AD

AD 740876

Research and Development Technical Report

ECOM-0212-2

PROJECT RATRAN
RESEARCH ON THE ESTIMATION OF TRAJECTORIES FROM RADAR DATA

SECOND QUARTERLY REPORT

Prepared by

R. S. Berkowitz, M. Plotkin, D. Luber

S. Gottesman

FEBRUARY 1972

Distribution Statement
Approved for public release; distribution unlimited.

1ST QTLY REP.
AD 731499

ECOM

UNITED STATES ARMY ELECTRONICS COMMAND · FORT MONMOUTH, N.J.

CONTRACT DAAB07-71-C-0212

UNIVERSITY OF PENNSYLVANIA

The Moore School of Electrical Engineering (Moore School Report 72-23), Philadelphia, Pennsylvania 19104

77

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

UNCLASSIFIED

MIL-R-55611(EL)

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The Moore School of Electrical Engineering University of Pennsylvania Philadelphia, Pennsylvania 19104	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

Project RATRAN - Research on the Estimation of Trajectories from Radar Data

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Second Quarterly Progress Report

5. AUTHOR(S) (First name, middle initial, last name)

Raymond S. Berkowitz, et al

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1972	75	2

8a. CONTRACT OR GRANT NO. DAAB07-71-C-0212 b. PROJECT NO. 1S6-63719-DK-72-02-01 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ECOM-0212-2 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) Moore School Report No. 72-23

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Army Electronics Command Fort Monmouth, New Jersey 07703 AMSEL.CT-R

13. ABSTRACT

The general purpose of the work reported here is to obtain the best possible feasible signal processing algorithms for estimating from radar data characteristics and trajectory parameters of bodies moving in the air.

By analyzing the characteristics of a family of projectiles, a set of approximate simplified dynamics equations is obtained (Section 2) that can be used for extrapolation and backtracking with radar determinable parameters. The formulation includes terms which account for both drag and drift. Tentative numerical results indicate small resultant backtracking errors due to drift but somewhat larger errors due to drag, especially when the projectile's velocity passes thru mach 1 during the observation period.

In Section 3, formulation is developed which can be programmed to determine the effects of random radar errors on trajectory estimation and backtracking accuracy.

Section 4 presents the formulation necessary for constructing an extended Kalman filter and smoother algorithm for extracting pertinent state vectors and trajectory parameters from radar data. Specific inputs required and relationships are given in form suitable for programming.

Six appendices are included, giving some of the mathematical details, describing modifications and other experience with the modified point-mass dynamics program, and a glossary of the main symbols used.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Figure 3

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Simplified-Dynamics Equations						
Modified Point-Mass Equations						
Drag Acceleration						
Drift Acceleration						
Universal Functions						
Backtracking						
Radar Errors						
Launch-Point Estimation						
Error Covariances						
Filter						
Smoother						

Figure 3 (Continued)

TR ECOM-0212-2
FEBRUARY 1972

Reports Control Symbol
OSD-1366

PROJECT RATRAN

RESEARCH ON THE ESTIMATION OF TRAJECTORIES FROM RADAR DATA

SECOND QUARTERLY REPORT
1 August 1971 to 31 October 1971

CONTRACT NO. DAAB07-71-C-0212
DA PROJECT NO. 1S6-63719-DK-72-02-01

DISTRIBUTION STATEMENT

(Approved for Public Release;
distribution unlimited)

Prepared by

R. S. Berkowitz
M. Plotkin
D. B. Luber
S. Gottesman

The Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia, Pennsylvania 19104

For

U. S. ARMY ELECTRONICS COMMAND, FORT MONMOUTH, N. J. 07703

Moore School Report No. 72-23

ABSTRACT

The general purpose of the work reported here is to obtain the best possible feasible signal processing algorithms for estimating from radar data characteristics and trajectory parameters of bodies moving in the air.

By analyzing the characteristics of a family of projectiles, a set of approximate simplified dynamics equations is obtained (Section 2) that can be used for extrapolation and backtracking with radar determinable parameters. The formulation includes terms which account for both drag and drift. Tentative numerical results indicate small resultant backtracking errors due to drift but somewhat larger errors due to drag, especially when the projectile's velocity passes thru mach 1 during the observation period.

In Section 3, formulation is developed which can be programmed to determine the effects of random radar errors on trajectory estimation and backtracking accuracy.

Section 4 presents the formulation necessary for constructing an extended Kalman filter and smoother algorithm for extracting pertinent state vectors and trajectory parameters from radar data. Specific inputs required and relationships are given in form suitable for programming.

Six appendices are included, giving some of the mathematical details, describing modifications and other experience with the modified point-mass dynamics program, and a glossary of the main symbols used.

FOREWORD

This report describes work done from 1 August 1971 through 31 October 1971 at the Moore School of Electrical Engineering, University of Pennsylvania, under contract number DAAB07-71-C-0212 with U. S. Army Electronics Command for research entitled "The Estimation of Trajectories from Radar Data". The cognizant technical personnel at USA ECOM are Dr. Leonard Hatkin, head of the Radar Technical Area and Mr. Duane Sheppard, CSTA and SI Laboratory, Evans Area Fort Monmouth, N.J. 07703.

TABLE OF CONTENTS

(continued)

APPENDICES

TABLES AND FIGURES

1.0 INTRODUCTION.

As indicated in the previous Quarterly Progress Report, this work is concerned with the general signal processing problem of estimating characteristics and trajectory parameters of a body moving in the air from noisy radar observations.

The main thrust during this last quarter has been the adaptation of a BRL program supplied to us by USAECOM: to provide trajectory data for a suitable family of ballistic projectiles, to calculate simulated radar coordinate data that can result from different geometrical configurations, and to form the basis of "simplified dynamics" equations to be used in filter-smoothing algorithm design and for backtracking simulation and evaluation.

Additional results have been obtained concerned with evaluation of irreducible estimation errors due to radar measurement errors and the development of the necessary formulation for the optimal smoothing, filtering, and backtracking algorithms for the estimation of necessary trajectory parameters.

1.1 Summary, main results.

Section 2.0 to follow describes the development of simplified-dynamics equations, that are suitable for approximate trajectory calculations in terms of measurable parameters. Starting with the BRL modified point-mass equations and a representative set of projectile characteristics, simplified expressions for drag force and for drift force were evolved. For drag a universal coefficient curve was calculated that together with a projectile variable scale parameter that must be estimated enabled the approximate drag force to be calculated. For drift, a normalized parameter differential equation with a universal curve coefficient was postulated requiring an initial value estimation from the data. Magnus force was assumed negligible. In Section 2, the necessary formulation is developed and a set of tests are described which indicate the effectiveness of the simplified-dynamics equations in backtracking along the trajectory to the launch point. It is shown that drift force errors are effectively accounted for but drag forces still lead to errors that are perhaps larger than desirable. Associated with this section are Appendix A, giving the BRL Modified point mass equations, Appendix B showing the Universal Aerodynamic Functions evolved for drag and drift force effects and Appendix C commenting on the computational aspects of the work.

Section 3.0 presents the formulation necessary to determine theoretical limitations on accuracy of trajectory parameter estimates (launch-point state and aerodynamic parameters) due to random radar noise errors. Associated with this section is Appendix D which gives the formulation required for transformations between radar-centered and gun-centered coordinate systems. Formulation to account for radar bias errors will be presented in a subsequent report.

Section 4.0 presents the general filter-smoother algorithm developed to estimate trajectory parameters from the noisy radar data. Following the extended Kalman filter technique an augmented time-varying state-vector is defined having eight components consisting of target position and velocity vector components as well as drag and drift parameters. The formulation is set up for calculating optimal estimates of present value (filter) and initial value (smoother) of the augmented state vector updated on a quasi-real time basis. Initial work assumes availability of radar data for off-line processing but once algorithms are efficiently programmed, the feasibility of true real-time processing will be assessed. Provision is made for use of simulation as well as analytical techniques for performance and feasibility evaluation of the algorithms developed. Results are presented in a form convenient for programming. Specific definition of matrix elements required is given in Appendix E. A detailed theoretical development will be presented in the next quarterly report.

Conclusions and plans are described in Section 5.

2.0 SIMPLIFIED-DYNAMICS EQUATIONS.

BRL (Ballistics Research Laboratory) Report No. 1314., Reference [1], presents the modified point-mass equations used in computing firing tables. These equations are described briefly in Appendix A.

In order to compute backward in time from the observed projectile positions to the launch point, some equations governing the projectile motion must be used. If the projectile type and its aerodynamic characteristics were known, the modified point-mass equations could be used for computing backward along the trajectory. Under the conditions of interest in the present study, nothing is known about the projectile except what is deduced from its observed behavior. It is assumed in the present study that the accuracy and reproducibility requirements on artillery fire must result in generally similar aerodynamic characteristics for all artillery shells, so that the 105mm, 155mm, 175mm, and 8 inch projectiles -- called for brevity the standard projectiles -- have characteristics generally similar to any effective artillery shell. In particular it is assumed that the standard projectiles can be imbedded, in a sense made specific below in connection with equations (5) and (12), in the family of all projectiles, the different members of the family differing from one another only in numerical values of certain parameters.

Casual examination of the point-mass equations in Appendix A shows that the trajectory is affected by three aerodynamic forces, causing drag, drift, and the Magnus effect, in addition to gravity and the Coriolis "force". Among them they involve seven functions of Mach number and at least five constants, that can vary from projectile to projectile. To determine all these functions and constants from observation of the behavior of an unknown projectile would require accuracies in and durations of radar observations far in excess of what is achievable. The present study assumes that it will be enough to estimate, from the observed projectile behavior, the principal (zero-yaw) component of the drag acceleration and the drift acceleration. The conditions under which this assumption is valid are discussed in 2.4 below. In all previous work of which the writers are aware, only the zero-yaw drag acceleration has been estimated.

2.1 Drag acceleration.

The direction of the drag force is in the direction opposite to that of the instantaneous velocity. Its magnitude is, for the four standard projectiles,

$$(\rho/C_i)K_{Di}(M)v^2 \qquad i = 1,2,3,4 \tag{1}$$

In equation (1) $K_{Di}(M)$ is a function of Mach number characteristic of the projectile, ρ is the density of the air mass, and v is the projectile air speed. The ballistic coefficient C_i is an empirical constant approximately equal to ratio m/d^2 but generally also a function of firing charge.* Examination of the four $K_{Di}(M)$ functions, which are illustrated in Appendix B (actually the ratios K_{Di}/C_i are plotted for nominal values of C_i), shows that the functions are

*BRL data for the standard projectiles indicate that for the 155mm, 175mm, and 8 inch shells C_i is a function of charge; C_i is also a function of quadrant elevation for the 155mm projectile.

approximately proportional, i.e. the ratio of any two functions, or drag curves, is a constant nearly independent of Mach number. It is this characteristic of the functions K_{Di} that has simplified the problem of estimating the drag effects for an unknown projectile. It permits the selection of a function $\bar{K}_D(M)$, which will be devoted as the universal drag function, that is independent of projectile. The universal curve has the property that the ratio

$$(K_{Di}(M)/C_i)/\bar{K}_D(M) \qquad i = 1,2,3,4 \tag{2}$$

is nearly constant with Mach number, and can therefore be approximated by a constant

$$c_i = (K_{Di}(M)/C_i)/\bar{K}_D(M) \; . \tag{3}$$

Substituting equation 3 into equation (1), we have that the drag acceleration for the standard projectiles is approximately

$$\rho \; c_i\bar{K}_D(M)v^2 \tag{4}$$

The assumption mentioned above that the four standard missiles can be imbedded in the family of all projectiles means, with respect to zero-yaw drag acceleration, that for any projectile the drag acceleration will be representable by

$$c\bar{K}_D(M)v^2 \tag{5}$$

where the value of c is to be estimated from the observed projectile motion. The function $\bar{K}_D(M)$ is selected as described in 2.3 below.

2.2 Drift acceleration. It can be seen from Appendix A that the vector acceleration due to drift is

$$-(a_iK_i(M)N/v^2)(\underset{\sim}{v} \times \dot{\underset{\sim}{u}}) \tag{6}$$

for the i^{th} standard projectile, with $K_i(M)$ a function of Mach number, N the spin angular velocity, $\underset{\sim}{v}$ the vector of projectile velocity relative to the ground. As indicated in the appendix, the coefficient a_i is a function of several physical parameters characteristic of the projectile and the so called "lift factor" ℓ, a scale factor function of the firing charge.

The spin angular velocity is governed by the expression

$$\dot{N} = -b_i \rho K_{Ai}(M) v\, N \tag{7}$$

where b_i is again a projectile characteristic constant and $K_{Ai}(M)$ another function of Mach number. Evaluation of the constants b_i in (7), using data not shown in Reference [1] but given in BRL data associated with the computer programs for the modified point-mass computations, shows that the values for the four standard projectiles are 0.0255, 0.0193, 0.0166, and 0.0157 ft^2/pound. Similar evaluation of the functions $K_{Ai}(M)$ shows that the same function is given by BRL for three of the four projectiles; its value is 0.007 (dimensionless) at $M = 0$ and it falls smoothly to about half the value at $M = 2.5$. For the 8-inch projectile, the value is given as 0.005, constant with Mach number.

Clearly the situation is less favorable for (7) than for the drag acceleration, with regard to embedding the four standard projectiles in some general functional form with small relative errors of approximation. Fortunately the drift effects are smaller than the drag effects to start with, and larger relative errors are tolerable. In the simplified-dynamics equations it is assumed that

$$\dot{N} = -0.020 \rho \bar{K}_A(M) vN \tag{8}$$

is a tolerable universal approximation to (7), with $\bar{K}_A(M)$ taken as the function $K_{Ai}(M)$ for the 105mm, 155mm, and 175mm projectiles.

The expression (6) may be written

$$-(a_i/c_i)(c_i K_i(M)\, N/v^2)(\underline{v} \times \dot{\underline{u}}) \qquad i = 1,2,3,4 \tag{9}$$

where, as explained in 2.3 below, a function $\bar{K}(M)$ and four constants c_i are used (these c_i's are unrelated to the c_i's of (3) and (4); they merely serve an analogous purpose) such that for each $i = 1,2,3,4$, $\bar{K}(M)$ approximates $c_i K_i(M)$ as closely as possible. Again, the relative accuracy will be poor by comparison with the relative accuracies achieved for the drag approximations, but as will be seen in 2.4 below the use of the drift terms in the simplified-dynamics equations gives results considerably better than does ignoring the drift.

The drift acceleration can now be taken as

$$-(a_i/c_i)\bar{K}(M)(N/v^2)(\underset{\sim}{v} \times \dot{\underset{\sim}{u}}) \qquad i = 1,2.3,4 \tag{10}$$

where a_i/c_i is a single parameter and $\bar{K}(M)$ a universal function, and where N is given by (8). As in the passage from (4) to (5), we could imbed the values of the parameter

$$r_i = a_i/c_i \tag{11}$$

in a family of parameter values, and for the case of an unknown projectile estimate the value of r from observed drift acceleration effects assumed to be governed by an expression

$$-r\bar{K}(M)(N/v^2)(\underset{\sim}{v} \times \dot{\underset{\sim}{u}}) \tag{12}$$

together with (8), in which only r needs to be estimated. However, although (8) through (12) are useful for exhibiting the underlying reasoning, it is more convenient in the computations to define a new variable s(t) to be estimated, as part of the state vector, than to estimate the parameter r.

Let

$$s(t) = rN(t) = aN(t)/c \tag{13}$$

by definition. Then (8) becomes

$$\dot{s} = -0.020\rho\bar{K}_A(M)vs \tag{14}$$

and the drift acceleration (12) becomes

$$-(s/v^2)\bar{K}(M)(\underset{\sim}{v} \times \dot{\underset{\sim}{u}}) \tag{15}$$

Equation (14) and expression (15) in the simplified-dynamics equations replace (7) and (6) of the modified point-mass equations.

