A COMPARISON OF FILTERING ALGORITHMS FOR GPS SATELLITE NAVIGATION APPLICATION

B. D. Tapley, J. G. Peters, and B. E. Schutz

Aerospace Engineering and Engineering Mechanics
University of Texas

ABSTRACT

The limited word size proposed for the GPS navigation computers could precipitate the problem of filter divergence in a standard sequential estimation algorithm. To insure filter reliability during the periods when the GPS observations are being processed, a square-root filter algorithm has been adopted for the GPS navigation computer. Several formulations of square-root filtering algorithms have been developed during the past fifteen years. In implementing the filters, the covariance matrix can be propagated by either integrating the state transition matrix or by integrating a differential equation for the square root of the covariance matrix. The various approaches have different characteristics with regard to computer execution time, accuracy of the estimate, computer storage requirements, and the effort required to code and validate the algorithm. In this investigation the Potter, Carlson-Cholesky, and UDU square-root filters are compared with the standard extended Kalman filter. The characteristics of the algorithms are compared by simulating the application of a phase one GPS system to the determination of a LANDSAT-D Satellite.

A COMPARISON OF FILTERING ALGORITHMS FOR GPS SATELLITE NAVIGATION APPLICATION

B. D. Tapley[1], J. G. Peters[2], B. E. Schutz[3]

The Universtiy of Texas at Austin

INTRODUCTION

The limited word size used in contemporary microprocessor design may lead to problems in autonomous satellite navigation applications. The numerical error introduced when the navigation computations are performed with a short wordlength computer can lead to divergence of a standard extended sequential estimation algorithm. To insure filter reliability for applications where GPS observations are being processed, a square root filter algorithm has been adopted for the GPS navigation computer. Several formulations of square root filtering algorithms have been developed during the past fifteen years. This investigation describes a preliminary comparison of three square root filter formulations with the standard extended Kalman filter. Initial results are obtained regarding the relative computation speed and accuracy of these algorithms in a simulation of LANDSAT-D navigating with a Phase I GPS constellation.

This summary is an overview of results obtained in the study. A complete discussion of the simulation procedure and numerical results can be found in [1].

---

† The research was supported by the National Aeronautics and Space Administration at Goddard Space Flight Center under Grant No. NSG 5154.

[1] Professor, Department of Aerospace Engineering and Engineering Mechanics.

[2] Research Assistant, Department of Aerospace Engineering and Engineering Mechanics.

[3] Associate Professor, Department of Aerospace Engineering and Engineering Mechanics.

ALGORITHMS CONSIDERED

The algorithms compared in this initial study are listed in Table 1. They are classified according to one of four ways of representing the state error covariance matrix and one of two methods of time propagation of this covariance matrix.

The four methods of covariance representation are, briefly:

1. standard Kalman formulation [2],

2. the UDU algorithm [5], which decomposes the state error covariance into an upper unitary matrix, U, a diagonal matrix, D, and U^T,

3. the Carlson-Cholesky algorithm [4], in which the covariance is decomposed into an upper or lower triangular matrix, W, and its transpose W^T; and

4. the Potter algorithm [3] which decomposes the covariance into a general nxn square root covariance S and its transpose, S^T.

The two methods of performing time updates of the covariance are referred to as the transition matrix method and the direct integration method. In the former technique, a system of variational equations is integrated from one measurement epoch to the next, to obtain a transition matrix. The time update of the covariance is performed by the proper multiplication of the covariance by the transition matrix and by the addition of process noise. The process noise matrix is obtained in this study by an approximate analytic integration of a diagonal spectral level density matrix. The direct integration methods involve the numerical integration of the state error covariance matrix directly. Direct integration is often considered for time propagation because, in general, fewer equations must be integrated between observation epochs (n(n+1)/2 versus nxn for transition matrix methods). For square root algorithms which use a transition matrix method, the inclusion of process noise at each measurement epoch requires a retriangularization of the square root covariance matrix. This extra computational burden is avoided in the direct integration methods as the process noise effects are included directly in the integration of the differential equations.

The direct integration algorithm for the Potter filter requires an nxn matrix inversion at each integration step. As a result of the extreme numerical penalty resulting from the inversion operation, a directly integrated Potter algorithm is not considered.

OBSERVATION SIMULATION PROCEDURE

The filter algorithms were tested using a series of simulated range and range-rate observations made from the LANDSAT-D satellite to navigation satellites of a Phase I GPS constellation. The structure of the dynamic models used in the simulation of the observations is outlined in Table 2. The dynamic model of the motion of LANDSAT-D includes the geopotential effects of a non-spherical earth and the effects of atmospheric drag. The geopotential model used is GEM7 truncated to order and degree 8. The drag on the satellite is modeled as the function of a ballistic coefficient, the atmospheric density and the square of the user's velocity relative to the atmosphere. The atmospheric density is determined from an exponential density model.

In the observation simulation model, the GPS satellites are assumed to be in two-body circular orbits about a point mass earth. No other perturbations are assumed to affect them.

