

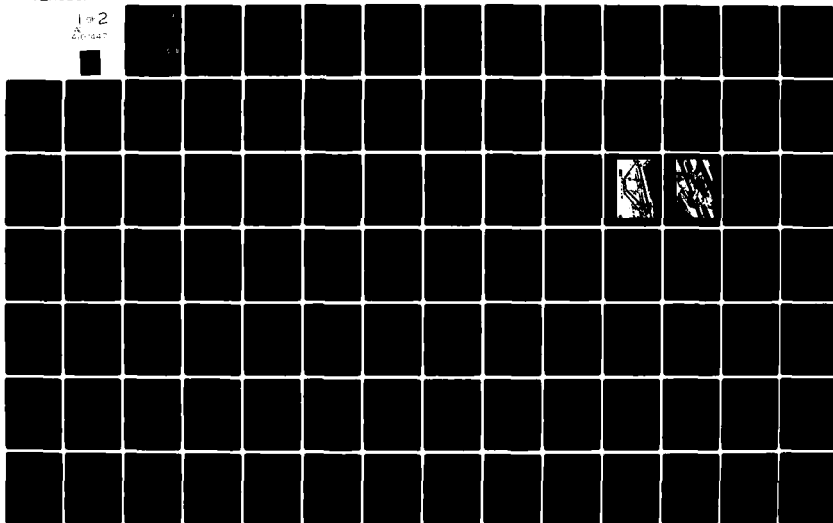
AD-A107 447

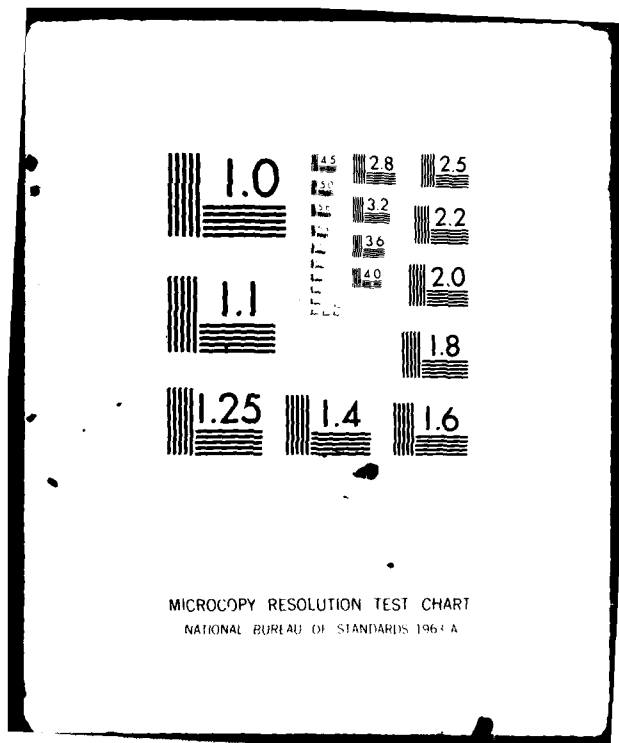
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH F/O 12/1
IDENTIFICATION OF WHEELSET/RAIL CREEP COEFFICIENTS FROM DYNAMIC--ETC(U)
JAN 79 W N HERZOG
AFIT-CI-79-254T

UNCLASSIFIED

NL

1 of 2
210-047





AD A107447

AFIT-CI-79-254T

LEVEL II

IDENTIFICATION OF WHEELSET/RAIL CREEP COEFFICIENTS
FROM DYNAMIC RESPONSE DATA USING THE MAXIMUM
LIKELIHOOD PARAMETER IDENTIFICATION TECHNIQUE

by

William N. Herzog

Princeton University
School of Engineering and Applied Science
Department of Mechanical and Aerospace Engineering

Submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering from Princeton University, 1978.

Prepared by:

William N. Herzog
William N. Herzog

Approved by:

Larry M. Sweet
Professor Larry M. Sweet

DTIC
ELECTE
NOV 12 1981

DTIC FILE COPY

January 1979

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

01-00001 10 26 16

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 79-254T	2. GOVT ACCESSION NO. AD-A107447	3. RECIPIENT'S CATALOG NUMBER 447
4. TITLE (and Subtitle) Identification of Wheelset/Rail Creep Co-Efficients from Dynamic Response Data Using the Maximum Likelihood Parameter Identification Technique		5. TYPE OF REPORT & PERIOD COVERED THESIS/UNSUBMITTATION
7. AUTHOR(s) William N. Herzog		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Princeton University		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE Jan 1979
		13. NUMBER OF PAGES 119
		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 20 OCT 1981		
18. SUPPLEMENTARY NOTES APPROVED FOR PUBLIC RELEASE; IAW AFR 190-17 Fredric C. Lynch FREDRIC C. LYNCH, Major, USAF Director of Public Affairs Air Force Institute of Technology (AIC) Wright-Patterson AFB, OH 45433		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) ATTACHED 81 10 26 164		

ABSTRACT

This thesis explores the application of the maximum likelihood parameter identification technique to determine the wheel/rail creep coefficients using dynamic response data. The equations of motion for the dynamically scaled wheelset are presented and the reduced form of the maximum likelihood equations as applicable to the dynamically scaled wheelset model are developed. The maximum likelihood equations were formulated into a maximum likelihood algorithm which was implemented in Fortran IV. Using simulated wheelset data, the effects of a random input representation of the track versus a deterministic input with uncertainty representation are determined. The effects of various levels of measurement noise are also examined. This preliminary analysis indicates that the deterministic representation of the track input yields better results. Representing the track as a random track input requires further investigation into the effects of longer data records and smaller time steps on the performance of the maximum likelihood algorithm.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

ACKNOWLEDGMENTS

I would like to thank the Guggenheim Foundation for providing me with the opportunity to complete my thesis research at Princeton University.

I am most indebted to my advisor Professor Larry M. Sweet for his continuing interest and guidance during both the experimental and theoretical phases of my research. I am also grateful for his suggestions during the preparation of the final draft of this thesis. I would also like to express special thanks to Professor Robert F. Stengel for introducing me to Maximum Likelihood theory and for providing me with many practical insights into its use.

I gratefully acknowledge the support I received from Bill Putman, John Kukon, Ted Griffith and John Grieb, all members of the research and technical staff of the Princeton Flight Mechanics facilities. Their insights into the practical aspects of engineering and their guidance during the experimental phase of this research program are most appreciated.

I would like to express my sincerest thanks to Randy Nichols whose assistance during the computer programming part of my research was invaluable and is most deeply appreciated.

I would like to thank Karen Praul for the time she devoted to accurately typing the final draft of this thesis.

This thesis carries 1408-T in the records of the Department of Mechanical and Aerospace Engineering.

TABLE OF CONTENTS

	<u>Page</u>
Abstract	i
Acknowledgments	ii
Table of Contents	iii
List of Figures	v
List of Tables	viii
Nomenclature	ix
Introduction	1
Chapter 1. Maximum Likelihood Theory	4
1.1 General Form of the Maximum Likelihood Equations	4
1.2 Application of the Maximum Likelihood Equations to the Wheelset	11 11
1.3 Maximum Likelihood Algorithm	15
1.4 Test Data Generation	19
Chapter 2. Experimental Program	21
2.1 Objectives	21
2.2 Experimental Facility	21
2.3 Transducers	22
2.4 Experimental Procedure	34
2.5 Measurement Noise	35
2.6 Filtering	36
2.7 Digitization	36
Chapter 3. Results	39
3.1 Outline of Test Cases	39

	<u>Page</u>
3.1.1 Explanation of Test Case Data	39
3.1.2 Purpose of Each Test Case	41
3.2 Method of Testing the Maximum Likelihood Program	43
3.3 Test Case Results	46
3.3.1 Likelihood Function	46
3.3.2 Gradient and Fisher Information Matrix	67
3.4 Application of Results to Actual Wheelset Data	72
References	76
Appendix A. Derivation of $\frac{\partial \hat{x}(t)}{\partial \theta}$	77
Appendix B. Solution to $\frac{\partial P(t_i)}{\partial \theta}$	78
Appendix C. Definitions of Variables in F and G Matrices	81
Appendix D. Simplification of $\frac{\partial P(t_i)}{\partial \theta}$ for Wheelset Problem	82
Appendix E. Listing of the Maximum Likelihood Program	83
Appendix F. List of APL Functions Used to Generate Wheelset Simulated Data	112
Appendix G. List of the Program Used to Operate the A/D Converter .	116
Appendix H. Plots of the Likelihood Function, Observation Term and Bias Term for Test Cases 1 through 9	119

LIST OF FIGURES

Chapter 1

Figure 1.1 Maximum Likelihood Algorithm

Figure 1.2 Development Form of Maximum Likelihood Algorithm

Chapter 2

Figure 2.1 Side View of Wheelset Idler-Carriage Relationship

Figure 2.2 Top View of Wheelset Idler-Carriage Relationship

Figure 2.3 Wheelset Instrumentation

Figure 2.4 Transducer Signal Conditioning

Figure 2.5 Definition of Lateral Displacement

Figure 2.6 Calibration of Number One Displacement Transducer

Figure 2.7 Calibration of Number Two Displacement Transducer

Figure 2.8 Sample Responses of Wheelset Model to Random Track Inputs

Figure 2.9 Sampling Interval

Chapter 3

Figure 3.1 Representations of Track Input

Figure 3.2 Computer Output for the Last Iteration of Test Case Number 8

Figure 3.3 Plot of the Likelihood Function for Test Case 2

Figure 3.4 Plot of the Observation Term for Test Case 2

Figure 3.5 Plot of the Bias Term for Test Case 2

Figure 3.6 Plot of the Observation Term for Test Case 4

Figure 3.7 Plot of the Observation Term for Test Case 6

Figure 3.8 Plot of the Bias Term for Test Case 4

Figure 3.9 Plot of the Bias Term for Test Case 6

- Figure 3.10 Plot of the Likelihood Function for Test Case 4
- Figure 3.11 Plot of the Likelihood Function for Test Case 6
- Figure 3.12 Plot of the Likelihood Function for Test Case 7
- Figure 3.13 Average Slope Method for Checking the Gradient and Second Partial of the Likelihood Function

Figure 3.14 Plot of the Gradient for Test Case 4

Figure 3.15 Plot of the Gradient for Test Case 6

Appendix H

Figure H.1 Plot of the Likelihood Function for Test Case 1

Figure H.2 Plot of the Observation Term for Test Case 1

Figure H.3 Plot of the Bias Term for Test Case 1

Figure H.4 Plot of the Likelihood Function for Test Case 2

Figure H.5 Plot of the Observation Term for Test Case 2

Figure H.6 Plot of the Bias Term for Test Case 2

Figure H.7 Plot of the Likelihood Function for Test Case 3

Figure H.8 Plot of the Observation Term for Test Case 3

Figure H.9 Plot of the Bias Term for Test Case 3

Figure H.10 Plot of the Likelihood Function for Test Case 4

Figure H.11 Plot of the Observation Term for Test Case 4

Figure H.12 Plot of the Bias Term for Test Case 4

Figure H.13 Plot of the Likelihood Function for Test Case 5

Figure H.14 Plot of the Observation Term for Test Case 5

Figure H.15 Plot of the Bias Term for Test Case 5

Figure H.16 Plot of the Likelihood Function for Test Case 6

Figure H.17 Plot of the Observation Term for Test Case 6

Figure H.18 Plot of the Bias Term for Test Case 6

Figure H.19 Plot of the Likelihood Function for Test Case 7

- Figure H.20 Plot of the Observation Term for Test Case 7
- Figure H.21 Plot of the Bias Term for Test Case 7
- Figure H.22 Plot of the Likelihood Function for Test Case 8
- Figure H.23 Plot of the Observation Term for Test Case 8
- Figure H.24 Plot of the Bias Term for Test Case 8
- Figure H.25 Plot of the Likelihood Function for Test Case 9
- Figure H.26 Plot of the Observation Term for Test Case 9
- Figure H.27 Plot of the Bias Term for Test Case 9

LIST OF TABLES

Chapter 2

Table 2.1 Filter Cutoff Frequencies

Chapter 3

Table 3.1 Summary of Maximum Likelihood Program Testing

Table 3.2 Q and R Matrices In Test Cases 1 Through 5

Table 3.3 Summary of Results for Test Cases

NOMENCLATURE

A	a general matrix
B	covariance matrix of v
$E[f]$	expected value of f
F	system dynamics matrix for dynamically scaled wheelset model
G	system input matrix for dynamically scaled wheelset model
H	measurement matrix for dynamically scaled wheelset model
K	Kalman gain matrix
K_y, K_ψ	lateral (yaw) spring constants, n/m ($n-m/rad$)
L	likelihood function
M	Fisher information matrix
M	mass of dynamically scaled wheelset model, kg
N	end of summation index for likelihood function
$N(x,y)$	Gaussian distribution with mean x and covariance matrix y
P	error covariance of Kalman Filter state estimates
Q	covariance matrix of random input disturbance
R	covariance matrix of measurement noise
T	period of a signal
V	translational velocity of the dynamically scaled wheelset model
<u>Y</u>	transformation used to solve matrix Riccati equation and $\partial P/\partial \theta$ using a Kalman-Engler solution technique
a	element of a general matrix
b	bias in the yaw potentiometer mounted on the dynamically scaled wheelset model

c	lateral damping coefficient, n-sec/m
e	electrical bias added to the yaw potentiometer output by the on-board analog computer
f	creep coefficient, n
i	summation index, product index
j	imaginary part of a complex number
l	distance from wheelset c.g. to contact point between wheel and rail, m
m	slope of the straight line function which describes the characteristics of the displacement transducers
p	probability density
r_o	wheel rolling radius, m
t	time
<u>u</u>	deterministic input into a dynamic system
<u>v</u>	measurement noise vector
<u>w</u>	random input into a dynamic system
<u>x</u>	vector of state variables for a dynamic system
$\hat{x}$	vector of estimates of the state variables of a dynamic system
y	lateral displacement of wheelset, m
α	wheel conicity, rad
β	factor which multiplies the nominal values for the creep coefficients to get the true values for the creep coefficients
Γ	matrix which multiplies the deterministic input in the Kalman Filter equation for the propagation of the state variable estimates
<u>Y</u>	transformation used to derive $\partial P / \partial \theta$ using the Kalman-Engler solution technique
Δ	small increment
δ	distance from rail to inertial reference

δ_1, δ_2	distance of left and right rails from fixed reference, m
$\bar{\delta}$	distance of track centerline from reference $(= (\delta_1 + \delta_2)/2)$, m
ϵ	error criterion
ξ	term of the likelihood function
η	yaw potentiometer signal after processing on the on-board analog computer
$\underline{\theta}$	vector of unknowns to be identified using the maximum likelihood parameter identification technique
Θ	matrix which multiplies the measurement vector in the Kalman Filter equation for the propagation of the state variable estimates
Λ	matrix which multiplies the state variable estimates in the equation for the propagation of the sensitivity vector $\partial \underline{\hat{x}} / \partial \underline{\theta}$
Π	product
Σ	summation
T	matrix which multiplies the measurement vector in the equation for the propagation of the sensitivity vector $\partial \underline{\hat{x}} / \partial \underline{\theta}$
Φ	state transition matrix
Ψ	matrix which multiplies the deterministic input in the equation for the propagation of the sensitivity vector $\partial \underline{\hat{x}} / \partial \underline{\theta}$
Ω	axle rotational velocity $(= V/r_o)$, sec^{-1}
$\underline{\lambda}$	transformation used in solving the matrix Riccati equation using the Kalman-Engler solution technique
v	the innovation $\underline{z} - H\underline{\hat{x}}$
σ	eigenvalues of the dynamically scaled wheelset model system dynamics matrix

ϕ element of Φ
 ψ yaw angle of dynamically scaled wheelset model

Subscripts

i discrete point in time
L left side
n last of a sequence of discrete value
r right side
y lateral
ss steady state
 ψ yaw
o starting point
11 subscript of longitudinal creep coefficient
22 subscript of lateral creep coefficient

Sub-subscripts

o nominal value

Superscript

* true value

INTRODUCTION

The dynamic response of railroad vehicles to track irregularity inputs is determined by interactions between the vehicle body and suspension components, kinematic constraints dependent on the profiles of the wheels and rails, and friction forces generated at the wheel/rail interface. These latter forces, known as creep forces, result from small differential velocities between wheel and rail in directions tangential and lateral to the wheel. Linearized dynamic models for rail vehicles represent creep forces using the slope of the creep force versus relative velocity function at the origin; the slopes are known in the literature as creep coefficients.

