

NASA CONTRACTOR REPORT 177471

GUIDANCE SIMULATION AND TEST SUPPORT
FOR DIFFERENTIAL GPS FLIGHT EXPERIMENT

BY

G. J. GEIER, P. V. W. LOOMIS AND
A. CABAK

(NASA-CR-177471) GUIDANCE SIMULATION AND
TEST SUPPORT FOR DIFFERENTIAL GPS FLIGHT
EXPERIMENT (Theory and Applications
unlimited Corp.) 47 D

N90-10021

CSCL 17G

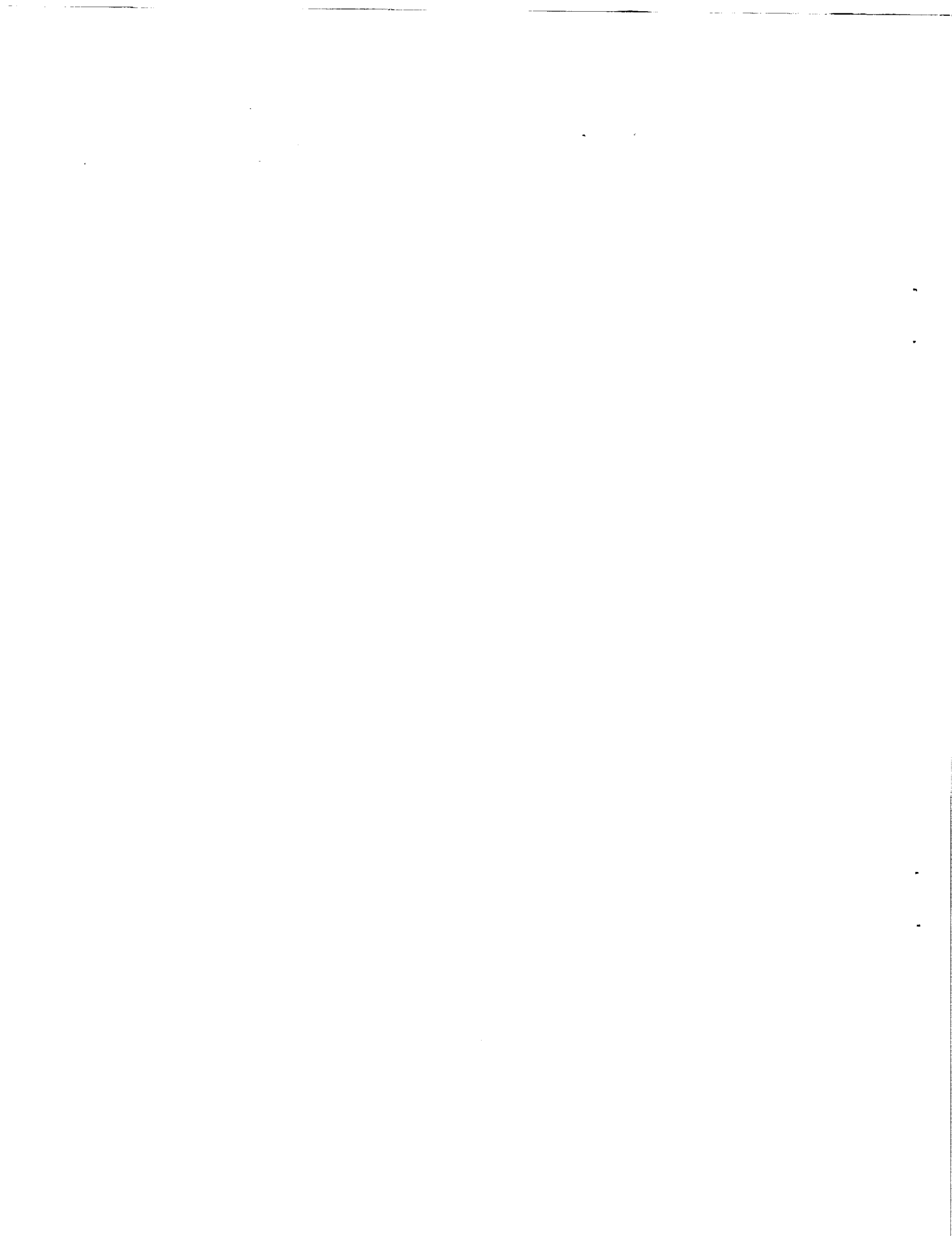
63/04

Uncl us
0237154

CONTRACT NAS 2 -12378
OCTOBER 1987



Date for general release October 1989



GUIDANCE SIMULATION AND TEST SUPPORT
FOR DIFFERENTIAL GPS FLIGHT EXPERIMENT

BY

G. J. GEIER, P. V. W. LOOMIS
AND A. CABAK

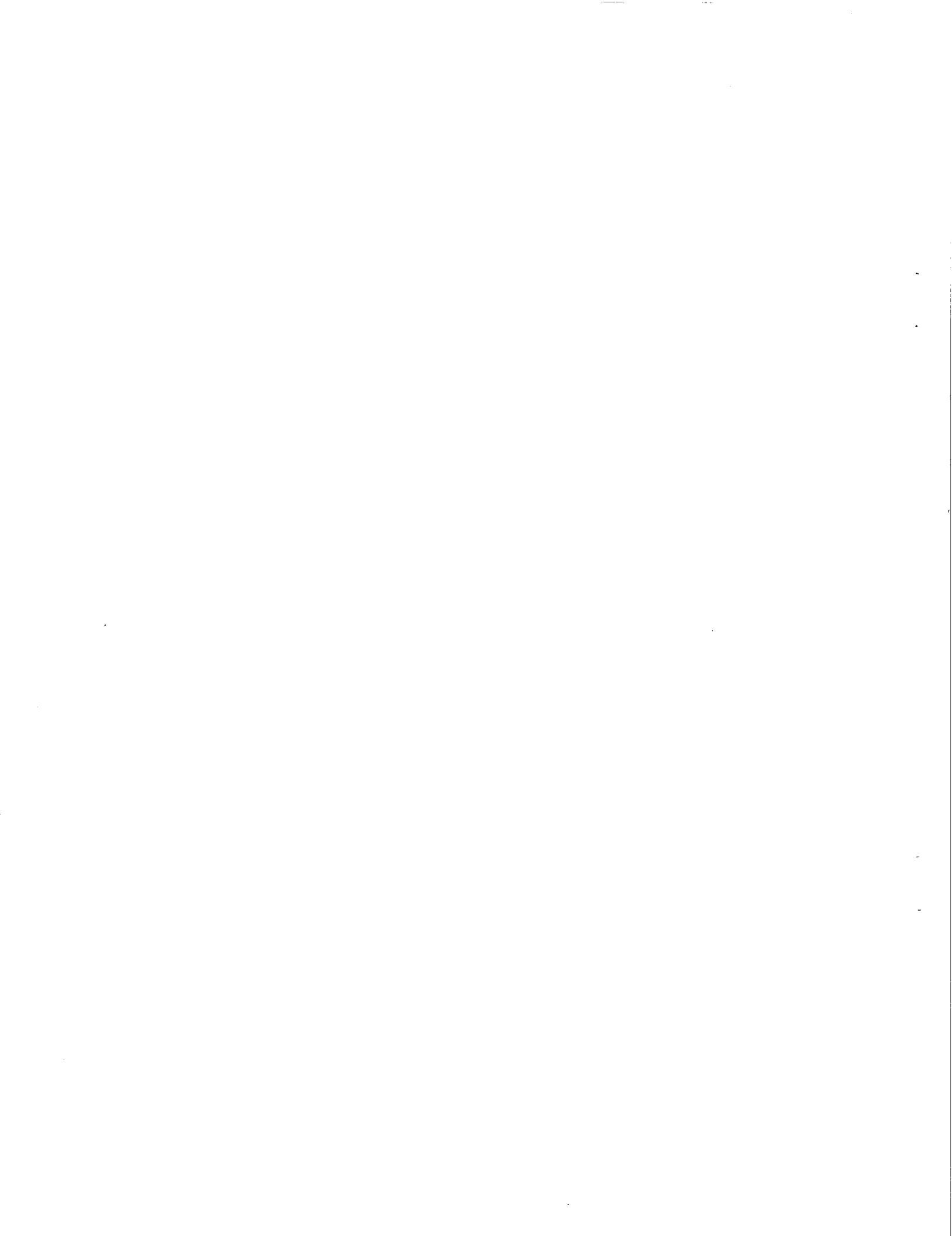
THEORY AND APPLICATIONS UNLIMITED
LOS GATOS, CALIFORNIA

PREPARED FOR
AMES RESEARCH CENTER
UNDER CONTRACT NAS 2-12378
OCTOBER 1987



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035



PREFACE

Enclosed in this final report are three separate Task reports: "Task I: Kalman Filter Flight Software;" "Task II: Landing Glidpath Steering Guidance," and "Task III: Kalman Filter Flight Software Data Analysis." Each Task report is self-contained, with its own introduction and summary sections.

1.0 KALMAN FILTER FLIGHT SOFTWARE

1.1 INTRODUCTION

Under Task I of its current contract to NASA-Ames, TAU Corporation has designed, coded, and tested a Kalman filter to optimally make use of GPS (in a differential mode) accelerometer data, and barometric and radar altimeter measurements. The filter will run in real time onboard the SH-3G helicopter, residing in a PDP-11/34M computer, and process raw pseudo and deltarange measurements generated by the Magnavox Z-set. The purpose of the helicopter flight tests is to investigate the extent to which differential GPS (perhaps inertially or otherwise aided) can provide the navigation accuracy required for Category I landings. The current FAA Navigation System Accuracy Standards are severe, requiring ± 3 meters 2 sigma in vertical position. In order to facilitate accuracy comparisons, the filter can be configured in the following modes: unaided (i.e., GPS only), GPS-aided by a vertical accelerometer, GPS-aided by the accelerometer and a barometric altimeter, and GPS-aided by the accelerometer and a radar altimeter.

In this report, the Kalman filter software interfaces are described in detail, followed by a description of the Kalman filter algorithm, including the basic propagation and measurement update equations. The performance of the filter, as exhibited by flight tests conducted to date, is then reviewed and discussed. Finally, recommendations for further data analysis activities and possible enhancements to the Kalman filter algorithm are made.

1.2 KALMAN FILTER SOFTWARE INTERFACES

The interfaces for the TAU Kalman filter software have been previously documented [1,2]. The relevant tables from [1,2] are repeated herein to make this report more nearly self-contained. Table 2-1, abstracted from [1], lists the Kalman filter-related variables required for post-flight analysis. Note that program variable names are provided where needed. Table 2-2, abstracted from [2], lists the Kalman filter outputs which are required for post flight data analysis.

Table 2-1 Kalman Filter Data Requirements

Data Item Source (Variable Name)	Definition	Units	Comments
TXBEG (B33DAT)	User Time	Seconds	"User Time" will be used by the TAU software to determine when data blocks are omitted.
PRNGE (B33DAT)	Pseudo Range "valid at beginning of last Delta Pseudo Range measurement interval"	Meters	The time of validity of the pseudo range measurement is "User Time" (above).
DRNGE (B33DAT)	Delta Pseudo Range Measurement (over the last 0.22 secs), divided by 0.22 secs	Meters/Sec	NASA/Sterling Software is required to convert the delta range measurement to a rate measurement.
DCODE (B33DAT)	Delta Pseudo Range Confidence Factor	0-2	TAU will use this information to adjust the Kalman filter's measurement variance.
RCODE (B33DAT)	Pseudo Range Confidence Factor	0-7	(same comment as for DCODE)
ALEPFG (B33DAT)	Satellite Position Computation Flag	True - almanac used False - ephemeris used	TAU may make use of this to adjust the filter measurement noise variance and to compute a bias range error associated with the use of the almanac from a particular satellite.

(MORE)

Table 2-1 Kalman Filter Data Requirements (Continued)

Data Item Source (Variable Name)	Definition	Units	Comments
SE1(3) (B33DAT)	Satellite Runway Coordinate (RCS) position (at time of reception) at beginning of delta pseudo range measurement interval	Meters	NASA/Sterling Software is required to convert the satellite positions from ECEF coordinates at the time of transmission.
	Satellite RCS position rates (at time of reception) at beginning of delta pseudo range measurement interval	Meters	NASA/Sterling Software is required to convert the satellite velocities from ECEF coordinates at the time of transmission.
NA	Average of the vertical accelerometer readings since the last data block transfer (<u>nominally four sets</u>)	Meters/Sec ²	NASA/Sterling Software has averaged the accelerometer outputs after performing the scale factor and bias compensations; gravity compensation may be performed by either TAU or NASA/S.S. (TBD).
NA	Baro Altimeter indicated altitude	Meters	NASA/Sterling Software has performed any/all necessary compensations/calibrations.
NA	Radar Altimeter indicated	Meters	(same comment as above)
NA	Radar Altimeter tracking flag	0: not tracking 1: radar tracking	

Table 2-2. Kalman Filter Data Analysis Requirements

<u>Item</u>	<u>Dim</u>	<u>Prec</u>	<u>Program Name</u>	<u>Comments</u>
GPS Time	1	SP	-	Variable already recorded by NASA
Kalman Filter State Vector	9	SP	-	Variable already recorded by NASA
Baro-Altimeter Indicated Alt	1	SP	-	Variable already recorded by NASA
Radar-Altimeter Tracking Flag	1	SP	-	Variable already recorded by NASA
Radar-Altimeter Indicated Alt	1	SP	-	Variable already recorded by NASA
Satellite ID	1	SP	-	Variable already recorded by NASA
PR Meas Residual	1	SP	PRMESR	
PR Meas Residual Variance	1	SP	PRMESV	
ΔPR Meas Residual	1	SP	DRMESR	
ΔPR Meas Residual Var	1	SP	DRMESV	
Baro Altitude Meas Residual	1	SP	BAMESR	
Baro Altitude Meas Res Var	1	SP	BAMESV	
Radar Altitude Meas Residual	1	SP	BAMESR	Note that the radar and baro altimeter measurement residuals and residual variances have the same program name; this is allowable since the measurements will never be processed simultaneously.
Radar Altitude Meas Res Var	1	SP	BAMESV	
Square Root Covariance Matrix	45	SP	UD(45)	Upper triangular elements only
Process Noise Covariance Matrix	9	SP	Q	Only of interest if the Kalman filter is adaptive; diagonal elements only. Already recorded by NASA
Accelerometer Incremental	3	SP	-	Increments over the last 1.2 secs

ORIGINAL PAGE IS
OF POOR QUALITY

1.3 KALMAN FILTER ALGORITHM DESCRIPTION

1.3.1 MODES OF OPERATION

The Kalman filter was designed with an eye toward using low-cost instrumentation to improve the vertical axis error inherent in the GPS system. Two instruments were studied: a barometric altimeter and a single, vertically mounted accelerometer. It was felt that these instruments would supply the maximum worth in vertical axis accuracy enhancement. The filter was designed to run in a matrix of modes: unaided; aided by baro-altimeter; aided by vertical accelerometer; and aided by both altimeter and accelerometer. A short description of operation of the aiding sensors follows.

1.3.1.1 Baro-Altimeter

The baro-altimeter is subject to a slowly varying, time-correlated bias error due to barometric pressure as well as a more complicated bias error due to aerodynamics during turns. The latter error source was too difficult to model within the computer data and code memory constraints and consequently neglected. The first error source is estimated in the filter and used to compensate the baro-altimeter outputs.