TABLE 1

ALGORITHMS COMPARED

Transition Matrix Methods:

- EKF($\dot{\Phi}$)
- UDU($\dot{\Phi}$)
- CARLSON-CHOLESKY($\dot{\Phi}$)
- POTTER($\dot{\Phi}$)

Direct Integration Methods:

- EKF($\dot{P}$)
- UDU($\dot{U}\dot{D}$)
- CARLSON-CHOLESKY($\dot{W}$)

TABLE 2

OBSERVATION SIMULATION MODEL

User Model:

- 8 × 8 Geopotential (GEM7)
- Atmospheric Drag
- Clock Errors

GPS Satellites Model:

- Two-Body Propagation Model
- Clock Errors

The performance of the user's clock and the GPS satellites' clocks are important factors in determining potential navigation accuracy. The phase error for each of the clocks is modeled in the simulation as the sum of three terms:

1. a deterministic phase error modeled as a first-order polynomial,
2. the integral of an exponentially correlated frequency error, and
3. a random walk phase error.

The frequency error of each clock is the sum of derivatives of the first two terms of the phase error.

Initial conditions for the Phase I GPS and for LANDSAT-D are shown in Table 3 and Table 4, respectively. The true anomaly of LANDSAT-D and the difference in the epoch times of the orbital elements can be varied to alter the pattern of GPS satellites visible from LANDSAT-D. The LANDSAT-D epoch elements are specified at a GPS system time of $t = 0$. The GPS epoch elements are specified at a GPS system time of -7200 seconds.

Given these initial conditions, and given the dynamic models described previously, the history of GPS satellite visibility shown in Figure 1 was generated. Over the 21,500 sec. simulation period, the number of GPS satellites visible to LANDSAT-D varies from zero to six. From this visibility history, a set of simulated range and range-rate observations was determined for processing by the navigation simulation program. The observations were generated on the CDC6600/6400 system at the University of Texas at Austin.

TABLE 3

PHASE I GPS CONFIGURATION

Satellite	Long. of Asc. Node (Deg.)	Mean Anomaly (Deg.)
1	-130.	0.
2	-130.	40.
3	-130.	80.
4	110.	40
5	110.	80.
6	110.	120.