The values of these coefficients has been determined from Hertzian contact theory by Kalker (4) and by measurements under steady-state conditions (9). However, previous analysis (12) of dynamic response data has indicated that better agreement between theory and experiment is obtained if values for creep coefficients are used that are up to 50% lower than determined in (4) and (9). The "best" values for creep coefficients under dynamic conditions have not been determined with precision; such determination would be of great value to the rail research technical community.

The objective of this thesis is the estimation of creep coefficients from experimental vehicle response data using advanced parameter identification techniques. To focus the effort on estimation of creep coefficients, data used are measurements of the response of a simplified scale model vehicle, whose characteristics and parameters are well-understood, exclusive of the

creep coefficients. Use of the simplified model, in this case a wheelset (single axle with two wheels and lateral and yaw suspension, capable of two degrees-of-freedom motion), permits examination of the fundamental kinematic hunting and instability characteristic of rail vehicles, without the effects of more complex vehicle configurations obscuring the results. The use of dynamically scaled models (9) allows better control of experimental conditions than expected in full scale.

The method used to estimate the creep coefficients is the maximum likelihood parameter identification technique. This technique has been used to identify aircraft stability derivatives successfully for several years, and represents the state-of-the-art in parameter estimation. The specific objective of this thesis is to develop and implement the maximum likelihood method for dynamically scaled wheelset models. In addition some insight into the best way to conduct dynamic experiments involving the dynamically scaled wheelset model was desired.

This research program was divided into two phases. The first was the experimental phase during which the dynamic wheelset experiments were conducted. The purpose of the experiments was to obtain data that could be processed using a maximum likelihood processor. The second stage of the research project was a developmental one. During this part of the research program the maximum likelihood algorithm was developed and implemented into a Fortran IV computer program. The final phase of research was the test phase during which simulated wheelset data was generated using a computer model of the dynamically scaled wheelset model. The maximum likelihood computer program was then checked out using this simulated data.

The experimental phase was conducted first so that as much information as possible about the wheelset model could be obtained prior to the development

of the maximum likelihood program. The actual wheelset data taken was not analyzed using the maximum likelihood program. The results for the simulated data suggest that more research needs to be done with simulated data so that more conclusive results can be drawn when the actual wheelset data is processed.

The body of this thesis is divided into three chapters. The first presents the generalized maximum likelihood equations and then develops the reduced equations that apply to the dynamically scaled wheelset model. The second chapter discusses the experimental phase of the research program. The knowledge gained from these initial experiments will be used to develop an improved experimental program for the next series of data. The final chapter presents the results obtained for the simulated data, and the interpretation of these results in terms of actual wheelset data.

Chapter 1

MAXIMUM LIKELIHOOD THEORY

1.1 General Form of the Maximum Likelihood Equations

The maximum likelihood parameter identification technique is used to identify unknown parameters of a dynamic system given measurements of some or all of the system's state variables. This method maximizes the probability of the given measurements conditioned on the vector of unknowns. The following development of the maximum likelihood equations will be for a general case, linear, time-varying system. Except where otherwise noted the equations in this section were taken from References (8) and (3).

The state-space form of the differential equations for a linear dynamic system is

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + L(t)\underline{u}(t) + G(t)\underline{w}(t) \quad (1.1)$$

where $\underline{x}(t)$ = state variable vector

$\underline{u}(t)$ = known input vector

$\underline{w}(t)$ = input disturbance vector

The measurement process is described by the equation

$$\underline{z}(t) = H(t)\underline{x}(t) + \underline{v}(t) \quad (1.2)$$

where $\underline{z}(t)$ = measurement vector

$\underline{x}(t)$ = state variable vector

$\underline{v}(t)$ = measurement noise vector

The foundation of the maximum likelihood method is maximizing the conditional probability of the vector for unknowns given the measurements of the states.

$$\text{maximize } p(\underline{\theta} | \underline{z}_n)$$

where p denotes the probability density

$\underline{\theta}$ = vector of unknowns to be identified

$\underline{z}_n$ = sequence of measurements

Maximizing $p(\underline{\theta} | \underline{z}_n)$ is equivalent to maximizing the log likelihood function.

$$L = \ln[p(\underline{\theta} | \underline{z}_n)] \quad (1.3)$$

where L $\triangleq$ log likelihood function

Sequential application of Baye's Rule to the conditional probability of $\underline{\theta}$ given $\underline{z}_n$ yields:

$$p(\underline{\theta} | \underline{z}_n) = p(\underline{z}_n | \underline{\theta}) p(\underline{\theta} | \underline{z}_{n-1}) \quad (1.4)$$

$$= p(\underline{z}_n | \underline{\theta}) p(\underline{z}_{n-1} | \underline{\theta}) p(\underline{\theta} | \underline{z}_{n-2})$$

$$= p(\underline{z}_n | \underline{\theta}) p(\underline{z}_{n-1} | \underline{\theta}) p(\underline{z}_{n-2} | \underline{\theta}) p(\underline{\theta} | \underline{z}_{n-3})$$

$$= \prod_{i=1}^N p(\underline{z}_i | \underline{\theta}) p(\underline{\theta} | 0) \quad (1.5)$$

Because the state vector is dependent upon the vector of unknowns the equation

$$p(\underline{\theta} | \underline{z}_n) = \prod_{i=1}^N p(\underline{z}_i | \underline{\theta}) p(\underline{\theta} | 0)$$

can be re-written as

$$p(\underline{\theta} | \underline{z}_n) = \prod_{i=1}^N p(\underline{z}_i | \underline{x}_i(\underline{\theta})) p(\underline{\theta} | 0) \quad (1.6)$$

Assuming the input disturbance ($\underline{w}(t)$) and the measurement noise ($\underline{v}(t)$) are gaussian, then the formula for multivariate gaussian conditional distribution can be applied because $p(\underline{z}_i | \underline{x}_i(\underline{\theta}))$ is gaussian.

$$p(\underline{z}_i | \underline{x}_i(\underline{\theta})) = \frac{1}{(2\pi)^{N/2} |\underline{B}_i|^{1/2}} \cdot e^{-\frac{1}{2}(\underline{z}_i - \underline{E}[\underline{z}_i])^T \underline{B}_i^{-1}(\underline{z}_i - \underline{E}[\underline{z}_i])} \quad (1.7)$$

In equation 1.7 $E[\underline{z}_i] = H_i \hat{\underline{x}}_i$ where $\hat{\underline{x}}$ is the output of a Kalman Filter. The matrix B_i is defined as follows:

$$\begin{aligned} B_i &= E[(\underline{z}_i - E[\underline{z}_i])(\underline{z}_i - E[\underline{z}_i])^T] \\ B_i &= E[(\underline{z}_i - H_i \hat{\underline{x}}_i)(\underline{z}_i - H_i \hat{\underline{x}}_i)^T] \\ B_i &= H_i P_i H_i^T + R_i \end{aligned} \quad (1.8)$$

Combining equations 1.3, 1.6 and 1.7 yields:

$$L = \ln \left\{ \prod_{i=1}^N \frac{1}{2\pi^{(N/2)} |B_i|^{(1/2)}} \cdot e^{-\frac{1}{2}(\underline{z}_i - H_i \hat{\underline{x}}_i)^T B_i^{-1}(\underline{z}_i - H_i \hat{\underline{x}}_i)} \right\} \cdot p(\underline{\theta}|0) \quad (1.9)$$

$$\begin{aligned} L &= -\frac{1}{2} \sum_{i=1}^N \{(\underline{z}_i - H_i \hat{\underline{x}}_i)^T B_i^{-1}(\underline{z}_i - H_i \hat{\underline{x}}_i) + \ln|B_i|\} \\ &\quad - \left(\frac{N}{2}\right) \ln(2\pi) + \ln[p(\underline{\theta}|0)] \end{aligned} \quad (1.10)$$

$$L = -\frac{1}{2} \sum_{i=1}^N \{(\underline{z}_i - H_i \hat{\underline{x}}_i)^T B_i^{-1}(\underline{z}_i - H_i \hat{\underline{x}}_i) + \ln|B_i|\} + \text{constant} \quad (1.11)$$

The maximum likelihood problem involves maximizing equation 1.11 with respect to theta, subject to the constraints of the Kalman Filter equations. The Kalman Filter equations must be satisfied because the state estimate $\hat{\underline{x}}_i$ and the state covariance (P_i) are outputs of the Kalman Filter. A summary of the continuous time, time varying Kalman Filter equations are given below (5).

System Model:

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + L(t)\underline{u}(t) + G(t)\underline{w}(t) \quad (1.12)$$

Measurement Model:

$$\underline{z}(t) = H(t)\underline{x}(t) + \underline{v}(t) \quad (1.13)$$

State Estimates:

$$\hat{\underline{x}}(t) = F(t)\hat{\underline{x}}(t) + L(t)\underline{u}(t) + K(t) \underline{z}(t) - H(t)\hat{\underline{x}}(t) \quad (1.14)$$

Error Covariance Propagation:

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - P(t)H^T(t)R^{-1}(t)H(t)P(t) \quad (1.15)$$

Kalman Gain Matrix:

$$K(t) = P(t)H^T(t)R^{-1}(t) \quad (1.16)$$

$$\text{where } \underline{v}(t) \sim N(0, R(t)) \quad (1.17)$$

$$\text{and } \underline{w}(t) \sim N(0, Q(t)) \quad (1.18)$$

The notation used in equation 1.17 denotes that $\underline{v}(t)$ is a random sequence with a gaussian probability density with a mean value and variance defined below.

$$E[\underline{v}(t)] = 0 \quad (1.19)$$

$$E[(\underline{v}(t) - E[\underline{v}(t)])(\underline{v}(t) - E[\underline{v}(t)])^T] = R(t) \quad (1.20)$$

Equation 1.14 is solved on the digital computer for discrete values of $\hat{x}_1$ using equations 1.21 through 1.24.

$$\hat{x}(t_0 + \Delta t) = \Phi(t_0, \Delta t)\hat{x}(t_0) + \Theta(t_0)\underline{z}(t_0) + \Gamma(t_0)\underline{u}(t_0) \quad (1.21)$$

$$\text{where } [F(t_0) - K(t_0)H(t_0)]\Delta t \Phi(t, \Delta t) = e \quad (1.22)$$

$$\Theta(t_0) = \Phi(t_0, \Delta t)[F(t_0) - K(t_0)H(t_0)]^{-1} \cdot [I - \Phi^{-1}(t_0, \Delta t)]K(t_0) \quad (1.23)$$

$$\Gamma(t_0) = \Phi(t_0, \Delta t)[F(t_0) - K(t_0)H(t_0)]^{-1} \cdot [I - \Phi^{-1}(t_0, \Delta t)]L(t_0) \quad (1.24)$$

The error covariance propagation equation (1.15) is solved using the Kalman-Engler solution.

$$P(t_0 + \Delta t) = [\phi_{\lambda y}(t_0, \Delta t) + \phi_{\lambda \lambda}(t_0, \Delta t)P(t_0)] \cdot [\phi_{yy}(t_0, \Delta t) + \phi_{y\lambda}(t_0, \Delta t)P(t_0)]^{-1} \quad (1.25)$$

where

$$\begin{bmatrix} \underline{y}(t_0 + \Delta t) \\ \underline{\lambda}(t_0 + t) \end{bmatrix} = \begin{bmatrix} \phi_{yy}(t_0, \Delta t) & \phi_{y\lambda}(t_0, \Delta t) \\ \phi_{\lambda y}(t_0, \Delta t) & \phi_{\lambda \lambda}(t_0, \Delta t) \end{bmatrix} \cdot \begin{bmatrix} \underline{y}(t_0) \\ \underline{\lambda}(t_0) \end{bmatrix} \quad (1.26)$$

and

$$\Phi(t_o, \Delta t) = \begin{bmatrix} \phi_{yy}(t_o, \Delta t) & \phi_{y\lambda}(t_o, \Delta t) \\ \phi_{\lambda y}(t_o, \Delta t) & \phi_{\lambda\lambda}(t_o, \Delta t) \end{bmatrix} \quad (1.27)$$

The equations for $\dot{\underline{y}}(t)$ and $\dot{\underline{\lambda}}(t)$ are given in equation 1.28.

$$\begin{bmatrix} \dot{\underline{y}}(t) \\ \dot{\underline{\lambda}}(t) \end{bmatrix} = \begin{bmatrix} -F^T(t) & H^T(t)R^{-1}(t)H(t) \\ G(t)Q(t)G^T(t) & F(t) \end{bmatrix} \begin{bmatrix} \underline{y}(t) \\ \underline{\lambda}(t) \end{bmatrix} \quad (1.28)$$

Therefore

$$\Phi(t_o, \Delta t) = \exp \begin{bmatrix} -F^T(t_o) & H^T(t_o)R^{-1}(t_o)H(t_o) \\ G(t_o)Q(t_o)G^T(t_o) & F(t_o) \end{bmatrix} \Delta t \quad (1.29)$$

The linear system of equations in 1.29 is derived in Reference (3).