2.3 Choice of universal functions. In passing from (9) to (10), it is necessary to find a function $\bar{K}(M)$ and four constants c_i, $i = 1,2,3,4$, so that each of four given functions $K_i(M)$ is approximated by $c_i\bar{K}(M)$. Below is described the basis for choosing the universal function $\bar{K}(M)$ and the four constants c_i. The same problem occurs in passing from (3) to (4), with $C_iK_{Di}(M)$ given instead of $K_i(M)$, $i = 1,2,3,4$.

One way to achieve the desired result is to minimize

$$S = \sum_i \int_{M_a}^{M_b} w_i(M)\{c_i\bar{K}(M) - K_i(M)\}^2 dM \tag{16}$$

where (M_a,M_b) is the range of interest of M and $w_i(M)$ is any set of weighting functions that permit assigning importance to certain i, to certain subranges of M, or both. In what follows, all integrals are over (M_a,M_b) and all sums are over i.

For a given set $(c_1,\ldots,c_4)$, the $\bar{K}(M)$ that minimizes (16) must satisfy

$$S(\bar{K}(M) + \delta\bar{K}(M)) = S(\bar{K}(M)) \tag{17}$$

for an arbitrary small variation $\delta K(\bar{M})$, by the reasoning basic to the calculus of variations. To first-order effects, (17) implies

$$\int\Sigma\{w_i(M)(c_i\delta\bar{K}(M))(c_i\bar{K}(M) - K_i(M)) \, dM = 0 \tag{18}$$

which can hold for arbitrary small $\delta\bar{K}(M)$ only if

$$\Sigma c_i w_i(M)(c_i\bar{K}(M) - K_i(M)) = 0 \qquad (19)$$

or

$$\bar{K}(M) = \Sigma c_i w_i(M)K_i(M)/\Sigma c_i^2 w_i(M) \qquad (20)$$

Or if $\bar{K}(M)$ is fixed, the c's that minimize must satisfy $\partial S/\partial c_i = 0$, $i = 1,2,3,4$, or

$$2\int w_i(M)\bar{K}(M)(c_i\bar{K}(M) - K_i(M))dM = 0 \qquad i = 1,2,3,4 \qquad (21)$$

from which

$$c_i = \int w_i(M)\bar{K}(M)K_i(M)dM/\int w_i(M)(\bar{K}(M))^2 dM \qquad i = 1,2,3,4 \qquad (22)$$

By applying (20) and (22) alternately, since each application reduces S, one can find a succession of c_i's and functions $\bar{K}(M)$ until successive changes in S are as small as desired. However there is no unique solution to the problem as stated above, because for any real $k \neq 0$, multiplication of each c by k and division of $\bar{K}(M)$ by k leaves S unchanged. It is convenient to make the solution unique. This is done by using the following computation procedure:

(a) Take an arbitrary f(M) to begin with -- for example $K_1(M)$.

(b) Use (22), with f(M) in place of $\bar{K}(M)$, to compute a set of c's.

(c) Normalize the c's by dividing each by their mean, so that the normalized c's will satisfy $\Sigma c_i = 4$.

(d) Using the normalized c's, use (20) to compute $\bar{K}(M)$.

(e) Use (22), and normalize as in (c), to compute a new set of c's.

(f) Repeat (d) and (e) until the changes in an iteration are satisfactorily small.

The above procedure was used with the $w_i(M) = 1$ for all M and i in order to determine the universal drag function $\bar{K}_D(M)$ and the corresponding four constants c_i. The nominal ballistic coefficients for the 155mm, 175mm, and 8 inch projectiles and the correct single value for the 105mm projectile given in Appendix B were assumed for the computation. It should be noted that the above choice of weighting functions although an obvious one, is quite arbitrary. Other choices could have been made, possibly in order to shape the universal curve to minimize drag computation errors in the transonic region for specific expected target projectiles.

Only one cycle of the procedure was used for determining the universal drift function $\bar{K}(M)$ and the associated constants r_i. Justified by our assumption that large relative inaccuracies are acceptable in drift, $w_i(M)$ was taken to be the Dirac delta function $\delta(M-M_o)$ with $M_o = 0.8$ for all i. As was previously indicated, the computations were scaled by the nominal lift coefficients of Appendix B.

2.4 Test of the simplified-dynamics equations. One limiting factor in the performance of the system studied by Project RATRAN is the amoung of error incurred in using the simplified-dynamics equations for the backward integration along the trajectory. To measure the amount of this error, it is intended to use the modified point-mass equations to compute a trajectory up to some point (x,y,z,t), and then to use the simplified-dynamics equations for computing backward, as these equations will be used in practice for computing the estimated launch point. The difference in x and z coordinates between the point on the backward trajectory at which y = 0 and the starting point for the forward trajectory computation is the error due to the use of the simplified-dynamics equations, plus numerical errors in computation; the effects of the latter are estimated by an auxiliary computation in which the modified point-mass equations are used in the backward computation as well as in the forward computation.

In the backward computation, the correct parameter value c_i is used in the drag computations and the correct starting value is used for s(t), the drift variable. In both cases the constants were properly scaled to account for the differences between the nominal drag and lift coefficients of Appendix B and true values for the particular projectile cases considered. There will be errors, in practice, in the estimates of c_i and s(t), but these errors are not chargeable to use of the simplified dynamics equations.

The forward-backward computation test on the simplified-dynamics equations has been performed for several trajectories; results are shown in Table 1. Characteristics of the test trajectories used are shown in Figure 1.

ERRORS IN BACKTRACKING TO LAUNCH POINT

GUN TYPE	INITIAL VELOCITY (M/SEC)	QUADRANT ELEVATION (MILS)	TIME AT REVERSAL (SEC)	HEIGHT AT REVERSAL (M)	DRIFT AT REVERSAL (M)	BRL EQUATIONS		SIMPLIFIED DYNAMICS INCLUDING DRAG AND DRIFT		SIMPLIFIED DYNAMICS INCLUDING DRAG ONLY	
						ERROR IN X(M)	ERROR IN Z(M)	ERROR IN X(M)	ERROR IN Z(M)	ERROR IN X(M)	ERROR IN
175	914.4	860	10	5400.488	14.967	0.066	-0.170	1.059	-3.184	1.636	-12.80
			20	8888.957	57.935	0.131	-0.564	11.418	-10.571	14.419	-58.52
			30	11064.484	128.932	0.671	-1.075	40.234	-17.616	48.826	-147.72
			40	12139.016	229.533	1.432	-1.628	92.829	-15.644	111.479	-284.0
155	563.9	601	10	2134.808	14.105	-0.053	-0.223	-3.274	-0.136	-0.456	-16.59
			20	2883.896	55.764	0.182	-0.489	-18.177	-3.737	-22.057	-68.88
			30	2634.535	125.134	-0.271	-0.828	36.317	-3.533	48.561	-154.89
			40	1466.620	217.963	2.326	-1.207	166.655	-2.210	183.672	-255.40
105	302.0	600	10	1112.902	8.807	0.026	-0.046	2.200	0.137	2.620	-8.95
			20	1190.328	37.579	0.054	-0.165	13.040	1.122	15.169	-42.15
			30	329.441	84.445	-0.058	-0.358	16.561	1.777	20.891	-91.52
105	465.0	477	10	1237.613	9.785	0.064	-0.107	-16.109	1.075	-14.588	-10.83
			20	1335.064	40.980	0.470	-0.312	-21.103	0.607	-17.620	-50.85
			30	497.609	93.142	0.594	-0.642	-57.007	-0.860	-50.439	-115.58
155	463.3	468	10	1272.378	13.602	-0.019	-0.158	-16.778	-0.926	-16.776	-14.54
			20	1405.705	52.675	0.217	-0.348	19.402	-2.400	23.480	-57.16
			30	591.172	114.741	0.727	-0.659	98.879	-1.886	106.314	-122.82
155	268.2	390	10	483.199	10.315	0.036	-0.044	4.777	0.570	5.543	-10.36

TABLE 1 BACKTRACKING ERRORS FOR A SELECTION OF TRAJECTORIES

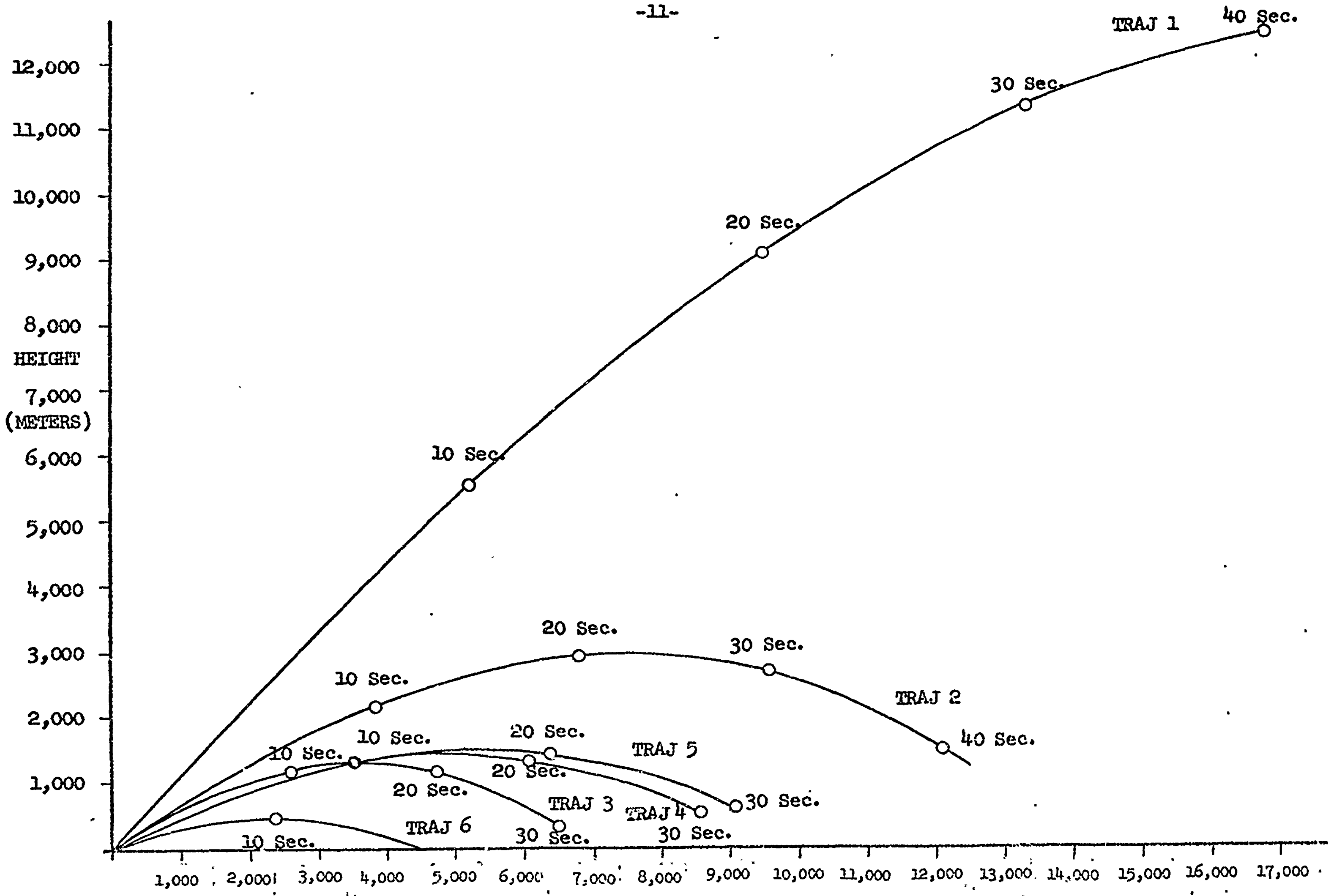


FIGURE 1 Characteristics of Test Trajectories

The table summarizes the results of three kinds of computations. In all three, the BRL modified point-mass program was used to compute trajectories to some preselected time, the "time at reversal". From the state parameters at the time of reversal, three backtracking computations were made, proceeding backward in time to zero elevation. The three differ as follows. In the case of most interest, the backward computation used the simplified dynamics equations; the results are given in the columns headed "simplified dynamics including drag and drift". Magnus effects were not included in the backward computations. In order to show what would happen if no drift effects are included, drift force was set equal to zero in another set of backward computations, reported under "simplified dynamics including drag only". Finally, to estimate errors due to numerical round-off, a third set was made using the complete BRL equations; the results are shown under "BRL equations".

The tables show that, for the trajectories tried, (a) the effects of computational errors are negligible, (b) drift is the principal source of error if only drag effects are included in the backward computations, and (c) drift error effects can be reduced to amounts small by comparison with drag error effects, insofar as errors in the simplified-dynamics equations are concerned. (Whether the filter can estimate the drift-variable well enough to take advantage of the situation remains to be seen).

For the trajectories so far tested, the errors due to the use of the simplified dynamics equations are below 25 meters in estimated launch point for all trajectories in which the radar observations can be made within 20 second of firing.

3.0 IRREDUCIBLE LAUNCH-POINT ESTIMATION ERRORS DUE TO RANDOM RADAR ERRORS.

3.1 Purpose. On the assumption that the simplified-dynamics equations represent the true behavior of the projectile, this section describes how to compute the covariance matrix of the error in launch-point estimation for least squares estimators. The simplified-dynamics equations are defined, and the errors incurred in their use discussed, in section 2. For the purposes of this section, it is enough to say that they involve a drag variable and a drift parameter to be estimated.

It is assumed here that the only source of error is the random radar errors. Bias radar errors will be investigated separately. The errors in launch-point estimation here studied will depend only on the covariance matrix of the radar errors, the trajectory-radar geometry, and the schedule of radar observations. They are sampling errors, in the sense that if enough independent radar observations could be made, the errors in launch-point estimation due to the radar errors could be made as small as desired.

This section, then, tells how to compute the performance of the best possible (least-squares) signal-processing algorithms in coping with random radar errors, in deducing what needs to be known about the projectile and its trajectory. Later, when the performance of individual algorithms (e.g., Kalman filters) is investigated, the results of the computations described below will provide limits on the performance of the algorithms. The limits could be reached if the covariance matrix of the radar errors is known and if enough radar observations could be made and if computation time and computer memory were not limited. Comparison between the limiting performance here considered and actual algorithm performance will be used in guiding decisions on whether to accept performance of a particular algorithm or to seek improvement at the cost of computer capacity.

3.2 Outline of method. The following is well-known: If there are N observed column-vectors (radar observations)

$$\tilde{\underline{r}}_k = \underline{r}_k + \delta\underline{r}_k \qquad k = 1,\ldots,N \qquad (23)$$

where $\underline{r}_k$ is a true value and each $\delta\underline{r}$ is a zero-mean normally-distributed error vector independent of the others, and if there is to be estimated a vector $\underline{v}$ (augmented initial state vector) that satisfies

$$\underline{r}_k = A_k\underline{v} \qquad k = 1,\ldots,N \qquad (24)$$

for given matrices A_k, then the maximum-likelihood estimator for $\underline{v}$ is

$$\hat{\underline{v}} = S^I \sum_k A_k^T R_k^I \tilde{\underline{r}}_k . \tag{25}$$

where the superscripts I and T denote respectively the inverse and transpose operations,

$$S = \sum_{k=1}^{n} A_k^T R_k^I A_k, \tag{26}$$

and

$$R_k = E\{\delta \underline{r}_k \, \delta \underline{r}_k^T\} \tag{27}$$

is the covariance matrix for the errors in $\underline{r}_k$.