Inclination: 63°

Eccentricity: 0.0

TABLE 4

LANDSAT-D EPOCH ORBITAL ELEMENTS

$a \equiv 7.086901 \times 10^6$ m

$e \equiv 0.001$

$i \equiv 98^\circ.181$

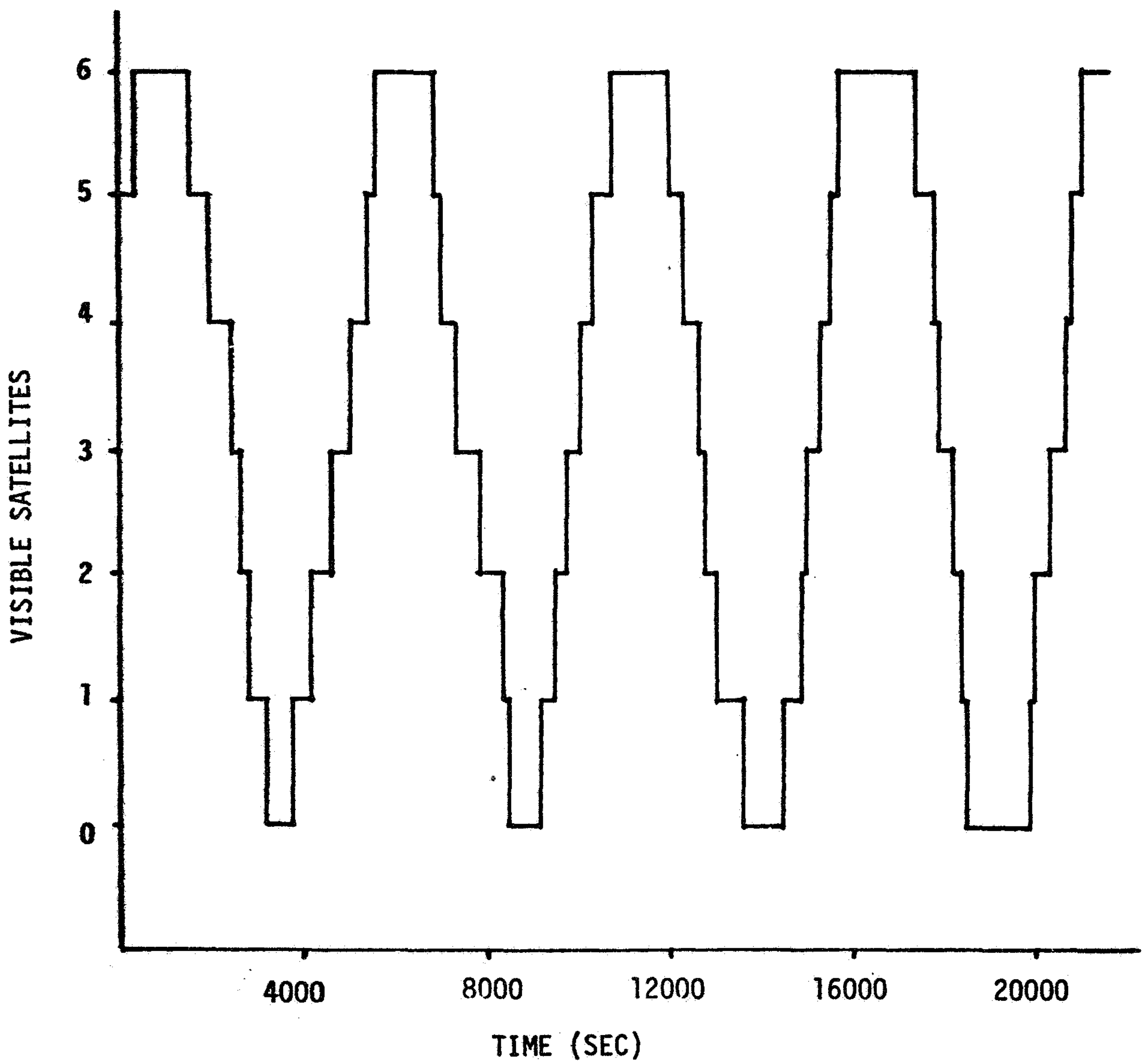
$\Omega \equiv 354^\circ.878$

$\omega \equiv 180^\circ.$

f(true anomaly) $\equiv -185^\circ.$

FIGURE 1

GPS SATELLITE VISIBILITY VS TIME



FILTER STRUCTURE

The limited size of satellite on-board computer necessitates the use of simplified dynamic models in the navigation filter. Table 5 shows the dynamic model structure assumed by the filter for the user satellite and for the GPS satellites.

The geopotential affecting the user satellite is of lower degree and order than that used to generate the simulated observations. The drag acceleration calculation uses an exponential model of atmospheric density, as does the observation generation program, but the ballistic coefficient of the user satellite is estimated. The bias of the user's clock at a given time is predicted by a linear equation. The coefficients of this equation, the clock bias and drift at an epoch, are estimated. The estimation of the ballistic coefficient and of the clock parameters attempts to account for modeling errors caused by the reduced size of the filter's geopotential.

The filter prescribes a two-body point mass geopotential as the dynamic model for the GPS satellites. Each GPS satellite's clock bias is predicted between observation epochs by a linear equation whose coefficients are predetermined to fit the clock's error behavior. For these simulations, the clock coefficients have been set to zero. This implies that the filter assumes perfect GPS clocks. Therefore, timing errors from all sources are accounted for in the estimation of the user's clock coefficients.

Preliminary simulations have been performed with an eleven-state navigation filter. The differential equations of the states are shown in Table 6. The eleven states are: position (1-3), velocity (4-6), ballistic coefficient (7), drag correlation parameter (8), user clock bias (9), user clock drift (10), and clock drift correlation parameter (11). Important

TABLE 5

FILTER SIMULATION MODEL

User Model:

- 4 × 4 Geopotential (GEM7)
- Atmospheric Drag (Ballistic Coefficient Estimated)
- Clock Error Polynomial (Coefficients Estimated)

GPS Satellites Model:

- Two-Body Propagation Model
- Clock Errors (Predetermined Coefficients)

TABLE 6

FILTER DIFFERENTIAL EQUATIONS

11-STATE FILTER:

$\dot{\bar{r}}$	$= \bar{v}$	(position)
$\dot{\bar{v}}$	$= \bar{a}_g + \bar{a}_d + \xi_v$	(velocity)
$\dot{d}$	$= -\beta_d\, d + \xi_d$	(ballistic coefficient)
$\dot{\beta}_d$	$= \xi_{\beta_d}$	(drag correlation parameter)
$\dot{b}_\rho$	$= b_{\dot{\rho}}$	(user clock bias)
$\dot{b}_{\dot{\rho}}$	$= -\beta_{b_{\dot{\rho}}}\, b_{\dot{\rho}} + \xi_{b_{\dot{\rho}}}$	(user clock drift)
$\dot{\beta}_{b_{\dot{\rho}}}$	$= \xi_{\beta_{b_{\dot{\rho}}}}$	(drift correlation parameter)

assumptions in the form of the differential equations are that the ballistic coefficient and user clock drift are first-order Gauss-Markov processes and that the correlation parameters of the Markov processes are random walks. Additionally, the user clock's bias is modeled as the integral of its drift.

Parameters in the forcing functions of the equations in Table 6 are the following:

$$\bar{a}_g \equiv \text{gravitational acceleration}$$

$$\bar{a}_d \equiv \text{drag acceleration.}$$

The $\bar{\xi}_i$ are white noise forcing functions with statistics

$$E[\bar{\xi}_i] = 0 \quad ; \quad E[\bar{\xi}_i \bar{\xi}_i^T] = Q_i$$

FILTER PERFORMANCE

Figures 2 and 3 are plots of position error magnitude and velocity error magnitude of the estimates versus time. Two trends in the error history are evident from the examination of the plots. Catastrophic increases in the error magnitudes occur during periods of poor satellite visibility (less than four satellites in view). A comparison of Figures 1, 2 and 3 reveals the correlation between satellite visibility and error magnitude. The peaks of these errors reach over one kilometer in position, and over 1 meter per second in velocity.