It should be noted that the filter calibrates the baro-altimeter to "GPS coordinates." Simply put, the filter estimates the difference between the GPS altitude and the baro-altimeter altitude, then subtracts the difference from the baro-altimeter altitude. Consequently, if the GPS altitude is in error, the baro-altimeter is calibrated to agree with the GPS error. The baro-altimeter is used principally because it has much better response than the Z-set in determining altitude under dynamic conditions; the altimeter produces a relatively noise-free altitude at better than 1 Hz, in contrast to the relatively noisy altitude update from the Z-set at 4.8 second intervals. The altimeter acts as a smoothing influence on the GPS position, much as a good vertical velocity measurement would.

To prevent the second error source (due to aerodynamics during turns) from contaminating the altimeter bias estimation, the filter would not estimate baro bias until the landing approach was initiated. The trajectory was then assumed to be fairly straight with little probability of long turns with the associated baro-altimeter dynamics error. The remaining baro bias was expected to be fairly constant during the twenty or thirty seconds of landing approach, long enough for the filter to estimate baro bias and calibrate the altimeter.

It was noted during the experiment that the barometric altimeter, once calibrated at the airfield for the day's flights, was actually more accurate than non-differential GPS. In practical situations, however, the baro-altimeter will not be calibrated at the landing site and the GPS receiver will be run in differential mode. Thus, the altimeter will be effectively calibrated in real-time to differential GPS altitude, which is very accurate for the landing approach scenario.

1.3.1.2 Vertical Accelerometer

A vertical accelerometer was also explored as a means of vertical accuracy enhancement. During landing approach the pitch and roll attitude of the helicopter is fairly stable and near level. By mounting the accelerometer near the GPS antenna and inclining it slightly to compensate for nominal pitch angle during descent, the very precise acceleration measurements can be used to smooth the GPS vertical axis. Addition of horizontal accelerometers was considered and rejected in the early phases of the experiment because of the resulting requirement for heading information, creating a need to tie into an all-axis attitude system. The concept of all-axis instrumentation is very sound and must be examined in terms of implementation cost; a single vertical accelerometer at the antenna was judged to be innovative, low-cost, and practical, so this was the main avenue pursued.

Attitude excursions from the vertical during landing approach were expected to be minor. To illustrate the effect of vertical excursion on a vertically mounted accelerometer, a coordinated turn at a bank angle of 0.1 radians (5.7 degrees) induces an acceleration on the vertical accelerometer of 1.005 g's rather than the true acceleration of 1.000 g. This error of roughly five cm/sec/s sec downward causes the navigation filter to have a small downward velocity bias. This bias was deemed acceptable and was to be compensated by adding extra process noise into the accelerometer vertical channel. The extra process noise allows the velocity estimate to recover quickly from the resulting acceleration mismodeling error through the GPS range-rate measurements.

Initial information indicated that the accelerometer could be calibrated at the beginning of each flight. The resulting error was found to be too large in flight test, however, so an accelerometer calibration state was proposed. The filter would be able to estimate the accelerometer bias during flight. As with the altimeter bias, the estimation was limited to the landing approach, because high bank angles in the turns leading up to the approach would cause major vertical excursion errors as outlined above and invalidate the accelerometer bias estimate.

1.3.2 FILTER PROCESSING EQUATIONS

Design of the filter was influenced by the severely restrictive code and data memory constraints imposed by the real-time hardware. Much of these constraints were removed by definition of a leaner interface between Z-set and computer, but this occurred after the filter design had been frozen. Shortcuts due to the memory constraints will be noted as they occur; these areas have potential for improvement in light of the looser constraints.

Because memory was more constrained than throughput, it was decided early on to utilize the U-D filter software implementation developed by Bierman [3]. This not only allows use of single precision without numerical instability in the gain computations but also is designed for minimal data storage. A slight loss of throughput is experienced; tests on similar filters at TAU indicate the Bierman filter formulation runs roughly 15-20% slower than a simple Kalman implementation if both are in single precision.

The filter state comprises position and receiver clock phase error, velocity and receiver clock frequency error, and altimeter and/or accelerometer bias. Table 3.1 defines the variables used in the following two sections. Table 3.2 [4] illustrates the operation of the filter; although the Bierman algorithm is different in equation form, it performs equivalent operation. The filter is a two-stage process; the state and covariance is propagated from measurement to measurement through time, and the measurement is incorporated into the estimate and the covariance is updated. A filter is well-defined by the propagation and measurement models it uses. These models are described below with some rationale as to their development.

1.3.2.1 Filter Propagation Model

The filter state in the software is arranged as $x, v_x, y, v_y, z, v_z, \phi, f$, and/or ninth and tenth states for the two biases. For convenience the elements of the F, H, Q matrices will be indexed by the state variable rather than the numerical index; for instance, since x is the first state and z is the fifth state $F(1,5)$ will be denoted $F(x,z)$. The propagation model for the states is:

$$\begin{aligned} \dot{r}_x &= v_x \\ \dot{v}_x &= w_x \\ \dot{r}_y &= v_y \\ \dot{v}_y &= w_y \\ \dot{r}_z &= v_z \\ \dot{v}_z &= w_z, \text{ or } = z - b_z + w_z \text{ if accelerometer inputs are available} \\ \dot{\phi} &= f \\ \dot{f} &= w_\phi \\ \dot{b}_h &= w_h \\ \dot{b}_z &= 0 \end{aligned}$$

where $w_v = (w_x, w_y, w_z)$, w_ϕ , and w_h are independent zero-mean Gaussian random variables of spectral density q_x, q_y, q_z, q_ϕ , and q_h , respectively.

The corresponding dynamics matrix $F = (\delta \dot{x} / \delta x)$ is all zeros except for the (x, v_x) , (y, v_y) , (z, v_z) , and (ϕ, v_ϕ) elements, which are all 1.0. If accelerometer bias is also included in the state, the (v_z, b_z) element is equal to -1.0, corresponding to the effect of accelerometer error on vertical acceleration.

As can be seen from the differential equations, the propagation is very linear in nature. For this reason a linearized filter was implemented.

The state is propagated linearly, and the error covariance also. Covariance propagation is performed by

$$P_k = \Phi_k P_{k-1} \Phi_k^t + Q_k \quad \text{where}$$

$$\Phi(\tau) = \text{ex}(F\tau) = I + F\tau + F^2\tau^2/2 \quad \text{and}$$

$$\Phi_k = \Phi(t_k - t_{k-1}) = \Phi(\Delta t); \quad \text{and process noise covariance is described by}$$

$$\Phi_k = \int_{\tau=0}^{\Delta t} \Phi(\tau) Q \Phi^t(\tau) d\tau$$

$$= Q\Delta t + (FQ + QF^t)(\Delta t^2/2) + (F^2Q + 2FQF^t + QF^2t)(\Delta t^3/6) + \dots$$

In the implemented software, an approximation is used for Q_k :

$$Q_k = (I + F\Delta t/2) Q (I + F^t\Delta t/2)\Delta t$$

This approximation underestimates the process noise contribution and causes the filter to lag a little more than an optimal filter would. Also, if the filter runs for extended periods of time, divergence may occur. Both consequences were considered negligible for the relatively short times involved in helicopter landing approach. The purpose of the approximation is to reduce the rank of the matrix Q_k from eight to four in the unaided filter. Process noise injection in the Bierman algorithm is done by a rank-one algorithm that must be invoked as many times as the Q_k matrix has rank, so the approximate Q_k allows roughly half the computation. Since the process noise injection is relatively computationally expensive and the software was originally very throughput-constrained, the filter was designed toward this compromise. In light of the work done by Sterling Software and TAU Corporation in reducing throughput and memory problems, it is recommended that the filter be reconfigured slightly to invoke the full process noise matrix. Although the performance during landing will probably not improve significantly, the filter created will have more legacy in non-approach navigation.

If an accelerometer is present, the vertical velocity is propagated from time point to time point by integrating the accelerometer output, instead of the a priori assumption that the vertical velocity is roughly constant. Having an actual measurement rather than a seat of the pants guess gives much higher confidence in the velocity propagation, of course, so the process noise uncertainty injected into the vertical channel is typically much lower - decreasing from values on the order of meters per second to values on the order of centimeters per second. This allows greater averaging over time in the vertical channel, providing a much less noisy error signal coming out of the vertical channel.

To recap the propagation process:

1. The new measurement time tag is received and Δt is computed;
2. The state estimate is propagated to the new time, integrating accelerometer outputs if any;
3. The error covariance is propagated to the new time;
4. Process noise is added into the forward, right, down (x, y, z) channels and the clock phase error channel independently; also, if an altimeter is present, the baro-altimeter channel as well.

1.3.2.2 Filter Measurement Model

The measurement process computes a measurement update to the state, Δx . In the Bierman structure, Δx is stored explicitly in the UD structure in the last elements, numbered $(N(N+1)/2 + 1)$ to $(N(N+3)/2)$. A representation of the covariance is stored in the first $N(N+1)/2$ elements of the UD structure, where N is the number of states.

Each call to the Bierman subroutine UMEAS incorporates the information in a single scalar measurement into the state update and updates the covariance to reflect the additional information. Each scalar measurement must be uncorrelated to the other scalar measurements as well as the state estimate. The measurement update equations require as input:

the pre-measurement covariance, P ;
the measurement noise covariance, R ;
the measurement gradient vector, H ;
the measurement, ΔZ ;
the state estimate, Δx ;

and output:

the post-measurement covariance, P ;
the updated state estimate, Δx .

In the NASA implementation, the measurement update is linearized about the post-propagation (pre-measurement) state "whole" estimate, as is done in an extended Kalman filter. At the beginning of a measurement cycle (defined as a collection of two or three simultaneous measurements) the state "update" estimate in the filter, Δx , is reset to zero at the start, and the measurements are linearized by subtracting the nominal measurements which are calculated as a function of the whole state estimate. At the end of the measurement process, the state update estimate is added to the state whole estimate in preparation for propagation to the next measurement time.

The Bierman UMEAS subroutine interface contains the following arguments:

1. The UD vector structure, containing covariance and state update, is updated.
2. The H vector is input. The nominalized measurement $\Delta Z = Z - h(x)$ is stored in the last element of H .
3. The measurement noise variance R is input.
4. Two scratch vectors are required. Bierman routines have no internal vector storage.
5. The pre-measurement residual variance is output. $(HPH^t + R)$
6. A measurement editing threshold can be input. This edits a measurement whose pre-measurement residual is greater than the square root of the product of the threshold and the pre-measurement residual variance.

The NASA filter uses three different measurement types: pseudorange, deltarange, and baro-altitude. Each requires a separate call to UMEAS according to the models described below. Accelerometer outputs are considered as part of the process or dynamical system and treated deterministically in the propagation model.

1.3.2.2.1 Pseudorange

The GPS signal is a code impressed on a carrier signal. The code is repeated every millisecond, and is about 300 km in wavelength. GPS receivers measure code phase with a noise standard deviation of about $5 \text{ m} \cdot \sqrt{\text{sec}}$ or better with various types of correlators. In the NASA Z-set, code phase of one satellite is measured every 1.2 seconds, with a full cycle four satellites requiring 4.8 seconds. Code phase measurement error is very uncorrelated (white noise) in the Z-set.

Pseudorange is the basic code phase measurement, and is usually defined as the sum of the transit time between GPS receiver and the relative clock error between the two. Some definitions account for the various compensatable propagation errors; the Z-set, for instance,

subtracts a simply modeled estimate of tropospheric delay. For filter purposes, however, the range and the clock phase error are the only quantities observable through the filter, so they are included in the state.

Defining range $s_{SV} = | \underline{r} - \underline{r}_{SV} |$

$$PR = h(\underline{x}) = s_{SV} + \phi + \text{noise.}$$

The measurement gradient vector is

$$\underline{H} = ((x-x_{SV})/s_{SV}, (y-y_{SV})/s_{SV}, (z-z_{SV})/s_{SV}, 1, 0,0,0,0,0,0).$$

The measurement error variance R is roughly $(5m)^2$.

1.3.2.2.2 Deltarange

The **deltarange** is the range-rate, as measured from the carrier, integrated over a short interval of time. Carrier tracking loops are much more accurate than code tracking loops, with accuracies mostly in the centimeter per second range; the Z-set **deltarange** seems to be quantized at about a foot per second accuracy. In the NASA implementation **deltarange** is converted to range-rate by dividing by the appropriate time interval. The measurement sequence of the Z-set is to measure the code phase for 980 milliseconds, then turn off the code phase lock loop and track carrier phase for roughly 220 milliseconds. Thus the range-rate measurement is not coincidental with the range measurement but rather an average range rate over a short period of time lagging the range lags by a fraction of a second. For filter modeling purposes the two measurements are assumed to be instantaneous and simultaneous rather than average and lagged, a good assumption in terms of the accuracy attainable from the Z-set.

The measurement is described by

$$DR = h(\underline{x}) = (v-v_{SV})^t(\underline{r}-\underline{r}_{SV})/s_{SV} + f + \text{noise.}$$

and the measurement gradient vector is

$$\underline{H} = (*,*,*,0, (x-x_{SV})/s_{SV}, (y-y_{SV})/s_{SV}, (z-z_{SV})/s_{SV}, 1, 0,0).$$

where the terms denoted by * are very small and negligible in normal navigation (they are necessary only in high precision survey applications).

The measurement noise variance R is roughly $(2m/\text{sec})^2$

1.3.2.2.3 Baro-altitude

The last measurement, an optional one, is baro-altitude. The measurement is simply

$$A = h(\underline{x}) = z + b_h + \text{noise}$$

and the measurement gradient vector is

$$\underline{H} = (0,0,1,0,0,0,0,0,1,0).$$

The measurement noise variance R is less than $(1m)^2$.

The characteristics of the baro-altimeter bias have been noted in the previous sections. The filter is intended to start with little or no a priori knowledge of the baro bias and create an estimate of the bias by averaging the difference between the GPS derived altitude and the baro-altitude over a long history of readings. It should be noted that the baro-altimeter will be calibrated to the GPS altitude whether or not the GPS altitude has large, small, or no error. As a consequence, the baro-altimeter is no help to accuracy when the GPS error is large, such as is the case when there are no differential corrections.