Under the assumption that the errors $\delta\underline{r}_k$, and the errors $\delta\hat{\underline{v}}$ that they cause in $\hat{\underline{v}}$, are small, the RATRAN launch point estimation problem can be identified with the linear problem just described as follows. The vector $\underline{v}$ has eight components: the coordinates in gun-axes x_o, y_o, and $z_o = 0$ and the velocity components $\dot{x}_o$, $\dot{y}_o$, and $\dot{z}_o$ at the time t_o of launch; a constant parameter c characteristic of the projectile; and a drift parameter ζ_o, the initial value of a variable used in the simplified-dynamics equations for drift computations. For notational convenience these eight numbers are denoted also by $v_1, \ldots, v_8$ respectively. The vector $\underline{r}_k$ of radar observations at time t_k has either three or four components, depending on whether doppler is not or is used, the other three being the range and two angles. (Four different radar types will be considered: range-α-β and range-azimuth-elevation, each with and without doppler.) Then

$$A_k = (a_{ij}^{(k)}) \tag{28}$$

where

$$a_{ij}^{(k)} = \partial r_i^{(k)} / \partial v_j, \tag{29}$$

the change in the ith radar coordinate at time t_k due to small unit change in the jth component of the initial augmented state vector $\underline{v}$.

Equation (25) implies that

$$E\{\delta\hat{\underline{v}}\ \delta\hat{\underline{v}}^T\} = S^I \tag{30}$$

is the covariance matrix for the error

$$\delta\hat{\underline{v}} = \hat{\underline{v}} - \underline{v} \tag{31}$$

in the maximum-likelihood estimator $\hat{\underline{v}}$. By the assumptions of normality and linearity, $\hat{\underline{v}}$ is also the least-squares estimator.

Since the purpose of RATRAN is not to estimate $\underline{v}$ at a particular value t = 0 of time, with respect to the radar observation times, but rather at a particular value z = 0 of projectile altitude -- and in general the two conditions will not concur -- $\hat{\underline{v}}$ is adjusted by

$$\begin{aligned} \hat{x}_L &= \hat{x}_o - \hat{z}_o\ \dot{x}_o/\dot{z}_o \\ \hat{y}_L &= \hat{y}_o - \hat{z}_o\ \dot{y}_o/\dot{z}_o \end{aligned} \tag{32}$$

or

$$\begin{pmatrix} \hat{x}_L \\ \hat{y}_L \end{pmatrix} = L\hat{v} \tag{33}$$

with

$$L = \begin{pmatrix} 1 & -\dot{x}_o/\dot{y}_o & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\dot{z}_o/\dot{y}_o & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{34}$$

Because the adjustment (32), is small, the true value of L may be used, and for convenience will be used, in the computations here described, in place of the estimated value that would have to be used in the actual radar system.

The covariance matrix for the error launch-point estimate, using least-squares estimation, is therefore

$$P = \begin{pmatrix} E\delta\hat{x}_L^{\,2} & E\delta\hat{x}_L\delta\hat{y}_L \\ E\delta\hat{x}_L\delta\hat{y}_L & E\delta\hat{y}_L^{\,2} \end{pmatrix} = E(L\,\underline{\hat{v}}\,\underline{\hat{v}}^T\,L^T) = LS^I\,L^T \qquad (35)$$

The computation of P is the subject of the remainder of this section.

3.3 Inputs to the computations. Each computation of P is for the following inputs: A projectile trajectory, requiring specification of the projectile and its initial velocity vector; the position of the gun and its direction of fire relative to the radar; the choices with regard to the radar -- namely the use of doppler, the choice of angular coordinates, the radar altitude, and (only if range-α-β is used) the antenna array tilt angle; the covariance matrices R_k for the random radar errors; and the set of observation times t_k.

The projectile trajectories will be computed using the BRL modified point-mass computer program as adapted for the simplified-dynamics equations. The drag parameter $v_7 = c$ and the initial value $v_8 = \zeta_o$ of the drift variable must be specified. The gun coordinates z_g, x_g relative to the radar (y is vertical and x forward, as explained in Appendix D), and the azimuth firing angle A2 in the rectilinear radar-axes system must be given; these numbers are specified as though the earth were flat, but the trajectory and the radar coordinates are computed for a spherical earth. The covariance matrices R_k will be specified either as constant or as a function of radar coordinates, never as time functions as might have been inferred from the definition relative to the observation times t_k. The computer program will provide for the possibility that the elements R_k are to be computed from the radar coordinates at the time t_k. The matrices R_k will be 4-by-4, if doppler is used; if doppler is not used, the computations are simplified as discussed in 3.5.3 below. The choice of observation times t_k that the program will be able to accomodate is discussed in 3.5.1.

3.4 Values for radar error covariances. Inputs to this analysis program representing the variances of the random radar estimation errors can be described as follows. Logical system operation will consist of a sequence of independent "looks" at each target, each look consisting of a burst of radar pulses during which range, two angles, and range rate may be estimated. Aside from bias errors, which will be handled separately in this study, the different measurement errors can be considered to be independent random variables with zero mean values being effectively independent also from look to look. The following formulas will be applicable to this end (see for example Skolnik [2], Chapter 10).

$$\sigma_R^2 = \frac{C^2}{4\gamma_B^2(2E/N_o)} + \sigma_{R_C}^2 \qquad \text{Range error variance} \tag{36}$$

$$\sigma_\theta^2 = \frac{\lambda^2}{\gamma_A^2(2E/N_o)} + \sigma_{\theta_C}^2 \qquad \text{Angle error variance} \tag{37}$$

$$\sigma_{\dot{R}}^2 = \frac{\lambda^2}{4\gamma_T^2(2E/N_o)} + \sigma_{\dot{R}_C}^2 \qquad \text{Range rate error variance} \tag{38}$$

In the above expressions, $2E/N_o$ is effectively signal-to-noise ratio determined by transmitted power, antenna gain and target range and cross section; λ is the wavelength of the transmitted signal; C is the velocity of light; γ_B is the effective bandwidth of the transmitted signal; γ_A is the effective aperture dimension of the radar antenna; γ_T is the effective duration of the coherent signal used for doppler estimation; $\sigma_{R_C}^2$, $\sigma_{\theta_C}^2$, $\sigma_{\dot{R}_C}^2$ are the non-range-dependent components of measurement error variances, due to (unavoidable) system imperfections.

A convenient simplification is to make a preliminary set of calculations with a nominal system at a nominal range R_o, then make use of the fact that $2E/N_o$ varies inversely as range to the fourth power so that

$$2E/N_o = (2E/N_o)_o \ (R_o/R)^4 \tag{39}$$

Then equations (36), (37), and (38) simplify as follows:

$$\sigma_R^2 = \sigma_{R_o}^2 (R/R_o)^4 + \sigma_{R_C}^2 \tag{40}$$

$$\sigma_\theta^2 = \sigma_{\theta_o}^2 (R/R_o)^4 + \sigma_{\theta_C}^2 \tag{41}$$

$$\sigma_{\dot{R}}^2 = \sigma_{\dot{R}_o}^2 (R/R_o)^4 + \sigma_{\dot{R}_C}^2 \tag{42}$$

where $\sigma_{R_o}^2$, $\sigma_{\theta_o}^2$ and $\sigma_{\dot{R}_C}^2$ are variances of errors due to thermal noise, calculated at the nominal range R_o.

The above quantities are then used as the main-diagonal elements of the matrix R_k referred to in section 3.3 above; all off-diagonal elements being zero.

3.5 Details of the computations. Let k be arbitrary but fixed, so that it need not be exhibited in the notation. Then (28) and (29) may be expressed, using notation common in control theory, by

$$A^T = \partial \underline{r}^T / \partial \underline{v}. \tag{43}$$

Let $\underline{u} = \underline{u}_k = \underline{u}(t_k)$ be the vector whose elements are the values at time t_k of the projectile coordinates x_k, y_k, z_k and their time-derivatives in the gun-axes system.

$$\underline{u}_k^T = (x_k \; y_k \; z_k \; \dot{x}_k \; \dot{y}_k \; \dot{z}_k) \tag{44}$$

Clearly $\underline{u}$ is determined by $\underline{v}$, for given t_k, and $\underline{r}$ is determined by $\underline{u}$, and so

$$\partial \underline{r}^T / \partial \underline{v} = (\partial \underline{u}^T / \partial \underline{v})(\partial \underline{r}^T / d\underline{u}) \tag{45}$$

or $A = CB$ where

$$B^T = \partial \underline{u}^T / \partial \underline{v}, \quad C^T = \partial \underline{r}^T / \partial \underline{u} \tag{46}$$

in the notation of (43).

The 6-by-8 matrix B is independent of the type or location of the radar, expressing only the relation between the initial augmented state vector of the projectile and a later state vector. The elements of B will be computed and stored. The computation of the elements b_{ij} of B require nine executions of the program for computing trajectories. One of the nine is called the reference trajectory. The reference trajectory begins with the nominal initial conditions. In each of the other eight, one of the eight components $v_1, \ldots, v_8$ of the initial augmented state vector is perturbed

$$v_j \to v_j + \delta v_j \qquad j = 1, \ldots, 8 \tag{47}$$

For each k, the values of

$$b_{ij} = b_{ij}(t_k) \qquad i = 1, \ldots, 6$$

are given by

$$b_{ij} = (1/\delta v_j)(\text{perturbed } u_1 - \text{reference } u_1) \tag{48}$$

with the right-hand side evaluated at time t_k. The perturbations δv_j are input constants, to be determined by preliminary trials not defined here in detail, their purpose being to determine values δv_j for which the differences between perturbed and unperturbed u-values are above the round-off error noise level, but are small enough for (48) to approximate a partial derivative. Forty-eight b's will be computed and stored for each k.

In addition to the 48 matrix elements b_{ij}, there will also be stored the six vector elements $u(t_k)$ of (44), as computed for the reference trajectory, for use in computing the elements

$$c_{ij}^{(k)} = c_{ij}(t_k), \qquad k = 1,\ldots,N \tag{49}$$

of C_k, as explained below. Each trajectory thus requires storing 54N numbers.

The matrices C_k will not be stored between runs. They will be computed as needed, from input constants and the values of the vectors $\underline{u}(t_k) = \underline{u}_k$. The first step in the computation of each C_k is the computation of the projectile position vector

$$\underline{w}_k = \begin{pmatrix} x_{rk} \\ y_{rk} \\ z_{rk} \end{pmatrix} = M \begin{pmatrix} x(t_k) \\ y(t_k) \\ z(t_k) \end{pmatrix} + \underline{d} \tag{50}$$

in a right-handed rectilinear coordinate system with origin at the radar.* In (50), the radar coordinate system has the y-axis along the vertical at the radar, positive upward, and the x-axis along the nominal forward (zero-azimuth) radar direction. The values of $x(t_k)$, $y(t_k)$ and $z(t_k)$, three of the six elements of the previously-stored vector $\underline{u}_k$, are the components of the projectile position vector in the right-handed rectilinear system that the trajectory computations use. This system has its origin at the gun and its y-axis positive upward along the local vertical, and its xy-plane contains the initial-velocity vector. M is the 3-by-3 matrix of the rotation needed to make the gun and radar axes parallel, and $\underline{d}$ is the position vector of the gun in the rectilinear radar axes system. M and $\underline{d}$ are computed in terms of input constants as follows:

$$M = (m_{ij}), \qquad i,j = 1,2,3 \tag{51}$$

$$\underline{d}^T = (d_1\ d_2\ d_3) \tag{52}$$

$$g = (z_g^{\,2} + x_g^{\,2})^{1/2} \tag{53}$$

*See Appendix D for a simplified discussion of the coordinate conversion used.

$$C3 = -x_g/g \tag{54}$$

$$S3 = -z_g/g \tag{55}$$

$$C1 = (S3)\ \sin A2 + (C3)\ \cos A2 \tag{56}$$

$$S1 = -(S3)\ \cos A2 + (C3)\ \sin A2 \tag{57}$$

$$S4 = g/R, \tag{58}$$

where R is the radius of the earth.

$$C4 = 1 - (1/2)(S4)^2 \tag{59}$$

$$m_{11} = (C1)(C3)(C4) - (S1)(S3) \tag{60}$$

$$m_{12} = -\ (C3)(S4) \tag{61}$$

$$m_{13} = -\ (S1)(C3)(C4) - (C1)(S3) \tag{62}$$

$$m_{21} = (C1)(S4) \tag{63}$$

$$m_{22} = C4 \tag{64}$$

$$m_{23} = -\ (S1)(S4) \tag{65}$$

$$m_{31} = (C1)(S3)(C4) + (S1)(C3) \tag{66}$$

$$m_{32} = -\ (S3)(S4) \tag{67}$$

$$m_{33} = -\ (S1)(S3)(C4) + (C1)(C3) \tag{68}$$

$$d_1 = x_g \tag{69}$$

$$d_2 = -g(S4)/2 - h_r \tag{70}$$

$$d_3 = z_g \tag{71}$$

In (71), h_r is the altitude of the radar (in excess of the altitude of the gun).

The matrix M, with components given by (60) through (68), is also used to compute

$$\dot{\underline{w}}_k = \begin{pmatrix} \dot{x}_{rk} \\ \dot{y}_{rk} \\ \dot{z}_{rk} \end{pmatrix} = M \begin{pmatrix} \dot{x}(t_k) \\ \dot{y}(t_k) \\ \dot{z}(t_k) \end{pmatrix} \tag{72}$$

The vectors $\underline{w}_k$ and $\dot{\underline{w}}_k$ are used to compute the radar range at time t_k

$$r_k = \left(\underline{w}_k^T\, w_k\right)^{1/2} \tag{73}$$

the direction cosines

$$C5 = x_{rk}/r_k \tag{74}$$

$$C6 = y_{rk}/r_k \tag{75}$$

$$C7 = z_{rk}/r_k\ , \tag{76}$$

the range rate

$$\dot{r}_k = (C5)\dot{x}_{rk} + (C6)\dot{y}_{rk} + (C7)\dot{z}_{rk}\ , \tag{77}$$

and the ratios

$$C8 = \dot{x}_{rk}/\dot{r}_k \tag{78}$$

$$C9 = \dot{y}_{rk}/\dot{r}_k \tag{79}$$

$$C10 = \dot{z}_{rk}/\dot{r}_k \tag{80}$$

By definition (see figs. 1 and 2)

$$\alpha = \text{arc cos}\{(C5)\cos\gamma + (C6)\sin\gamma\} \tag{81a}$$

where γ is the antenna array tilt angle.