The second trend is observable by studying the long-term history of the errors. Although the errors decrease significantly when good satellite visibility is recovered, there is a long-term growth in the navigation error recorded during the periods of good visibility. As

FIGURE 2

RSS POSITION ERROR VS TIME

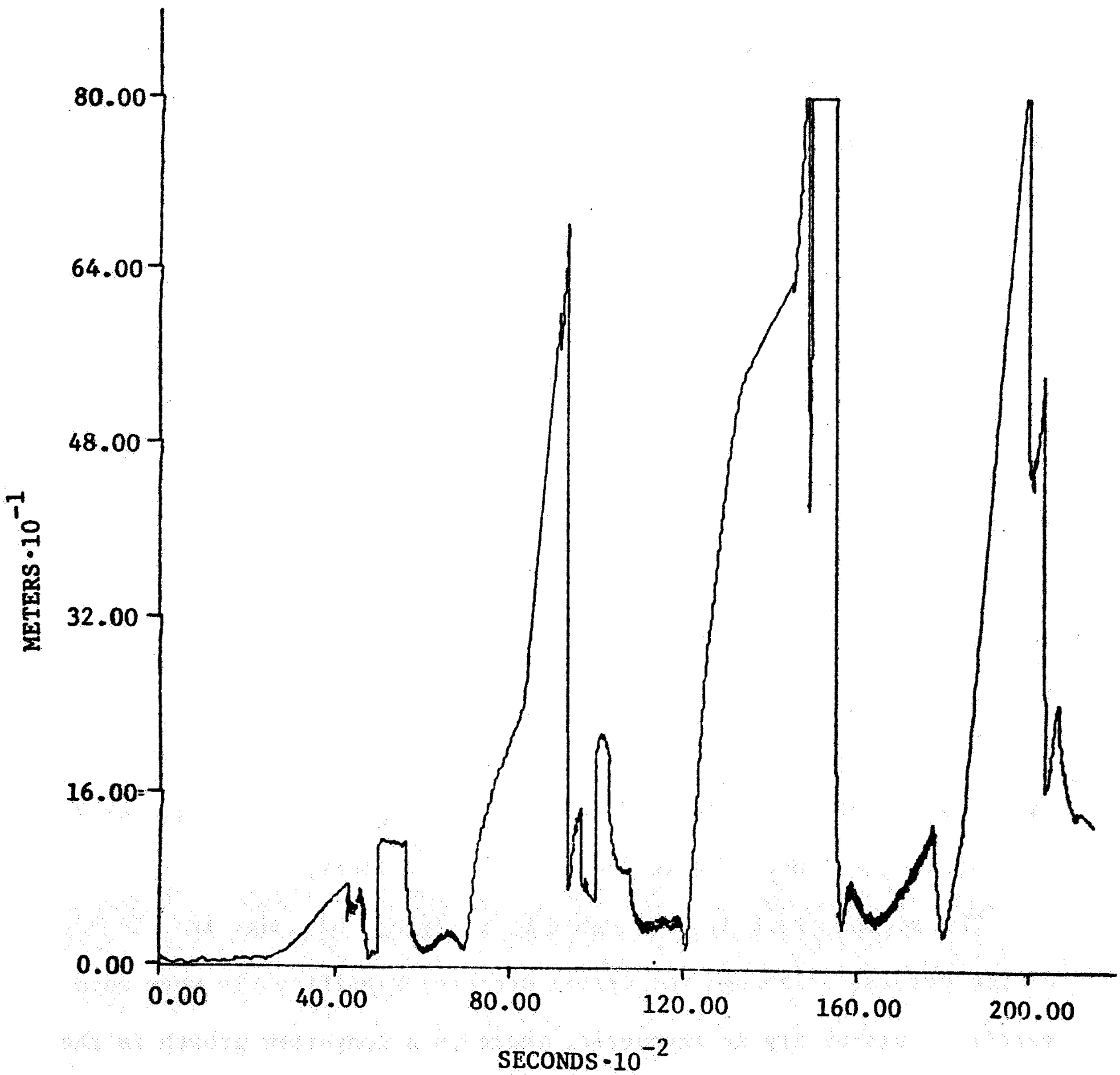
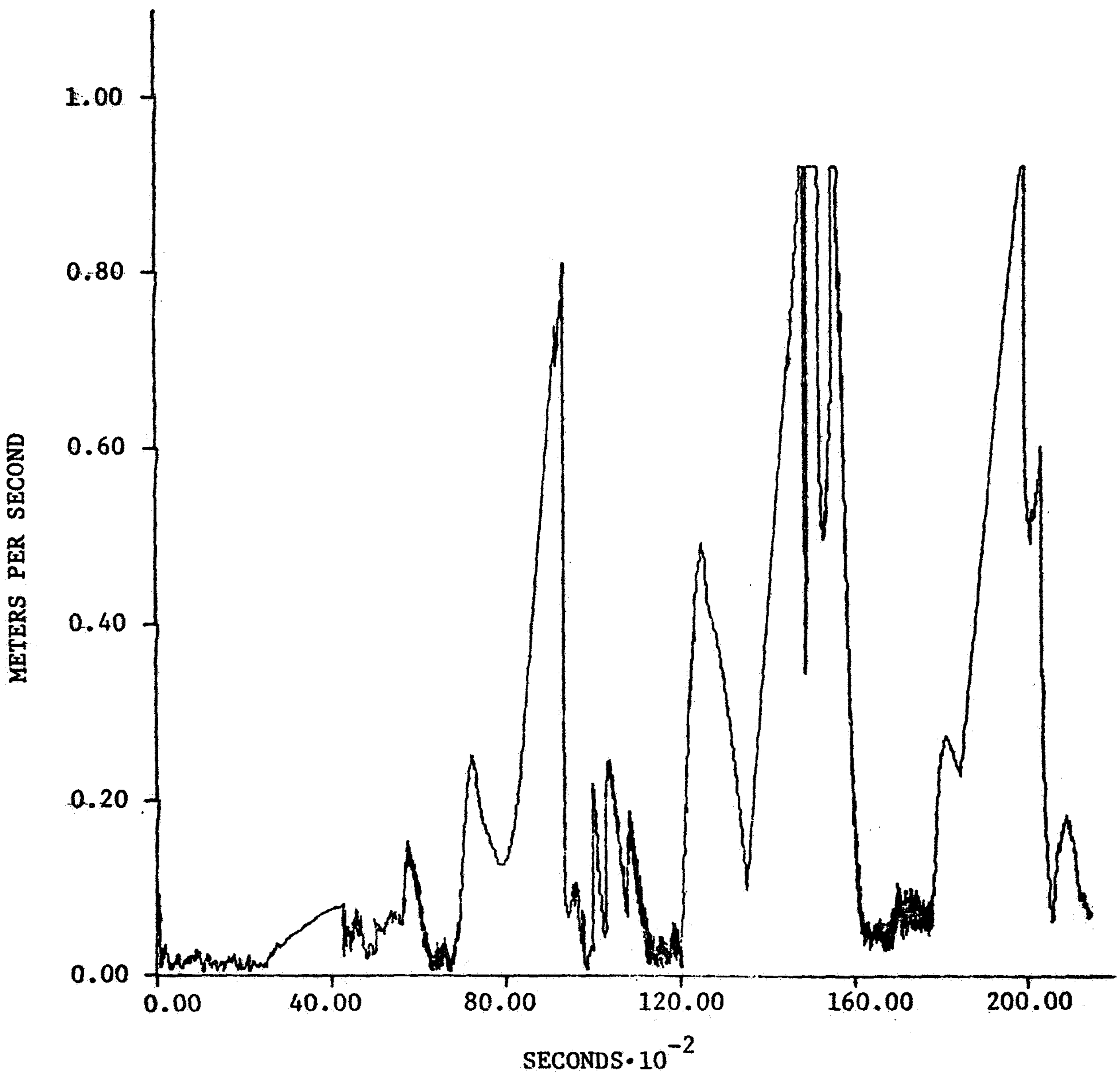