The likelihood function can be expanded into a Taylor series with only three terms because the likelihood function for Gaussian conditional probability distribution is quadratic.

$$L(\theta) = L(\theta_o) + \left. \frac{\partial L}{\partial \theta} \right|_{\theta_o} (\theta - \theta_o) + (\theta - \theta_o)^T \left[\frac{1}{2} \left. \frac{\partial^2 L}{\partial \theta^2} \right|_{\theta_o} \right] (\theta - \theta_o) \quad (1.30)$$

Taking the partial derivative of $L(\theta)$ with respect to θ and setting it equal to zero to find the maximum:

$$\left. \frac{\partial L}{\partial \theta} \right|_{\theta^*} = \left. \frac{\partial L}{\partial \theta} \right|_{\theta_o} + \left. \frac{\partial^2 L}{\partial \theta^2} \right|_{\theta_o} (\theta^* - \theta_o) = 0 \quad (1.31)$$

where θ^* is the value of θ that results in the maximum of the likelihood function. Solving for θ^* yields:

$$\underline{\theta}^* = \underline{\theta}_o - \left[\left. \frac{\partial^2 L}{\partial \theta^2} \right|_{\theta_o} \right]^{-1} \left[\left. \frac{\partial L}{\partial \theta} \right|_{\theta_o} \right]^T \quad (1.32)$$

In equation 1.32 $\Delta\theta$, or the step in theta is equal to:

$$\Delta \underline{\theta} = - \left[\frac{\partial^2 L}{\partial \underline{\theta}^2} \bigg|_{\underline{\theta} = \underline{\theta}_0} \right]^{-1} \left[\frac{\partial L}{\partial \underline{\theta}} \bigg|_{\underline{\theta} = \underline{\theta}_0} \right]^T \quad (1.33)$$

$\frac{\partial L}{\partial \underline{\theta}}$ is defined as the gradient and is equal to:

$$\begin{aligned} \frac{\partial L}{\partial \underline{\theta}} = & \sum_{i=1}^N v^T(t_i) B^{-1}(t_i) \frac{\partial v(t_i)}{\partial \underline{\theta}} \\ & - \frac{1}{2} v^T(t_i) B^{-1}(t_i) \frac{\partial B(t_i)}{\partial \underline{\theta}} B^{-1}(t_i) v(t_i) \\ & + \frac{1}{2} \text{tr} \left[B^{-1}(t_i) \frac{\partial B(t_i)}{\partial \underline{\theta}} \right] \end{aligned} \quad (1.34)$$

where

$$v(t_i) = \underline{z}(t_i) - H(t_i) \hat{\underline{x}}(t_i) \quad (1.35)$$

$$\frac{\partial v(t_i)}{\partial \underline{\theta}} = -H(t_i) \frac{\partial \hat{\underline{x}}(t_i)}{\partial \underline{\theta}} - \frac{\partial H(t_i)}{\partial \underline{\theta}} \quad (1.36)$$

and

$$\begin{aligned} \frac{\partial B(t_i)}{\partial \underline{\theta}} = & H(t_i) \left[P(t_i) \frac{\partial (H^T(t_i))}{\partial \underline{\theta}} + \frac{\partial P(t_i)}{\partial \underline{\theta}} H^T(t_i) \right] \\ & + \frac{\partial H(t_i)}{\partial \underline{\theta}} P(t_i) H^T(t_i) + \frac{\partial R(t_i)}{\partial \underline{\theta}} \end{aligned} \quad (1.37)$$

The term $\frac{\partial^2 L}{\partial \underline{\theta}^2}$ is defined as the Fisher information matrix.

$$\begin{aligned} \frac{\partial^2 L}{\partial \underline{\theta}^2} = & \sum_{i=1}^N \frac{\partial v^T(t_i)}{\partial \underline{\theta}} B^{-1}(t_i) \frac{\partial v(t_i)}{\partial \underline{\theta}} \\ & - (2) v^T(t_i) B^{-1}(t_i) \frac{\partial B(t_i)}{\partial \underline{\theta}} B^{-1}(t_i) \frac{\partial v(t_i)}{\partial \underline{\theta}} \\ & - \frac{1}{2} \text{tr} \left[B^{-1}(t_i) \frac{\partial B(t_i)}{\partial \underline{\theta}} B^{-1}(t_i) \frac{\partial B(t_i)}{\partial \underline{\theta}} \right] \end{aligned} \quad (1.38)$$

The term $\frac{\partial \hat{\underline{x}}(t_i)}{\partial \underline{\theta}}$ in equation 1.36 is referred to as the sensitivity term

and is derived from the Kalman Filter state estimate equation (1.14).

$$\begin{aligned} \frac{\partial \hat{x}(t)}{\partial \theta} = & F(t) \frac{\partial \hat{x}(t)}{\partial \theta} + \frac{\partial F(t)}{\partial \theta} \hat{x}(t) + \frac{\partial L(t)}{\partial \theta} u(t) - K(t)H(t) \frac{\partial \hat{x}(t)}{\partial \theta} \\ & - K(t) \frac{\partial H(t)}{\partial \theta} \hat{x}(t) + \frac{\partial K(t)}{\partial \theta} z(t) - \frac{\partial K(t)}{\partial \theta} H(t) \hat{x}(t) \end{aligned} \quad (1.39)$$

See Appendix A for the derivation of equation 1.39.

Equation 1.39 can be solved for discrete values of $\frac{\partial \hat{x}(t)}{\partial \theta}$ on the digital computer using the following equations.

$$\begin{aligned} \frac{\partial \hat{x}(t_1)}{\partial \theta} \triangleq \frac{\partial \hat{x}(t_0 + \Delta t)}{\partial \theta} = & \Phi(t_0, t) \frac{\partial \hat{x}(t_0)}{\partial \theta} + \Lambda(t_0) \hat{x}(t_0) \\ & + \Psi(t_0) u(t_0) + T(t_0) z(t_0) \end{aligned} \quad (1.40)$$

where

$$\Phi(t_0, \Delta t) = e^{[F(t_0) - K(t_0)H(t_0)]\Delta t} \quad (1.41)$$

$$\begin{aligned} \Lambda(t_0) = & \Phi(t_0, \Delta t) [F(t_0) - K(t_0)H(t_0)]^{-1} \\ [I - \Phi^{-1}(t_0, \Delta t)] \left[\frac{\partial F(t_0)}{\partial \theta} - K(t_0) \frac{\partial H(t_0)}{\partial \theta} - \frac{\partial K(t_0)}{\partial \theta} H(t_0) \right] \end{aligned} \quad (1.42)$$

$$\Psi(t_0) = \Phi(t_0, \Delta t) [F(t_0) - K(t_0)H(t_0)]^{-1}.$$

$$[I - \Phi^{-1}(t_0, \Delta t)] \left[\frac{\partial L(t_0)}{\partial \theta} \right] \quad (1.43)$$

$$T(t_0) = \Phi(t_0, \Delta t) [F(t_0) - K(t_0)H(t_0)]^{-1}.$$

$$[I - \Phi^{-1}(t_0, t)] \left[\frac{\partial K(t_0)}{\partial \theta} \right] \quad (1.44)$$

The term $\frac{\partial P(t_1)}{\partial \theta}$ in equation 1.37 is solved using equation 1.45.

$$\begin{aligned} \frac{\partial P(t_0 + t)}{\partial \theta} = & \left[\phi_{yy}(t_0, \Delta t) + \phi_{yy}(t_0, \Delta t) \frac{\partial P(t_0)}{\partial \theta} \right] \cdot \\ & \left[\phi_{yy}(t_0, \Delta t) + \phi_{yy}(t_0, \Delta t) \frac{\partial P(t_0)}{\partial \theta} \right]^{-1} \end{aligned} \quad (1.45)$$

Equation 1.45 is an iterative solution to $\frac{\partial P(t_o + t)}{\partial \theta}$ and is derived in Appendix B.

The final unknown term in the sensitivity equation is $\frac{\partial K(t)}{\partial \theta}$. This term is derived by differentiating equation 1.16 with respect to theta

$$\frac{\partial K(t)}{\partial \theta} = P(t) \left[\frac{\partial H^T(t)}{\partial \theta} R^{-1}(t) + H^T(t) \frac{\partial R^{-1}(t)}{\partial \theta} \right] + \frac{\partial P(t)}{\partial \theta} H^T(t) R^{-1}(t) \quad (1.46)$$

1.2 Application of the Maximum Likelihood Equations to the Wheelset

The equations developed in the last section apply to any linear dynamic system. In this section those equations will be tailored to the dynamically scaled wheelset problem.

The wheelset equations of motion are given in equations 1.47 through 1.50.

$$\dot{\underline{x}}(t) = \underline{F}\underline{x}(t) + \underline{G}\underline{w}(t) \quad (1.47)$$

where

$$\underline{F} = \begin{bmatrix} \frac{-2f_{22} + cV}{MV} & \frac{2f_{22}}{M} & \frac{-K_y}{M} \\ 0 & -\frac{K_{\phi}}{2f_{11}l^2} & -\frac{\alpha V}{kr_o} \\ 1 & 0 & 0 \end{bmatrix} \quad \underline{x} = \begin{bmatrix} \dot{y} \\ y \\ (y - \delta) \end{bmatrix} \quad (1.48)$$

$$\underline{G} = \begin{bmatrix} \frac{c}{T} \\ 0 \\ -1 \end{bmatrix} \quad \underline{w} = \frac{\dot{\delta}}{\delta} \quad (1.49)$$

and

$$\underline{w}(t) \sim N(0, Q) \quad (1.50)$$

See Reference (11) for a detailed development of the wheelset equations of motion. Appendix C contains definitions for the variable of the F and G matrix.

The wheelset measurement equation is given in 1.51

$$\underline{z}(t) = H \underline{x}(t) + \underline{v}(t) \quad (1.51)$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.52)$$

and

$$\underline{v}(t) \sim N(0, R) \quad (1.53)$$

The system dynamics matrix (F), the system random input matrix (G), and the system measurement matrix (H), the covariance matrix for the random input disturbance (Q) and the covariance matrix for the measurement noise (R) are all time-invariant.

The creep coefficients f_{11} and f_{22} in the system dynamics matrix are the parameters to be identified by the maximum likelihood method.

For the purposes of simplifying the maximum likelihood equations the actual creep coefficients will be defined as a multiple of their nominal values.

$$f_{22} = \beta_{22} f_{22_0} \quad (1.54)$$

$$f_{11} = \beta_{11} f_{11_0} \quad (1.55)$$

$$\text{where, for simplicity} \quad \beta_{22} = \beta_{11} = \beta \quad (1.56)$$

The nominal values for the creep coefficients have been identified in other research on railroad vehicle stability (4).

Combining equations 1.48 and 1.54 through 1.56 the system dynamics matrix is:

$$F = \begin{bmatrix} \frac{-2\beta f_{22_0} + cV}{MV} & \frac{2\beta f_{22_0}}{M} & \frac{-K_y}{V_1} \\ 0 & \frac{-K_y V}{2\beta f_{11_0} \ell^2} & \frac{-\alpha V}{2r_0} \\ 1 & 0 & 0 \end{bmatrix} \quad (1.57)$$

Since the system dynamics matrix contains the only parameter to be identified the vector of unknowns $\underline{\theta}$ reduces to a scalar.

$$\underline{\theta} = \beta \quad (1.58)$$

The maximum likelihood equations of the last section can be greatly simplified for two reasons:

- 1) the wheelset is a time-invariant system.
- 2) the system dynamics matrix (F) contains the only parameter to be identified.

A summary of the time-invariant Kalman Filter Equations for the wheelset system is given below

State Estimates:

$$\dot{\hat{x}}(t) = F\hat{x}(t) + Kz(t) - KH\hat{x}(t) \quad (1.59)$$

Error Covariance Propagation:

$$\dot{P}(t) = FP(t) + P(t)F^T + GQG^T - P(t)H^TR^{-1}HP(t) \quad (1.60)$$

Kalman Gain Matrix:

$$K = P_{ss}H^TR^{-1} \quad (1.61)$$

$$\underline{v}(t) \sim N(0, R) \quad (1.62)$$

$$\underline{w}(t) \sim N(0, Q) \quad (1.63)$$

Equation 1.21, the solution to the Kalman filter estimates equation (1.14), reduces to

$$\hat{x}(t_o + \Delta t) = \Phi(\Delta t) \hat{x}(t_o) + \Theta z(t_o) \quad (1.64)$$

where

$$\Phi(\Delta t) = e^{[F-KH]\Delta t} \quad (1.65)$$

$$\Theta = \Phi(\Delta t)[F-KH]^{-1} \cdot [I - \Phi^{-1}(\Delta t)]K \quad (1.66)$$

P_{ss} in equation 1.61 is the steady state solution to equation 1.67

$$P(t_o + \Delta t) = [\phi_{\lambda y}(\Delta t) + \phi_{\lambda \lambda}(\Delta t) P(t_o)] \cdot [\phi_{yy}(\Delta t) + \phi_{y\lambda}(\Delta t) P(t_o)]^{-1} \quad (1.67)$$

This equation is the same as equation 1.25 except that ϕ is no longer dependent

on t_0 , but is a function only of Δt , as shown in equation 1.68.

$$\Phi(\Delta t) = \exp \begin{bmatrix} -F^T & H^T R^{-1} H \\ G Q G^T & F \end{bmatrix} \Delta t \quad (1.68)$$

Because all of the elements of the matrix in equation 1.68 are time invariant, the iterative solution for P has to be solved only once. This value of P_{s_0} is used to calculate the Kalman gain matrix according to equation 1.61. The Kalman gain matrix for a time invariant system such as the wheelset remains constant for the entire Kalman gain problem.

Taking into account the two simplifications listed above, the maximum likelihood equations reduce to:

$$L = -\frac{1}{2} \left[\sum_{i=1}^N v^T(t_i) B^{-1} v(t_i) + \ln|B| \right] \quad (1.69)$$

The constant term in equation 1.69 can be dropped because it shifts the likelihood function up or down but does not change the location of the maximum of the likelihood function.

$$\frac{\partial L}{\partial \underline{\theta}} = \sum_{i=1}^N v^T(t_i) B^{-1} \frac{\partial v(t_i)}{\partial \underline{\theta}} - \frac{1}{2} v^T(t_i) B^{-1} \frac{\partial B}{\partial \underline{\theta}} B^{-1} v(t_i) + \frac{1}{2} \text{tr} [B^{-1} \frac{\partial B}{\partial \underline{\theta}}] \quad (1.70)$$

$$\frac{\partial v(t_i)}{\partial \underline{\theta}} = -H \frac{\partial \hat{x}(t_i)}{\partial \underline{\theta}} \quad (1.71)$$

$$\frac{\partial B}{\partial \underline{\theta}} = H \frac{\partial P}{\partial \underline{\theta}} H^T \quad (1.72)$$

$$\begin{aligned} \frac{\partial^2 L}{\partial \underline{\theta}^2} = & \sum_{i=1}^N \frac{\partial v^T(t_i) B^{-1}}{\partial \underline{\theta}} \frac{\partial v(t_i)}{\partial \underline{\theta}} - (2) v^T(t_i) B^{-1} \frac{\partial B}{\partial \underline{\theta}} B^{-1} \frac{\partial v(t_i)}{\partial \underline{\theta}} \\ & - \frac{1}{2} \text{tr} [B^{-1} \frac{\partial B}{\partial \underline{\theta}} B^{-1} \frac{\partial B}{\partial \underline{\theta}}] \end{aligned} \quad (1.73)$$

$$\frac{\partial \hat{x}}{\partial \underline{\theta}} = F \frac{\partial \hat{x}(t)}{\partial \underline{\theta}} + \frac{\partial F}{\partial \underline{\theta}} \hat{x}(t) - KH \frac{\partial \hat{x}(t)}{\partial \underline{\theta}} - \frac{\partial K}{\partial \underline{\theta}} z(t) - \frac{\partial K}{\partial \underline{\theta}} H \hat{x}(t) \quad (1.74)$$

$$\frac{\partial \hat{x}(t_i)}{\partial \theta} = \frac{\partial \hat{x}(t_o + \Delta t)}{\partial \theta} = \phi(\Delta t) \frac{\partial \hat{x}(t_o)}{\partial \theta} + \Lambda \hat{x}(t_o) + T \underline{z}(t_o) \quad (1.75)$$

where

$$\phi(\Delta t) = e^{[F-KH]\Delta t} \quad (1.76)$$

$$\Lambda = \phi(\Delta t)[F-KH]^{-1} \cdot [I - \phi^{-1}(\Delta t)] \left[\frac{\partial F}{\partial \theta} - \frac{\partial K}{\partial \theta} H \right] \quad (1.77)$$

$$T = \phi(\Delta t)[F-KH]^{-1} \cdot [I - \phi^{-1}(\Delta t)] \left[\frac{\partial K}{\partial \theta} \right] \quad (1.78)$$

The term $\frac{\partial P}{\partial \theta}$ is the steady state solution to equation 1.79.

$$\frac{\partial P(t_o + \Delta t)}{\partial \theta} = [\phi_{yy}(\Delta t) + \phi_{yy}(\Delta t) \frac{\partial P(t_o)}{\partial \theta}] \cdot [\phi_{yy}(\Delta t) + \phi_{yy}(\Delta t) \frac{\partial P(t_o)}{\partial \theta}]^{-1} \quad (1.79)$$

Equation 1.79 is a time invariant form of equation 1.45 and since $\frac{\partial P}{\partial \theta}$ is time invariant for the wheelset problem, equation 1.79 is solved only once during each iteration of the maximum likelihood processor. Equation 1.79 is derived in Appendix D.