$$\beta = \text{arc cos}(C7) \tag{82a}$$

The purpose of (50) through (82a) is the computation of $c_{ij}(t_k)$, $i = 1,\ldots,4$; $j = 1,\ldots,6$; $k = 1,\ldots,N$, where c_{ij} is the partial derivative of the ith radar coordinate with respect to the jth component of $\underline{u}(t_k)$ as defined by (44). If the first element of $\underline{r}$ is taken to be the range r and the fourth to be the range rate $\dot{r}$, in both the range-α-β case and the range-azimuth-elevation case, then the first and fourth rows of C will be the same in the two cases. They are given by

$$C_{11} = C5 \tag{83}$$

$$C_{12} = C6 \tag{84}$$

$$C_{13} = C7 \tag{85}$$

$$C_{14} = C_{15} = C_{16} = 0 \tag{86}$$

$$C_{41} = C_{42} = C_{43} = 0 \tag{87}$$

$$C_{44} = C8 \tag{88}$$

$$C_{45} = C9 \tag{89}$$

$$C_{46} = C10 \tag{90}$$

In the range-α-β case,

$$\underline{r}_k^T = (r_k\ \alpha_k\ \beta_k\ \dot{r}_k). \tag{91a}$$

Equations (81a) and (82a) give α and β. Performing the partial differentiations with respect to the u_j's gives

$$C_{21} = \{-((C6)^2 + (C7)^2)\cos\gamma + (C5)(C6)\sin\gamma\}/DENOM \tag{92a}$$

$$C_{22} = \{(C5)(C6)\cos\gamma - ((C5)^2 + (C7)^2)\sin\gamma\}/DENOM \tag{93a}$$

$$C_{23} = (C7)((C5)\cos\gamma + (C6)\sin\gamma)/DENOM \tag{94a}$$

$$C_{24} = C_{25} = C_{26} = 0 \tag{95}$$

where

$$DENOM = \{(x_{rk}\sin\gamma - y_{rk}\cos\gamma)^2 + z_{rk}^{\;2}\}^{1/2} \tag{96}$$

$$C_{31} = (C5)(C7)/RHO \tag{97a}$$

$$C_{32} = (C6)(C7)/RHO \tag{98a}$$

$$C_{33} = -1/RHO \tag{99a}$$

$$C_{34} = C_{35} = C_{36} = 0 \tag{100}$$

where

$$RHO = \{\, x_{rk}^{\;2} + y_{rk}^{\;2}\,\}^{1/2} \tag{101}$$

In the range-azimuth-elevation case,

$$\underline{r}_k^T = (r_k \; AZ_k \; EL_k \; \dot{r}_k) \tag{91b}$$

where

$$AZ_k = \text{arc tan } (z_{rk}/x_{rk}) \tag{81b}$$

$$EL_k = \text{arc sin } (y_{rk}/r_k) \tag{82b}$$

Performing the partial differentiations gives

$$C_{21} = -z_{rk}/(x_{rk}^2 + z_{rk}^2) \tag{92b}$$

$$C_{22} = 0 \tag{93b}$$

$$C_{23} = x_{rk}/(x_{rk}^2 + z_{rk}^2) \tag{94b}$$

$$C_{24} = C_{25} = C_{26} = 0 \tag{95}$$

$$C_{31} = -x_{rk} \, y_{rk}/r_k \tag{97b}$$

$$C_{32} = 1/r_k \tag{98b}$$

$$C_{33} = -y_{rk} \, z_{rk}/r_k(x_{rk}^2 + z_{rk}^2) \tag{99b}$$

$$C_{34} = C_{35} = C_{36} = 0 \tag{100}$$

3.5.1 The observation times t_k. As has been said, there will be first computed and stored 48 elements of B_k and six of $\underline{u}_k$ for time $t_1,\ldots,t_k,\ldots,t_n$. The matrix

$$S = \sum_{k=1}^{n} A_k^T R_k^I A_k \tag{26}$$

is approximated, for evenly-spaced

$$t_k = t_1 + (k-1)\tau \, , \tag{102}$$

by

$$S \approx A_1^T R_1^I A_1/2 + (1/\tau) \int_{t_1}^{t_n} A^T R^I A \, dt + A_n^T R_n^I A_n/2, \tag{103}$$

where the integral has been equated to its trapezoidal-sum approximation. Equation (103) shows that the number n of the observations need not be carefully selected, because for given end-points t_1 and t_n, (103) can be used over a range of intermediate spacings τ, as long as they are small enough to warrant the integral-sum approximation. Since there is no reason to expect rapid changes in the product $A^T(t)R^I(t)A(t)$, it seems safe to take the values of t_k perhaps 1/2 second apart, for the purpose of storing the elements of B_k and $\underline{u}_k$, and to use the stored values for computing the integral in (103), after which (103) may be used with arbitrary τ to compute S for other observation rates, by means of

$$S(\tau) = A_1^T R_1^I A_1/2 + A_n^T R_n^I A_n/2$$
$$+ (\tau_o/\tau)\{S(\tau_o) - A_1^T R_1^I A_1/2 - A_n^T R_n^I A_n/2\} \tag{104}$$

where τ is the spacing of interest and τ_o is the spacing used in the stored values, so that $S(\tau_o)$ is given by (26). Of course the end-points t_1 and t_n must be the same for the two spacings τ_o and τ.

The values of t_1 and t_n will be selected after some preliminary investigation, for each trajectory, on the earliest time the projectile will be visible to the radar and the latest time for which the accumulated errors in the simplified-dynamics equations are of tolerable magnitudes.

3.5.2 Inversion of the matrix S. From (34) and (35) it is clear that only the first three rows and columns of S^I are of interest. If the 8-by-8 matrix S is partitioned

$$S = \begin{pmatrix} S_1 & S_2 \\ S_2^T & S_3 \end{pmatrix} \tag{105}$$

where S_1 is 3-by-3, S_2 is 3-by-5, and S_3 is 5-by-5, its inverse has in its upper-left-hand corner the matrix

$$\tilde{S}_1 = \left(S_1 - S_2\, S_3^I\, S_2^T\right)^I . \tag{106}$$

The computation of (106) requires a 5-by-5 inversion and a 3-by-3 inversion, in place of an 8-by-8 inversion for the entire matrix S. The use of (106) is preferable. Then

$$P = \tilde{L}\, \tilde{S}_1\, \tilde{L}^T \tag{107}$$

replaces (35), with

$$\tilde{L} = \begin{pmatrix} 1 & -\dot{x}_o/\dot{y}_o & 0 \\ 0 & -\dot{z}_o/\dot{y}_o & 1 \end{pmatrix} \tag{108}$$

3.5.3 Effect of not using doppler. If doppler is not used, the computations are considerably simplified. The matrices have the following dimensions:

	With doppler		Without doppler	
Matrix	rows	columns	rows	columns
S, S^I	8	8	8	8
A	4	8	3	8
R, R^I	4	4	3	3
B	6	8	3	8
C	4	6	3	3

In particular, only the first three rows of B need be computed and stored, if it is known that doppler will not be used. It is intended to use the full system, as described above for the case in which doppler is used, at least for the first few trajectories studied, to assess roughly the value of using doppler. If doppler is found not to be of great value, it is intended to omit it from most of the computations for economy, and perhaps later to restore it for studying its effects on the best filters found. If doppler is omitted, so are equations (77), (78), (79), (80), (87), (88), (89), and (90), and $i = 4,5,6$ in (41).

3.5.4 Summary of the computations. The sequence of computations is as follows. First, preliminary investigations are needed to establish the values of δv_j, $j = 1,\ldots,8$, in (48) and of $t_1,\ldots,t_n$ for each trajectory.

The values of $\underline{u}_k$, (44), are computed for the reference trajectory and the matrices B_k, defined in (46) and computed by (48), are determined and stored.

The matrix M and the vector $\underline{d}$ are computed, by (51) through (71), for the gun-radar geometry. M and $\underline{d}$ are used in (72) through (80), together with the stored values of $\underline{u}_k$, to determine the projectile state-vector in a rectilinear, radar-centered coördinate system.

Then (81) through (100), with the a-version of equations (81), (82), (91), (92), (93), (96), (97), and (98), give the matrices C_k for the range-α-β radar coordinate system. The same equations, using the b-version, gives the matrices C_k for the range-azimuth-elevation coordinate system. In both cases, $A_k = C_k B_k$ gives the matrics A_k needed in (26).

Then for the spacing $\tau_o = t_{k+1} - t_k$ of radar observations, (26) gives the matrix S; (104) is used for any other evenly-spaced radar observations. Finally (105) through (108) give the covariance matrix P of the least squares launch-point estimator.

4.0 FILTER-SMOOTHER DESIGN

4.1 Introduction. This section will be devoted to a discussion of the nonlinear filtering/smoothing algorithm for estimating instantaneous state variables that will be implemented by the University in the next quarter. The first subsection contains a short mathematical description of the algorithms to be employed. The second will describe the basic problem formulation. This will include a discussion of the equations of motion, state variables, and covariance matrices. Finally the third subsection will describe implementation concepts and future plans.

4.2 Estimation Equations. The nonlinear filter/smoother specifies estimates of a given past state and the current n-dimensional state of a dynamical system observed discretely in the presence of additive Gaussian white noise. The system is assumed to be described by the stochastic vector differential equation

$$dx_t = f(x_t, t)dt + g(x_t, t)d\beta_t \tag{109}$$

where β_t represents a Brownian motion process with covariance matrix

$$E\{d\beta_t, d\beta_t^T\} = Q_t \, dt \tag{110}$$

The corrupted m-dimensional ($m \le n$) measurement vector z_ℓ is related to the state by the expression

$$z_\ell = h(x_\ell) + v_\ell \tag{111}$$

where

$$x_\ell = x_t, \; t = t_\ell \, . \tag{112}$$

and

$$E\{v_\ell, v_k^T\} = R_\ell \, \delta_{\ell k} \, . \tag{113}$$

An a priori estimate of the state $\tilde{x}_o$ with covariance $\mathrm{cov}(x_o, x_o)$ is also assumed given.

The algorithm is composed of a set of discretized extrapolation equations and a set of estimate update equations. The extrapolation equations describe the behavior of the current state estimate $\hat{x}_\ell$ and the associated covariance matrices between observation samples. The update equations determine improved estimates of the current state and the prescribed previous state $x_k (k < \ell)$. These estimates and modifications to the covariance matrices reflect the new information available in z_ℓ.

We shall denote the best estimate of state x_k based on a realization of measurements $Z_\ell = \{z_1, \ldots, z_\ell\}$ as $\hat{x}_{k|\ell}$. The conditional covariance will be denoted by $\mathrm{cov}(x_k, x_k | Z_\ell)$.

The extrapolation equations for

$$t_\ell = t_{\ell-1} + \Delta t$$

are given as follows

$$\hat{x}_{\ell|\ell-1} = \hat{x}_{\ell-1|\ell-1} + \Delta t \, f(\hat{x}_{\ell-1|\ell-1}, t_{\ell-1}) + \frac{\Delta t^2}{2} F_{\ell-1|\ell-1} f(\hat{x}_{\ell-1|\ell-1}, t_{\ell-1}) \quad (114)$$

$$\mathrm{cov}(x_\ell, x_\ell | Z_{\ell-1}) = \Phi_{\ell|\ell-1} \mathrm{cov}(x_{\ell-1}, x_{\ell-1} | Z_{\ell-1}) \Phi^T_{\ell|\ell-1} + g(\hat{x}_{\ell-1|\ell-1}, t_{\ell-1}) Q'_{\ell-1}$$

$$g^T(\hat{x}_{\ell-1|\ell-1}, t_{\ell-1}) \quad (115)$$

$$\mathrm{cov}(x_k, x_\ell | Z_{\ell-1}) = \mathrm{cov}(x_k, x_{\ell-1} | Z_{\ell-1}) \Phi^T_{\ell|\ell-1} \quad (116)$$

$F_{\ell-1|\ell-1}$ is the nxm Jacobian matrix of the $f(\cdot\,, t)$ vector function with elements

$$\left[F_{\ell-1|\ell-1}\right]_{i,j} = \left.\frac{\partial f_i(x,t)}{\partial x_j}\right|_{x = \hat{x}_{\ell-1|\ell-1}} \quad (117)$$

$$\Phi_{\ell|\ell-1} = I + \Delta t\ F_{\ell-1|\ell-1} \tag{118}$$

$$Q'_{\ell-1} = \Delta t\ Q(t_{\ell-1}) \tag{119}$$

where I is the nXn identity matrix.

Defining the gain term

$$S_\ell = H_\ell^T\ (H_\ell\ \mathrm{cov}(x_\ell, x_\ell|Z_{\ell-1})H_\ell^T + R_\ell)^{-1} \tag{120}$$

$$\left[H_\ell\right]_{i,j} = \frac{\partial\ h_i(x)}{\partial\ x_j}\Bigg|_{x = \hat{x}_{\ell|\ell-1}}\ , \tag{121}$$

We have for the update equations

$$\hat{x}_{\ell|\ell} = \hat{x}_{\ell|\ell-1} + \mathrm{cov}(x_\ell, x_\ell|Z_{\ell-1})S_\ell(z_\ell - h(\hat{x}_{\ell|\ell-1})) \tag{122}$$

$$\hat{x}_{k|\ell} = \hat{x}_{k|\ell-1} + \mathrm{cov}(x_k, x_\ell|Z_{\ell-1})S_\ell(z_\ell - h(\hat{x}_{\ell|\ell-1})) \tag{123}$$

$$\mathrm{cov}(x_\ell, x_\ell|Z_\ell) = \mathrm{cov}(x_\ell, x_\ell|Z_{\ell-1})(I - S_\ell\ H_\ell\ \mathrm{cov}(x_\ell, x_\ell|Z_{\ell-1})) \tag{124}$$

$$\mathrm{cov}(x_k, x_\ell|Z_\ell) = \mathrm{cov}(x_k, x_\ell|Z_{\ell-1})(I - S_\ell\ H_\ell\ \mathrm{cov}(x_\ell, x_\ell|Z_{\ell-1})) \tag{125}$$

An expression for updating the approximate covariance for the smoothed estimate also exists. The expression is however, not required for implementing equation 123. Its utility is restricted to that of evaluating the quality of the estimate. Since algorithm evaluation will be an important part of the project, we include the equation

$$\text{cov}(x_k,x_k|Z_\ell) = \text{cov}(x_k,x_k|Z_{\ell-1}) - \text{cov}(x_k,x_\ell|Z_{\ell-1})S_\ell\ H_\ell\ \text{cov}^T(x_k,x_\ell|Z_{\ell-1}). \quad (126)$$

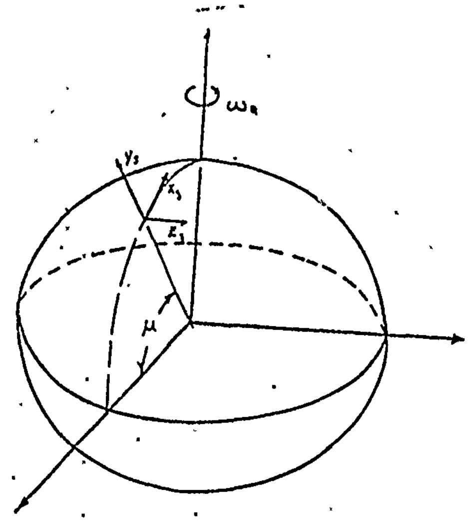
The algorithm is initiated by the given a priori information

$$\hat{x}_{o|o} = \bar{x}_o \quad (127)$$

and

$$\text{cov}(x_o,x_o|Z_o) = \text{cov}(x_o,x_o) \quad (128)$$

4.3 <u>Equations of Motion</u>. We assume the radar to be located at an altitude h_o and at laditude μ on a rotating spherical earth as illustrated in Figure 2. A moving



x_g - North

y_g - Local vertical

z_g - East

Fig 2 Radar Location

Cartesian coordinate system x_g, y_g, z_g will be defined centered at this site such that the x_g axis points north and the y_g axis points up along the local verticle.

Figure 3 illustrates a second coordinate system x_r, y_r, z_r

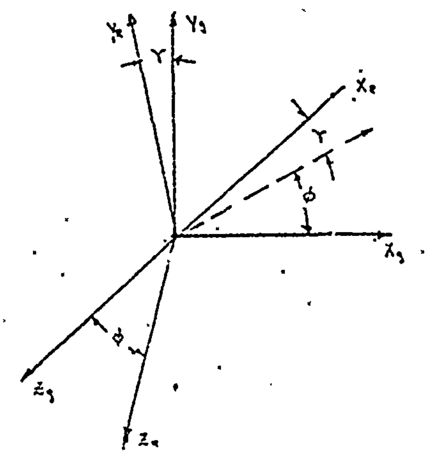


Fig 3 Radar Centered Coordinate Systems

obtained from the first by a rotation $-\Phi$ in azimuth and γ in elevation. In this system the x_r axis is normal to the radar face pointing outwards and the y_r and z_r axes are in the plane of the face.