FIGURE 3

RSS VELOCITY ERROR VS TIME



an example, the position error before the first period of poor satellite visibility is at or below the 20 meter level. After the third period of poor visibility (approximately 16,000 sec.) position error has grown to approximately 100 meters. This indicates a secular or long period trend in the position error. A similar trend exists in the history of the velocity error magnitude.

Although both error trends are significant and are deserving of further study, the goal of this investigation has been to consider relative efficiencies of different algorithms. Therefore, current simulations have not been directed toward the removal of the error problems. This is to be the topic of follow-on studies.

INITIAL NUMERICAL RESULTS

The time propagation and measurement update algorithms described in the previous sections were tested on the 21,500 sec. arc of observations, using four different numerical integration schemes. The four schemes are:

1) a variable step RK(7)8 with integration tolerances: absolute error = 10^{-6}, relative error = 10^{-10},
2) a variable step RK(2)4 with integration tolerances: absolute error = 10^{-2}, relative error = 10^{-6},
3) a fixed step RK8 with a step size of 6 seconds, and
4) a fixed step RK4 with a step size of 6 seconds.

Simulations with these integration algorithms are intended to establish bounds on errors for navigation computations performed on the CDC6600 system. The results of further studies using the lower order integrators proposed for the actual GPS computer can be compared to the results shown here to determine the loss in navigation accuracy caused by using lower order integration methods.

Initial navigation simulations, using the different filter algorithms over the entire 21,500 sec. observation arc, produced position error histories nearly identical to that in Figure 2. Specifically, no algorithm reduced the catastrophic position error growth during the third data drop-out (approximately 12,000 sec.), or removed the long term error growth. Therefore, for the purpose of reducing computer use, the simulations for performance comparisons were run over the first 10,000 seconds of observations only.

The results of the initial set of simulations appear in Tables 7 through 10. The pertinent values tabulated are: the total computation time for time update, the total computation time for measurement update, the total computation time (the sum of time update and measurement update computation times), the total computation time normalized by the fastest total time, and the RMS error of the position estimate for the 10,000 seconds of data. All computation times are tabulated in milliseconds, and RMS errors are written in meters. The algorithms are listed in the order of increasing computation time.

The results for the different filtering algorithms can be compared on the basis of computation time, estimation accuracy, and algorithm stability. Briefly, a comparison of the algorithms reveals the following trends.