Equation 1.46 which defines $\frac{\partial K(t)}{\partial \theta}$ reduces to:

$$\frac{\partial K}{\partial \theta} = \frac{\partial P}{\partial \theta} \bigg|_{ss} H^T R^{-1} \quad (1.80)$$

1.3 Maximum Likelihood Algorithm

The equations of the preceding section were implemented into a Fortran IV computer program. This section discusses the algorithm used to solve the wheelset maximum likelihood problem on the digital computer.

The maximum likelihood method is a batch processor in that the entire set of measurements is used for each iteration of the maximum likelihood equations (8). Each iteration of the maximum likelihood algorithm produces an estimate of the value of the parameter to be identified which will maximize

the likelihood function given in equation 1.69. This estimate $\underline{\theta}^*$ then replaces $\underline{\theta}_0$ within the program and another iteration is made. When the following condition is met, iteration is stopped.

$$\underline{\theta}^* - \underline{\theta}_0 = \Delta \underline{\theta} < \epsilon \quad (1.81)$$

During the development phase of a maximum likelihood processor it is helpful to calculate the value of the likelihood function (L), the gradient ($\frac{\partial L}{\partial \underline{\theta}}$), and the Fisher information matrix ($\frac{\partial^2 L}{\partial \underline{\theta}^2}$) for a range of $\underline{\theta}$ where the maximum of the likelihood function is expected to lie. Changes in the flowchart of Figure 1.1 are shown in Figure 1.2. The flowchart at the bottom of the dotted box replaces the $\Delta \theta$ test at the end of the flowchart in Figure 1.1. This procedure is beneficial in that the maximum likelihood program can be checked out without the risk of an infinite loop being set up because of a convergence problem.

No mention has been made so far of the method used to calculate the state transition matrix (STM). On a digital computer, a very fast and accurate way to calculate the STM is through a power series expansion.

$$\Phi(\Delta t) = I + A\Delta t + \frac{1}{2!} A^2(\Delta t)^2 + \frac{1}{3!} A^3(\Delta t)^3 + \dots + \frac{A^i(\Delta t)^i}{i!} \quad (1.82)$$

The series is truncated when the following condition is satisfied

$$\left(\sum_{i=1}^{N+1} A^i(\Delta t)^i \frac{1}{i!} + I \right) - \left(\sum_{i=1}^N A^i(\Delta t)^i \frac{1}{i!} + I \right) < \epsilon \quad (1.83)$$

where ϵ is some error criterion chosen by the user. A value of

$$\epsilon = 1.0 \cdot 10^{-10} \quad (1.84)$$

is usually considered to be sufficient. A listing of the maximum likelihood program is contained in Appendix E.

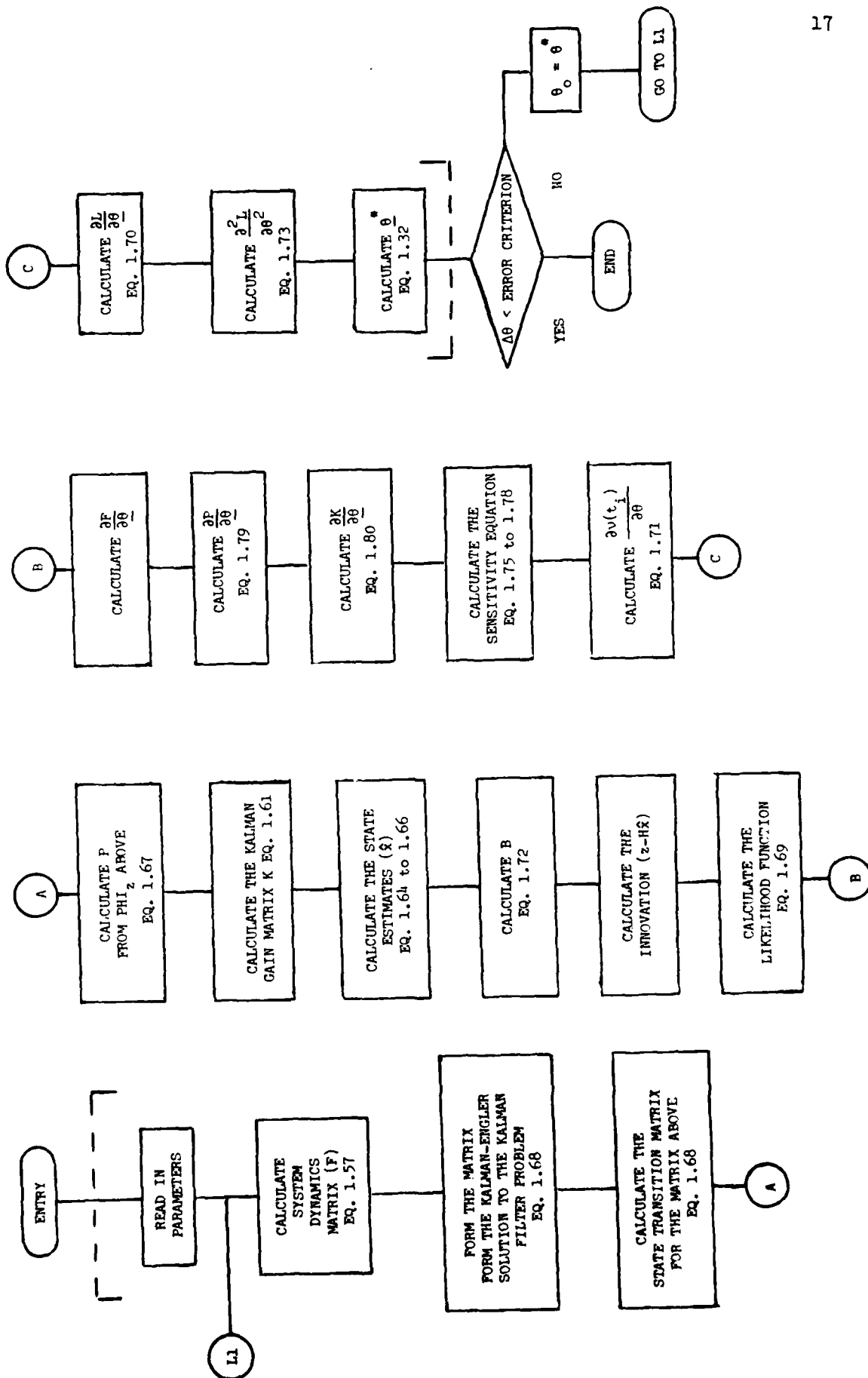


Figure 1.1 Maximum Likelihood Algorithm (8).

1.4 Test Data Generation

In order to ensure that the maximum likelihood computer program implement the maximum likelihood algorithm properly, it was tested using simulated wheelset data. The data was generated using the state transition matrix approach to solve the following equation.

$$\dot{\underline{x}} = \underline{F}\underline{x} + \underline{G}w \quad (1.85)$$

where F and G are specified by equations 1.57 and 1.49 respectively. Due to limitations in computer storage a total of 1000 data points for each state was all that could be generated.

The time step (Δt) was chosen by examining the eigenvalues of the F matrix. Typical values for the poles of the wheelset model are:

$$\begin{aligned} \sigma_1 &= -210.7 \text{ sec}^{-1} \\ \sigma_{2,3} &= -0.423 \pm j 5.982 \text{ sec}^{-1} \end{aligned}$$

These poles indicate that transient responses are characterized by a lightly damped sinusoid ($\omega_n = 6.0 \text{ sec}^{-1}$; $\zeta = 0.07$) plus a rapidly converging exponential decay ($\tau = \frac{1}{\sigma_1} = 4.7 \text{ msec}$). In the frequency domain a sharp resonant peak occurs at the natural frequency, with an additional breakpoint at $\omega = 1/\tau$.

As the estimates of the creep coefficients change, the damping ratio of the low frequency resonance changes, as does the location of the higher frequency real pole. In selecting the time interval ΔT and the record length $N\Delta T$ it is desirable for information on both high and low frequency modes to be present in the data. To obtain six data points per cycle at the higher frequency $f = \sigma_1/2\pi$ a ΔT is chosen to be 0.005 sec. Since computer storage constraints limited the record length to $N = 1000$ points, the record length is 5 seconds, which corresponds to 4.8 cycles of the low frequency mode.

Large values for Δt were tried but the state transition matrix approach used in the maximum likelihood program had convergence problems. A $\Delta t = .005$ was the largest time step that could be used without causing numerical problems.

In addition to the random track input $\underline{w}$ in equation 1.85, there was also measurement noise added to the state vectors.

$$\underline{z} = H\underline{x} + \underline{v} \quad (1.86)$$

$\underline{w}$ and $\underline{v}$ were generated by summing 10 random number vectors which were generated by an APL* operator. The random vectors generated by successive runs of this APL operator are not correlated. Ten of these random vectors were summed to force the distribution of $\underline{u}$ and $\underline{w}$ to be approximately Gaussian (1). Once these gaussian random vectors were formed they could be manipulated to force their means and mean squared values to equal what the user desired.

Although maximum likelihood theory requires $\underline{w}$ and $\underline{v}$ to be white noise because these vectors were of finite length they were bandwidth limited. Also since only 10 random vectors were summed, the probability distributions of $\underline{w}$ and $\underline{v}$ were not exactly gaussian, only approximately gaussian. A listing of the APL functions used to generate the simulated wheelset data is contained in Appendix F.

* APL: A Programming Language. Implemented on the Princeton University IBM 370/158 time-sharing system.

Chapter 2

EXPERIMENTAL PROGRAM

2.1 Objective

The experimental program was designed to make measurements of the wheelset state variables. The third order wheelset model has the following state variables:

$\dot{y}$ = lateral velocity

ψ = yaw angle

$(y-\bar{\delta})$ = wheelset to rail centerline relative lateral displacement

See Reference (11) for a detailed development of the wheelset equations of motion.

2.2 Experimental Facility

The entire experimental program was carried out on the Princeton Dynamic Model Track. Previous research at the Dynamic Model Track involved the development and validation of the dynamically scaled wheelset model and the measurement, restraint and propulsion system for the wheelset (9).

The experimental apparatus consisted of the wheelset model and the idler carriage. Both of these components rode on the 400 foot LEXAN track. The idler carriage surrounded the wheelset but was dynamically isolated from it. The idler carriage provided restraints to prevent the wheelset from completely derailing. The idler carriage was also part of the linkage that connects the wheelset to the propulsion unit. Figures 2.1 and 2.2 diagram the wheelset-idler carriage relationship.

The propulsion system was a hydraulically operated drive unit that rode along a steel I-beam guideway above the LEXAN track. The hydraulic drive unit had a feedback control system for maintaining a constant, user-selected velocity.

Suspended from the propulsion unit on instrument racks were the on-board analog computer, the bridge amplifiers, and the portable four track FM analog cassette recorder. The bridge amplifiers were used to amplify the output of the lateral accelerometer.

2.3 Transducers

Figure 2.3 is a diagram of transducer placement on the wheelset and Figure 2.4 traces the output of the transducers through the signal conditioning equipment to the recorder.

Lateral velocity could not be measured directly, therefore an accelerometer was used to measure lateral acceleration. A bridge amplifier was used to excite the differential type accelerometer and to amplify the output of the accelerometer. The lateral acceleration signal will have to be integrated on a digital computer to obtain lateral velocity.

The yaw angle was measured directly using a potentiometer. The equation relating potentiometer output (v) to the yaw angle (ψ) is:

$$\psi = mv + b \quad (2.1)$$

where m is the slope of the straight line described by this function and b is the y-intercept. The intercept b varied from day to day due to amplifier drift. A pre-run and post-run procedure was developed so that the value of b could be determined at the start of every day.