Table 3-1. Filter Variable Definition

$\underline{x} = (\underline{r}, \phi, \underline{v}, f, b_h, b_z)$			
\underline{r}	=	position, RCS (Forward-right-down) frame	meters
ϕ	=	receiver clock phase error	meters
\underline{v}	=	velocity, RCS	m/sec
f	=	receiver clock frequency error	m/sec
b_h	=	altimeter bias	meters
b_z	=	accelerometer bias	m/s/s
PR_i	=	pseudorange from i^{th} satellite	meters
DR_i	=	deltarange from i^{th} satellite	m/sec
h_i	=	barometric altitude	meters
z	=	vertical accelerometer output	m/s/s
\underline{r}_{sv}	=	satellite position, RCS	meters
\underline{v}_{sv}	=	satellite velocity, RCS	m/sec
Δt	=	time since last measurement	sec

Table 3-2. Summary of Continuous-Discrete

Linearized Kalman Filter

System Model	$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), t) + \underline{w}(t) ; \quad \underline{w}(t) \sim N(0, Q(t))$
Measurement Model	$\underline{z}_k = \underline{h}_k(\underline{x}(t_k)) + \underline{v}_k ; \quad k = 1, 2, \dots ; \quad \underline{v}_k \sim N(0, R_k)$
Initial Conditions	$\underline{x}(0) \sim N(\hat{\underline{x}}_0, P_0)$
Other Assumptions	$E[\underline{w}(t) \underline{v}_k^T] = 0$ for all k and all t Nominal trajectory $\bar{\underline{x}}(t)$ is available
State Estimate Propagation	$\dot{\hat{\underline{x}}}(t) = \underline{f}(\bar{\underline{x}}(t), t) + F(\bar{\underline{x}}(t), t)[\hat{\underline{x}}(t) - \bar{\underline{x}}(t)]$
Error Covariance Propagation	$\dot{P}(t) = F(\bar{\underline{x}}(t), t)P(t) + P(t)F^T(\bar{\underline{x}}(t), t) + Q(t)$
State Estimate Update	$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k[\underline{z}_k - \underline{h}_k(\bar{\underline{x}}(t_k)) - H_k(\bar{\underline{x}}(t_k))[\hat{\underline{x}}_k(-) - \bar{\underline{x}}(t_k)]]$
Error Covariance Update	$P_k(+) = [I - K_k H_k(\bar{\underline{x}}(t_k))] P_k(-)$
Gain Matrix	$K_k = P_k(-) H_k^T(\bar{\underline{x}}(t_k)) [H_k(\bar{\underline{x}}(t_k)) P_k(-) H_k^T(\bar{\underline{x}}(t_k)) + R_k]^{-1}$
Definitions	$F(\bar{\underline{x}}(t), t) = \left. \frac{\partial \underline{f}(\underline{x}(t), t)}{\partial \underline{x}(t)} \right _{\underline{x}(t) = \bar{\underline{x}}(t)}$ $H_k(\bar{\underline{x}}(t_k)) = \left. \frac{\partial \underline{h}_k(\underline{x}(t_k))}{\partial \underline{x}(t_k)} \right _{\underline{x}(t_k) = \bar{\underline{x}}(t_k)}$

1.4 SUMMARY

In the absence of aiding sensors, the filter developed for this project was substantially similar in structure to the filter internal to the Z-set. The demonstrated improvements in performance were probably due to the fact that the Z-set filter is self-tuning, which always results in compromised performance when compared against a filter tuned for the specific dynamical system.

The results shown here represent the status of filter development at the time of the last flight test in October 1986. At that flight test the basic eight state filter was still under test in the real-time environment. Among the aiding sensors, the accelerometers were never fully integrated into the filter solution because the residual bias was much more severe than originally thought; the baro-altimeter was not exercised because the basic filter was only being tested on the last flight; and the differential GPS feature was never fully exercised because of lack of confidence in the GPS reference station.

1.4.1 EXPECTED PERFORMANCE

Theoretically, addition of accelerometer and altimeter either singly or together should aid the filter considerably, and, in fact, the data collected during the last series of flight tests looks extremely promising. These instruments are low-noise, quick response sensors to complement the noisy, somewhat slower response of the GPS receiver. The result is a filter solution that responds more quickly to real dynamics instead of mistaking the dynamics effects as measurement noise. Differential GPS, of course, is necessary for the project to succeed, since standalone GPS is not sufficiently accurate to perform landing approaches.

Data collected during the experiment and processed in-house at NASA has shown that the altimeter and accelerometer to show excellent measurement characteristics over short periods of time, such as in landing approach. Experiments in differential GPS at TAU show a potential of sub-meter static accuracy in the static situations. The combination of instruments should produce accuracies on the order of two to three meters (standard deviation) accuracy in the vertical axis.

1.4.2 RECOMMENDATIONS FOR FURTHER STUDY

Although not currently in the software, accelerometer bias should be added to the filter. This report contains the guidelines in expanding the filter propagation to include the new bias state. Because the real-time software is no longer throughput- or memory-constrained, the expansion of the Q matrix to full rank would also be advised. These two additions would improve the performance of the filter marginally and allow it to be used in normal flight as well as during the landing approach.

Expansion of the system to include other sensors should also be examined. This study should concentrate on sensors that are similar to those found on helicopters, such as level sensors, compasses, and inexpensive accelerometers. The helicopter tested at NASA already contains such sensors, so that the expansion of the system requires only extended software and some hardware data interfaces.

2.0 LANDING GLIDEPATH STEERING GUIDANCE

2.1 INTRODUCTION

This task represents a fairly modest effort to refine the lateral and vertical steering algorithms developed under the previous contract. Specifically, the following refinements have been made to the steering algorithms, and their effectiveness either flight tested or examined by simulation:

- Modification of the waypoint switching logic to enable handling of more diverse flight test scenarios.
- Further data smoothing, including a second order turn propagation.
- Compensation of measured delays in the Control Display Indicator (CDI).

The following section of this report reviews each of these refinements, discussing their implementation, simulation based testing, and flight testing (if applicable). In the final section, the conclusions which can be drawn from the experiences of flight testing and simulation are summarized.

2.2 STEERING DISPLAY REFINEMENTS

2.2.1 WAYPOINT SWITCHING LOGIC

The revised waypoint switching logic can best be described by reference to Figure 1. The revision was motivated by a desire to improve the robustness of the steering algorithm, i.e., to enhance its performance under off-nominal conditions (e.g., in situations where navigation error might be excessive). Basically, five specific regions are established: four of these result in an error signal being computed based upon one of the four waypoints on the path; the fifth region results in the setting of an error flag. The default dimensions for the square regions are 0.25 mile on a side. Whenever the center region is entered, a message to return to the Glide Path Intercept Point (GPIP) is set.

Flight test experience with the revised switching logic has proven it to be satisfactory; no specific problems worth noting were encountered. Possible future enhancements which were not funded as part of this effort were the inclusion of additional tests based upon velocity and relating to the effects of solution error on the switch boundaries. Velocity-based testing would ensure that the helicopter was heading in the right direction, and inform the pilot if he was not. Such tests could be based upon a simple dot product of the estimated horizontal velocity vector with a preferred direction along each leg. Avoidance of an apparent backing up across the boundary of a region (induced by solution noise) could be accomplished by only permitting transitions to occur in single directions.

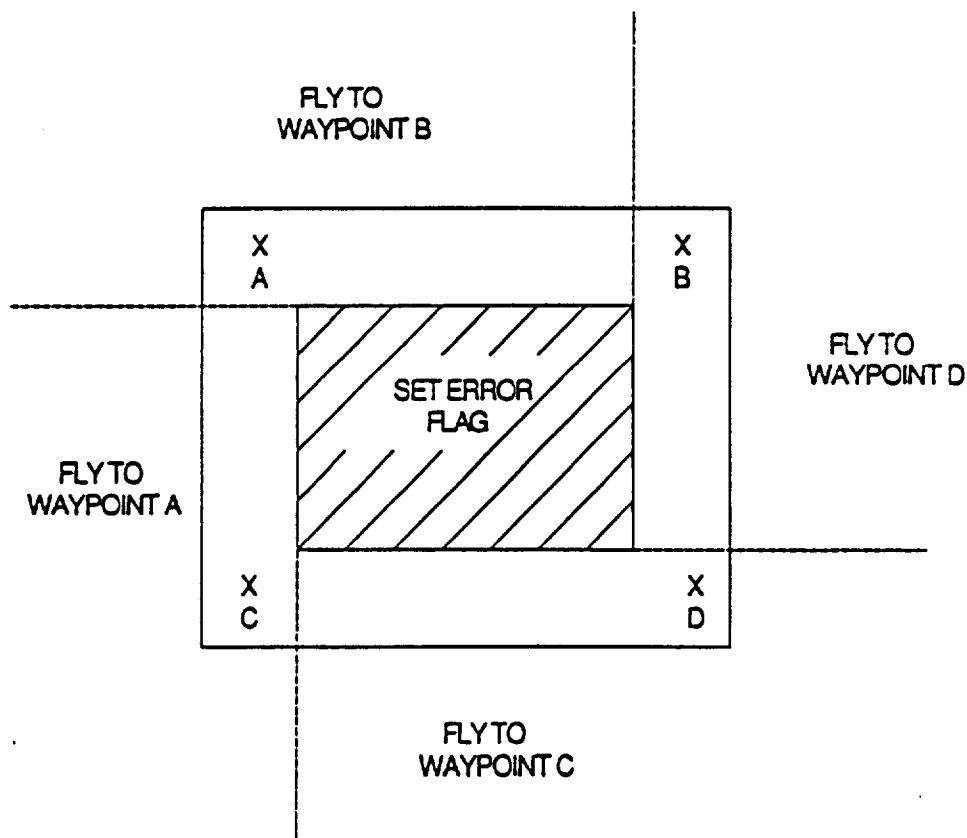


Figure 1. Illustration of Waypoint Switching Logic

2.2.2 DATA SMOOTHING

Additional data smoothing was considered as a means of updating the CDI more frequently than once every 1.2 seconds. Specifically, the effectiveness of a second order propagation model, which estimated acceleration by numerically differentiating the derived velocity estimates, was used to extrapolate the CDI inputs at 10 Hz. Figure 2 illustrates the approach.

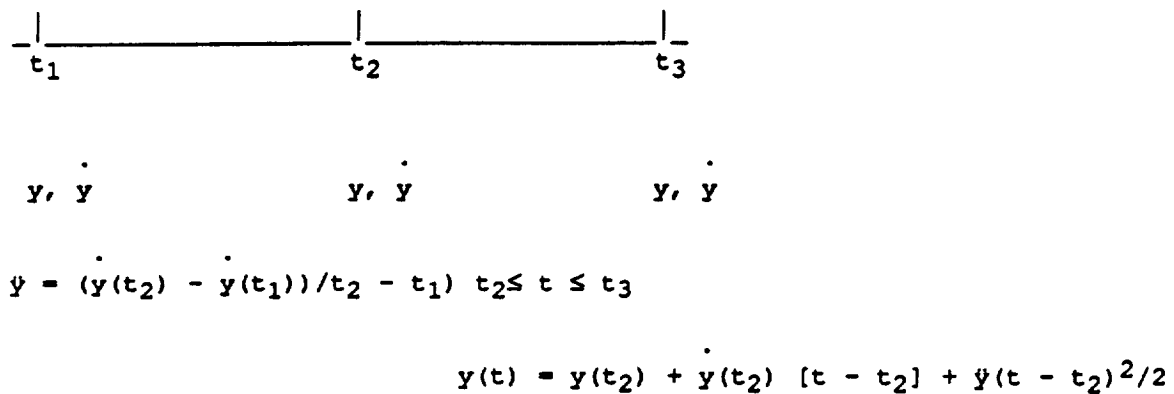


Figure 2. Second Order Steering Command Propagation

In the figure, y could represent either the lateral or vertical steering error which is sent to the CDI from the Kalman filter, and t_1 , t_2 , and t_3 represent the 1.2 second time epochs. Note the 1.2 second lag inherent in estimating acceleration (\ddot{y}); derivative information calculated over the interval (t_1, t_2) is used over (t_2, t_3) . The effectiveness of this approach was examined using TAU's landing simulation. Results are presented as Figures 3, 4, 5, and 6. The first two figures correspond to a "benign" landing profile, while the second two correspond to a "worst case" profile. The distinction between these cases lies in the amount of high frequency helicopter motion present. By comparison of Figures 3 and 4, it is demonstrated that some improvements in the smoothness of the steering display commands can be realized by inclusion of the second order propagation term. On the other hand, comparison of Figures 5 and 6 demonstrates that, in the "worst case" environment, incorporation of the second order propagation can degrade the smoothness of the commands. Based on these results, the second order terms were not included in the flight tests; in fact, the first order interpolation was not included, resulting in an update to the CDI display being computed only once every 1.2 seconds.

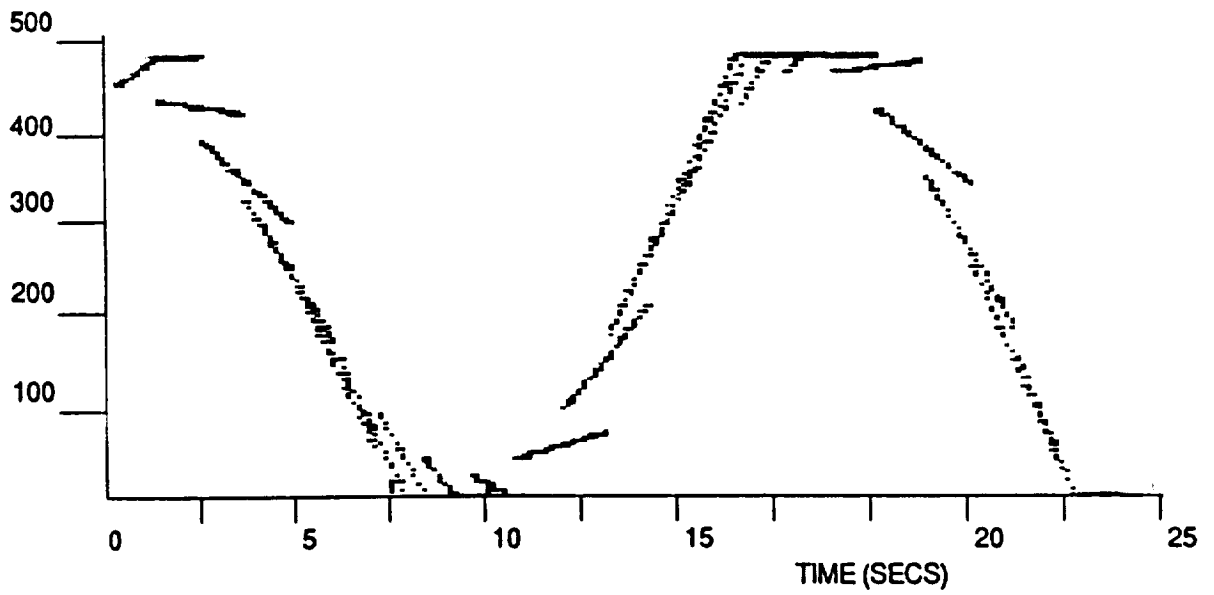
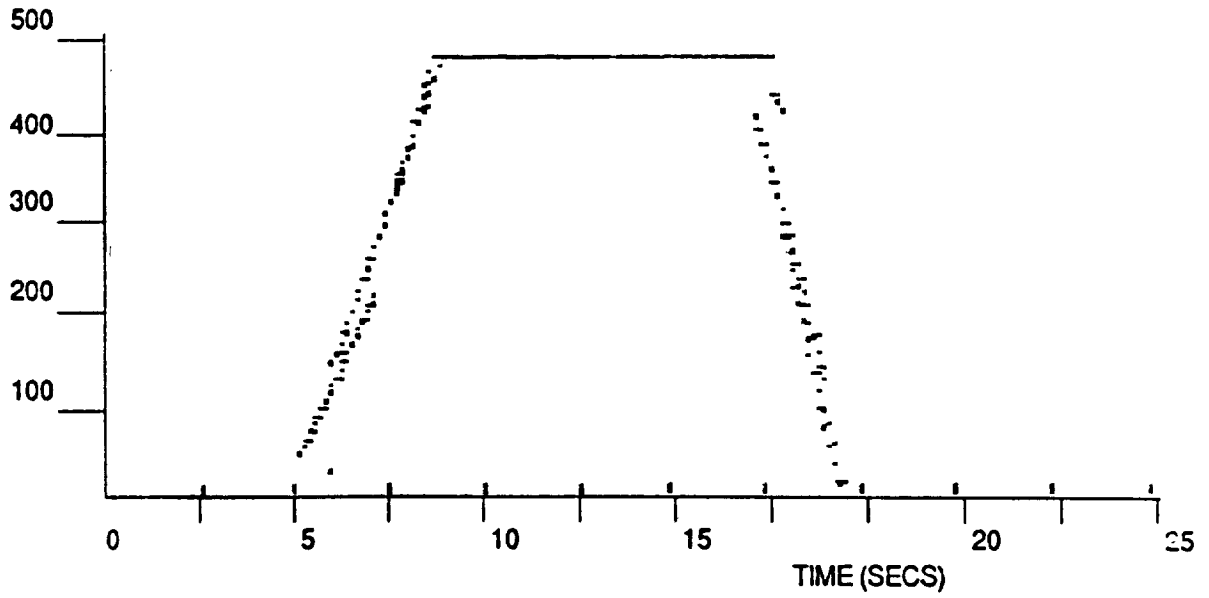


Figure 3. Computed Steering Commands for Benign Landing Profile Using First Order Extrapolation.