The simplified dynamic equations of motion in the x_r, y_r, z_r coordinate system are

$$\begin{bmatrix} \ddot{x}_r \\ \ddot{y}_r \\ \ddot{z}_r \end{bmatrix} = \underbrace{-\rho\ C\ \bar{K}_D(\bar{M})v \begin{bmatrix} v_{xr} \\ v_{yr} \\ v_{zr} \end{bmatrix}}_{\text{drag}} \underbrace{-\frac{S\ \ddot{K}(M)}{v^2} \begin{bmatrix} v_{yr}\ g_z - v_{zr}\ g_y \\ v_{zr}\ g_x - v_{xr}\ g_z \\ v_{xr}\ g_y - v_{yr}\ g_x \end{bmatrix}}_{\text{drift}} + \underbrace{2\omega_R U \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{z}_r \end{bmatrix}}_{\text{coriolis}} + \underbrace{\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}}_{\text{gravity}} \qquad (129)$$

and

$$\dot{S} = -.004\ \rho\bar{K}_A\,(M)VS. \tag{130}$$

where

$$h = h_o + x_r \sin\gamma + y_r \cos\gamma + \frac{(x_r^{\,2}\cos^2\gamma + y_r^{\,2}\sin^2\gamma + z_r^{\,2})}{2(R_e + h_o)}$$

R_e = Radius of the earth

$$g_o = -9.80665\ [1-.0026\ \cos 2\mu]\ (R_e/(R_e + h_o))^2$$

$$g_x = g_o[\sin\gamma + \{x_r(1 - 3\sin^2\gamma) + y_r(-3\sin\gamma\cos\gamma)\}/(R_e + h_o)]$$

$$g_y = g_o[\cos\gamma + \{x_r(-3\sin\gamma\cos\gamma) + y_r(1 - 3\cos^2\gamma)\}/(R_e + h_o)]$$

$$g_z = g_o\, z_r/(R_e + h_o).$$

The skew symmetric corioulis matrix is defined by the upper off-diagonal terms

$$U_{ud} = \begin{matrix} 0 & \sin\Phi\ \cos\mu & \cos\Phi\sin\gamma\cos\mu - \cos\gamma\sin\mu \\ & 0 & \cos\Phi\cos\gamma\cos\mu + \sin\gamma\sin\mu \\ & & 0 \end{matrix} \tag{131}$$

and the projetile velocity relative to the air is given by the vector (of magnitude V):

$$V_{xr} = \dot{x}_r - W_s(\cos AZ \cos\gamma \cos\Phi + \sin AZ \cos\gamma \sin\Phi)$$

$$V_{yr} = \dot{y}_r - W_s(-\cos AZ \sin\gamma \cos\Phi - \sin AZ \sin\gamma \sin\Phi)$$

$$V_{zr} = \dot{z}_r - W_s(\cos AZ \sin\Phi - \sin AZ \cos\Phi) \tag{132}$$

where W_s is the magnitude of wind and AZ its azimuth direction (measured clockwise from north).

Atmospheric data are assumed available in the look-up format (as a function of h) prescribed for the BRL pointmass program. The universal aerodynamic curves are described by piecewise fourth order polynomials of mach number*

$$\begin{aligned} \bar{K}_D(M) &= \Sigma\, \bar{a}_i\, M^i \\ \bar{K}(M) &= \Sigma\, \bar{b}_i\, M^i \\ \bar{K}_A(M) &= \Sigma\, \bar{c}_i\, M^i \end{aligned} \tag{133}$$

4.4 State Variables. As indicated in the previous quarterly report, the unknown aerodynamic parameters will be treated as additional state variables to be identified by the estimation equations 122 and 123. Included in the simplified dynamics (eq. 129) are two such parameters C and S. The variable S is the product of two unknowns, an assumed locally constant drift parameter r similar to C and the projectile spin rate.

The state vector x_t is defined by the expressions

$$X_1 = x_r$$

$$X_2 = y_r$$

$$X_3 = z_r$$

---

* See Appendix B for plots of $\bar{K}_D(M)$ and $\bar{K}(M)$ as well as the coefficients a_i, b_i and c_i

$$
\begin{aligned}
X_4 &= \dot{y}_r \\
X_5 &= \dot{y}_r \\
X_6 &= \dot{z}_r \\
X_7 &= C \\
X_8 &= S
\end{aligned}
\tag{134}
$$

And from the equations of motion we obtain

$$
f(x) = \begin{bmatrix}
X_4 \\
X_5 \\
X_6 \\
D1V_x + D2(V_y g_z - V_z\ g_y) + 2\omega_R(U_{12}\ X_5 + U_{13}\ X_6) + g_x \\
D1V_y + D2(V_z g_x - V_x\ g_z) + 2\omega_R(U_{21}\ X_4 + U_{23}\ X_6) + g_y \\
D1V_z + D2(V_x g_y - V_y\ g_x) + 2\omega_R(U_{13}\ X_4 + U_{32}\ X_5) + g_z \\
0 \\
D3\ X_8
\end{bmatrix}
\tag{135}
$$

where

$$
\begin{aligned}
D1 &= -\rho\ X_7\ \bar{K}(M)V \\
D2 &= -\ X_8\ \bar{K}_D(M)/V^2 \\
D3 &= -\ .004\rho\ \bar{K}_A(M)V
\end{aligned}
\tag{136}
$$

The components of the Jacobian matrix F are given in Appendix E.

The x_r, y_r, z_r coordinate system was orginally selected because the phased array radar measurements are defined directly in this system as is illustrated in Fig. 4.

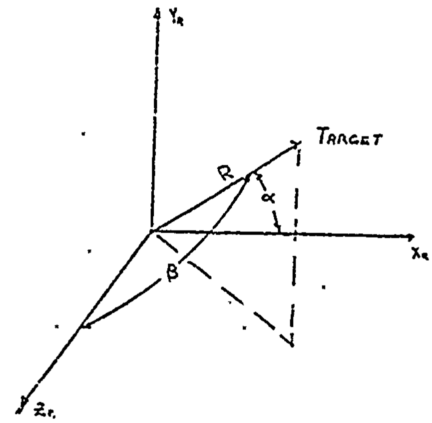


Fig 4 Observation Vector

In state variable notation these measurements (including optional doppler rate information) are

$$h(x) = \begin{bmatrix} R \\ \alpha \\ \beta \\ \dot{R} \end{bmatrix} = \begin{bmatrix} (X_1^2 + X_2^2 + X_3^2)^{1/2} \\ \cos^{-1}(X_1/R) \\ \cos^{-1}(X_3/R) \\ (X_1 X_4 + X_2 X_5 + X_3 X_8)/R \end{bmatrix} \tag{137}$$

Again the Jacobian matrix H_ℓ components are given in Appendix E.

4.5 Covariance Terms. In radar tracking applications the Brownian motion process β_t in equation 1 is used to represent random atmospheric effects and errors due to modelling approximations. Of the latter, the most significant is the assumption that C and r are constant functions of mach number. This is particularly true for the drift constant, however as illustrated in Fig 5 a constant C is also not an especially good approximation for $K_{Di}(M)/K_D(M)$ in the region about mach 1.

Although it has been shown that a best Q matrix for filter implementation must be determined experimentally, order of magnitude estimates for Q_{44} and Q_{55} may be determined as follows. We assume that at any time t for particular shell j C has been estimated such that

$$\hat{C} = K_{D_j}(M)/\bar{K}_D(M). \tag{138}$$

Then at time t+Δt due to the change in mach number (and thus the true value of the ratio of equation (138) there will be an error ϵ_{jD} introduced in $\hat{C}$ given by the expression

$$\epsilon_{jD} = \Delta t \frac{d}{dt} \left(K_{D_j}(M)/\overline{K_D}(M)\right) \tag{139}$$

$$= \frac{\Delta t \dot{V}}{Vs} \frac{\partial}{\partial M} \left[K_{D_j}(M)/\overline{K_D}(M)\right] \tag{140}$$

$$= \Delta t \frac{\dot{V}}{Vs \overline{K_D}(M)} \left[\frac{\partial}{\partial M} K_{D_j}(M) - C \frac{\partial \overline{K_D}(M)}{\partial M} \right] \tag{141}$$

Recalling the assumption

$$K_{D_1}^{(M)} = \sum_{0,4} a_i^j M^i \tag{142}$$

we obtain

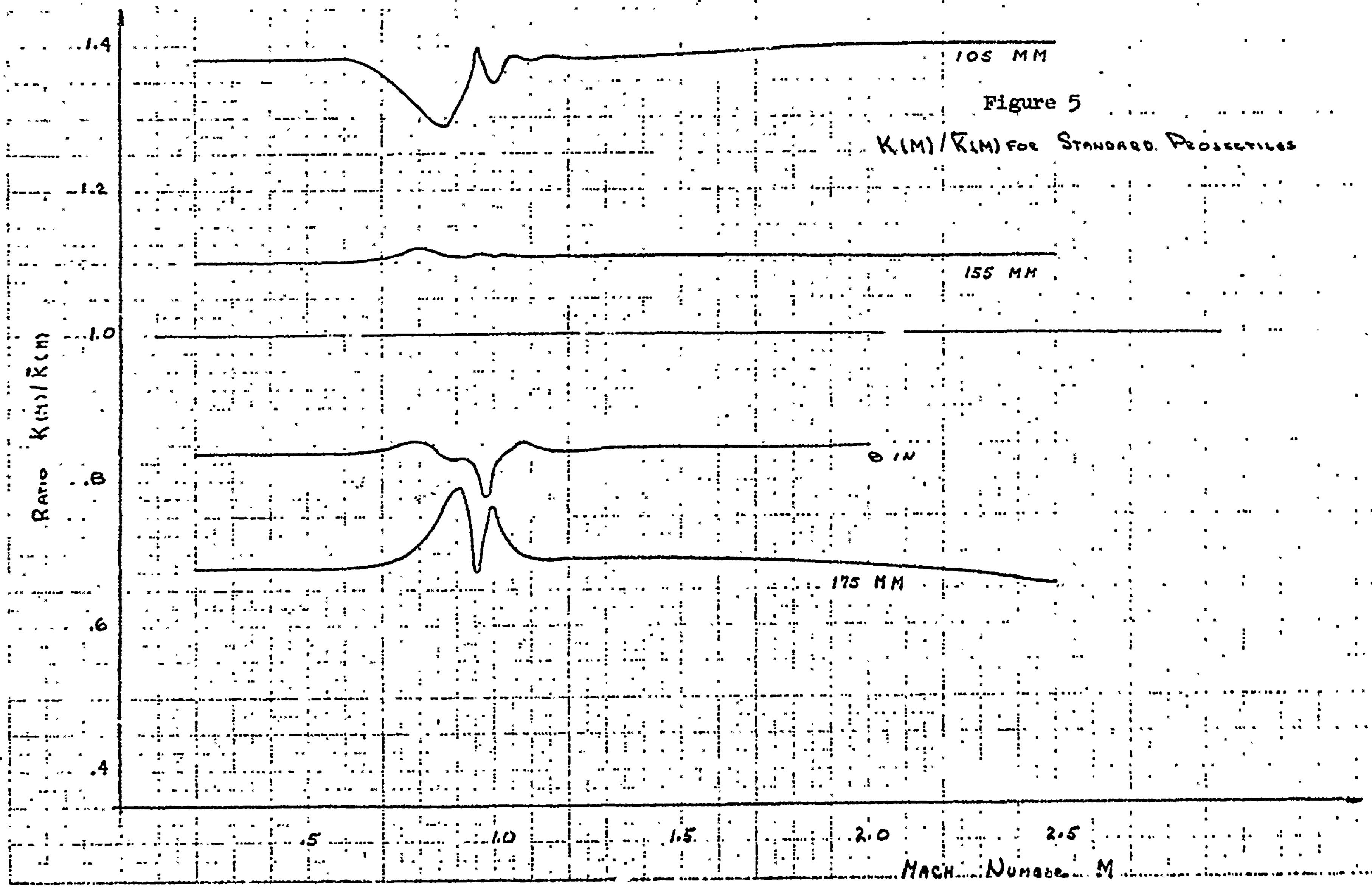


Figure 5

$K(M)/\bar{K}(M)$ for Standard Projectiles

$$\epsilon_{jD} = \Delta t \frac{\dot{V}}{V_s \, K_D(M)} \sum_{1,4} (a^j{}_i - D\bar{a}_i)_i \, M^{i-1} \tag{143}$$

where we approximate

$$\dot{V} = |\rho \, K_{D_j}(M) \, V \, \underline{V} + \underline{g}| \tag{144}$$

Rather obviously the development for the drift coefficient will be very similar to the above. By computing (for nominal ρ, V_s, and QE) error terms for drag and drift as a function of mach number for the given standard shells and then averaging the results; the following tentative $g(x)$ 8X5 and Q'5X5 matrices were determined

$$Q' = \Delta t^2 \begin{bmatrix} .0016 & & & & \\ & .0016 & & & \\ & & .0016 & & \\ & & & Q_{44} & \\ & & & & Q_{55} \end{bmatrix} \tag{145}$$

$$g(x) = \begin{bmatrix} 0 & & & & \\ 0 & & & & \\ 0 & & & & \\ 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & D3\,X_8 \end{bmatrix} \tag{146}$$

where

$$Q_{44} = \begin{cases} 10^{-5} & 1.14 < M \; ; \; M \le .7 \\ 10^{-(13.634 - 12.34M)} & .7 < M \le 1.01 \\ 10^{-(29.46M - 28.58)} & 1.01 < M \le 1.14 \end{cases}$$

$$Q_{55} = \begin{cases} .000196 & 1.10 < M \; ; \; M \leq .83 \\ 10^{-(13.382 - 11.666M)} & .83 < M \leq .96 \\ 10^{-(10M - 7.2} & .96 < M \leq 1.10 \end{cases} \qquad (147)$$

4.6 Implementation. The filter/smoother will be simulated using data generated by the programs described above in Section 2 of this report for a number of nominal projectile trajectories. The past state to be estimated by fixed point smoothing will be that which would correspond to X_o, the first point of the tracked portion of the trajectory. Experiments will be conducted as to the viability of using algorithm generated covariance matrices for error evaluation studies. Convergence properties of the algorithms relative to given Q matrices and higher order expansions (particularly h(x)) will be considered.

5.0 CONCLUSIONS AND PLANS FOR FUTURE WORK. As pointed out in Section 2 above, a set of simplified-dynamics equations has been obtained and programmed with corrections for both drag and drift. Tentative numerical results obtained so far indicate that these equations approximate rather well the effects of drift keeping the resultant lateral errors quite small. The drag term, however leads to longitudinal errors that are disappointingly large. This effect is most pronounced for cases where the point at which backtracking begins occurs after the projectile's velocity has crossed the mach 1 region. This appears to indicate that satisfactory results can only be obtained in such cases when it is possible to obtain valid tracks of the target while it is still going at speeds greater than mach 1. Work remaining to complete this effort must include a systematic investigation of the lateral and longitudinal errors to be expected in backtracking for a comprehensive set of trajectories. Also, additional thought will be applied to possible refinement of the drag correction so as to decrease the longitudinal errors to an acceptable value.

In Section 3 the formulation necessary to determine the magnitudes of irreducible errors due to radar observation noise has been developed and the necessary programming is underway. The purpose of this program is to determine the theoretical accuracies attainable in launch-point estimation, on the assumptions that the simplified-dynamics equations are correct. That is, it finds the minimal errors, using best possible filtering and smoothing without limits on the amount of computation required, that result from the existence of the radar noise. These minimal errors resulting from radar noise of course add to the errors introduced by use of the simplified-dynamics equations. This will include a determination of whether acceptable estimates can be made for the drift and drag parameters so as to permit effective use of the simplified-dynamics equations for backtracking.

The work in Section 4 constitutes a detailed formal procedure for performing the filtering and smoothing required to estimate necessary inputs for backtracking from a set of radar observations. Work has begun to create a computer program for the realization and evaluation of this algorithm.

6.0 REFERENCES

1. Lieske, R. F., Reiter, M. L., "Equations of Motion For A Modified Point Mass Trajectory," Ballistic Research Laboratories, Report No. 1314, March 1966.