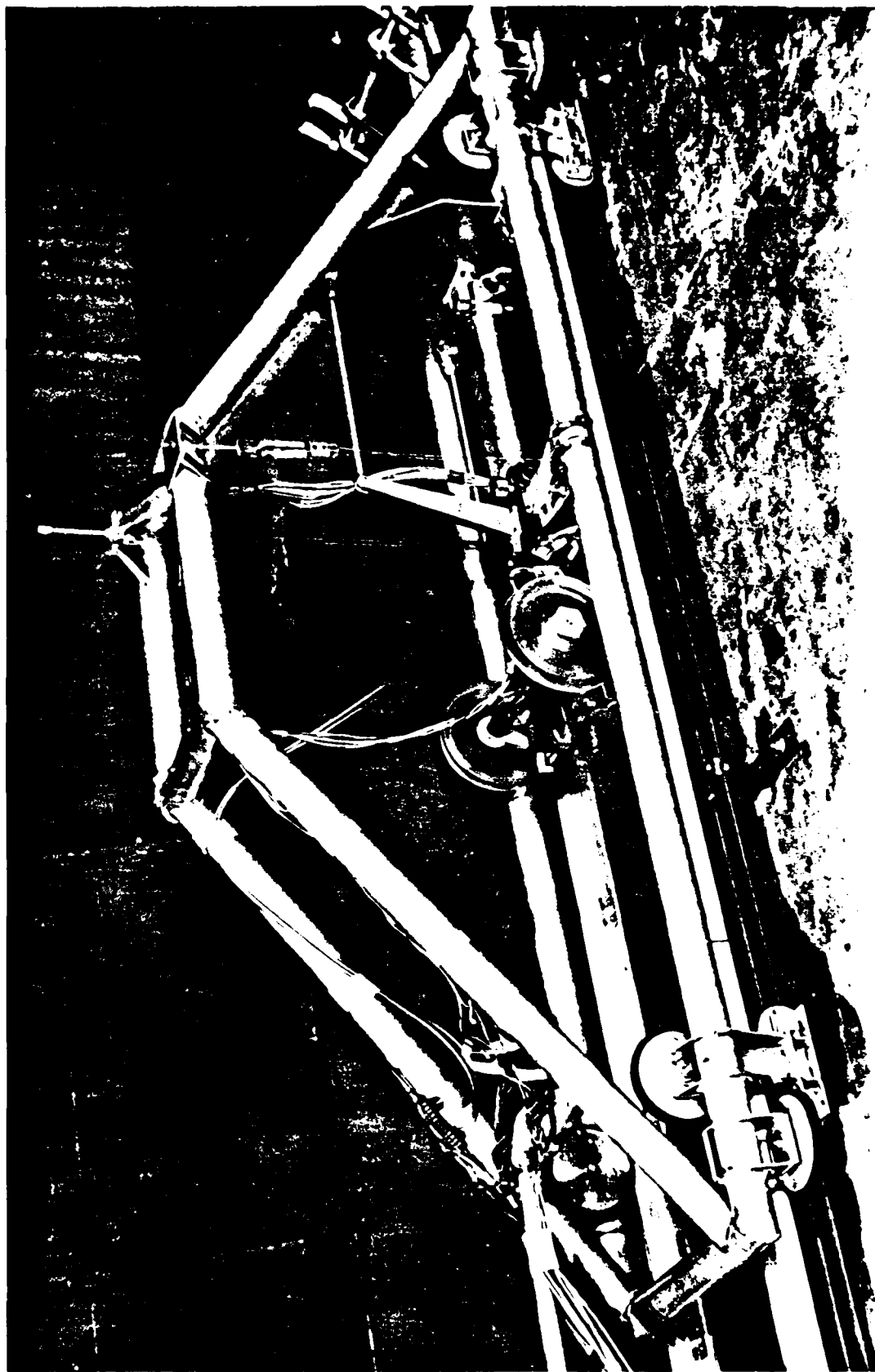


Figure 2.1a Side View of Wheel and
Carriage Relationship

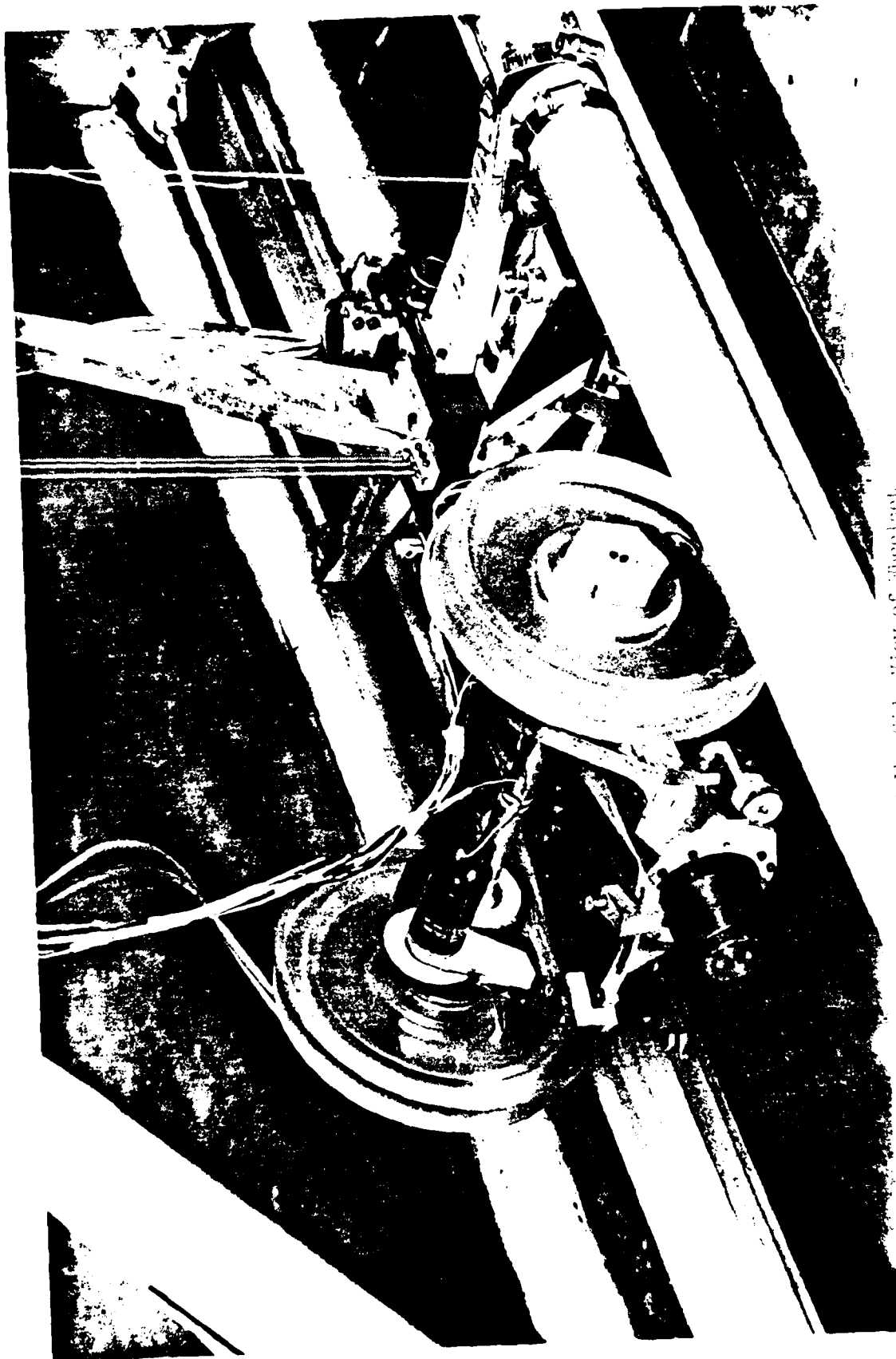


Figure 2.1b Side View of Mounting
Turret-Carriage Relat-
tionship

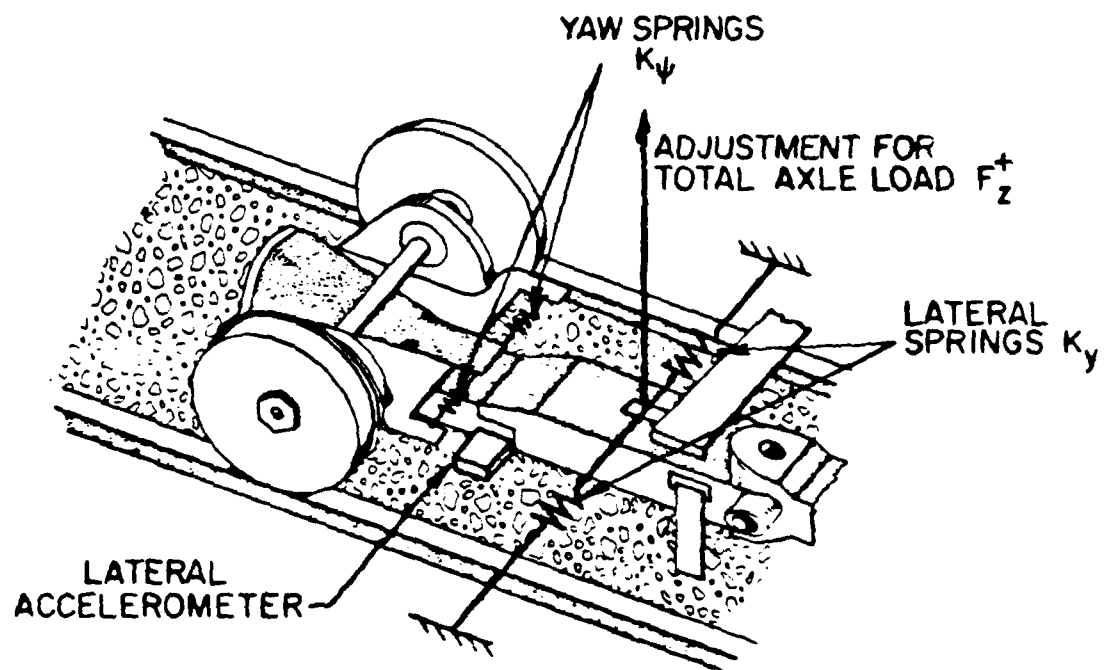


Figure 2.2 Top View of Wheelset Idler-Carriage Relationship.

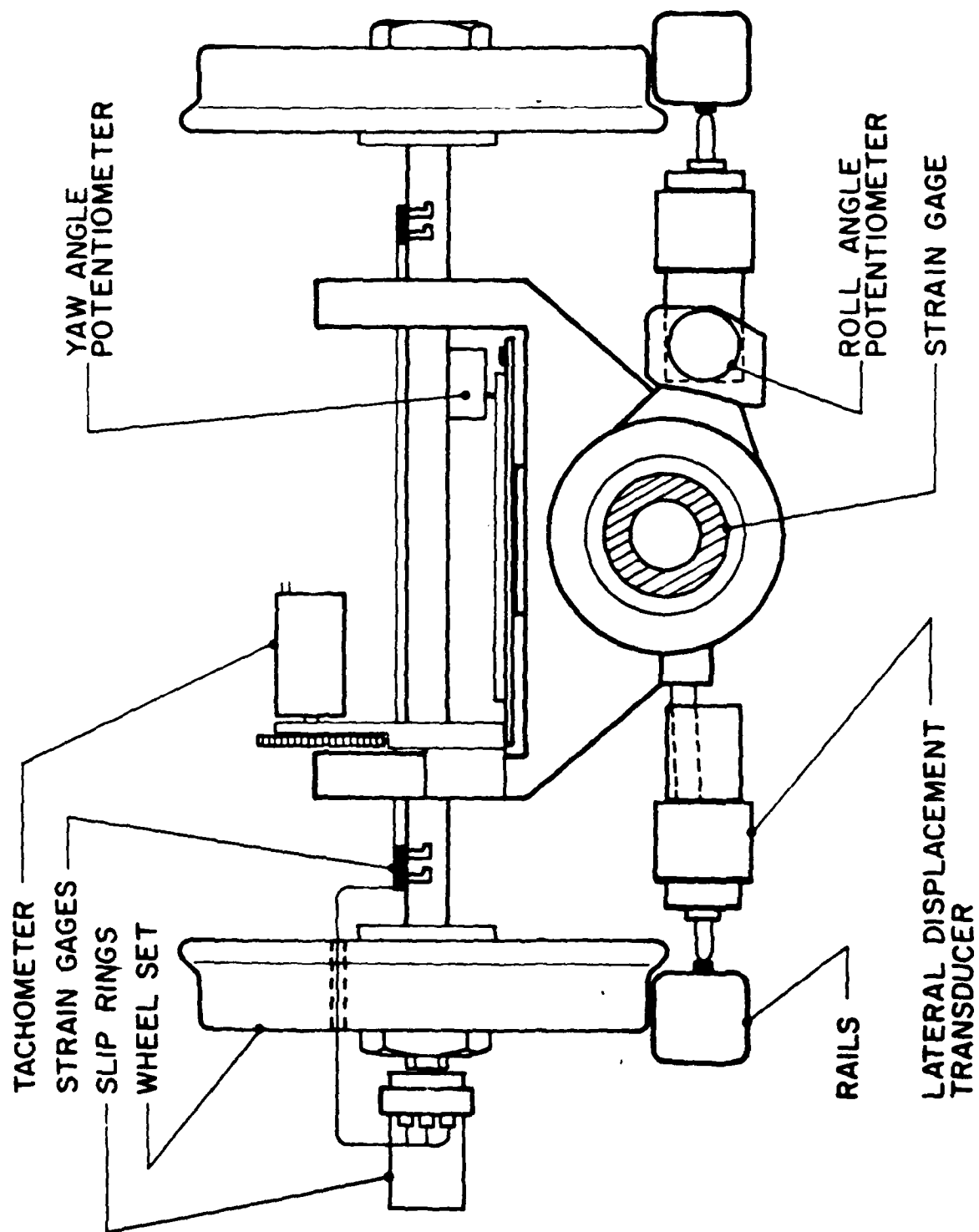


Figure 2.3 Wheelset Instrumentation.

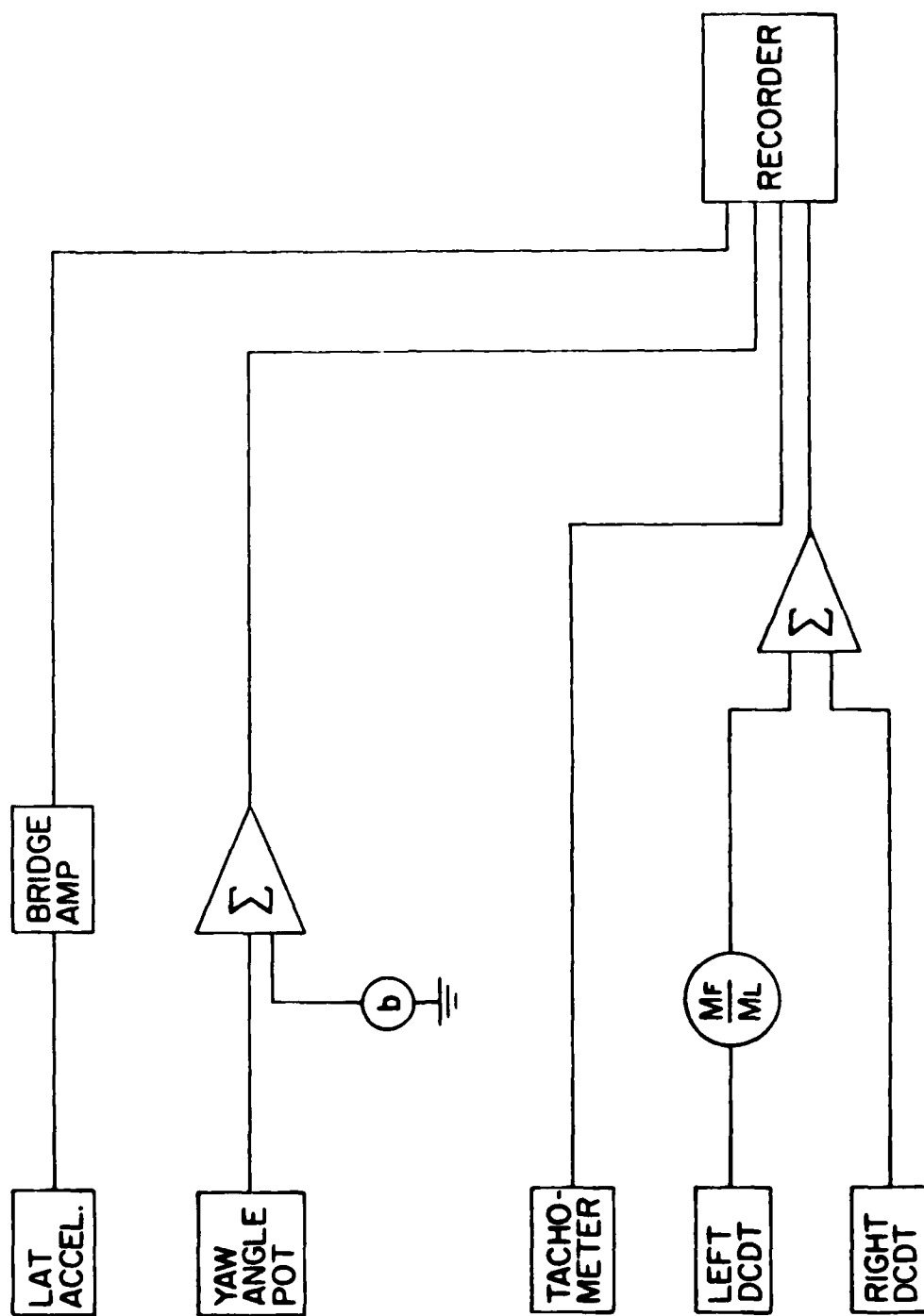


Figure 2.4 Transducer signal conditioning.