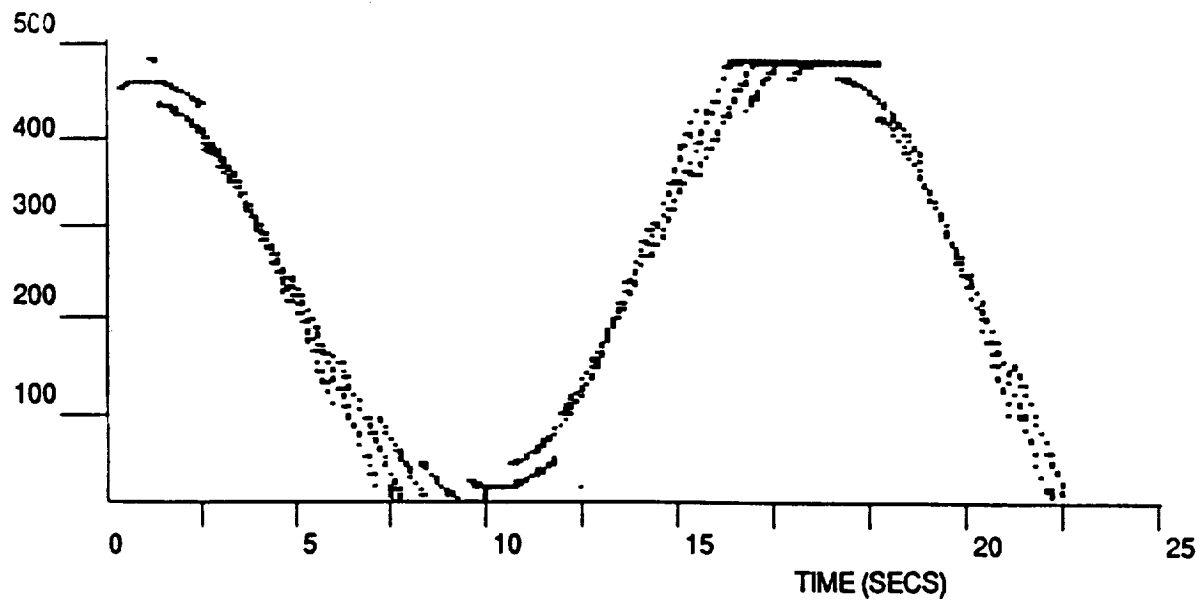
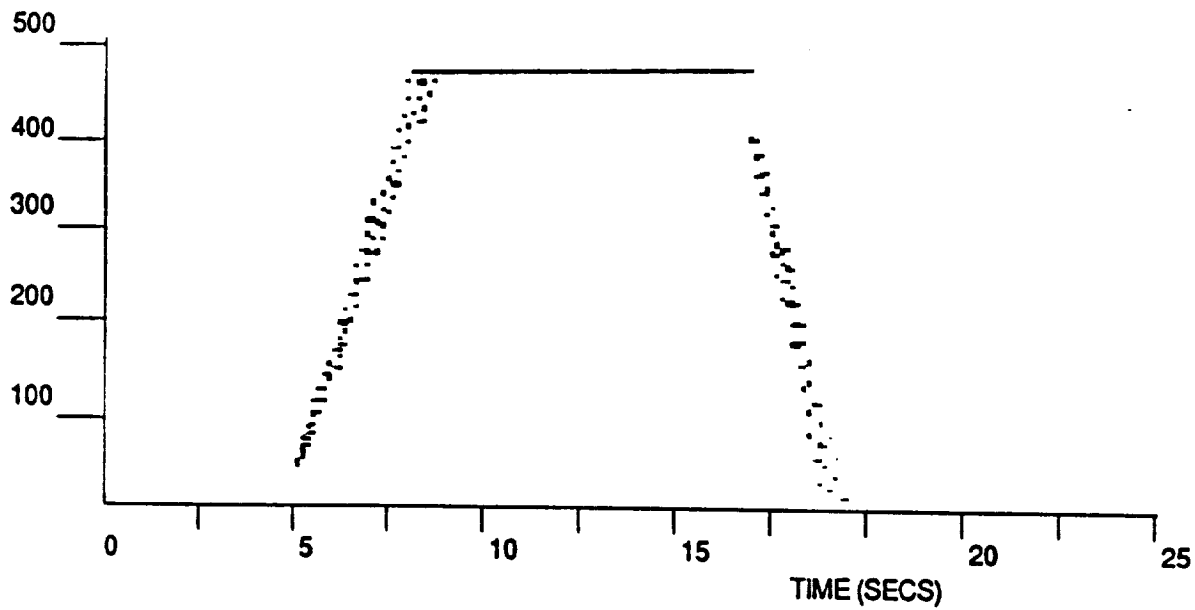


Figure 4. Computed Steering Commands for Benign Landing Profile Using Second Order Extrapolation.

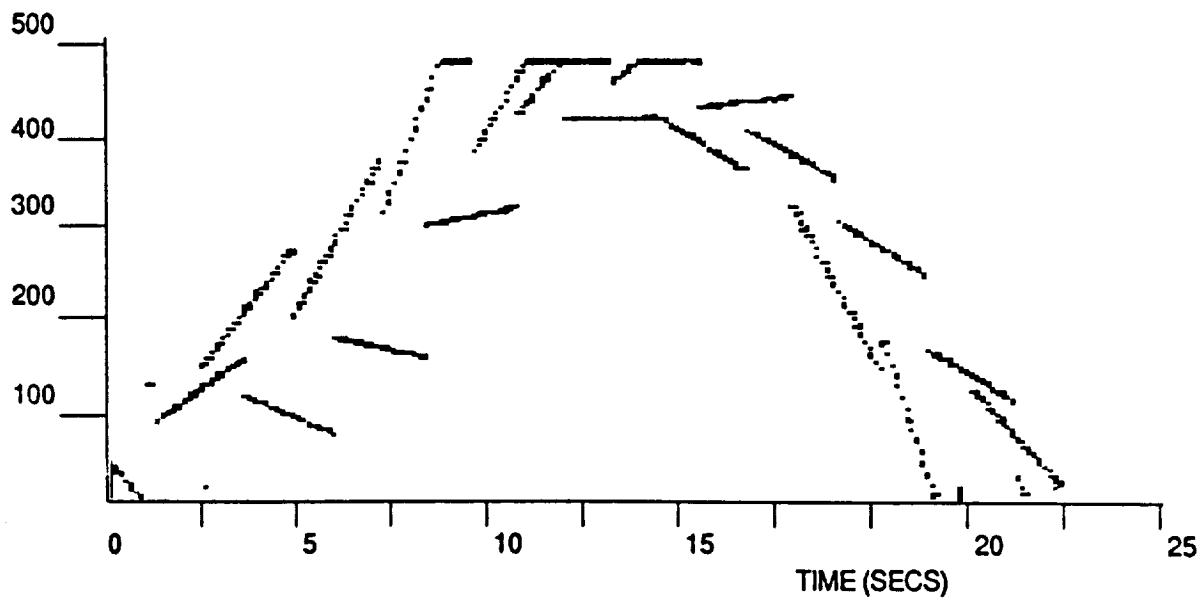
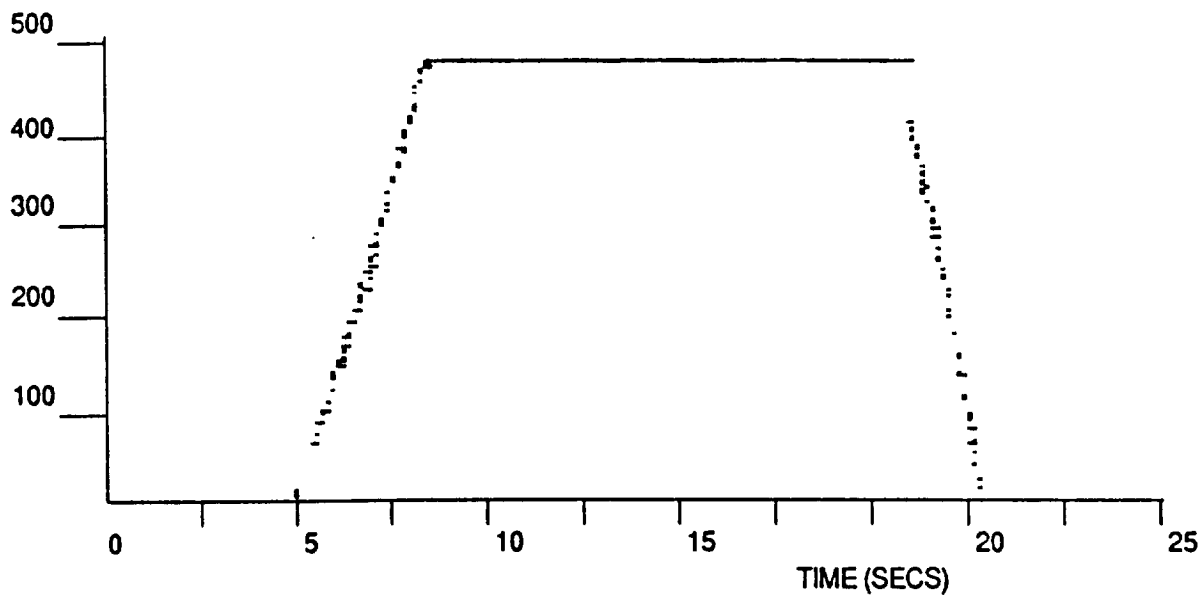


Figure 5. Computed Steering Commands for Worst Case Landing Profile Using First Order Extrapolation.

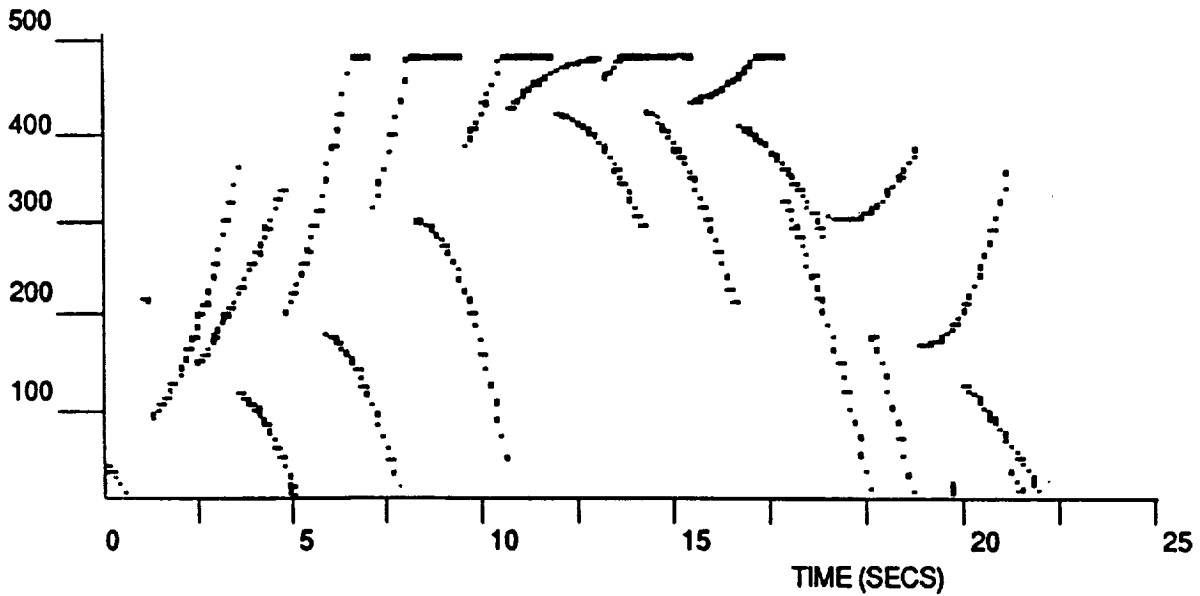
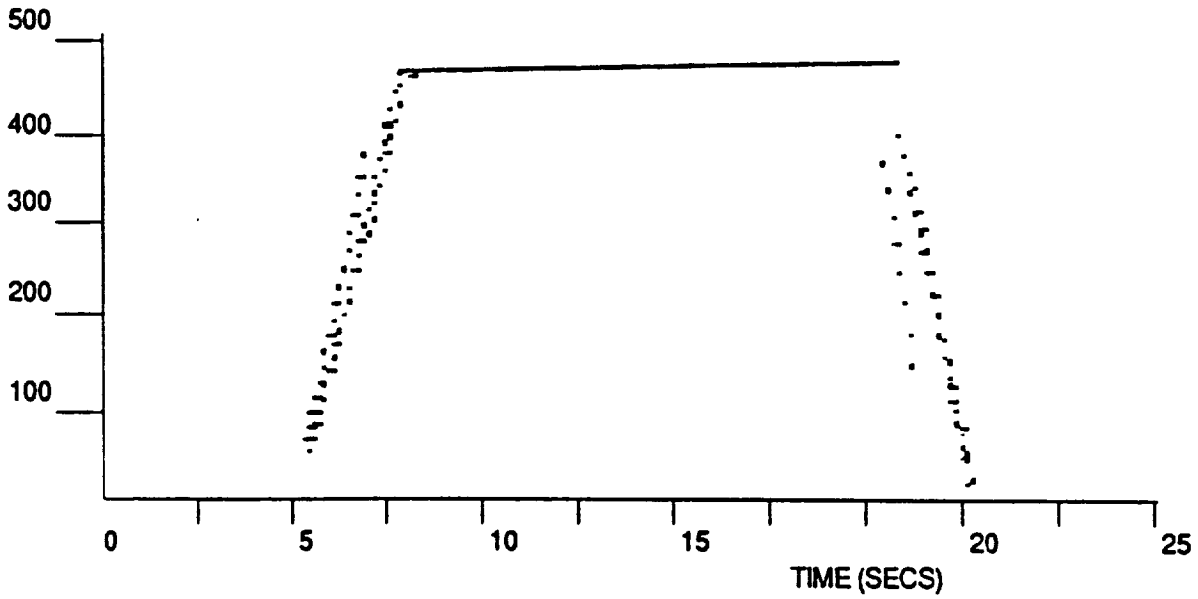


Figure 6. Computed Steering Commands for Worst Case Landing Profile Using Second Order Extrapolation.

