C_F y^* + C_F y^*
\end{align*}
\]

**WHERE**

\[
\begin{align*}
C_F &= S_{22} + C S_{12} \\
S_{22} &= -H x^{-1} G \\
S_{12} &= (-F^{-1} G) S_{22} \\
C_B &= -C
\end{align*}
\]

**OPEN-LOOP EQUILIBRIUM**

\[
\begin{align*}
x^* &= -F^{-1} G y^* \\
y^* &= H x x^* = -H x F^{-1} G y^* \\
u^* &= (-H x F^{-1} G)^{-1} y^* \\
&= - (H x F^{-1} G)^{-1} H x x^*
\end{align*}
\]
COMMAND AUGMENTATION SYSTEM (CONCLUDED)

IT CAN BE SHOWN THAT

\[ \dot{u}^* = - CB(F - G CF)^{-1} G CF \dot{y}^* + CF \dot{y}^* \]
\[ = (-HF^{-1} G)^{-1} \dot{y}^* \]

- \( \dot{u}^* \) INDEPENDENT OF GAIN \( c \)
- \( c \) DETERMINES RESPONSE
- \( \dot{x}^* \) LOCATION DETERMINED BY STATE EQUATION
- \( \dot{u}^* \) EXISTS FOR "NONSINGULAR COMMAND" EQUILIBRIUM
EFFECTS OF CONTROL SATURATION

The control saturation limits and open-loop dynamics determine the minimum and maximum values of state equilibrium. Saturation prevents the system from attaining the desired equilibrium and response, and it imposes bounds on achievable command vector \( y^* \). To avoid saturation, desired state equilibrium points must lie within the unsaturated region. However, this still does not guarantee that the trajectories would not enter a saturated-control region, for some initial conditions and/or commands. This is mainly determined by the eigenvectors in the unsaturated region. Thus, saturation enforces bounds on the command vectors for which equilibrium could actually be attained without saturation. Equilibrium cannot be attained for the command vectors for which the state equilibrium point is located in the saturated region.

Feedback gain \( C \) alters the response of the system, but it does not affect \( u^* \). The command vectors for which equilibrium can be attained are independent of feedback gain. \( H_x \) does not affect the state equilibrium or the feedback gain. It changes the prefilter gain which shapes input to achieve the desired equilibrium. \( u^* \) changes with the command vector; hence, the saturation boundaries change with commands.

\[
\begin{align*}
-u_M &< u^* < +u_M : \text{CONTROL SATURATION} \\
\begin{bmatrix} x^* \end{bmatrix} &= -F^{-1}G\begin{bmatrix} u^* \end{bmatrix} \\
-x-U_M &< x^* < x+U_M : \text{BOUNDS ON STATE EQUILIBRIUM} \\
Y^* &= H_x\begin{bmatrix} x^* \end{bmatrix} : \text{BOUNDS ON ACHIEVABLE COMMANDS} \\
-Y_U &< Y^* < Y_U : \text{BOUNDS ON ACHIEVABLE COMMANDS}
\end{align*}
\]

- DEPENDENCE OF STATE EQUILIBRIUM ON \( F^{-1}G \)
- EFFECTS OF VARIATIONS OF \( C, H_x \)
The stability boundaries for the minimum-control-energy (MCE) case are shown on the next three figures. Note the variations in saturation boundaries and location of $x^*$ with changes in the command vector. The region of stability remains unchanged. Equilibrium can be achieved only for those command vectors for which the equilibrium point lies within the saturated region. Invariance of stability boundaries with changes in command command vectors is a unique result for the saddle-point MCE case.
MCE CASE (CONCLUDED)

\[ Y = 2.2 \]
The stability boundaries are shown on the next three figures, where unlike the minimum-control-energy (MCE) case, the stability boundaries change with commands. The locations of equilibrium points in the saturated region do not change with the command vectors. The region of stability is biggest for zero command, i.e., the maximum region of stability is achieved for the stability augmentation case. For non-zero command vectors, the stability region shrinks. It reduces to zero when the desired equilibrium control exceeds the saturation limits.

For each command vector, trajectories seek separate equilibrium points; hence, the trajectories starting from the same initial conditions follow entirely different paths in the phase plane. For this reason, markedly different time-histories for different command vectors are obtained, though the system eigenvalues/eigenvectors remain unchanged.
STABILITY BOUNDARIES FOR UNSTABLE NODES

The next three figures show the stability boundaries for the case of unstable nodes for various command vectors. As noticed before, the saturation boundaries here also change with changes in commands. The sizes of limit cycles, which represent the saturation boundaries, also change. The region of stability is biggest for the stability augmentation system (SAS) case. For increasing command magnitudes, the steady-state equilibrium point moves away from the origin. The region of stability shrinks, and eventually it reduces to zero for commands that require $u^* \geq |u_m|$.

\[
F = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 3.5 \end{bmatrix}
\]
The stability boundaries for the case of unstable foci are unstable limit cycles, as shown on the next six figures. Apparently, there is little variation in the size of stability region with commands. The stable equilibrium point moves to the right of the origin with increasing command values, and the saturation boundaries also shift. At control saturation limits, this equilibrium point lies on the saturation boundary, but the region of stability does not shrink to zero, contrary to the cases of saddle-point and unstable nodes.