The variations in yaw angle during a run were smaller in magnitude than the value of the y-intercept. In order to obtain the best resolution on the cassette tape an electrical bias was summed with the yaw potentiometer output on the on-board analog computer. The value of the electrical bias was adjusted from day to day so that the output of the analog computer circuit equalled zero when the yaw angle was zero. The equation that describes the analog computer output is:

$$\eta = \frac{\psi - b}{m} + e \quad (2.2)$$

where η = the output of the analog computer

e = the electrical bias

A displacement transducer (DCDT) was mounted on each side of the wheelset. Each displacement transducer measured the distance of the wheelset from the inside edge of the track. Figure 2.5 diagrams the relationship between the variable measured by the DCDT's and the state variable $(y - \delta)$. The equations for the DCDT's are:

$$V_r = m_r (y - \delta_2) \quad (2.3)$$

$$V_L = m_L (y - \delta_1) \quad (2.4)$$

where V_r = voltage output of the right DCDT

V_L = voltage output of the left DCDT

m_r = slope of the straight line function

m_L = slope of the straight line function

Multiplying V_L by $\frac{m_r}{m_L}$ and adding V_r to V_L :

$$V_r + \frac{m_r}{m_L} V_L = m_r (y - \delta_r) + m_r (y - \delta_L) \quad (2.5)$$

This is the signal that was recorded during the experiment. The on-board

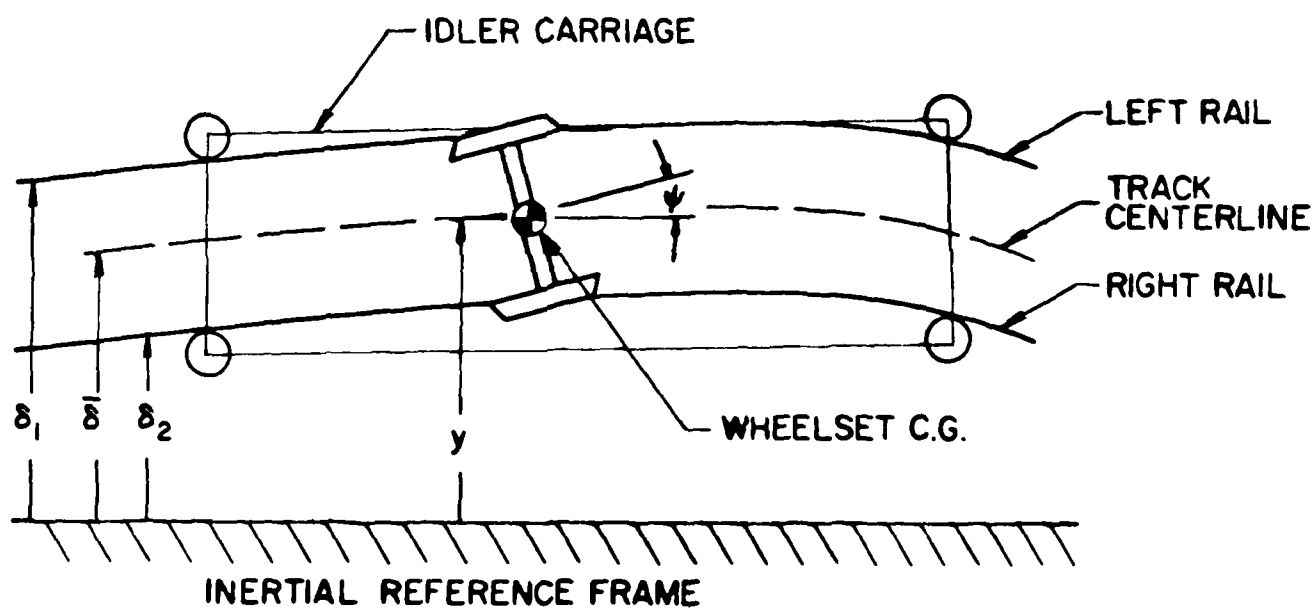


Figure 2.5 Definition of Lateral Displacement.

analog computer was used to multiply the left DCDT signal by $\frac{m_r}{m_L}$. The state variable $(y-\bar{\delta})$ can be derived from the recorded signal by dividing $V_r + \frac{m_r}{m_L} V_L$ by $2m_r$.

$$\frac{V_r + \frac{m_r}{m_L} V_L}{2 m_r} = \frac{2 m_r y - m_r (\delta_2 + \delta_1)}{2 m_r} \quad (2.6)$$

$$(y-\bar{\delta}) = y - \frac{(\delta_1 + \delta_2)}{2} = y - \bar{\delta} \quad (2.7)$$

The DCDT's have a linear range of operation which is much smaller than their total range of operation. The linear range of operation is slightly different for each DCDT as shown in Figures 2.6 and 2.7. Each DCDT was positioned so that its output was the center of its linear range when the wheelset was centered on the track. Because the DCDT's were not mounted symmetrically about the wheelset's longitudinal centerline, a bias was introduced into the measuring process.

$$V_L = m_L (y-\delta_L) + b \quad (2.8)$$

$$V_r = m_r (y-\delta_r) \quad (2.9)$$

where b = position bias

The same algebraic operations as previously yield:

$$\frac{m_r}{m_L} V_L = \frac{m_r}{m_L} m_L (y-\delta_L) + \frac{m_r}{m_L} b \quad (2.10)$$

$$V_r = m_r (y-\delta_r) \quad (2.11)$$

$$V_r + \frac{m_r}{m_L} V_L = m_r (y-\delta_L) + \frac{m_r}{m_L} b + m_r (y-\delta_r) \quad (2.12)$$

$$= 2m_r y - m_r (\delta_L + \delta_r) + \frac{m_r}{m_L} b \quad (2.13)$$

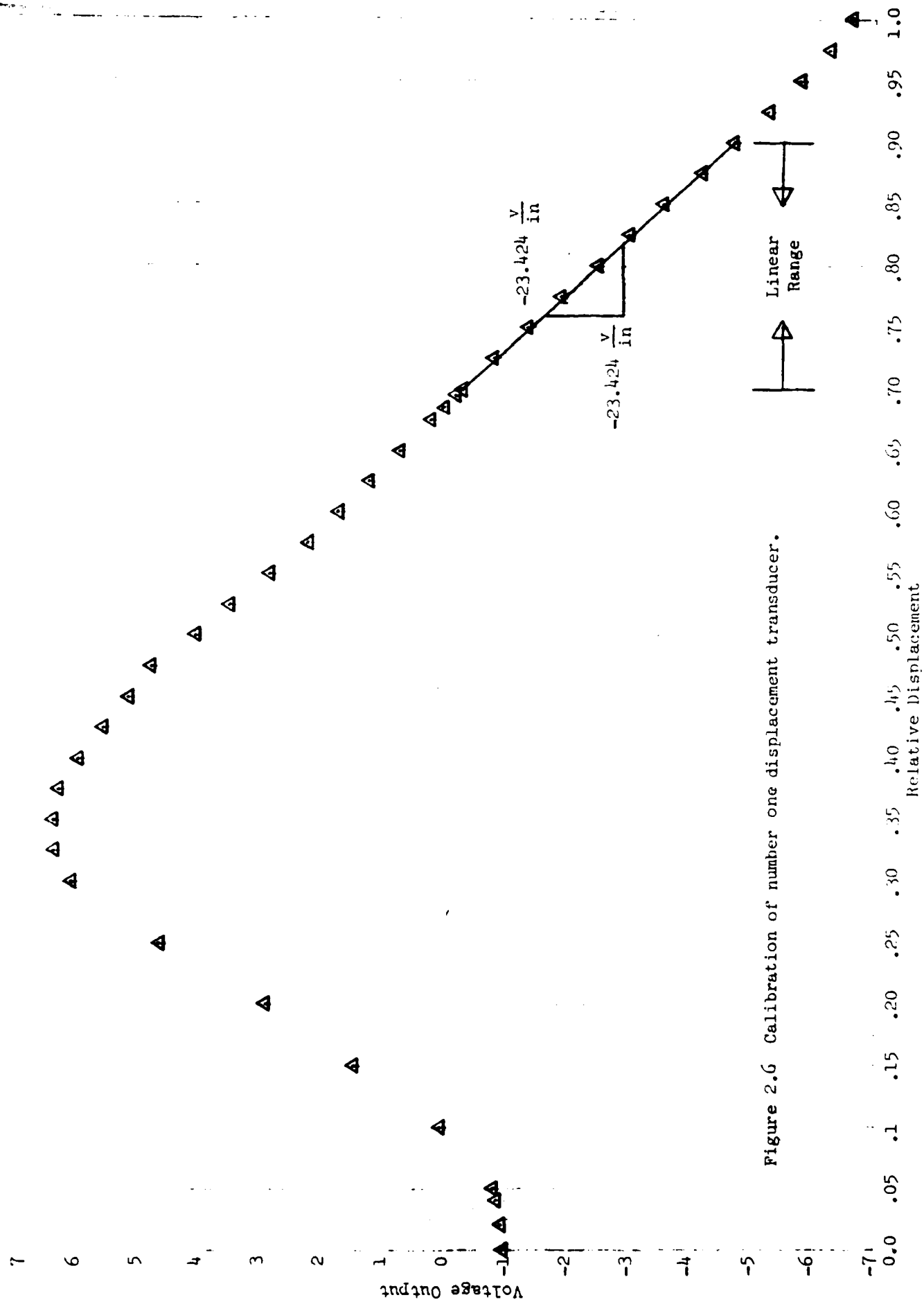


Figure 2.6 Calibration of number one displacement transducer.

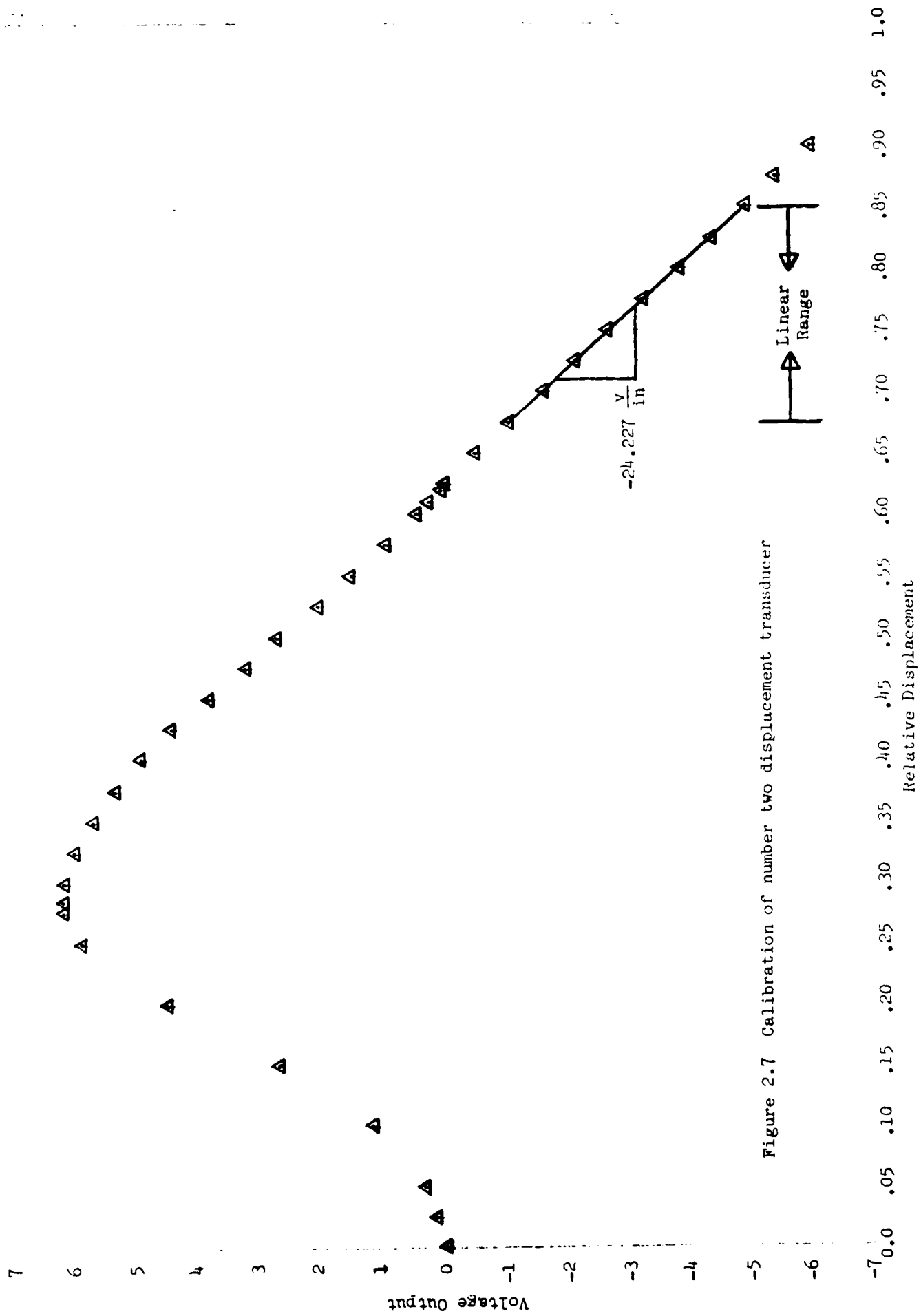


Figure 2.7 Calibration of number two displacement transducer

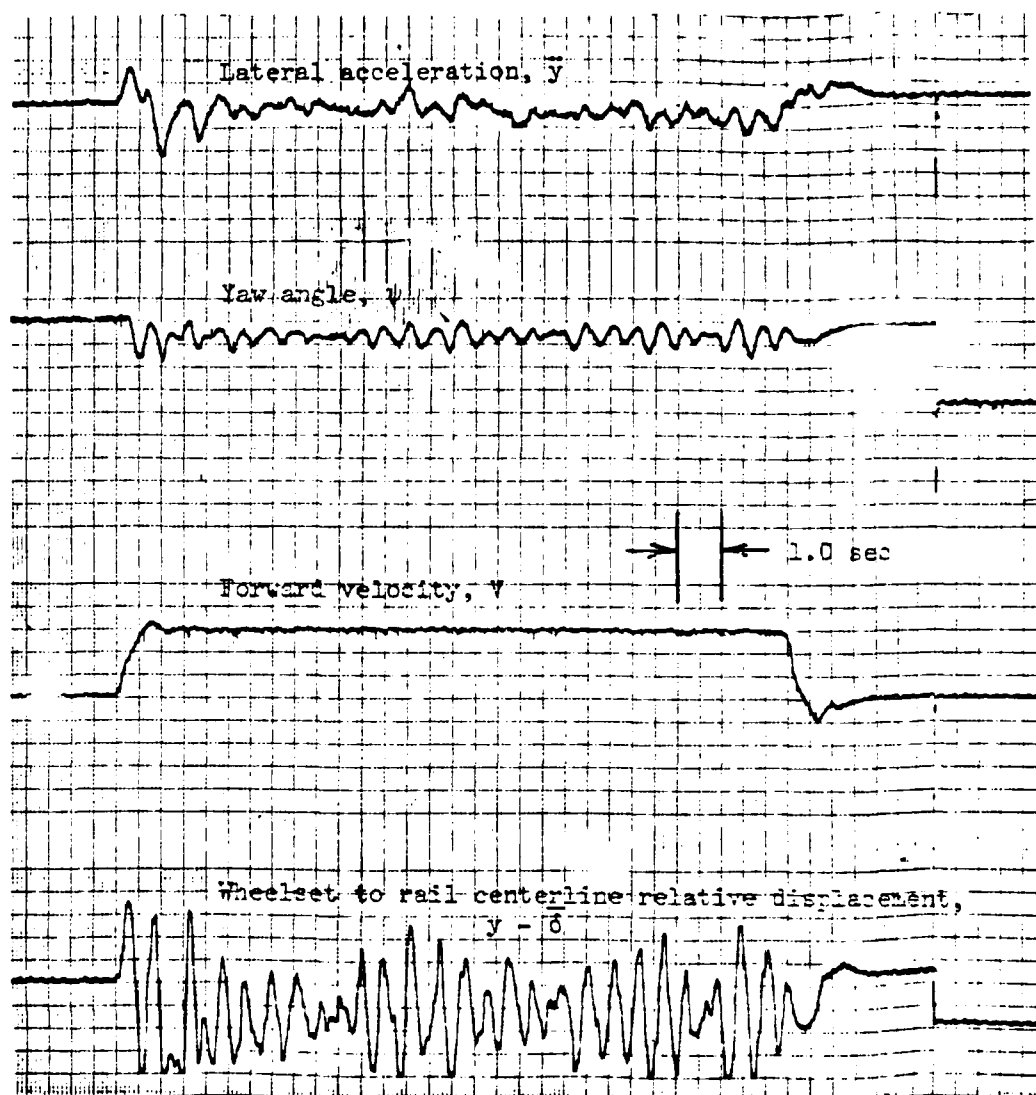


Figure 2.8 Typical transient response of wheelset to track inputs. Wheelset has lateral and yaw linear spring suspensions.