2.2.3 COMPENSATION FOR MEASURED CDI DELAYS

Delays inherent in the CDI imply that the pilot is using out-of-date information to steer the helicopter. The following development describes a relatively simple approach to compensating for this delay by using derivative information to induce appropriate "lead" into the commands to the CDI. In deriving the compensation equations, it is assumed that the CDI dynamics can be adequately represented by a first order lag, i.e., its transfer function is simply:

$$H_{CDI}(s) = \frac{Y_o(s)}{Y_i(s)} = \frac{1}{1 + \tau s} \quad (1)$$

where y_1 denotes the CDI input, and y_o denotes its output neglecting the nonlinear scaling. Ideally, we would like y_o (in the absence of the display nonlinearities) to be identical to y_1 . This can be accomplished, at least in theory, by replacing y_1 by y_1^1 , a version of y_1 "advanced" in time:

$$Y_1^1(s) = (1 + \tau s) Y_i(s) \quad (2)$$

Obviously, if $y_1(s)$ is replaced by $Y_1^1(s)$ in Equation (1), a transfer function from $Y_1(s)$ to $Y_o(s)$ which is identically one results. Practically speaking, Equation (2) can be implemented using the rate (i.e., velocity) information computed by the filter:

$$y_1^1(t) = \hat{y} + \tau \dot{y} \quad (3)$$

In addition, if acceleration information is generated, the velocity information fed to the CDI (since the displayed value is a function of both position and velocity) can also likewise be "advanced":

$$y_1^1(t) = \hat{y} + \tau \dot{y} \quad (4)$$

Thus, y_1^1 and y_1^1 can be used in place of y and \dot{y} in generating the display, in an effort to compensate for the measured delay (time constant) of the CDI (τ).

The effectiveness of the compensation was examined using the landing simulation, using a simple first order lag model (per Equation (1)) for the CDI.

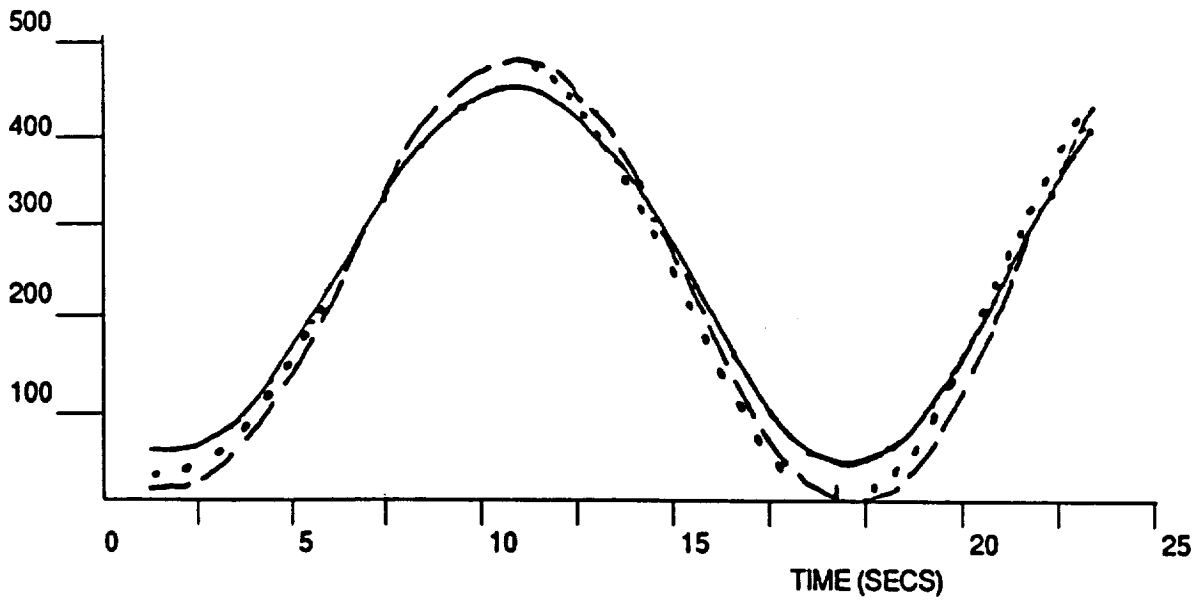
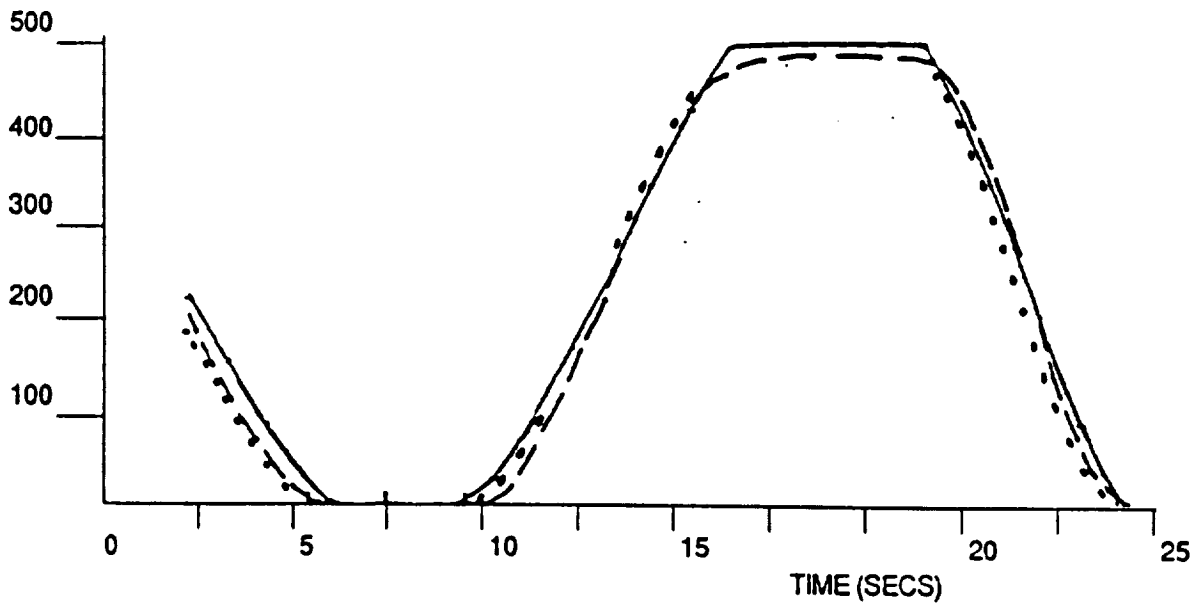


Figure 7. Steering Commands for "Perfect" CDI. _____
 Steering Commands for Position Compensated CDI. - - - - -
 Steering Commands for Position and Velocity Compensated CDI.

Simulation results are presented in Figure 7. The steering command plots correspond to an expanded time segment during a simulated landing approach with benign dynamics. The solid line steering commands appearing in Figure 7 correspond to a "perfect" CDI, i.e., one which instantaneously displays the commanded values. The steering commands plotted as dashed and dotted lines correspond to a simulated 1 second CDI time constant, with two versions of the compensation; in the first version (corresponding to the dashed line), only the position (y) is compensated, while in the second (corresponding to the dotted line), both position (y) and velocity (y) are compensated, the latter using the estimated acceleration. Both sets of results generally indicate that in the benign dynamic environment, the compensation works very well. Note that the one second time delay, corresponding roughly to one time tick, is barely discernible. The only distortion induced by either form of the compensation is a slight biasing of the vertical steering command when only position compensation is utilized. This biasing was not determined to be significant. Based on the previous results obtained using acceleration estimates in the worst case dynamic environment, the simpler position compensation was recommended for flight testing. The effective CDI time constant was measured by NASA to be 3 seconds and set as a default value in the software; however, any value (including zero, of course) could be set by the operator. Comparable simulation results can be derived for the 3 second delay but are not included here.

Flight testing experience with the compensation has not been encouraging and was generally inconclusive. This is due at least in part to the very large measured delay and to the general "noisiness" of the onboard solution. Of course, the compensation will only be effective if the onboard filter is tracking the helicopter's velocity well; any solution error will be scaled by the estimated CDI delay and used to drive the display.

3.0 SUMMARY AND CONCLUSIONS

As discussed in the previous sections, the refinements to the steering algorithm developed under this task have exhibited different levels of success in flight testing. The inclusion of a second order term in the command propagation was not incorporated for flight testing due to its poor performance in a "noisy" environment. The CDI compensation was exercised in several flight tests; however, results were generally inconclusive due primarily to navigation solution error. The most significant accomplishment was the modification of the waypoint switching logic, whose performance was successfully demonstrated, and is highlighted in the following paragraph, abstracted from the 8 August letter written by R. P. Denaro.

Comparison plots from the July 24 flight tests appear as Figures 8 and 9. The plots compare the steering display command (darkened line) with the sensed error, and indicate that the software is working as desired. Refer to the horizontal plot for the following explanation:

1. The curve marked "1" is the aircraft lateral position in the along-path frame.
2. The curve marked "2" is the command from the steering display.
3. You can usually expect the curves to be "mirror images." (We have corrected for the display bias to make these plots). That is, when the aircraft is left of course, the display says "turn right" (shown here).
4. The display will "peg" when the deviation exceeds a design value (max sealing).
5. Sudden shifts in position are due to path switching from leg to leg.
- 6&7. Note the small steering command, and even apparently wrong sign in the (7) case. This is because there is a rate term in the display. In other words, in (6), although the helicopter is far left of course, the "cut" or cross track rate is about right to smoothly recapture the path. In (7), the "cut" is too sharp, and the "fly left" command when still left of course is to avoid possible overshoot.
8. The same holds true here. Now we are left of course and increasing in rate. The command is a strong "turn right."

For the vertical plot, the very same set of observations can be made regarding the performance of the steering algorithm and so are not repeated here.

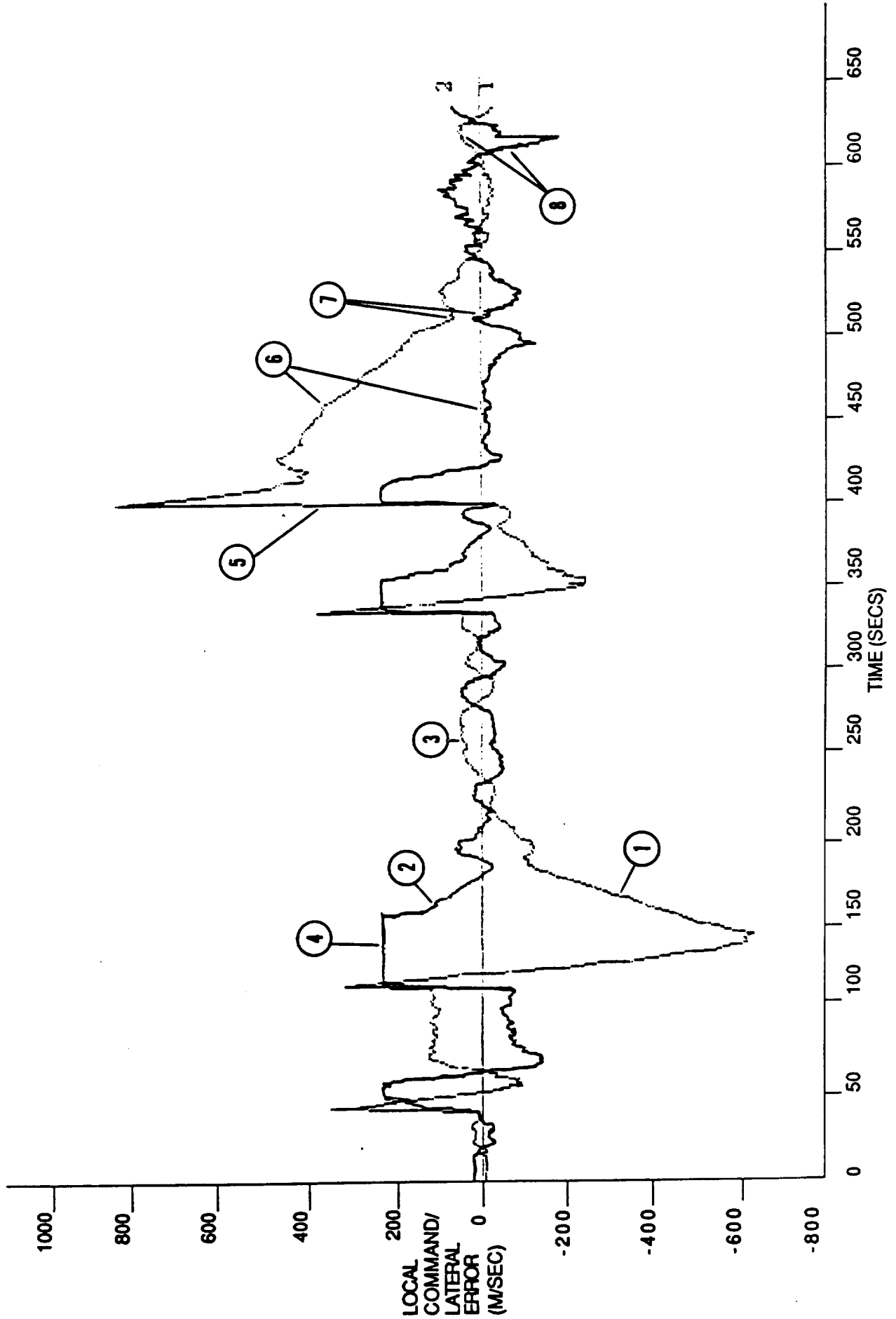


Figure 8. Input/Output Comparison for Lateral Steering Command

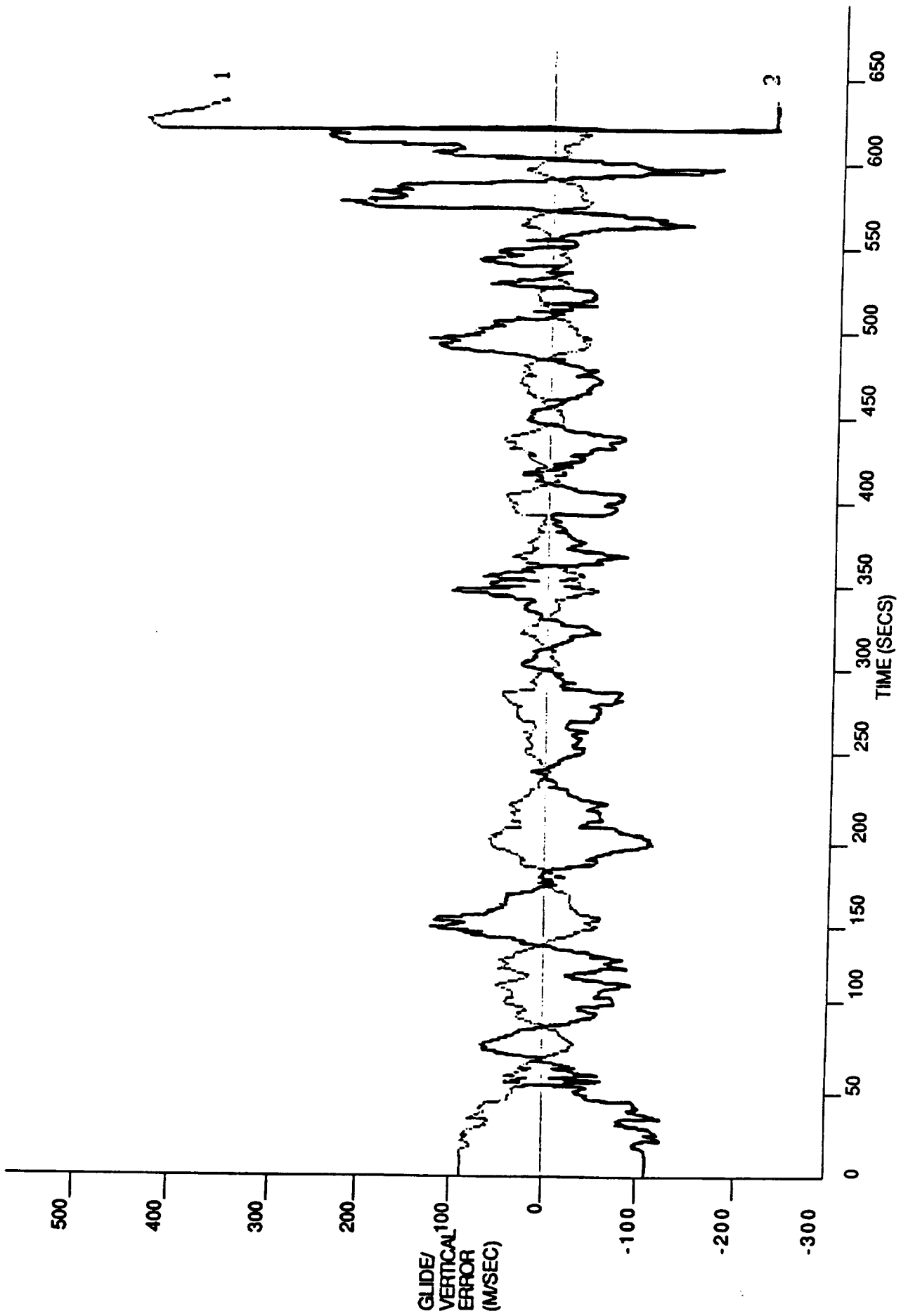


Figure 9. Input/Output Comparison for Vertical Steering Command

3.0 KALMAN FILTER FLIGHT SOFTWARE DATA ANALYSIS

3.1 INTRODUCTION

Under Task III of its current contract to NASA-Ames, TAU Corporation has participated in the analysis of the performance of the real-time Kalman filter. The performance has been compared to that of the Z-set filter in flight, and post flight comparisons with the laser tracker derived "truth" data (in addition to the Z-set) have been completed. Due to the problems and delays encountered in the flight test program, only a limited amount of data was made available for detailed analysis. Nevertheless, the current status of the flight test data analysis effort is discussed herein.