2. Skolnik, M. I., "Introduction to Radar Systems", McGraw-Hill Book Co. Inc., 1962.

Appendix A

Modified Point-Mass Equations

The equations here described are taken from Reference [1]. The total instaneous vector acceleration of the projectile's center of mass with respect to the ground is

$$\dot{\underset{\rightarrow}{u}} = \text{(drag term)} + \text{(drift term)} + \text{(Magnus term)}$$
$$+ \text{(gravity term)} + \text{(Coriolis term)}.$$

The drag term is

$$-(\rho d^2/m)\left\{K_{D_o} + K_{D_\alpha}(Q\alpha_e)^2\right\} v \underset{\rightarrow}{v} ,$$

the drift term is

$$(\rho d^2/m)K_L v^2 \ell \underset{\rightarrow}{\alpha}_e ,$$

and the Magnus-effect term is

$$(\rho d^3/m)K_F NQ(\underset{\rightarrow}{\alpha}_e \times \underset{\rightarrow}{v})$$

where ρ is the air density,
d is the projectile diameter,
m is the projectile mass,
K_{D_o}, K_{D_α}, K_L, and K_F are functions of Mach number characteristic of the projectile,

Q is the projectile yaw drag factor,
ℓ is the projectile lift factor,
N is the spin angular velocity,
$\underset{\rightarrow}{v}$ is the vector of projectile velocity with respect to air,

and $\underset{\rightarrow}{\alpha}_e$ is an approximation for the angle of repose.

Auxiliary equations are used for computing N and $\underset{\rightarrow}{\alpha}_e$:

$$\dot{N} = (\rho d^4/A)K_A N v$$

where A is the projectile's axial moment of inertia and K_A is a function of Mach number.

$$\underset{\rightarrow}{\alpha}_e = (\alpha_b - \alpha_a)(\underset{\rightarrow}{v} \times \dot{\underset{\rightarrow}{u}}) - \alpha_b(\underset{\rightarrow}{v} \times \underset{\rightarrow}{g})$$

where $\underset{\rightarrow}{g}$ is the gravity vector,

$$\alpha_a = AK_L N/(\rho d^3 K_L\ K_M\ v^4 + \rho d^5\ K_F\ K_T\ N^2 v^2)$$

$$\alpha_b = mK_T N/(\rho d K_L\ K_M\ v^4 + \rho d^3\ K_F\ K_T\ N^2 v^2\ )$$

and K_M, K_T are functions of Mach number.

From data not given in Reference [1] but used in the programs that compute the modified point-mass equations for the 105mm, 155mm, 175mm and 8 inch projectiles, $K_T = 0$, giving at once the simplification

$$\underset{\rightarrow}{\alpha}_e = -(AN/\rho d^3 K_M v^4)(\underset{\rightarrow}{v} \times \dot{\underset{\rightarrow}{u}})$$

and the drift acceleration term becomes

$$-(An/mdK_M v^2)K_L \ell(\underset{\rightarrow}{v} \times \dot{\underset{\rightarrow}{u}}).$$

In 2.2, K_L/K_M is called $K_i(M)$ and $A\ell/md$ is called a_i.

Computation of the drag coefficients K_{Do} and $K_{D\alpha}(Qa\hat{e}^2)$ indicate that for $ae \leq 2^\circ$ the sum behaves to an order of magnitude as K_{Do} alone. Since simulations with the modified point mass model indicate that angles of $> 2^\circ$ occur only in the vicinity of apogee and then only for QE's $\geq$ 800 mils, it is assumed that the approximation

$$\dot{\underset{\rightarrow}{u}}\ \text{drag} = (-\rho/C)K_{Do}\ v\ \underset{\rightarrow}{v}$$

is always valid.

Appendix B

Universal Aerodynamic Functions

This appendix includes coefficients for and plots of the universal drag and drift functions employed in the estimation algorithms. For comparison, graphs of the nominal drag, $K_{Di}(M)/C_i$ and drift $a_i K_i(M)$ functions for the four standard projectile types employed in our analysis are also included. It can easily be seen that the universal drift function tends to be a poorer approximation than does the corresponding drag function.

The multiplying constants c_i and r_i determined with the universal curves are given along with their respective nominal ballistic coefficients c_i and lift factors ℓ_i in Table B.1. Computation of the appropriate c_i and r_i for any particular trajectory (charge and Q.E.) in the backtracking evaluation required the following scaling operations

$$c_{i_{true}} = c_{i_{nom}} (C_{i_{nom}} / C_{i_{true}})$$

and

$$r_{i_{true}} = r_{i_{nom}} (\ell_{i_{true}} / \ell_{i_{nom}})$$

where the c_i true and ℓ_1 true are available from BRL data.

Table B.1

Shell Diameter	Drag c_i	Nominal Ballistic Coef.	Drift r_i	Nominal Lift Factor
105 MM	1.38850	1.919	.661707	.863
155 MM	1.07657	2.331	1.12855	.963
175 MM	.69031	3.101	.7411013	1.009
8 IN	.84459	3.16	1.468419	.880

The coefficients for all three aerodynamic functions also given in this appendix as Table B.2 are for piecewise fourth order polynomials (of Mach number) fits to the respective curves. The particular coefficients are assumed valid between the previous break point mach number up to and including the value given in column six of the particular row of coefficient values. All values are given in the same format as is employed in the "aeropacks" of the BRL point mass program.

List of Figures

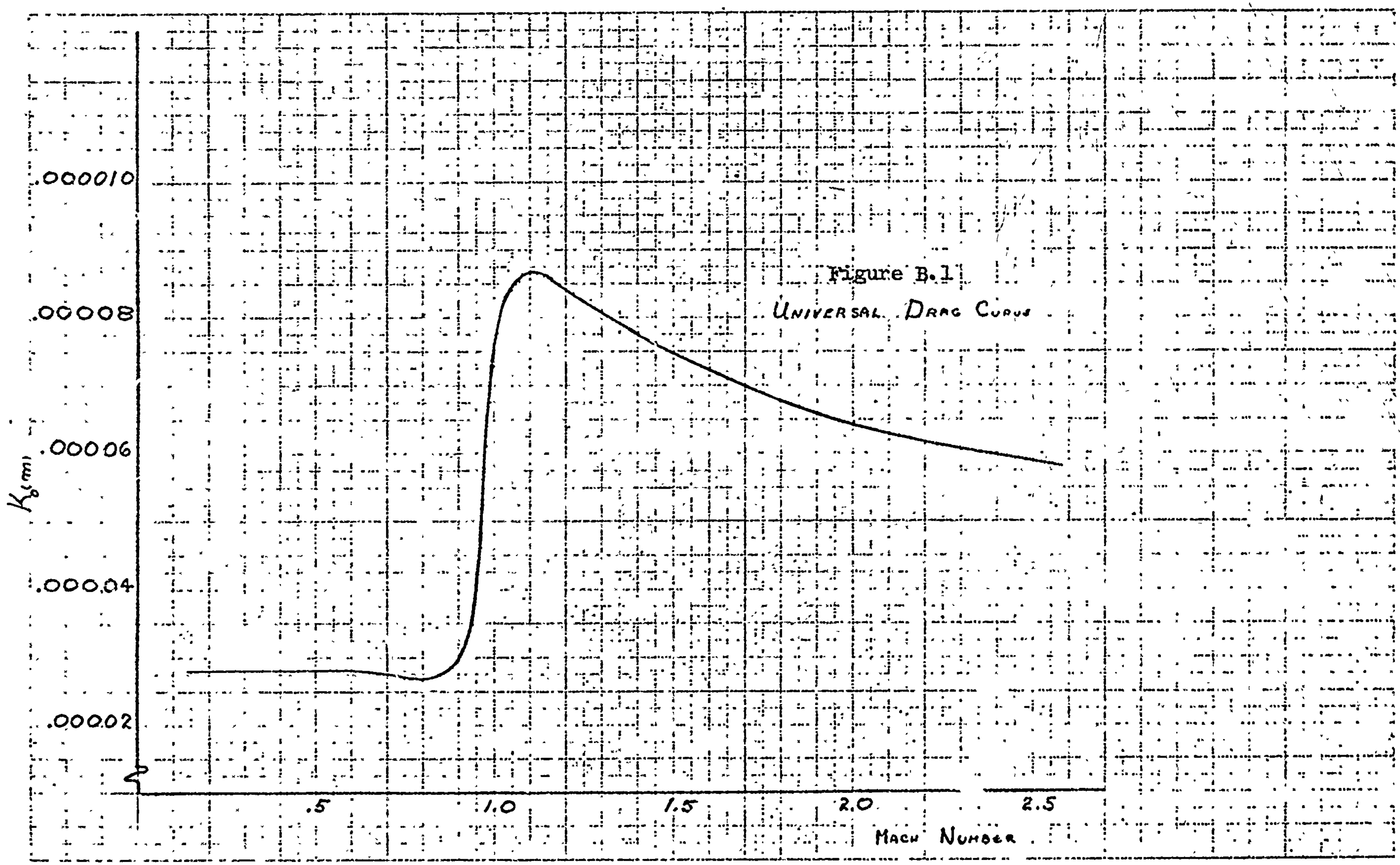


Figure B.1
UNIVERSAL DRAG CURVE

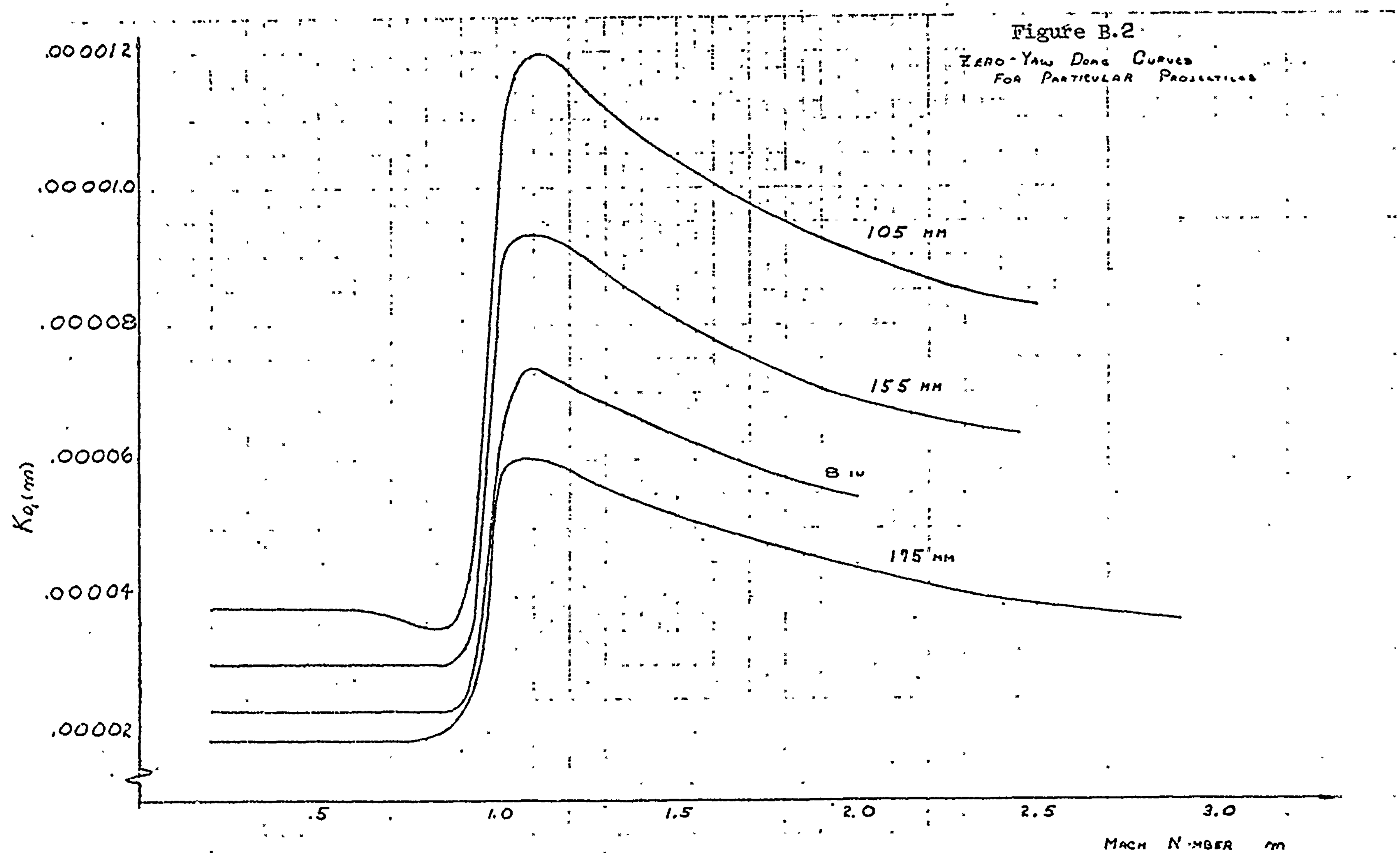


Figure B.2
ZERO-YAW DRAG CURVES
FOR PARTICULAR PROJECTILES
.000012
.00001.0
.00008
.00006
.00004
.00002
$K_{D_0}(m)$
105 MM
155 MM
8 IN
175 MM
.5
1.0
1.5
2.0
2.5
3.0
MACH NUMBER m