$$\frac{\frac{V_r + m_r V_L}{m_L}}{2m_r} = y - \frac{(\delta_L + \delta_r)}{2} + \frac{b}{2m_L} \quad (2.14)$$

$$= (y - \bar{\delta}) + \frac{b}{2m_L} \quad (2.15)$$

In addition to the three variables mentioned above, the wheelset axle angular velocity ($\ell = v/r_o$) in the rolling direction was also measured. Although the angular velocity is not a state variable, it is necessary to construct the velocity dependent state matrix F. The output of the tachometer was recorded directly with no processing on the on-board analog computer as with the lateral displacement and yaw angle signals. Sample responses from the above transducers are given in Figure 2.8.

2.4 Experimental Procedure

All of the above mentioned signals were recorded on a portable, four track FM analog cassette type recorder* which was carried on board the drive unit. The recorder was started and stopped manually before and after each run.

The bridge amplifier that powered the accelerometer was balanced at the beginning of each day to eliminate the drift in amplifier output that occurred from day to day.

The on-board analog computer had a built-in calibration circuit for the accelerometer bridge amplifier. This calibration circuit produced a known voltage that appeared to the accelerometer as an acceleration load. This acceleration load remained constant from day to day. In order to determine the input-output relationship of the accelerometer, the accelerometer output

* Philips Mini Log 4 Portable Analog Cassette Recorder.

was recorded during the calibration. This calibration test was performed before every data run to account for amplifier drift between data runs.

2.5 Measurement Noise

There were two sources of uncertainty in the data. The first of these was electrical noise. The bridge amplifier, analog computer and hydraulic pump for the propulsion unit all operate on 400 cycle alternating current.

Vibration of the cassette recorder during data runs was another source of noisy data. The steel I-beam guideway that supported the hydraulic drive unit had gaps between the I-beams. The vibration that resulted when the drive unit wheels hit these gaps was transmitted to the recorder, and showed up as noise on the data tapes. As mentioned earlier, the recorder was suspended from the drive unit on an instrument rack which had shock absorbers to help isolate the recorder from vibration. Nonetheless, some vibrations still affected the recorder and added noise to the data.

In addition to the two definite sources of noise, there is a third possible source of noise. It is known that there are small gaps between lengths of LEXAN rail. These small gaps are the result of contraction and expansion of the LEXAN material due to temperature changes. The 800 foot building which houses the Princeton Dynamic Model Track does not have a completely controlled environment; therefore, temperature variations of as much as 50°F are possible. Although no quantitative analysis has been carried out to verify this hypothesis, it is possible for the wheelset structural modes to be excited when the wheels hit these gaps. The transducers mounted on the wheelset, especially the lateral accelerometer, could detect this vibration, thereby adding noise to the data.

Finally to insure the validity of the data, the LEXAN track and wheels were cleaned before every run. Dirt and dust on the track or wheels could alter rail/wheel adhesion, in which case the wheelset model would be invalid. Absolute methanol was used as the cleaning agent because it left no residue on the track.

2.6 Filtering

To remove the noise due to the electrical system and vibration of the wheelset structural modes all of the signals were filtered off-line. The acceleration signal was filtered with a second order low-pass filter. The yaw angle and lateral displacement signals were filtered using seventh or eighth order band-pass filters, and the rotational velocity signal was filtered using a first order low-pass filter. The cutoff frequencies for the filters are given in Table 2.1.

Table 2.1
Filter Cutoff Frequencies

Signal	Filter Type	High Pass Cutoff	Low Pass Cutoff
Lateral Acceleration	low-pass	N/A	49.9 Hz
Yaw Angle	band-pass	.1 Hz	60 Hz
Rolling Angular Velocity	low-pass	N/A	20 Hz
Wheelset to Rail Relative Lateral Displacement	band-pass	.1 Hz	60 Hz

2.7 Digitization

After filtering the data had to be digitized so it could be stored and analyzed on a digital computer. As mentioned in the previous section, the

lowest permissible sampling rate was determined when the low-pass filter cutoff frequencies were chosen. It was decided to use a sampling rate of 120 samples/second per channel. The A/D converter used for digitization was a 12 bit sample and hold A/D converter.* This A/D converter was controlled by a minicomputer system.** The minicomputer had two memories for storing data (memory A and memory B) each of which had the capacity to store 512 data points. When the digitizing process started, the computer stored the digitized data in memory A. Once memory A was full the computer began filling memory B while at the same time writing the data in memory A to magnetic tape. Since the A/D converter was a sample and hold type, there was only a several nanosecond time delay between the sample for each channel. However, the switching process from memory A to memory B took 3 msec., therefore the sampling interval between the 512th data point and the 513th data point was 11.33 msec. The sampling interval between all other sets of four data points was 8.33 msec. Figure 2.8 depicts this shift in sampling interval. Appendix G is a listing of the program that controlled the A/D converter during the digitizing process.

* Preston GMAD-1 Analog to Digital Conversion System.

** Hewlett Packard HP 1000 System.

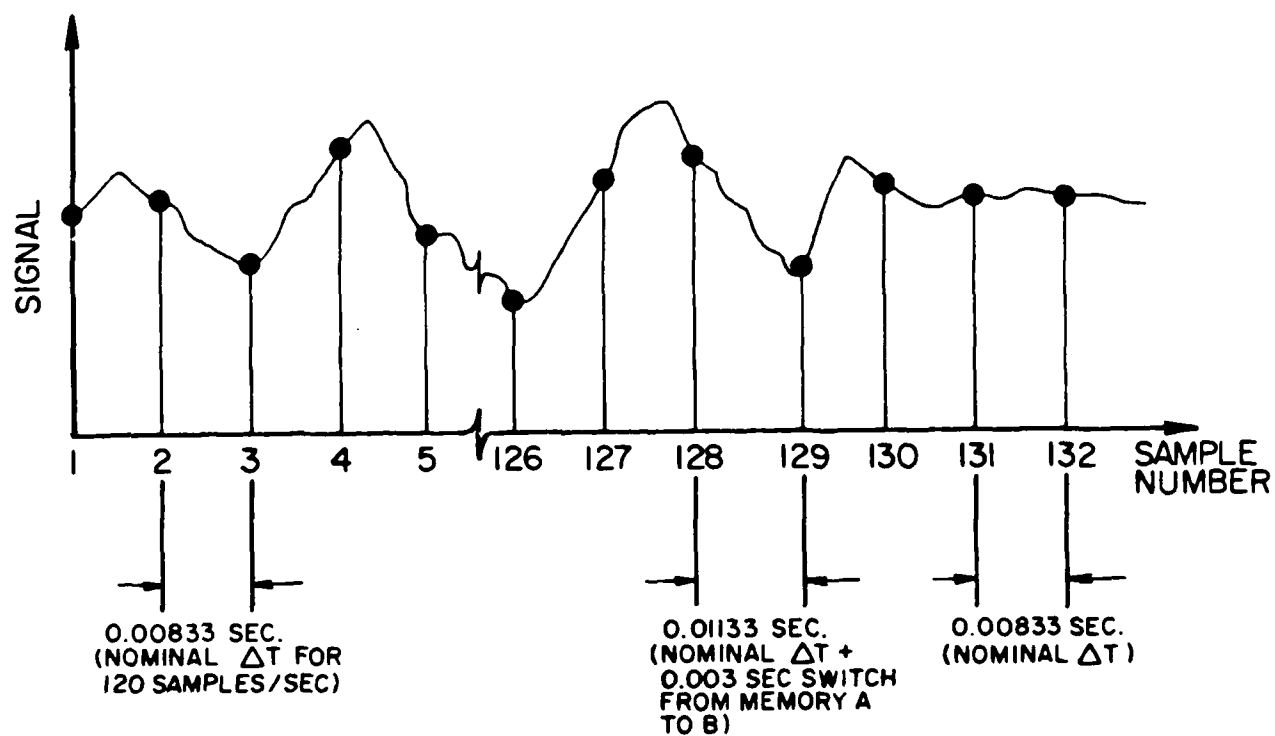


Figure 2.9 Effect of memory switching on sampling rate.

Chapter 3

RESULTS

3.1 Outline of Test Cases

In this chapter the results obtained from the maximum likelihood computer program using the simulated data will be presented. As mentioned in Chapter 1, the AFL program which generated the simulated data had provisions for any combination of deterministic input, random input, initial condition and measurement noise. Table 3.1 is a summary of the maximum likelihood program testing.

3.1.1 Explanation of Test Case Data

The LEXAN track lateral alignment is ideally a white noise velocity input, with a gaussian probability density. Because of these characteristics the track input does not have to be measured but can be described as a random input into the wheelset model. In this case, it is the Q matrix which gives the maximum likelihood processor all of its information about the track input. It is also possible to measure the track input, although there would be some uncertainty involved. If the track input were measured, then the measured values for the track input would become a deterministic input and the uncertainty in the track measurements would be described as a random track input.

For the first seven test cases the track input is treated as a random track input, with the only information describing it contained in the Q matrix.

In test cases 7 and 8 the track input is treated as a deterministic input. The random track input for these cases represents the uncertainty in the deterministic track input.

Table 3.1
Summary of Maximum Likelihood Program Testing

Test Case	Deterministic Track Input (u)	Initial Condition *	Random Track Input (w)		Measurement Noise (v)	
			Mean	Squared Value	Mean	Squared Value
1	0	$x(1)=x(2)=x(3)=10$	0	0	0	0
2	0	$x(1)=x(2)=x(3)=0$	0	1.0	0	0
3	0	$x(1)=x(2)=x(3)=10$	0	0	0	.01
4	0	$x(1)=x(2)=x(3)=0$	0	1.0	0	.01
5	0	$x(1)=x(2)=x(3)=10$	0	1.0	0	.01
6	0	$x(1)=x(2)=x(3)=0$	0	100.0	0	1.0
7	0	$x(1)=x(2)=x(3)=0$	0	100.0	0	.01
8	$E[u] = 0$ $E[u^2] = 100$	$x(1)=x(2)=x(3)=0$	0	.1	0	.1
9	$E[u] = 0$ $E[u^2] = 100$	$x(1)=x(2)=x(3)=0$	0	10.0	0	.1

* $\begin{bmatrix} \dot{y} \\ \psi \\ (y-\bar{y}) \end{bmatrix}$

$\underline{x} =$

In terms of generating the simulated wheelset data, the two representations of the track input are handled in the following way. For the case when the track input is considered to be a totally random input, a random vector is used as the input to the system model which generates the data. When this data is processed, the maximum likelihood program (specifically the Kalman Filter) is not told what the random vector was, the only information it is given is the covariance matrix for the vector Q .

When the track input is considered to be deterministic, the system model which generates the data has two random vectors as inputs. One of these random vectors represents the deterministic input; the other represents the uncertainty in the deterministic input. This time when the data generated by the system model are analyzed by the maximum likelihood processor, the Kalman Filter is given the vector that represents the deterministic input, but it is given only the covariance matrix for the vector that represents the uncertainty in the deterministic input. Figure 3.1 diagrams the generation of the simulated data for both representations of the track input. This figure is not designed to represent the entire process for generating simulated data (there can be measurement noise) nor is it designed to represent all of the inputs that go into the maximum likelihood processor. Its only purpose is to demonstrate the variations in representing the track input.

3.1.2 Purpose of Each Test Case

As stated previously, test cases 8 and 9 are different from the other seven test cases in how the track input is represented. Test case 9 is different from test case 8 in that the level of uncertainty in the deterministic track input is much higher. These two cases give some indication of how well the real track input would have to be measured to significantly increase the

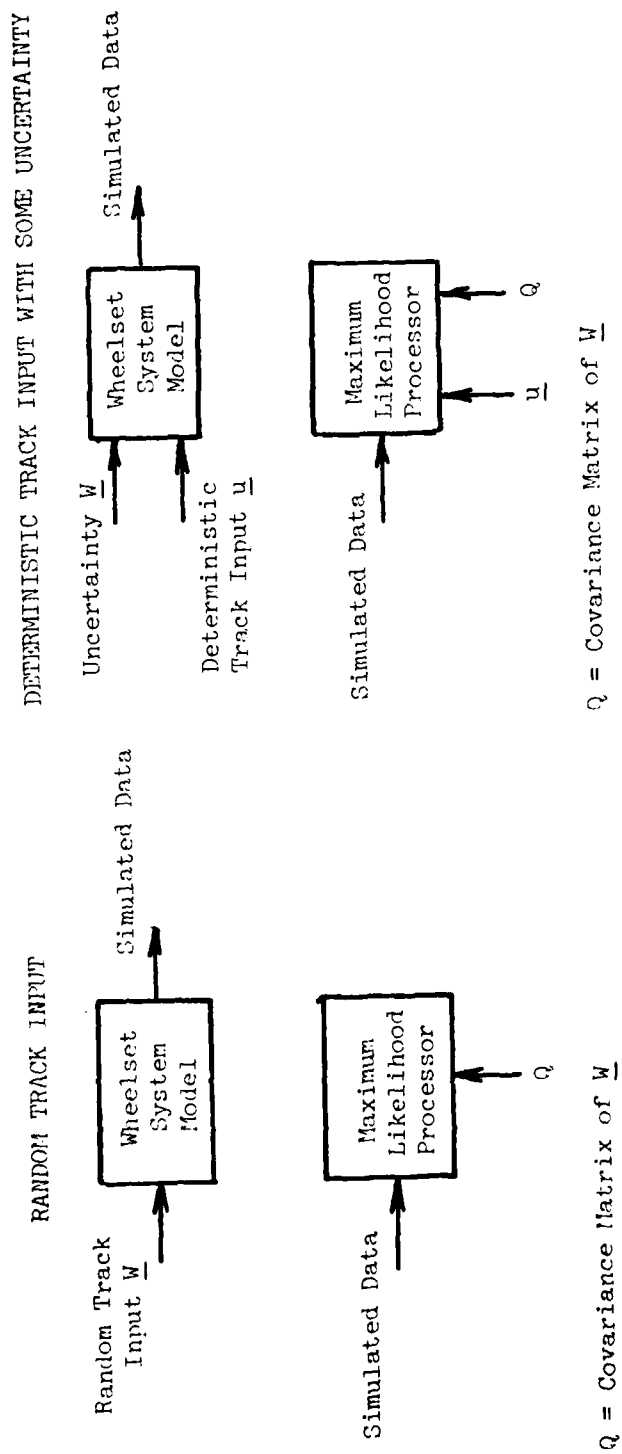


Figure 3.1 Representations of Track Input.

performance of the maximum likelihood processor over the case when the track is considered to be a random track input.