This report provides a summary of the data analysis work performed on data generated during the helicopter test flight on October 30, 1986.

As can be seen from these preliminary investigations, if properly tuned, an 8-state Kalman filter outperforms the Z-set in estimating the position and velocity of the aircraft.

A copy of the code used to perform the analysis was made available to NASA-Ames so that they could make additional runs and tune the filter to meet their needs.

3.2 POST-FLIGHT DATA ANALYSIS

Prior to the October 30, 1986, helicopter flight, TAU had no laser (truth) data with which to help tune the Kalman filter designed for the NASA flights. Thus, assuming that the Z-set, against which the filter was being compared in these flights, does a fairly accurate job of tracking the vehicle, the TAU filter was tuned to approximate the Z-set output as closely as possible, and the filter parameters were then slightly adjusted to yield smoother results. This "tuned" filter was used in the helicopter flight test on October 30, 1986.

Plots and data analysis results generated by NASA after the October 30 flight show TAU filter performance as inadequate. During two approaches, designated here as RUN3 and RUN5, that were successfully completed (no apparent equipment failure or satellite loss) the TAU filter appears to be outperformed by the Z-set -- especially so in the axis of most interest -- the vertical.

Having obtained the actual filter parameters used during the flight and the data recorded on board the helicopter at the time, TAU set out to try and duplicate the flight results and investigate the cause of poor performance, tuning, or structure. Filter tuning was performed on data collected during the period designated as RUN3.

Figures 1 and 2 show position and velocity errors on an axis-by-axis basis for the Z-set and the TAU filter, run with the filter parameters obtained from NASA for the helicopter flight test data on October 30. Figure 3 displays the overall position and velocity errors obtained for the run. Note that even though this filter is "poorly" tuned, overall position results are somewhat better than those produced by the Z-set, and velocity errors are only slightly worse.

ORIGINAL PAGE IS
OF POOR QUALITY

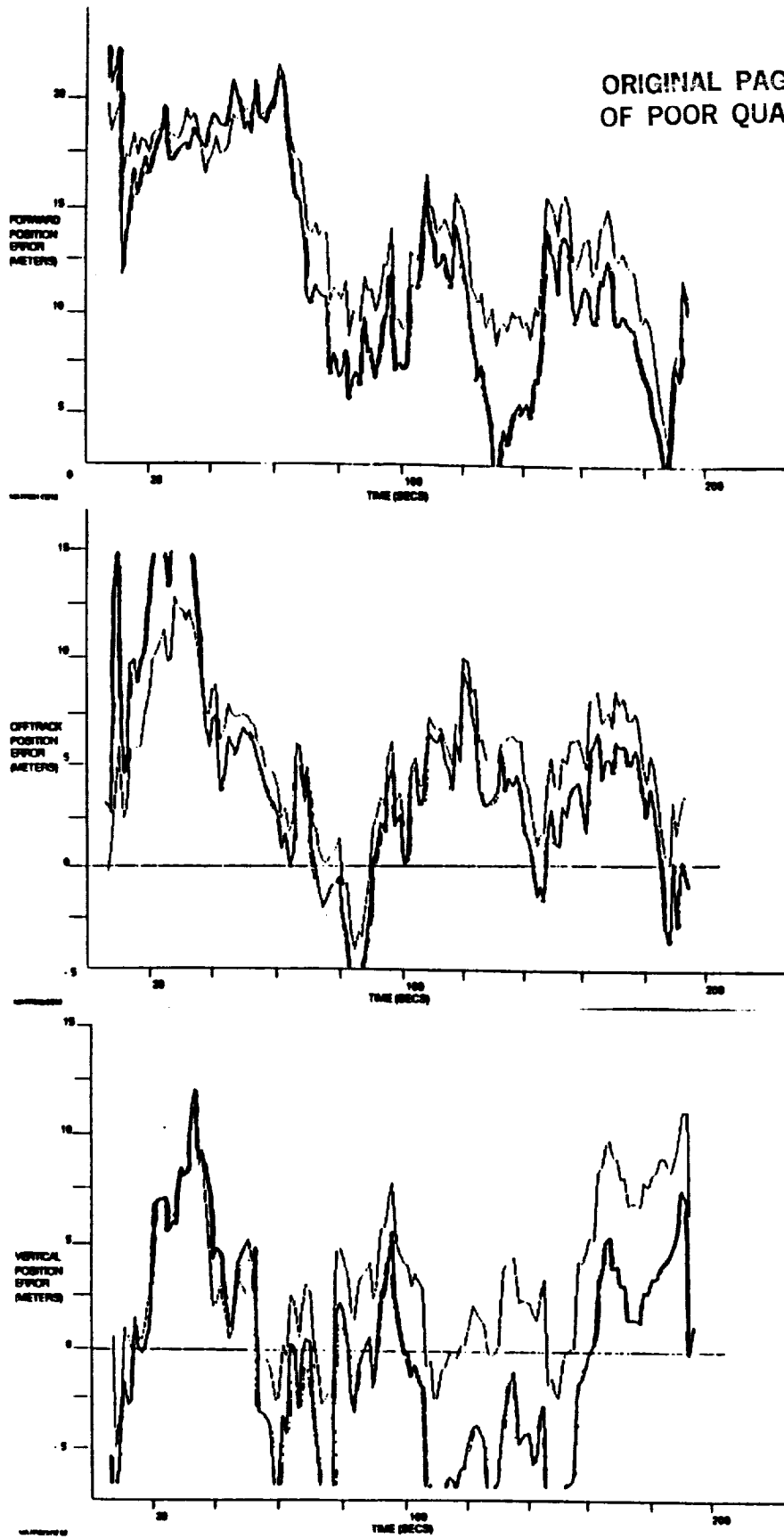


Figure 10. Position errors for the TAU filter (highlighted curves) and the Z-set, with parameters filter as defined for October 30 flight test.

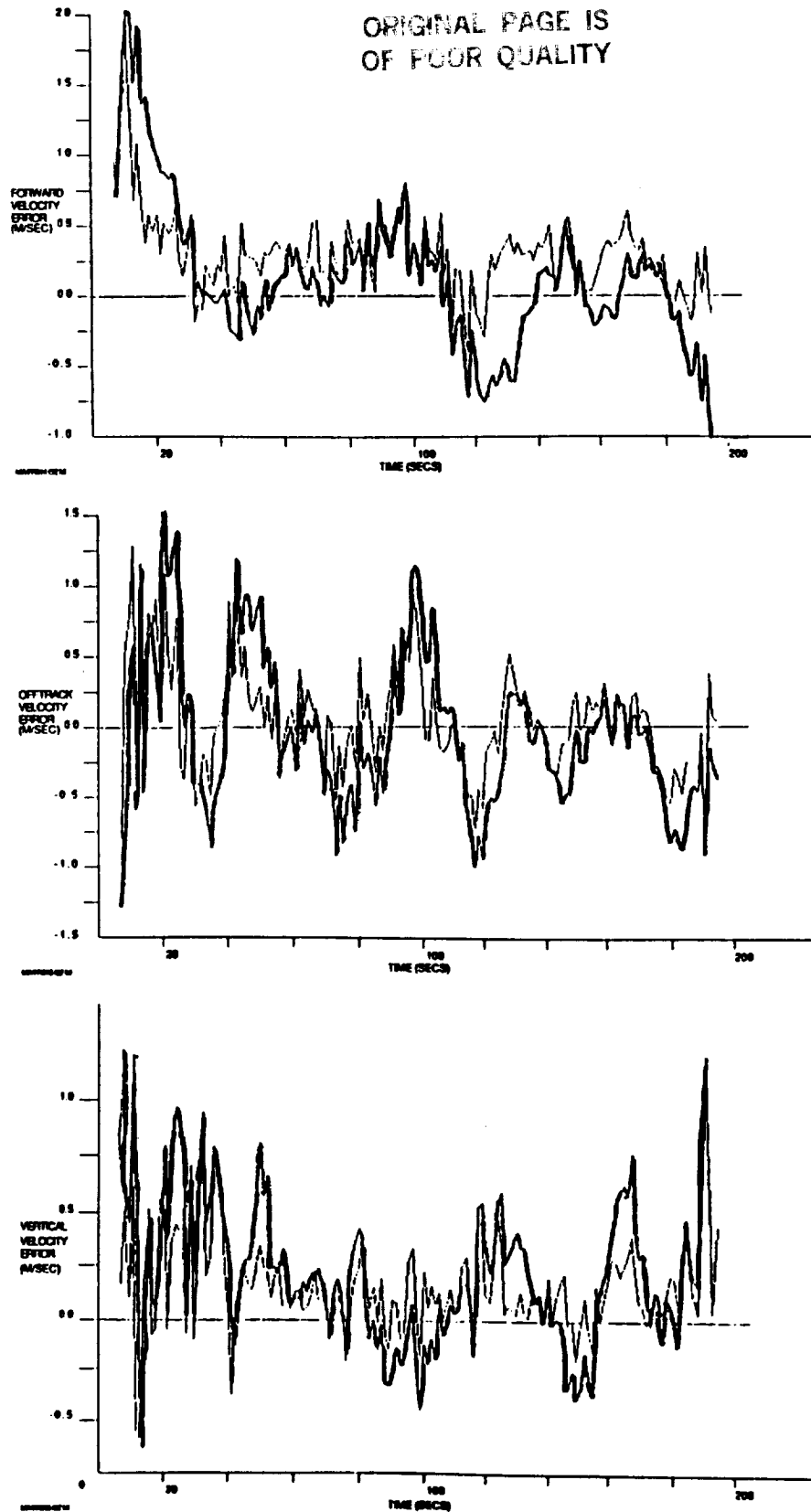


Figure 11. Velocity errors for the TAU filter (highlighted curves) and the Z-set, with parameters filter as defined for October 30 flight test.

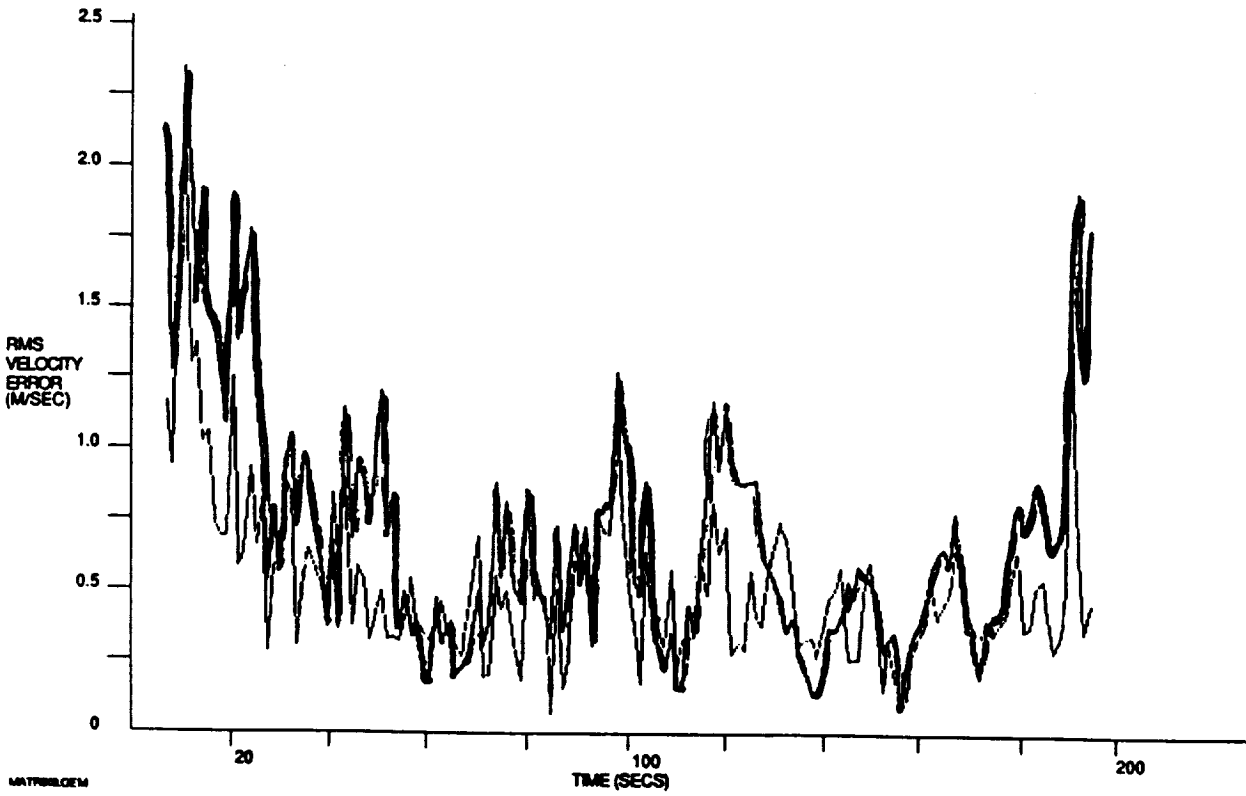
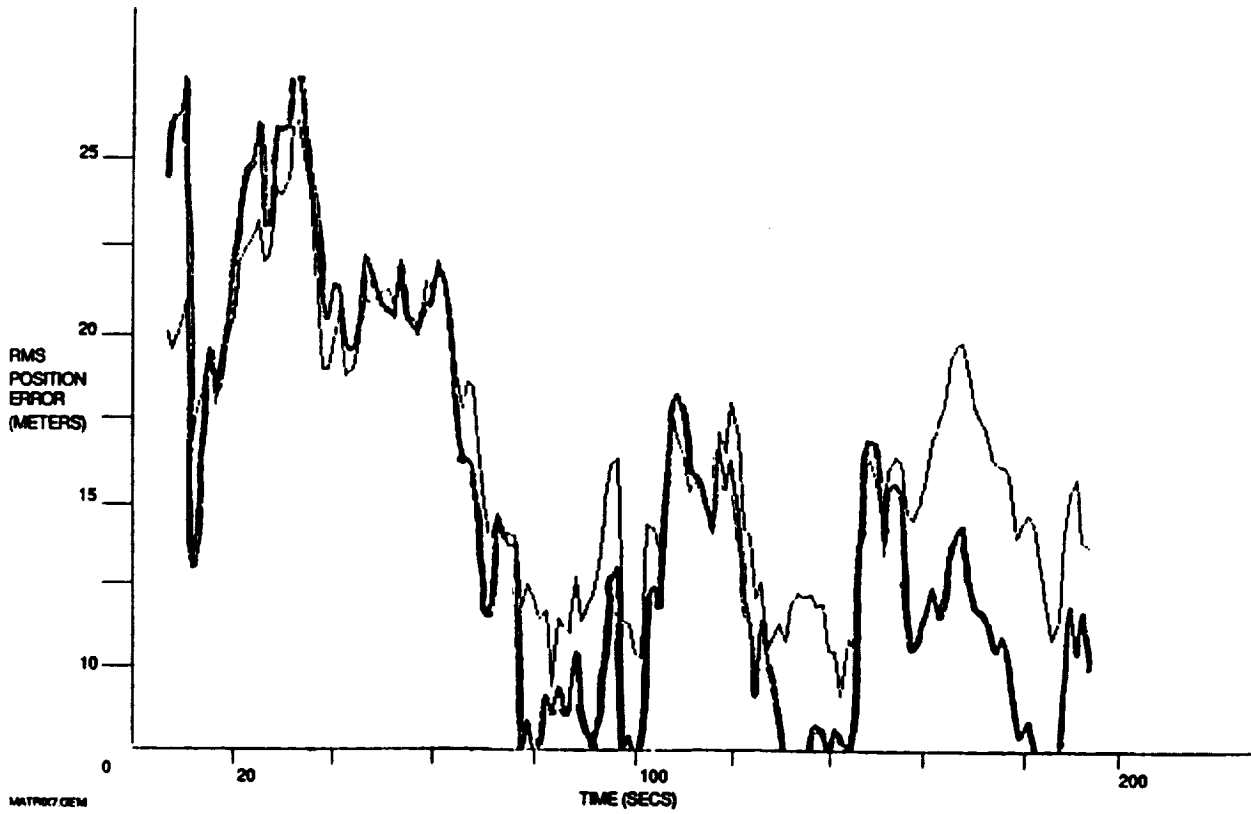