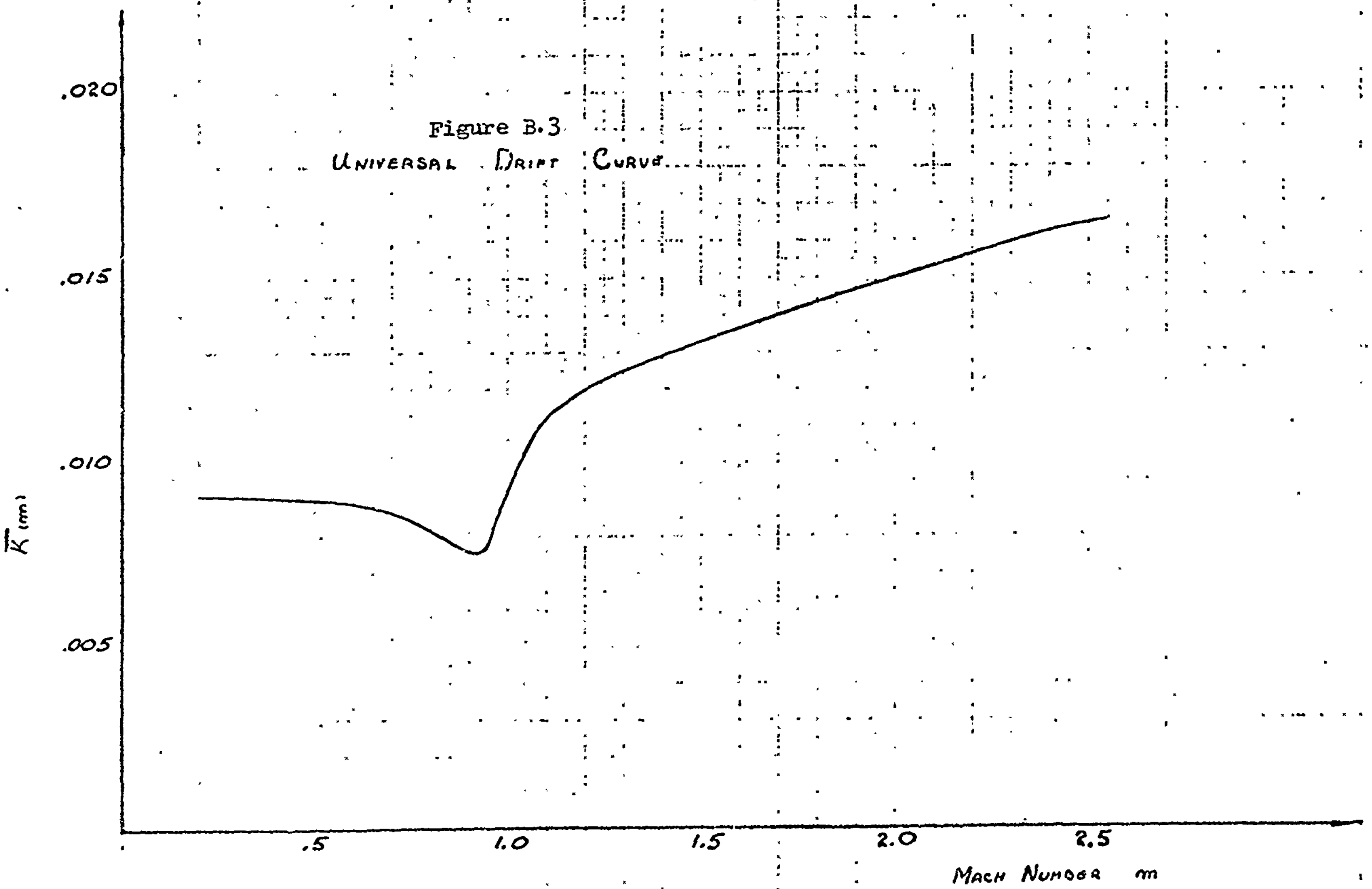


Figure B.3
Universal Drift Curve

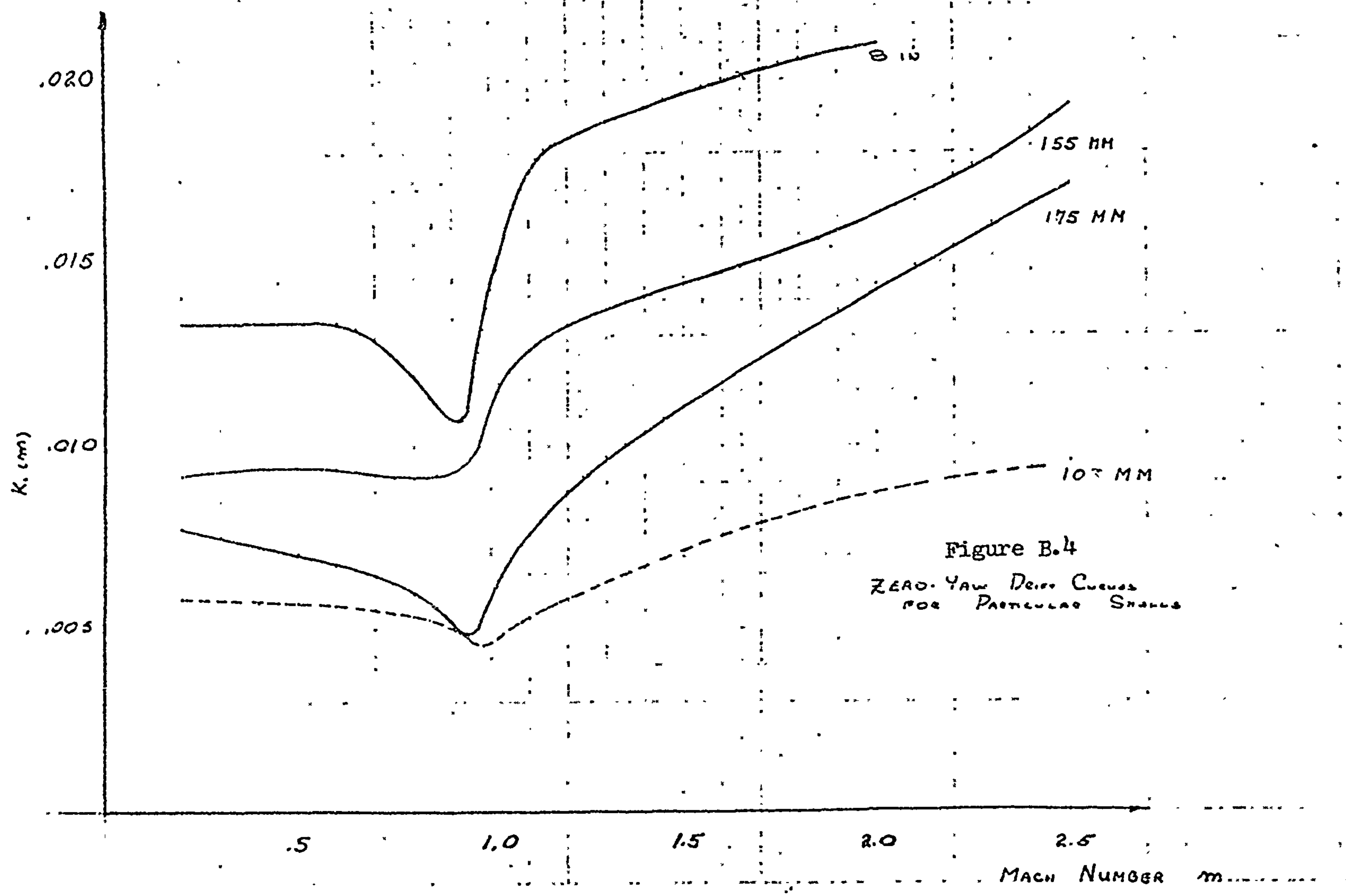


Figure B.4
Zero-Yaw Drift Curves for Particular Shells

UNIVERSAL DRAG CURVE

A(0)	A(1)	A(2)	A(3)	A(4)	UPPER MACH NO BREAK POINT
+27754000-04	00000000+ 0	00000000+ 0	00000000+ 0	00000000+ 0	+60000000+ 0
+22893676-03	-12863562-02	+30483832-02	-31694962-02	12178082-02	89000000 00
-28342963 00	12080492 01	-19264756 01	13620204 01	-36008687 00	97500000 00
78091593 02	-31639112 03	48068173 03	-32455698 03	82174843 02	10000000 01
-43060313 01	16931507 02	-24971760 02	16372837 02	-40264804 01	10400000 01
12834642 02	-48479677 02	68665363 02	-43221700 02	10201560 02	10800000 01
-42305762-02	13422046-01	-15539797-01	79504784-02	-15203564-02	14500000 01
17689915-03	-14035588-03	74594808-04	-21605570-04	26769062-05	40000000 01

UNIVERSAL DRIFT CURVE

A(0)	A(1)	A(2)	A(3)	A(4)	UPPER MACH NO BREAK POINT
94178026-02	-26634522-02	61690538-02	-32734184-02	-33347962-02	84000000 00
10546990 02	-48064499 02	82211585 02	-62499311 02	17816323 02	96500000 00
26615371 02	-10493487 03	15495878 03	-10156664 03	24936711 02	10700000 01
-48373662 00	14761935 01	-16536581 01	82453192 00	-15394547 00	15000000 01
15625846-01	-16050604-01	17887456-01	-69302356-02	96186882-03	40000000 01

UNIVERSAL SPIN CURVE

A(0)	A(1)	A(2)	A(3)	A(4)	UPPER MACH NO BREAK POINT
+70000000-02	-26504608-02	-90103102-03	+25286890-02	-11479416-02	+90000000+ 0
+67249870-02	-24994776-02	+71838136-03	-12021482-03	+80635505-05	+50000000+ 1

Table B.2 Universal Curve Coefficients

$$K(M) = \sum_{i=0,4} A(i)\, M^{i}$$

Appendix C

Computer Program Development

The starting point for this work was a FORTRAN program to compute modified-point-mass trajectories, produced at Ballistics Research Laboratory and furnished to Project RATRAN by USAECOM. In adapting this program to the Spectra-70 Computer at the Moore School several minor bugs were eliminated and several features were added to obtain desired results. A description of the main programs in use is as follows.

F-1. PTMASS - a module composed of 9 subroutines which computes the trajectory of a shell and the corresponding radar coordinates in either phased-array or range-azimuth-elevation form. This module includes the following subroutines:

A. PTMASS - the main calling program which calls all of the others and which reads in and stores the variable data - corresponding to a particular trajectory. This input data includes the necessary variables for computation of the radar coordinates.

B. AEROBL - a subroutine called by PTMASS which reads the aeroballistic data for a particular shell from a disk file, checks the data for errors, and stores it for later use in computation of the aerodynamic coefficients KDA, KDO, KLO, KM, KF, KA, and KLA.

C. TRAJ - a subroutine called by PTMASS which computes the velocity and position vectors of the trajectory at specific points in time, using a single-step predictor-corrector type of numerical integration, and which also computes the corresponding radar coordinates. It is possible to obtain either phased-array or range-azimuth-elevation radar coordinates by varying an input variable. This subroutine will compute the trajectory until one of the various "stop conditions" is satisfied. These stop conditions include termination of computation when time, the x-coordinate of the position vector, the range, the height of the up-leg, or the height of the down-leg have reached a given value, or at the summit of the trajectory. Other values printed out at each time interval include mach number, drag, lift, and magnus force.

D. COMPC - a subroutine called by PTMASS before TRAJ is called to compute the trajectory. COMPC computes the ballistic coefficient c as a function of the quadrant elevation.

E. COMPT - a subroutine called by PTMASS and TRAJ which computes the time of flight correction as a function of machine time.

F. COMPFS - a subroutine called by PTMASS which computes the fuze setting.

G. COMPL - a subroutine called by PTMASS before the trajectory is computed which computes the lift factor as a function of the quadrant elevation.

H. COMPNF - a subroutine called by PTMASS which computes the muzzle velocity correction factor N if the standard weight of the projectile is different from the actual weight.

I. COMPPT - a subroutine called by PTMASS which computes an adjustment to the initial velocity if propellant temperature is not 70°.

F-2. BACK.PTMASS - a module of 10 subroutines which computes the trajectory of a shell and has the option of stopping at any point in time and integrating backwards until height is zero using the simplified dynamics equations for drag and drift. This module includes many of the same routines as PTMASS - those that are different are explained in further detail below.

A. AGAIN - the main executive which calls all the others. AGAIN is similar to the routine PTMASS except that it does not have an insert allowing it to read radar data. It also differs from PTMASS in that it has a patch which calls AEROBL to read in the universal aeroballistic pack for use in backward integration.

B. AEROBL - same as before

C. AGAIN.TRAJ - a subroutine called by PTMASS to compute the trajectory. This routine is similar to TRAJ except that it does not compute corresponding radar coordinates at each point of the trajectory.

D. BTRAJ - a subroutine called by PTMASS which computes the trajectory backwards until height is zero using the simplified dynamics equations for drag and drift.

E. COMPC - same as before

F. COMPT - same as before

G. COMPFS - same as before

H. COMPL - same as before

I. COMPNF - same as before

J. COMPPT - same as before

F-3. AGAIN.PTMASS - a module of 10 subroutines which computes the trajectory of a shell until one of the various stop conditions is satisfied. This program has the option of stopping at any point in time and integrating backwards until height is zero using the standard BRL equations for drag, drift, and magnus force. This capability was designed into the module as a check on the backward computations of BACK.PTMASS with the simplified dynamics equations.

A. AGAIN - same as before

B. AEROBL - same as before

C. AGAIN.TRAJ - same as before

D. AGAIN.BTRAJ - a subroutine called by PTMASS which computes the trajectory backwards until height is zero using the standard equations for drag, drift, and magnus force.

E. COMPC - same as before

F. COMPT - same as before

G. COMPFS - same as before

H. COMPL - same as before

I. COMPNF - same as before

J. COMPPT - same as before

F-4. Considerations in implementing simplified drag and drift:

As originally designed, the program module AGAIN.PTMASS had the capability of stopping computation of a trajectory at any point in time [using a stop code of 4] and of integrating backwards until height was zero [using a stop code of 9]. It would have been easy to run the program with a stop code of 4 to some point in time and then re-run the program using the final values of the velocity and position vectors from the previous run as input to compute backwards with a stop code of 9. This approach, however, would have been wasteful of computer time and slower, as the entire program would have to be reloaded to compute backwards, and the input values for the second half of the run would have had to be written to a disk file, or punched into cards. Instead the module was modified so that upon reading a certain input card, the program automatically switches to a subroutine which integrates backwards using the simplified dynamics equations, taking the last values of the trajectory as a starting point.

The procedure is initiated by reading an A card in the input data. This signals the program to read in the universal aeroballistic pack for that particular shell type. Then the subroutine BTRAJ is called to perform the backward integration. The patch made to the program is shown below in lines 620-633.

```
      GOTO 110
C                    PROCEDURE FOR AN A CARD
C                    STORE NEW AEROBALLISTIC DATA
  112 IDSET=B(1)
C                    TEST READ NEW AEROPACK OR NOT
      IF(IDSET.EQ.32) GO TO 700
      CALL AEROBL(IDSET)
C                    PRINT HEADINGS
  700 IF(TEST.EQ.0) GO TO 701
      WRITE(6,406)
      GO TO 702
  701 WRITE(6,405)
  702 WRITE(6,450)
C                    COMPUTE THE BACKWARD TRAJECTORY WITH SIMPLIFIED EQUAT
      CALL BTRAJ(1,METRO)
      GO TO (43,113),IER
  113 GOTO(95,109,108),IERR
```

The subroutine BTRAJ was created from the rou+ TRAJ which computes trajectories under a variety of stop conditions A patch was made just after the entry point to the routine which sets all the necessary variables to enable backward integration to take place (stop code = 9). It was necessary to remove all statements which initialize the values of time, position, and acceleration for the final values from the forward run were to be used. Since these variables were all placed in the COMMON area of storage, the final values were automatically passed in when the subroutine was entered. The patch that was added to set the variables is shown below in lines 44-52 of the listing:

```
      DATA TOL/.01/
      METR=METRO
      GOTO(1000,2),JTR
C                    FIRST CALL PROCEDURE
C                    SET TIME TO FINAL VALUE FROM FORWARD RUN
 1000 TI=T
C                    SET STOP CODE TO 9
      ISTOP=9
C                    SET PRINT INTERVAL TO -1.0
      PINT=-1.
C                    SET FINAL VALUE TO ZERO
      FV=0.
      RECW=2.204622622/WT
```

Other changes made to the program to implement simplified drag and drift include a section to set up the constants used in the actual computation of drag and drift, as explained in another section of this report, and replacement of the equations to compute the forces of drag, drift, and magnus force with the new equations as shown below:

```
      ALPHA=SQRT(ALPHS)
C                    FIND COMPONENTS OF UDOT--THE ACCELERATION
C                    D1 IS DRAG
C                    D2 IS DRIFT
C                    D3 IS MAGNUS FORCE
  710 D1=-RHO*V*KDO*DDD
      D2 = NR*KLO/VSQ
      D3=0.
```

Appendix D

Coordinate Conversions

Project RATRAN has been using a computer program written by BRL, for generating projectile trajectories by means of the modified point-mass equations. In order to prepare for simulating the signal-processing filters to be designed, there have been added instructions to the BRL program that compute the center-of-mass projectile coordinates in radar axes, concurrently with the computations by the original program of the position in gun-axes. This appendix describes the radar-axes computations.

The gun-axes in the original program are a right-handed rectilinear system with origin at the gun, the y-axis positive upward along the local vertical, and the initial velocity vector in the x-y plane. The radar coordinates of the center of mass of the projectile are computed by rigid translations and rotations of coordinate systems until the projectile position is expressed in a right-handed rectilinear system with the origin at the radar, the y-axis positive upward along the local vertical, and the x-axis horizontal along the nominal zero-azimuth line of the radar. Then the rectilinear coordinates are transformed to range-α-β radar coordinates and also to range-azimuth-elevation. The earth is assumed to be a sphere of radius R. The radar elevation is not assumed to be the same as the gun elevation.