Further increase in commands moves the desired equilibrium point farther to the right, and another limit cycle emerges. The trajectories within it converge to this new limit cycle; those within the original limit cycle also converge to it. Thus, the new limit cycle is stable. This "inner" limit cycle grows with increase in commands, until it coincides with the "outer" limit cycle. Any further increase in command results in breaking of the closed stability region (or "bursting" of the limit cycle), making the entire region unstable. This peculiar result in the case of unstable foci is under further investigation.
UNSTABLE FOCI (CONTINUED)

\[ \gamma = 0.8 \quad \gamma = 1.4 \]

\[ \gamma = 1.8 \quad \gamma = 2.0 \]
UNSTABLE FOCI (CONTINUED)

\[ \gamma = 2.1 \]
\[ \gamma = 2.3 \]
\[ \gamma = 2.5 \]
\[ \gamma = 2.7 \]
UNSTABLE FOCI (CONCLUDED)

\[ y = 2.9 \]
CONCLUSIONS AND FUTURE WORK

The Stability Augmentation System (SAS) is a special case of the Command Augmentation System (CAS). Control saturation imposes bounds on achievable commands. The state equilibrium depends only on the open-loop dynamics and control deflection. The control magnitude to achieve a desired command equilibrium is independent of the feedback gain. A feedback controller provides the desired response, maintains the system equilibrium under disturbances, but it does not affect the equilibrium values of states and control.

The saturation boundaries change with commands, but the locations of the equilibrium points in the saturated region remain unchanged. Nonzero command vectors yield saturation boundaries that are asymmetric with respect to the state equilibrium. Except for the saddle-point case with MCE control law, the stability boundaries change with commands. For the cases of saddle-point and unstable nodes, the region of stability decreases with increasing command magnitudes; it is reduced to zero for commands that require steady-state control $u^* \geq |u_m|$.

The regions of stability are biggest for the SAS. In the case of unstable foci, the region of stability does not vanish at $u^* = u_m$. An "inner" limit cycle is obtained, which grows with increase in commands until it coalesces with the "outer" limit cycle. Any further increase in command breaks this closed stability boundary. For a fixed degree of stability, different commands cause markedly different responses because they seek different equilibrium states.

• STABILITY AUGMENTATION SYSTEM: A SPECIAL CASE OF
  COMMAND AUGMENTATION SYSTEM

• CONTROL SATURATION LIMITS ACHIEVABLE COMMANDS

• DEPENDENCE OF STATE EQUILIBRIUM ON OPEN-LOOP DYNAMICS
  AND CONTROL

• STATE EQUILIBRIUM, STEADY-STATE CONTROL: INDEPENDENT
  OF FEEDBACK GAIN

• FEEDBACK: TO ACHIEVE DESIRED RESPONSE: MAINTAIN
  EQUILIBRIUM IN PRESENCE OF DISTURBANCE
CONCLUSIONS

• SATURATION BOUNDARIES CHANGE WITH COMMAND VECTORS

• STABILITY BOUNDARIES CHANGE EXCEPT FOR MCE CASE

• BIGGEST REGION OF STABILITY FOR SAS

• REGION OF STABILITY REDUCES TO ZERO FOR $y = \pm y_{\text{MAX}}$ (EXCEPT IN THE CASE OF UNSTABLE FOCI)

• MARKEDLY DIFFERENT "LOOKING" RESPONSES FOR DIFFERENT COMMAND VECTORS

FUTURE WORK

• STABILITY BOUNDARIES FOR TWO-INPUT COMMAND AUGMENTATION SYSTEM

• DESIGN OF LATERAL-DIRECTIONAL COMMAND AUGMENTATION SYSTEM

• VARIATION OF STABILITY BOUNDARIES WITH FLIGHT CONDITIONS
This report summarizes the research conducted during 1984 under the NASA/FAA sponsored Joint University Program for Air Transportation Research. The material was presented at a meeting held at NASA Langley Research Center, Hampton, Virginia, January 18, 1985. The Joint University Program is a coordinated set of three grants sponsored by NASA Langley Research Center and the Federal Aviation Administration, one each with the Massachusetts Institute of Technology, Ohio University, and Princeton University. Completed works, status reports, and bibliographies are presented for research topics, which include navigation, guidance, control, and display concepts. An overview of the year's activities for each of the schools is also presented.