Computation Time

The computation times of the algorithms display the following general characteristics:

TABLE 7

RELATIVE ALGORITHM PERFORMANCE

RK(7)8 TOL: $10^{-6} - 10^{-10}$

	COMP. TIME (msec)				ACC(m)
ALG	TIME UPDATE	MEAS UPDATE	TOTAL UPDATE	NORMED UPDATE	RMS POS ERR
EKF($\dot{\Phi}$)	160.601	9.435	170.036	1.000	128.4
UDU($\dot{\Phi}$)	172.119	17.621	189.740	1.116	128.7
POTT($\dot{\Phi}$)	179.924	17.585	197.509	1.162	95.0
CARL($\dot{\Phi}$)	173.760	24.968	198.728	1.169	95.0
EKF($\dot{P}$)	357.728	9.442	366.720	2.157	129.0
CARL($\dot{W}$)	390.763	24.772	415.535	2.444	128.9
UDU($\dot{U}\dot{D}$)	530.734	18.286	549.020	3.229	128.4

TABLE 8

RELATIVE ALGORITHM PERFORMANCE

RK(2)4 TOL: $10^{-2} - 10^{-6}$

	COMP. TIME (msec)				ACC (m)
ALG	TIME UPDATE	MEAS UPDATE	TOTAL UPDATE	NORMED UPDATE	RMS POS ERR
EKF($\dot{\Phi}$)	67.310	10.215	77.525	1.000	128.4
UDU($\dot{\Phi}$)	77.234	17.629	94.863	1.224	128.7
POTT($\dot{\Phi}$)	85.569	18.308	103.894	1.340	95.0
CARL($\dot{\Phi}$)	78.683	25.256	103.939	1.341	95.0
CARL($\dot{W}$)	92.598	24.560	113.158	1.550	128.3
UDU($\dot{U}\dot{D}$)	144.782	19.499	164.281	2.119	140.1
EKF($\dot{P}$)	-----	----	-----	----	----

TABLE 9

RELATIVE ALGORITHM PERFORMANCE

RK 8, 6 SEC FIXED STEP

	COMP. TIME (msec)				ACC (m)
ALG	TIME UPDATE	MEAS UPDATE	TOTAL UPDATE	NORMED UPDATE	RMS POS ERR
EKF($\dot{\Phi}$)	231.178	8.957	240.135	1.000	128.4
UDU($\dot{\Phi}$)	243.286	18.493	261.779	1.090	128.7
POTT($\dot{\Phi}$)	252.709	17.675	270.384	1.126	95.0
CARL($\dot{\Phi}$)	245.504	26.009	271.513	1.131	95.0
CARL($\dot{W}$)	444.624	26.329	470.953	1.961	129.0
UDU($\dot{U}\dot{D}$)	479.770	18.191	497.961	2.074	128.8
EKF($\dot{P}$)	523.195	8.801	531.996	2.215	128.8

TABLE 10

RELATIVE ALGORITHM PERFORMANCE
RK 4, 6-SEC FIXED STEP.

	COMP. TIME (msec)				ACC(m)
ALG	TIME UPDATE	MEAS UPDATE	TOTAL UPDATE	NORMED UPDATE	RMS POS ERR
EKF($\dot{\Phi}$)	83.881	8.408	92.289	1.000	128.4
UDU($\dot{\Phi}$)	95.613	17.553	113.146	1.229	128.7
POTT($\dot{\Phi}$)	103.956	17.884	121.840	1.320	95.0
CARL($\dot{\Phi}$)	96.959	25.574	122.533	1.328	95.0
CARL($\dot{W}$)	130.071	25.632	155.703	1.687	128.6
EKF($\dot{P}$)	153.496	9.887	163.383	1.770	128.9
UDU($\dot{U}\dot{D}$)	-----	----	-----	----	----

1) transition matrix algorithms consistently have lower computation times for time update than do the direct integration methods,

2) the relative performance of computation times of the direct integration algorithms is not predictable from one set of integration parameters to the next, contrary to the predictability of the transition matrix methods,

3) for this computer system, the transition matrix square root methods do not suffer large computation time penalties relative to the EKF($\dot{\Phi}$) algorithm,

4) fixed step algorithms have larger computation times for time update and total computation time than do their variable-step counterparts.

<u>Accuracy</u>

Each algorithm generates an RMS position error of 128–129 meters, with two exceptions: the $\ddot{\mathrm{UD}}$ simulation with the RK(2)4 integrator records an RMS error of 140 meters and the Carlson($\dot{\Phi}$) and Potter($\dot{\Phi}$) algorithms each record 95 meter RMS errors. The increased error of the $\ddot{\mathrm{UD}}$ simulation results from the integration tolerances (10^{-2} and 10^{-6}) not being strict enough to maintain the error at the 128 meter level. The integrator exceeds the error limit set for each step by the tolerances. The 95 meter RMS errors of the Carlson($\dot{\Phi}$) and Potter($\dot{\Phi}$) algorithms are thought to result from an assumption in the coding of the noise update sections of these two algorithms. Currently, the reason for such a large drop in the position RMS is not known and, therefore, the error improvement offered by these algorithms should be viewed with caution.