Figure 12. Total position and velocity errors for the TAU filter (highlighted curves) and the with Z-set, filter parameters as defined for October 30 flight test.

Various filter parameter combinations were then run to improve on the performance of the initial tuning. Although different parameter combinations yielded significant improvements in various axes, the "best" overall improvement is presented in Figure 4 through 6.

Figure 4 shows the position errors for the retuned TAU filter. Note that the performance of the TAU filter in the horizontal axes is significantly better than that of the Z-set. Special notice should be taken of the improvement in the vertical obtained with the new tuning. Figure 5 shows a definite improvement in the velocity estimation over the original tuning with the new filter; note also, that the returned TAU filter shows better performance than the Z-set, particularly in the three "excursions" in accuracy over this time period. The 3-dimensional, spherical performance of the retuned and the originally-tuned TAU filters is compared in Figure 6.

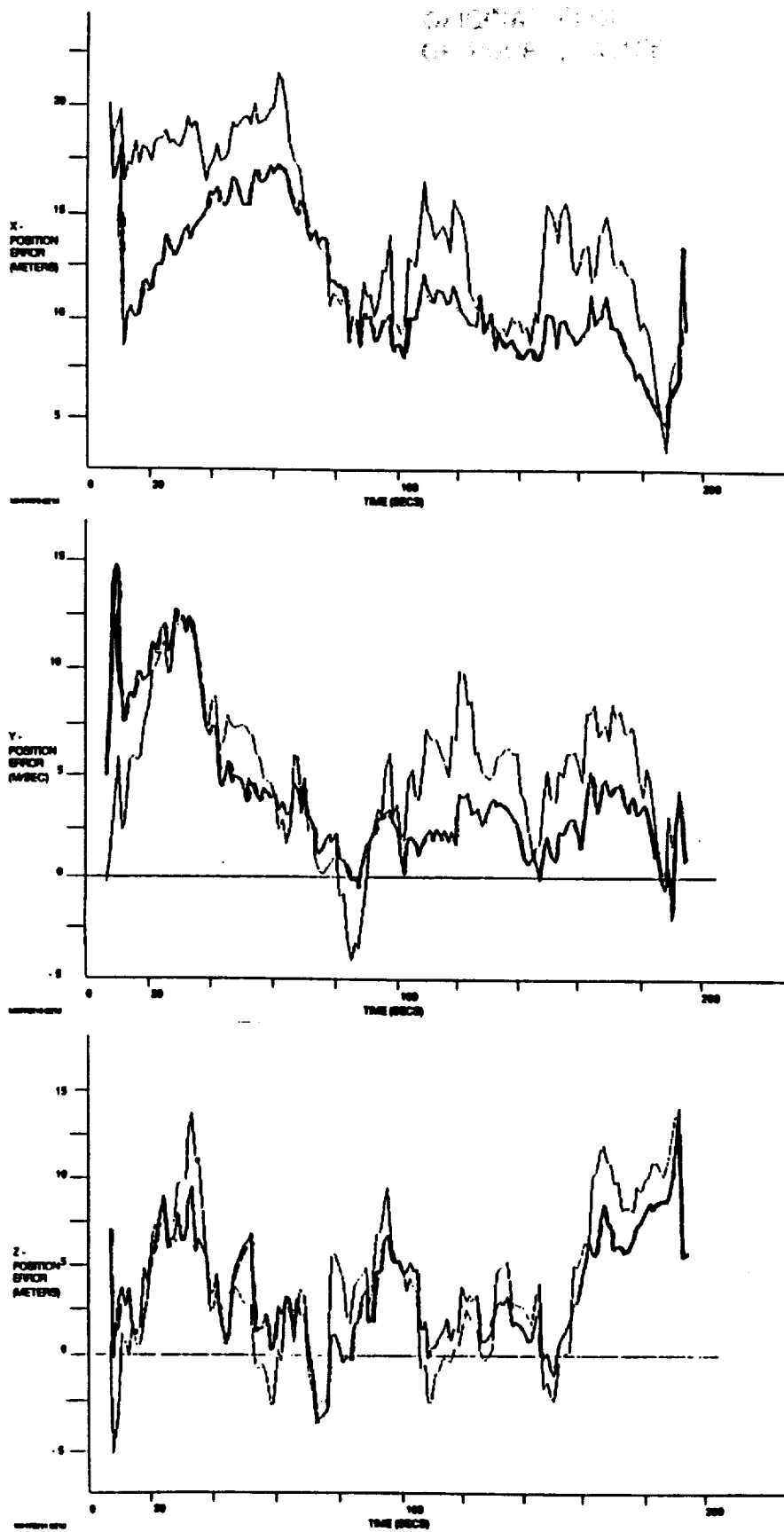


Figure 13. Position error comparison for the TAU filter (highlighted curves) and the Z-set.

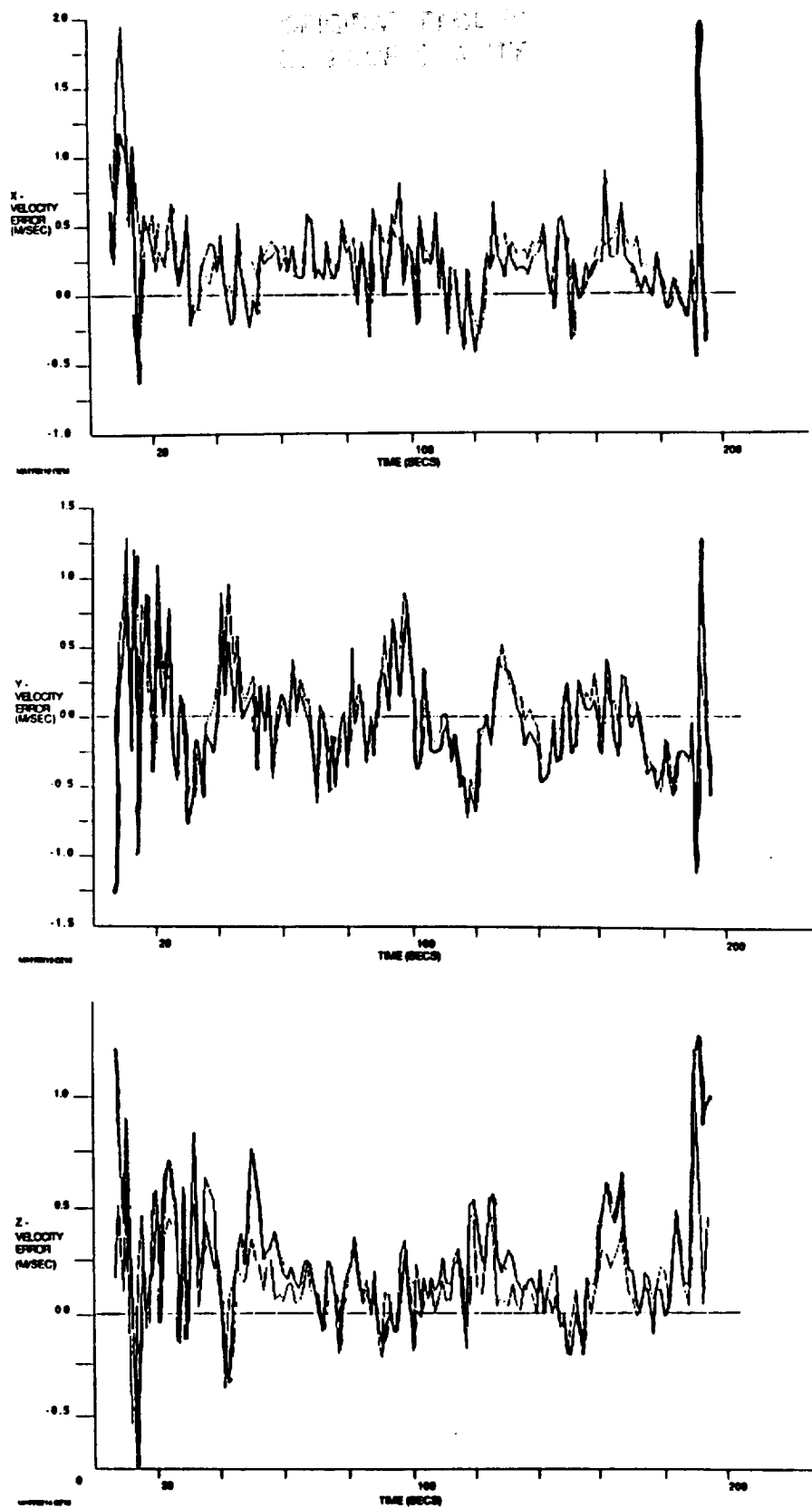


Figure 14. Velocity error comparison for the TAU filter (highlighted curves) and the Z-set.

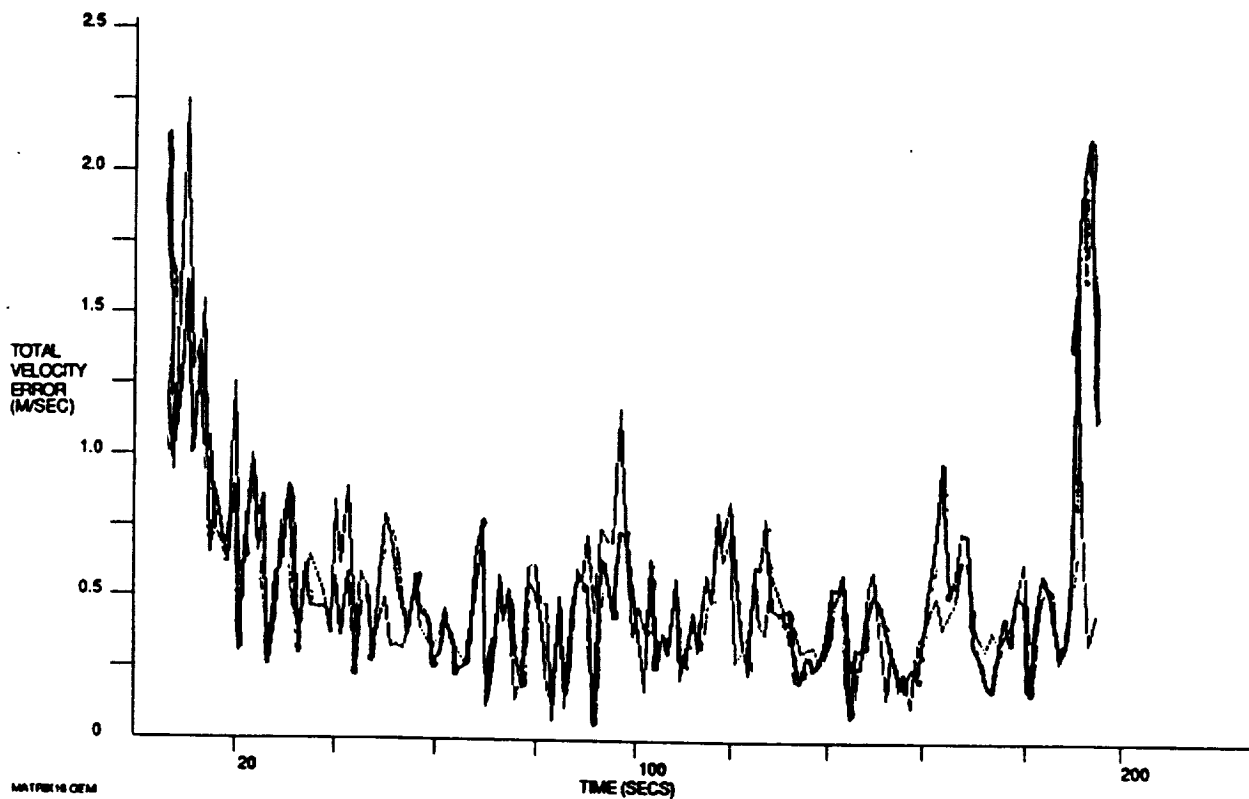
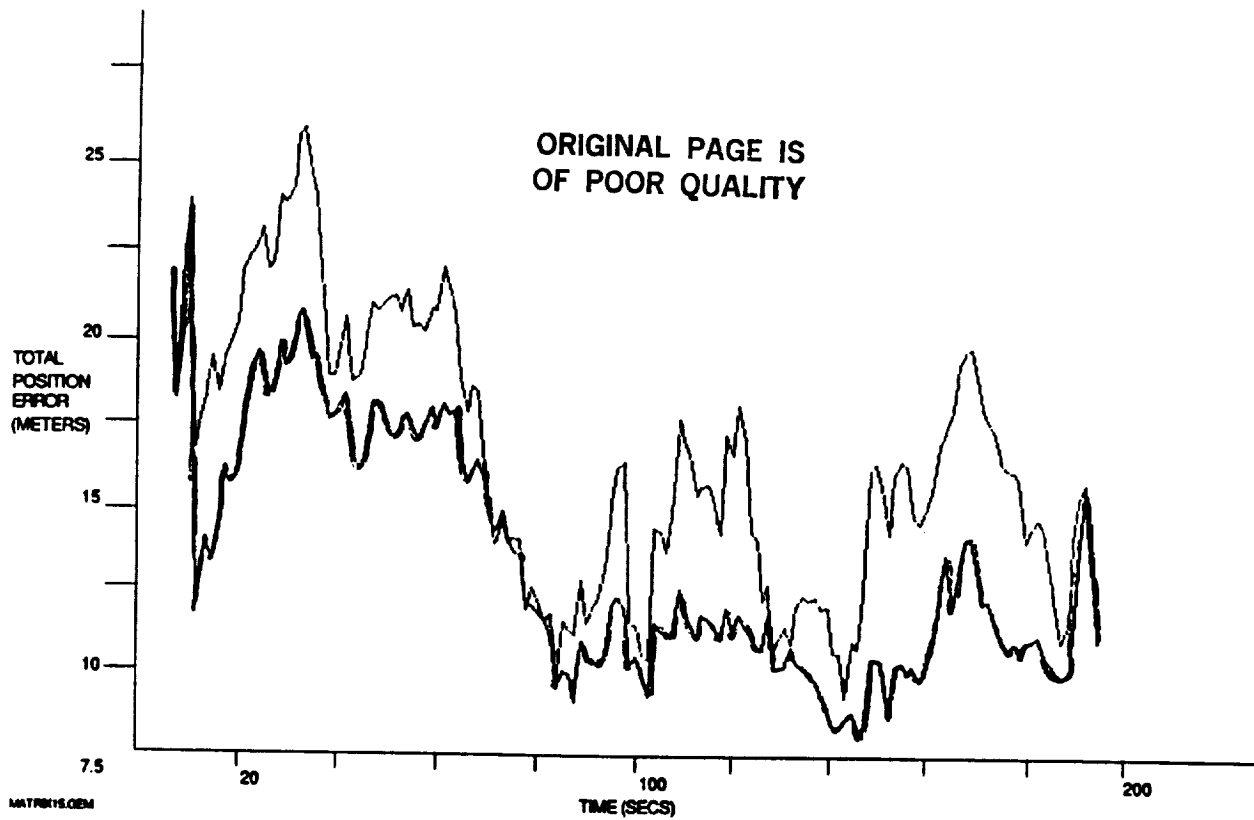


Figure 15. Total position and velocity error comparison for the TAU filter (highlighted curves) and the Z-set.