Three input constants, x_g, y_g, and A2, define the gun-axis system with respect to the radar coordinates. These three constants are most easily explained on the basis of a flat earth. A2 is measured clockwise looking down, from a line through the gun parallel to the radar x-axis, as shown in figure D-1. The position of the gun in the radar rectilinear coordinate system is at $z = x_g$, $x = y_g$. (The choice of the symbols x_g, y_g here and in the program annotations is inconsistent with the choice of axes, for historical reasons arising in the fact that information from BRL, including computer programs, uses a vertical y-axis and information from ECOM, in particular the gun location, uses a horizontal y-axis.)

The first coordinate transformation is a rigid rotation about the vertical, to make the gun x-y plane pass through the radar. The equations are

$$(x_g^2 + y_g^2)^{1/2}$$

$$C1 = (-x_g \sin A2 - y_g \cos A2)/d$$

$$S1 = (x_g \cos A2 - y_g \sin A2)/d$$

$$x_1 = x_o\,C1 - z_o\,S1$$

$$z_1 = x_o\,S1 + z_o\,C1$$

where x_o, y_o, z_o are the coordinates of the projectile center of mass as produced by the original BRL computer program.

Next there is a rigid rotation about the center of the earth, to make the y-axis pass through the radar; the origin is also moved upward a distance h, the height of the radar above the elevation of the gun. The equations are

$$\theta = d/R,$$

$$x_2 = x_1(1 - \theta^2/2) - y_o\theta - d$$

$$C3 = -y_g/d$$

$$S3 = -x_g/d$$

$$x_3 = x_2C3 - z_1S3$$

$$z_3 = x_2S3 + z_1Cs$$

$$y_3 = x_1\theta + y_o(1 - \theta^2/2) - \theta d/2 - h,$$

where h is the height of the radar above the altitude of the gun. The coordinates of the center of mass of the projectile in the rectilinear radar-axes system are x_3, y_3, z_3. The reason for not explicitly performing the rotation through θ in the usual manner is to avoid the round-off error of subtracting nearly-equal quantities of the order of R, the radius of the earth.

Concurrently with the computation of x_3, y_3, z_3 there are computed their time derivatives, using the time derivativex $\dot{x}_o$, $\dot{y}_o$, and $\dot{z}_o$ produced by the original

BRL program:

$$\dot{x}_1 = \dot{x}_o C1 - \dot{z}_o S1$$

$$\dot{z}_1 = \dot{x}_o S1 + \dot{z}_o C1$$

$$\dot{x}_2 = \dot{x}_1(1 - \theta^2/2) - \dot{y}_o \theta$$

$$\dot{x}_3 = \dot{x}_2 C3 - \dot{z}_1 S3$$

$$\dot{z}_3 = \dot{x}_2 S3 + \dot{z}_1 C3$$

The radar range

$$r = (x_3^2 + y_3^2 + z_3^2)^{1/2}$$

and the range rate

$$\dot{r} = (x_3\ \dot{x}_3 + y_3\ \dot{y}_3 + z_3\ \dot{z}_3)/r$$

are computed.

The phases-array radar angles are computed by

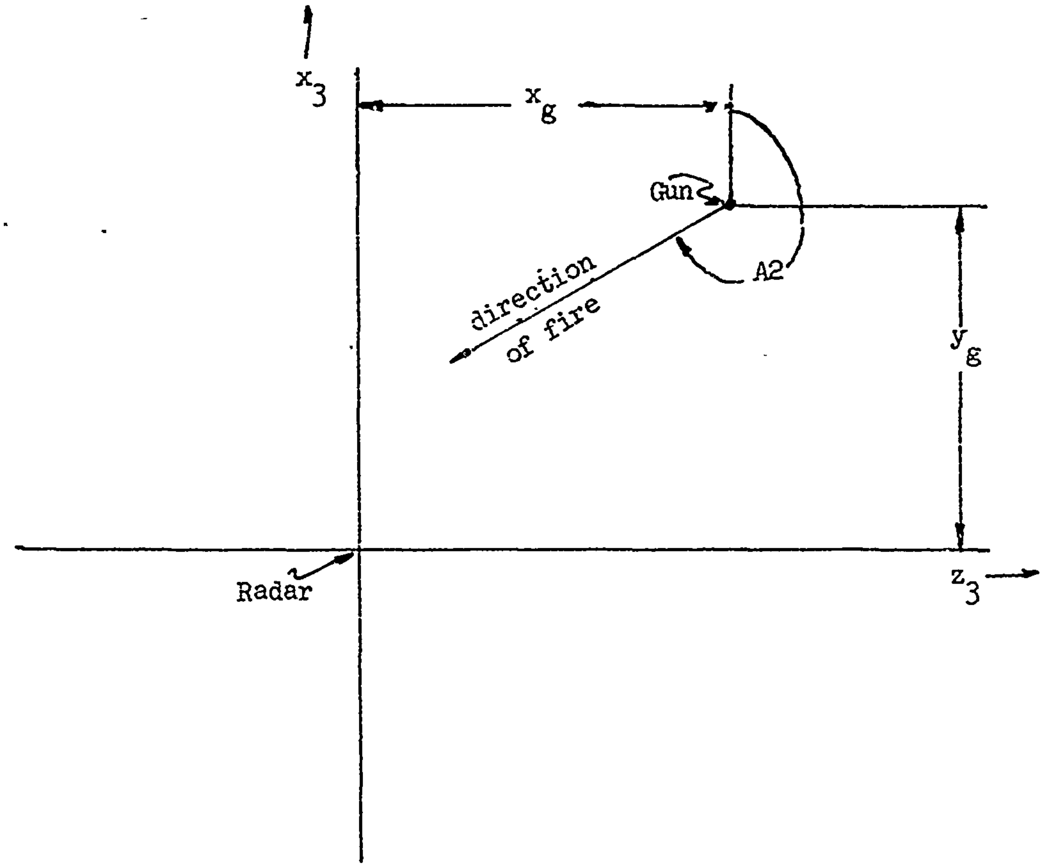
$$\alpha = \text{arc } \cos(x_3/r)$$

$$\beta = \text{arc } \cos(z_3/r)$$

The azimuth and elevation angles are

$$AZ = \text{arc } \tan(z_3/x_3)$$

$$EL = \text{arc } \sin(y_3/r)$$



<u>Position of Gun in Rectilinear</u>
<u>Radar Axes</u>

(Note: the gun coordinates are called (x_g, y_g) as shown above, in the program described in Appendix D. They are called (z_g, x_g) in section 2.)

Figure D-1

Appendix E

Components of Jacobian Matrices

A. F Matrix

$$[F]_{ij} = \partial f_i / \partial X_j$$

$$F = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ F_{41} & F_{42} & 0 & F_{44} & F_{45} & F_{46} & F_{47} & F_{48} \\ F_{51} & F_{52} & 0 & F_{54} & F_{55} & F_{56} & F_{57} & F_{58} \\ F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66} & F_{67} & F_{68} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ F_{81} & F_{82} & 0 & F_{84} & F_{85} & F_{86} & F_{87} & F_{88} \end{pmatrix}$$

$$F_{41} = \left[V_x \frac{\partial D_1}{\partial h} + (V_y g_z - V_z g_x) \frac{\partial D_2}{2h} \right] \sin\gamma$$

$$F_{42} = \left[V_x \frac{\partial D_1}{\partial h} + (V_y g_z - V_z g_x) \frac{\partial D_2}{\partial h} \right] \cos\gamma$$

$$F_{44} = D_1 + \left[V_x \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_y g_z - V_z g_y) \right] \frac{X_4}{V}$$

$$F_{45} = \left[V_x \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_y g_z - V_z g_y) \right] \frac{X_5}{V} + 2\omega U_{12} + D_2 g_z$$

$$F_{46} = \left[V_x \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_y g_z - V_z g_y)\right] \frac{X_6}{V} + 2\omega U_{13} - D_2 g_y$$

$$F_{47} = D_1 V_x / X_7$$

$$F_{48} = D_2 (V_y g_z - V_z g_y) / X_8$$

$$F_{51} = \left[V_y \frac{\partial D_1}{\partial h} + (V_z g_x - V_x g_z) \frac{\partial D_2}{\partial h}\right] \sin\gamma$$

$$F_{52} = \left[V_y \frac{\partial D_1}{\partial h} + (V_z g_x - V_x g_z) \frac{\partial D_2}{\partial h}\right] \cos\gamma$$

$$F_{54} = \left[V_y \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_z g_x - V_x g_z)\right] \frac{X_4}{V} + 2\omega U_{21} - D_2 g_z$$

$$F_{55} = D_1 + \left[V_y \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_z g_x - V_x g_z)\right] \frac{X_5}{V}$$

$$F_{56} = \left[V_y \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_z g_x - V_x g_z)\right] \frac{X_6}{V} + 2\omega U_{23} + D_2 g_x$$

$F_{57} = D_1 \dot{V}_y / X_7$

$F_{58} = D_2(V_z g_x - V_x g_z)/X_8$

$$F_{61} = \left[V_z \frac{\partial D_1}{\partial h} - (V_x g_y - V_y g_z) \frac{\partial D_2}{\partial h} \right] \sin\gamma$$

$$F_{62} = \left[V_z \frac{\partial D_1}{\partial h} + (V_x g_y - V_y g_z) \frac{\partial D_2}{\partial h} \right] \cos\gamma$$

$$F_{63} = 0$$

$$F_{64} = \left[V_z \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_x g_y - V_y g_x) \right] \frac{X_4}{V} + D_2 g_y + 2\omega U_{31}$$

$$F_{65} = \left[V_z \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_x g_y - V_y g_x) \right] \frac{X_5}{V} - D_2 g_x + 2\omega U_{32}$$

$$F_{66} = D_1 + \left[V_z \frac{\partial D_1}{\partial V} + \frac{\partial D_2}{\partial V} (V_x g_y - V_y g_x) \right] \frac{X_6}{V}$$

$F_{67} = D_1 V_z / X_7$

$F_{68} = D_2(V_x g_y - V_y g_x)/X_8$

$$F_{81} = X_8 \frac{\partial D_3}{\partial h} \sin\gamma$$

$$F_{82} = X_8 \frac{\partial D_3}{\partial h} \cos\gamma$$

$$F_{84} = \left(X_8 \frac{\partial D_3}{\partial V}\right) \frac{X_4}{V}$$

$$F_{85} = \left(X_8 \frac{\partial D_3}{\partial V}\right) \frac{X_5}{V}$$

$$F_{86} = \left(X_8 \frac{\partial D_3}{2V}\right) \frac{X_6}{V}$$

$$F_{88} = D_3$$

- -

where

$$\frac{\partial D_1}{\partial h} = D_1 \left[\frac{\partial\rho(h)/\partial h}{\rho(h)} \right] + \rho(h) X_7 \left[\sum_{1,4} i \; a_i \; M^{i+1} \right] \frac{\partial V_s}{\partial h}$$

$$\frac{\partial D_2}{\partial h} = \frac{X_8}{V^3} \left[\sum_{1,4} i \; b_i \; M^{i+1} \right] \frac{\partial V_s}{\partial h}$$

$$\frac{\partial D_3}{\partial h} = D_3 \left[\frac{\partial\rho(h)/\partial h}{\rho(h)} \right] + .004\rho(h) \left[\sum_{1,4} i \; c_i \; M^{i+1} \right] \frac{\partial V_s}{\partial h}$$

$$\frac{\partial D_1}{\partial V} = -\rho(h)\ X_7 \left[\sum_{0,4}(i+1)\ a_i\ M^i\right]$$

$$\frac{\partial D_2}{\partial V} = -\frac{X_8}{V^3}\left[\sum_{0,4}(i-2)b_i\ M^i\right]$$

$$\frac{\partial D_3}{\partial V} = -.004\ \rho(h)\left[\sum_{0,4}(i+1)\ c_i\ M^i\right]$$

$$\frac{\partial\ \rho(h)/\partial h}{\rho(h)}^{*} \cong -8.66473 \times 10^{-5} - 4.60557 \times 10^{-9}h + 1.26345 \times 10^{-15}h^2$$

$$\frac{\partial V_s}{\partial h}^{*} \cong -3.84 \times 10^{-3} \qquad h \leq 11,000$$

$$0 \qquad h > 11,000$$

* Assume U.S. Standard Atmosphere

B. H Matrix

$$H_{ij} = \partial h_i/\partial X_j$$

$$H = \begin{pmatrix} H_{11} & H_{12} & H_{13} & 0 & 0 & 0 & 0 & 0 \\ H_{21} & H_{22} & H_{23} & 0 & 0 & 0 & 0 & 0 \\ H_{31} & H_{32} & H_{33} & 0 & 0 & 0 & 0 & 0 \\ H_{41} & H_{42} & H_{43} & H_{44} & H_{45} & H_{46} & 0 & 0 \end{pmatrix}$$

$$H_{11} = X_1/R$$

$$H_{12} = X_2/R$$

$$H_{13} = X_3/R$$

$$H_{21} = -\sqrt{X_2^2 + X_3^2}/R^2$$

$$H_{22} = \frac{X_1 X_2}{R^2 \sqrt{X_2^2 + X_3^2}}$$

$$H_{23} = \frac{X_1 X_3}{R^2 \sqrt{X_2^2 + X_3^2}}$$

$$H_{31} = \frac{X_3 X_1}{R^2 \sqrt{X_1^2 + X_2^2}}$$

$$H_{32} = \frac{X_3 X_2}{R^2 \sqrt{X_1^2 + X_2^2}}$$

$$H_{33} = \sqrt{X_1^2 + X_2^2}/R^2$$

$$H_{41} = (X_4 R - \dot{R} X_1)/R^2$$

$$H_{42} = (X_5 R - \dot{R} X_2)/R^2$$

$$H_{43} = (X_6 R - \dot{R} X_3)/R^2$$

$$H_{44} = X_1/R$$

$$H_{45} = X_2/R$$

$$H_{46} = X_3/R$$

Appendix F

Glossary of Symbols

a_e	Projectile Yaw angle of repose
α, β	Radar angle measurements
AZ	Wind azimuth angle
C	Unknown drag parameter
d	Projectile diameter
D1	$- \rho\ C\ \bar{K}(M)V$
D2	$- S\ \bar{K}_D(M)/V^2$
D3	$- .004\ \rho\ K_A^*(M)V$
$f(x_t,t)$	State variable equations of motion
$F_{\ell-1\|\ell-1}$	Jacobian matrix $\partial f(x)/\partial x$
g_o	Sea level acceleration of gravity
g_x, g_y, g_z	Gravity vector components
$g(x,t)$	Matrix function
γ	Radar tilt angle
h	Target altitude
h_o	Radar site altitude
$h(x_\ell)$	Radar measurement vector
H_ℓ	Jacobian matrix $\partial h(x)/\partial x$
$K_i(M)$	Drift function
$K_{D_i}(M)$	Drag function — Projectile i
$K_iA(M)$	Spin function

$\bar{K}(M)$, $\overline{K}_D(M)$, $\bar{K}_A(M)$		Universal functions
l		Projectile lift factor
m		Projectile mass
M		Mach no.= V/Vs
μ		Laditude of radar
N		Projectile spin
ω_R		Earth rotational rate
ϕ		Radar pointing angle (clockwise from north)
$\phi_{\ell\|\ell-1}$	=	$I + \Delta t \; F_{\ell-1\|\ell-1}$
Q_t		State noise covariance matrix
Q_t'	=	$Q_t \; \Delta t$
r		Unknown drift parameter
ρ		Density of air
R_e		Radius of earth
R		Target range
$\dot{R}$		Range rate
S	=	$N \times r$
U		Coriolis matrix

V	Projectile velocity (W.R.T. air)
V_{xr}, V_{yr}, V_{zr}	Projectile velocity vector components
Vs	Speed of sound
x_g, y_g, z_g	Ground located Cartesian coordinate system
x_r, y_r, z_r	Radar located Cartesian coordinate system
X_t	State vector
$\hat{X}_\ell$	Estimated state vector
Z_ℓ	Radar measurement vector