With the exception of the two cases just mentioned, no algorithm pays a penalty in estimation error by decreasing the order of the numerical integrator.

Numerical Stability

No algorithm appears to offer any clear cut advantage in stability in the problem conditions tested and for this computer system. However, the transition matrix equations do appear to be smoother and are, therefore, easier to integrate than are equations of the direct integration methods. This is evidenced by the generally lower integration times of the transition matrix codes and by the failure of two of the direct methods to successfully finish simulations. Both vailures resulted from the discovery of a negative valued diagonal element in the covariance matrix brought about, it is believed, by an inability of the integrator to maintain an adequate single-step error, thus causing a fatal global error growth.

Additional Results

In the previously discussed set of navigation simulations, the integration of the state and state-error covariance matrix through data dropout periods is performed with one call to the numerical integrator. When only one integration call is made to span the entire data dropout, position error magnitudes during the dropout are not included in the navigation program's calculation of the total RMS position error. As these errors should be included in the RMS calculation, the navigation program has been modified to integrate in six-second intervals during dropouts and to calculate the position error at the end of each six-second interval. This means that numerical integration and process noise accumulation are performed in six-second increments until observations are obtained at the end of the dropout. As well as correcting the RMS position error calculation, this modification creates a better model of the operation of the actual GPS data processing system.

A plot of the RSS position error, as generated by the modified navigation program, is shown in Figure 4. Tables 11 and 12 contain the results of a set of navigation simulations in which the first 10,000 seconds of observations only were processed. The simulations were performed with each of the algorithms tested previously and with the RK(2)4 and RK4 integrators. The results given in Tables 11 and 12 are analogous to those contained in Tables 8 and 10, respectively, but were generated by the modified navigation program.

A comparison of the data in Tables 11 and 12 with those of Tables 8 and 10 shows no significant changes in the relative performance of the algorithms. The RMS position errors for most of the algorithms have grown from approximately 128 meters to approximately 166.5 meters because the position errors occurring during the data dropouts have been included in the calculation of the RMS error. The Carlson($\dot{\Phi}$) and Potter($\dot{\Phi}$) methods still have the lowest errors, but the values of the errors have experienced a growth roughly proportional to that experienced by the other algorithms. For the same reasons mentioned in the previous section, the lower RMS errors of these algorithms should be viewed with caution. Integration times and total computation times have increased as well. The increases result because the newly imposed six-second integration interval in the data dropouts forces a step size limit on the variable step integrator, causes more matrix retriangularizations for the transition matrix methods,

FIGURE 4

RSS POSITION ERROR VS TIME

(Modified Program)