To validate these conclusions on the robustness of the retuning of the TAU filter, that is, to determine that the filter was not tuned specifically to the data recorded for RUN3, the retuned and originally-tuned filters were rerun on the data period designated as RUN5. Once again the tuned TAU filter outperformed the Z-set, in this case by a substantial margin. Overall position and velocity errors for the tuned filter for RUN5 data are presented Figure 7.

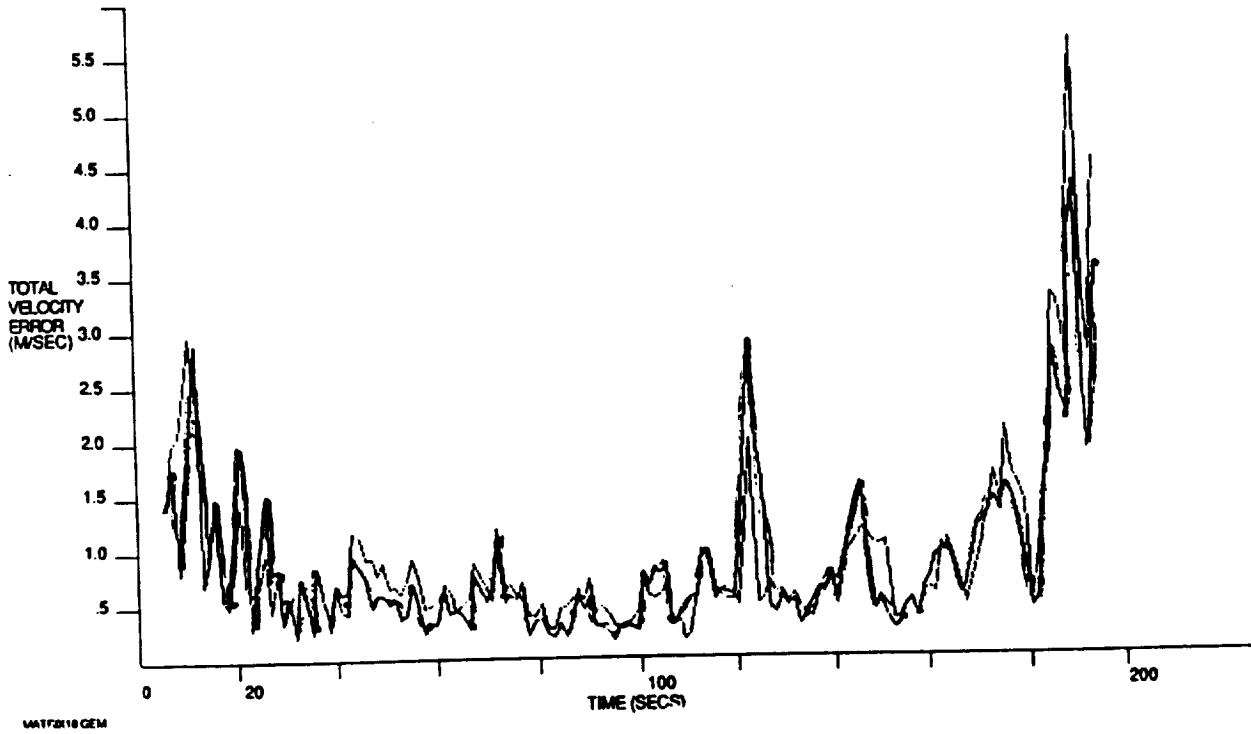
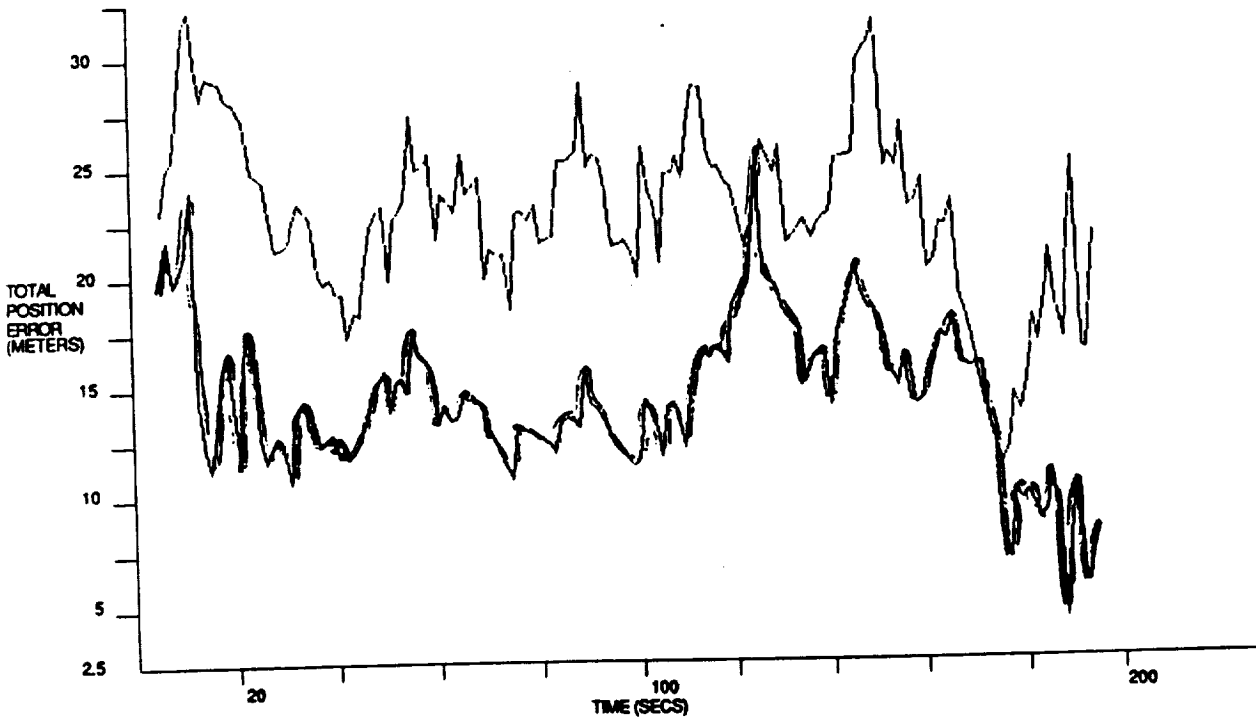


Figure 16. Total position and velocity errors for the retuned TAU filter and the Z-set.

ORIGINAL PAGE IS
OF POOR QUALITY

3.3 CONCLUSIONS

With proper tuning, the TAU filter shows definitely improved performance over the Z-set. The retuning also performed better against a second case. These results validate the performance of the TAU filter in meeting or bettering the Z-set performance prior to the addition of aiding sensors.

October 30 Flight Parameters: 8 - State Filter
Process Noise (Q-Values):

Pattern:

Q(1):	0.0 m	Q(2):	0.09 m/sec
Q(3):	0.0 m	Q(4):	0.09 m/sec
Q(5):	0.0 m	Q(6):	0.09 m/sec
Q(7):	0.0 nsec	Q(8):	0.000001 nsec/sec

Landing:

Q(1):	0.0 m	Q(2):	0.09 m/sec
Q(2):	0.0 m	Q(4):	0.09 m/sec
Q(5):	0.0 m	Q(6):	0.09 m/sec
Q(7):	0.0 nsec	Q(8):	0.000001 nsec/sec

Initial state estimates (1-sigma values):

STSIG(1):	100.0 m	STSIG(2):	10.0 m/sec
STSIG(3):	100.0 m	STSIG(3):	10.0 m/sec
STSIG(5):	100.0 m	STSIG(6):	10.0 m/sec
STSIG(7):	100.0 nsec	STSIG(8):	31.6227766 nsec/sec

Measurement noise estimates (1-sigma values):

Pseudorange:	12.0 m
Deltarange :	.8 m/sec

Retuned TAU Flight Parameters: 8 - State Filter
Process Noise (Q-Values):

Pattern:

Q(1):	0.0 m	Q(2):	1.00 m/sec
Q(3):	0.0 m	Q(4):	1.00 m/sec
Q(5):	0.0 m	Q(6):	0.01 m/sec
Q(7):	0.0 nsec	Q(8):	0.000001 nsec/sec

Landing:

Q(1):	0.0 m	Q(2)	1.00 m/sec
Q(2):	0.0 m	Q(4)	1.00 m/sec
Q(5):	0.0 m	Q(6)	0.10 m/sec
Q(7):	0.0 nsec	Q(8):	0.000001 nsec/sec

Initial state estimates (1-sigma values):

STSIG(1):	100.0 m	STSIG(2):	10.0 m/sec
STSIG(3):	100.0 m	STSIG(3):	10.0 m/sec
STSIG(5):	100.0 m	STSIG(6):	10.0 m/sec
STSIG(7):	100.0 nsec	STSIG(8):	10.0 nsec/sec

Measurement noise estimates (1-sigma values):

Pseudorange:	36.0 m
Deltarange :	.4 m/sec

Retuned TAU Flight Parameters: 9 - State Filter
Process Noise (Q-Values):

Pattern:

Q(1):	0.0 m	Q(2):	1.00 m/sec
Q(3):	0.0 m	Q(4):	1.00 m/sec
Q(5):	0.0 m	Q(6):	0.01 m/sec
Q(7):	0.0 nsec	Q(8):	0.000001 nsec/sec
Q(9):	0.01 m		

Landing:

Q(1):	0.0 m	Q(2):	1.01 m/sec
Q(2):	0.0 m	Q(4):	1.01 m/sec
Q(5):	0.0 m	Q(6):	0.10 m/sec
Q(7):	0.0 nsec	Q(8):	0.000001 nsec/sec
Q(9):	0.01		

Initial state estimates (1-sigma values):

STSIG(1):	100.0 m	STSIG(2):	10.0 m/sec
STSIG(3):	100.0 m	STSIG(4):	10.0 m/sec
STSIG(5):	100.0 m	STSIG(6):	10.0 m/sec
STSIG(7):	100.0 nsec	STSIG(8):	10.0 nsec/sec
STSIG(9):	100.0 m		

Measurement noise estimates (1-sigma values):

Pseudorange:	36.0 m
Deltarange:	.4 m/sec
Altimeter:	.1 m

SUMMARY


Under Tasks I and III of the subject contract, a Kalman filter was developed and flight tested. Its performance, in the absence of external sensors, was shown to be improved over that of the Z-set. Potential improvements which could be derived through incorporation of external sensors (e.g., a barometric altimeter and/or a vertical accelerometer) were examined but never flight tested. The Task I and III reports should be consulted for a more complete discussion of these results.

Under Task II, improvements to an existing steering algorithm were investigated; the relative merits of performing additional smoothing in the steering display were examined, and an improved waypoint switching logic was incorporated. The Task II Report should be consulted for a more complete discussion of these results.

REFERENCES

1. Geier, G. J., and Loomis, P. V. W., "Kalman Filter Data and Analysis Requirements," TAU Corporation Memorandum, 9 May 1986.
2. Geier, G. J., and Loomis, P. V. W., "Specification of Kalman Filter Data Requirements (Revision 1)," TAU Corporation Memorandum, 14 May 1986.
3. Bierman, G. J., Factorization Methods for Discrete Sequential Estimation, 1977.
4. Technical Staff, The Analytical Sciences Corporation, Applied Optimal Estimation, 1974.

ORIGINAL PAGE IS
OF POOR QUALITY

1. Report No. CR 177471	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Guidance Simulation and Test Support for Differential GPS Flight Experiment		5. Report Date October 1987	6. Performing Organization Code
		8. Performing Organization Report No.	
7. Author(s) G.J. Geier, P.V.W. Loomis and A. Cabak		10. Work Unit No. 505-66-11	
9. Performing Organization Name and Address Theory and Applications Unlimited 485 Alberto Way, Bldg D Los Gatos, CA 95030		11. Contract or Grant No. NAS2-12378	
		13. Type of Report and Period Covered NASA Contractors Report	
12. Sponsoring Agency Name and Address NASA Ames Research Center Moffett Field, CA 94035		14. Sponsoring Agency Code	
15. Supplementary Notes Point of Contact: Fred Edwards, Ames Research Center, M.S. 210-9 Moffett Field, CA 94035 - (415) 694-5437 or FTS 646-5437			
16. Abstract <p>This report consists of three separate tasks which supported the test preparation, test operations and post test analysis of the NASA Ames flight test evaluation of differential GPS. Task 1 consisted of a navigation filter design, coding and testing to optimally make use of Gps in a differential mode. The filter can be configured to accept inputs from external sensors such as an accelerometer and a barometric or radar altimeter. The filter runs in real time onboard a NASA helicopter. It processes raw pseudo and deltarange measurements from a single channel sequential GPS receiver. In the report the Kalman filter software interfaces are described in detail, followed by a description of the filter algorithm, including the basic propagation and measurement update equations. The performance during flight tests is reviewed and discussed.</p> <p>Task 2 describes a refinement performed on the lateral and vertical steering algorithms developed on a previous contract. The refinements include modification of the internal logic to allow more diverse inflight initialization procedures, further data smoothing and compensation for system induced time delays.</p> <p>Task 3 describes the TAU Corp participation in the analysis of the performance of the real time Kalman navigation filter. The performance was compared to that of the Z-set filter in flight and to the laser tracker position data during post test analysis. This analysis allowed a more optimum selection of the parameters of the filter.</p>			
17. Key Words (Suggested by Author(s)) NAVSTAR GPS NAVIGATION HELICOPTER APPROACH		18. Distribution Statement  Subject Category: 04	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) 4	21. No. of Pages 125	22. Price*