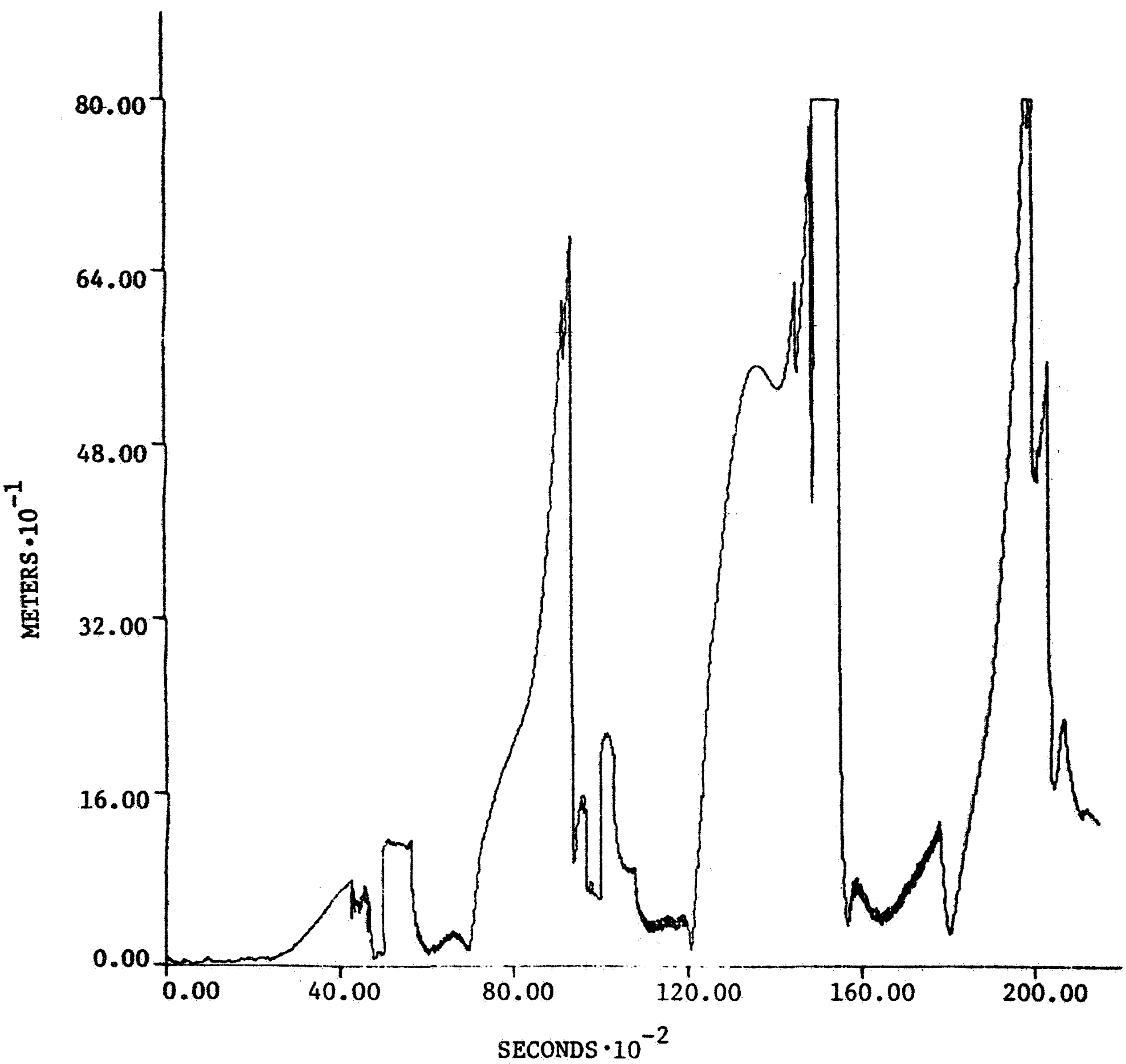


TABLE 11

RELATIVE ALGORITHM PERFORMANCE

RK(2)4, TOL: $10^{-2} - 10^{-6}$

ALG.	COMP. TIME (msec) TIME UPDATE	MEAS UPDATE	TOTAL UPDATE	NORMED UPDATE	ACC (m) RMS POS ERR
EKF($\dot{\Phi}$)	79.296	9.618	88.914	1.000	166.5
UDU($\dot{\Phi}$)	91.916	19.536	111.452	1.253	166.5
CARLSON($\dot{\Phi}$)	95.750	26.607	122.357	1.376	109.6
POTTER($\dot{\Phi}$)	104.488	19.361	123.849	1.393	109.6
EKF($\dot{P}$)	119.353	8.575	127.928	1.439	166.9
UDU($\dot{U}\dot{D}$)	225.471	19.697	244.968	2.755	166.9
CARLSON($\dot{W}$)	274.643	26.735	301.378	3.390	166.6

TABLE 12

RELATIVE ALGORITHM PERFORMANCE

RK4, 6-SEC FIXED STEP

ALG.	COMP. TIME (msec) TIME UPDATE	MEAS UPDATE	TOTAL UPDATE	NORMED UPDATE	ACC (m) RMS POS ERR
EKF($\dot{\Phi}$)	91.196	9.040	100.236	1.000	166.5
UDU($\dot{\Phi}$)	104.366	19.316	123.682	1.234	166.5
CARLSON($\dot{\Phi}$)	107.833	26.065	133.898	1.336	109.6
POTTER($\dot{\Phi}$)	116.608	19.320	135.928	1.356	109.6
CARLSON($\dot{W}$)	136.829	27.048	163.319	1.635	166.5
EKF($\dot{P}$)	163.799	8.652	172.451	1.720	166.0
UDU($\dot{U}\dot{D}$)	-	-	-	-	-

and generally creates a higher computation overhead due to a greater number of integrator calls.

The unexpected results in this new set of simulations are the improved performances of the $\dot{P}$ and $\ddot{UD}$ algorithms in conjunction with the RK(2)4 integrator. In this set of simulations, both methods have RMS errors consistent with the errors of the other algorithms (approximately 166.5 meters). Yet in the original simulations, those recorded in Table 8, the $\dot{P}$ method failed to complete its simulation run and the $\ddot{UD}$ code had a higher RMS error than did the other algorithms. The error improvements are believed to occur because the step size limit imposed by the six-second integration interval insures an accuracy in the integration of the $\dot{P}$ and $\ddot{UD}$ equations that the integration tolerances cannot insure.

SUMMARY

It is reemphasized that these results apply to simulations on the CDC6600 system only. Although it is believed that these results do establish important trends in the relative performance of these algorithms, additional simulations on other comuter systems, particularly those with small wordlengths, are essential. Such simulations are currently being carried out.

This article is intended as a general overview of the simulation method and initial results. A more exhaustive analysis of these results, as well as detailed descriptions of the simulation models and filter algorithms can be found in [1].

REFERENCES

1. B. D. Tapley, J. G. Peters, B. E. Schutz, "A Comparison of Square Root Estimation Algorithms for Autonomous Satellite Navigation," IASOM TR79-1, University of Texas at Austin, February 1979.

2. B. D. Tapley, "Statistical Orbit Determination Theory," presented at NATO Advanced Study Institute in Dynamical Astronomy, Cortina d'Ampezzo, Italy, August 1972.

3. R. H. Battin, "Astronautical Guidance," McGraw-Hill Book Company, Hightstown, New Jersey, 1964, pp. 388-389.

4. N. A. Carlson, "Fast Triangular Formulation of the Square Root Filter," AIAA Journal, Vol. 11, pp. 1239-1265, September 1973.

5. G. J. Bierman, "Factorization Methods for Discrete Sequential Estimation," Academic Press, New York, New York, 1976.