Test cases 1 through 5 were run to find out how various combinations of initial condition, random track input and measurement noise affect the maximum likelihood processor. Test case 1 through 3 were also run to find out how well that maximum likelihood processor performs when it is given incorrect information about the random track input and the measurement noise. In other words the simulated data for test cases 1 through 3 was generated with a random track input and measurement noise that had statistics as given in Table 3.1. However, when this data was analyzed using the maximum likelihood processor, the Kalman Filter was given incorrect information about the statistics of the $\underline{w}$ and $\underline{y}$ used to generate the data. This incorrect information took the form of an incorrect Q and R matrix. Table 3.2 shows the true Q and R matrices for $\underline{w}$ and $\underline{y}$ versus the Q and R matrices actually used by the Kalman Filter.

3.2 Method of Testing the Maximum Likelihood Parameter

Essentially the maximum likelihood program has 3 main parts. The first of these calculates the likelihood function based on the equation

$$L = -\frac{1}{2} \sum_{i=1}^N (\underline{v}^T(t_i) \underline{B}^{-1} \underline{v}(t_i) + \ln |\underline{B}|) \quad (3.1)$$

The likelihood function has two main components as can be seen in equation 3.1.

$$\xi_{\text{observation}} = -\frac{1}{2} \sum_{i=1}^N \underline{v}^T(t_i) \underline{B}^{-1} \underline{v}(t_i) \quad (3.2)$$

$$\text{and} \quad \xi_{\text{bias}} = -\frac{1}{2} \sum_{i=1}^N \ln |\underline{B}| \quad (3.3)$$

Table 3.2
Q and R Matrices for Test Cases 1 Through 5

Test Case	Random Track Input $\underline{w}$		Correct Value of Q for $\underline{w}$	Value of Q Actually Given to Maximum Likelihood Program	Measurement Noise $\underline{v}$		Correct R Matrix for $\underline{v}$	R Matrix Actually Given to Maximum Likelihood Processor
	Mean	Squared Value			Mean	Squared Value		
1	0	0	0	1.0	0	0	0	.01
2	0	1.0	1.0	1.0	0	0	0	.01
3	0	0	0	1.0	0	.01	.01	.01

The second part of the maximum likelihood equation calculates $\frac{\partial L}{\partial \theta}$ which is referred to as the gradient, and the third part of the program calculates $\frac{\partial^2 L}{\partial \theta^2}$ known as the Fisher Information Matrix. The inverse of the Fisher Information Matrix is defined as the Cramer-Rao Lower Bound on the covariance of the θ estimates. The maximum likelihood approaches the lower bound asymptotically (8). The gradient and the Fisher Information Matrix are used to calculate the estimate of the value of the parameter to be identified, according to the equation

$$\theta^* = \theta_0 - \Delta\theta \quad (3.4)$$

where $\Delta\theta = M^{-1} \left[\frac{\partial L}{\partial \theta} \right]^T$ and $M = \frac{\partial^2 L}{\partial \theta^2}$

As stated in Chapter 1, the maximum likelihood technique involves iterating through equations for L , $\frac{\partial L}{\partial \theta}$, $\frac{\partial^2 L}{\partial \theta^2}$, and θ^* until $\Delta\theta$ becomes smaller than some error criterion (ϵ). According to maximum likelihood theory as $\epsilon \rightarrow 0$, θ^* approaches the true identity of the unknown parameter. At the true value for θ^* the likelihood function has its maximum, and according to maximum likelihood theory, for the technique to converge the likelihood function must have a maximum. Because it was not certain that the likelihood function for all the test cases had a maximum, the maximum likelihood processor was not allowed to converge. Instead the likelihood function was given specific values of theta for which to calculate L , ξ_{bias} , $\frac{\partial L}{\partial \theta}$ and $\frac{\partial^2 L}{\partial \theta^2}$. Theta ranged from .50 to 1.45 in increments of .05. In all of the simulated data generated, the following value for the unknown parameter was used:

$$\beta = 1.00$$

therefore the peak in the likelihood function for all cases should be at

$$\theta = \beta = 1.00$$

For each value of theta in the range specified above, the maximum likelihood

program iterated through its equations once calculating:

- (1) L
- (2) ξ_{bias}
- (3) $\xi_{\text{observation}}$
- (4) $\frac{\partial L}{\partial \theta}$
- (5) $\frac{\partial^2 L}{\partial \theta^2}$
- (6) $\Delta \theta$
- (7) θ^*

Figure 3.2 is a sample of the computer program output for one iteration. After the maximum likelihood program finished the iteration for $\theta = 1.45$ the five quantities listed above are plotted for the range of theta.

3.3 Test Case Results

A summary of the results of the maximum likelihood program for test cases 1 through 9 are given in Table 3.3. These results will be discussed in three sections. The first section will discuss the likelihood function for each test case. The second section will discuss the results for the gradient and the Fisher Information Matrix and the third will examine the application of the test data results to the analysis of real wheelset data.

3.3.1 Likelihood Function

In this section the test cases will be discussed in three groups as indicated on the right hand side of Table 3.1.

Examining test cases 8 and 9 first, Table 3.1 shows that a deterministic track input as well as a random track input was used to generate the simulated

Figure 3.2 Computer output for the last iteration of test case number 8.

```

ISTEP= 20
THETA= 1.44999313

P MATRIX
-0.21115749E 03 0.61370386E 03 -0.12370279E 03
0.0 -0.42806154E 00 -0.12261953E 02
0.10000000E 01 0.0 0.0

Z MATRIX
0.21115749E 03 0.0 -0.10000000E 01
-0.61370386E 03 0.42806154E 00 0.0
0.12370279E 03 0.12261953E 02 0.0
0.64185696E 01 0.0 -0.80115962E 00
0.0 0.0 0.0
-0.80115962E 00 0.0 0.99999964E-01

19 ITERATIONS FOR STM.

STM DIFF MATRIX
0.0 -0.33881318D-20 0.0
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.67762636E-20 -0.54210109D-19
0.33881318D-20 0.0 0.0
0.0 0.0 0.0

KALMAN FILTER STM (PHIZ)
0.28725697D 01 -0.22740329E-03 -0.89864179D-02
-0.54506479D 01 0.10723531D 01 0.11381150D-01
0.95935641D 00 0.61336288D-01 0.99793038D 00
0.34601702D-01 -0.93414442D-04 -0.26461181D-02
0.17454588D-03 -0.31450230E-06 -0.15427495D-04
-0.68148958D-02 0.15442454E-04 0.50691631D-03

200 ITERATIONS. RICCATI SOLUTION DOES NOT CONVERGE.

RICCATI DIFFERENCE MATRIX.
0.78744964D-04 0.22717645E-04 -0.20399137D-04
0.22717645D-04 0.69047202D-05 -0.61998646D-05
-0.20399137D-04 -0.61998646E-05 0.55683564D-05

STATE COVARIANCE MATRIX (P)
0.42905962E 00 0.1360687CE 00 -0.19712276E-01
0.13606870E 00 0.4515712CE-01 -0.10056686E-01
-0.19712276E-01 -0.10056686E-01 0.16461197E-01

```

Figure 3.2 (continued)

KALMAN GAIN MATRIX (RKGM)
 0.42905970E 01 0.13606873E 01 -0.39712286E 00
 0.13606873E 01 0.45157129E 00 -0.10056686E 00
 -0.39712286E 01 -0.10056686E 00 0.16461200E 00

19 ITERATIONS FOR STM.

STM DIFF MATRIX
 -0.27755576D-16 0.0 0.0
 0.67762636D-20 0.0 0.0
 0.54210109D-19 -0.86736174D-18 0.0

SYSTEM STATE TRANSITION MATRIX (FHX)
 0.34703362E 00 0.18917809E 01 -0.44986582E 00
 -0.11093885E-01 0.99773932E 00 -0.61217483E-01
 0.30864351E-02 0.55509247E-02 0.99875748E 00

KALMAN FILTER SYSTEM MATRIX (FKF)
 -0.21544807E 03 0.61234302E 03 -0.12330566E 03
 -0.13606873E 01 -0.87963283E 00 -0.12161386E 02
 0.13571224E 01 0.10056686E 00 -0.16461200E 00

19 ITERATIONS FOR STM.

STM DIFF MATRIX
 -0.27755576D-16 0.0 0.0
 0.0 0.0 0.0
 0.0 -0.86736174E-18 0.0

STM FOR KALMAN FILTER SYSTEM (PHIK)
 0.33408942D 00 0.18615725E 01 -0.44243963D 00
 -0.42891837D-02 0.98796747D 00 -0.58943211D-01
 0.42553449D-02 0.81702378E-02 0.49744722D 00

MEASUREMENT COVARIANCE MATRIX (E)
 0.52905959E 00 0.13606870E 00 -0.39712276E-01
 0.13606870E 00 0.14515704E 00 -0.10056686E-01
 -0.39712276E-01 -0.10056686E-01 0.11646116E 00

NATURAL LOG OF THE DETERMINANT OF B= -0.50185900E 01

VALUE OF THE LIKELIHOOD FUNCTION (RLINE)
 RLINE= 0.14254326E 04

PARTIAL OF P WRT THETA (DPHTX)
 -0.14010136E 03 0.42124609E 03 0.0
 0.0 0.29521612E 03 0.0
 0.0 0.0 0.0

2 MATRIX FOR SOLUTION TO PARTIAL(P)/PARTIAL(THETA)
 0.21544807E 03 0.13606873E 01 -0.11971224E 01
 -0.61234302E 03 0.87963283E 00 -0.10056686E 00
 0.12130566E 01 0.12161386E 02 0.16461200E 00
 -0.50425720E 01 0.49333347E-01 0.13072910E 01 -0.21544407E 03 0.61234302E 03 -0.12330566E 03

Figure 3.2 (continued)

```

0.49333075E-01  0.26662237E-01  -0.29688976E-02  -0.13606073E 01  -0.87963283E 00  -0.12161386E 02
0.13072910E 01  -0.29688976E-02  0.0  0.13971224E 01  0.10056686E 00  -0.16461200E 00

```

6

```

&DINEN
N=
&END

```

19 ITERATIONS FOR STM.

STM DIAP MATRIX

```

0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0
0.0 -0.54210109D-19 0.0 -0.27755576D-16 0.0 0.0
-0.54210109D-19 0.0 0.0 0.0 0.0 -0.86736174D-18
0.0 0.0 0.0 0.0 0.0 0.0

```

STM FOR SOLUTION TO PARTIAL(P)/PARTIAL(THETA)

```

0.29103936D 01 0.1188656E-01 -0.1251381E-01 0.0 0.0 0.0
-0.54918858D 01 0.9892508E 00 0.1532659D-01 0.0 0.0 0.0
0.96643113D 00 0.6373233E-01 0.3979142D 00 0.0 0.0 0.0
-0.30804079D-01 0.4726974E-03 0.4068025E-02 0.3340894D 00 0.18615725D 01 -0.44243963D 00
0.29357490D-03 0.13265425E-03 -0.32378697D-04 -0.42891837D-02 0.98796747D 00 -0.58943211D-01
0.11635022D-01 0.19156478E-04 -0.17233599D-04 0.42553489E-02 0.81702378D-02 0.99744722D 00

```

200 ITERATIONS. RICCATI SOLUTION DOES NOT CONVERGE.

RICCATI DIFFERENCE MATRIX.

```

0.63945955D-05 0.20625284E-05 -0.91290541D-06
0.20625284E-05 0.66859567E-06 -0.27106848D-06
-0.91290541D-06 -0.27106848D-06 0.29388600D-06

```

PARTIAL OF P WRT THETA (DELTA P)

```

0.45337927E-01 0.20144747E-01 0.40192576E-03
0.20144747E-01 0.67489929E-02 -0.16518242E-02
0.40192576E-03 -0.16518242E-02 0.22276950E-02

```

PARTIAL OF MAGN WRT THETA (DELTA M)

```

0.45330436E 00 0.20144748E 00 0.40192567E-02
0.20144748E 00 0.67489922E-01 -0.16518246E-01
0.40192567E-02 -0.16518246E-01 0.22276953E-01

```

GRADIENT - PARTIAL OF THE LIKELIHOOD FUNCTION WRT THETA

```
DELTAJ= C.84175674E 02
```

FISHPR INFORMATION MATRIX

```
DELJ2= 0.82025391E 02
```

```
DTHETA=-0.10262146E 01
```

```
THETA= C.42377853E 00
```

Table 3.3

Summary of Results for Test Cases

Test Case	Likelihood Function Maximum θ =	Observation Term Maximum θ =	Bias Term Maximum θ =
1	.95	.95	N/A
2	N/A	.95	N/A
3	.95	.95	N/A
4	N/A	N/A	N/A
5	.95	.95	N/A
6	N/A	N/A	N/A
7	.65	N/A	N/A
8	.85	1.10	N/A
9	N/A	N/A	N/A

NOTE: N/A = not applicable, no maximum occurs or only a local maximum occurs.

wheelset data. As presented in Chapter 1, the wheelset equations of motion in state vector form are:

$$\dot{\underline{x}} = \underline{F}\underline{x} + \underline{G}\underline{w} \quad (3.5)$$

where $\underline{w}$ = random track input

The Kalman Filter state estimates for this case are given by the equation

$$\hat{\underline{x}} = (\underline{F}-\underline{K}\underline{H})\hat{\underline{x}} + \underline{K}\underline{z} \quad (3.6)$$

to incorporate deterministic and random track inputs the following changes in the above two equations are necessary.

$$\dot{\underline{x}} = \underline{F}\underline{x} + \underline{G}\underline{u} + \underline{G}\underline{w} \quad (3.7)$$

where $\underline{u}$ = deterministic track input

$\underline{w}$ = random track input due to the uncertainty in $\underline{u}$

$$\hat{\underline{x}} = (\underline{F}-\underline{K}\underline{H})\hat{\underline{x}} + \underline{K}\underline{z} + \underline{G}\underline{u} \quad (3.8)$$

Although $\underline{u}$ is a white noise input it is called deterministic because it was an input into the system model which generated the data (equation 3.7) and it was an input into the Kalman Filter of the maximum likelihood program (equation 3.8). $\underline{w}$ is called a random track input because it appears as an input into the wheelset system (equation 3.7) but it is not an input into the Kalman Filter (equation 3.8). The Kalman Filter in the maximum likelihood program has the $\underline{Q}$ matrix as its only source of information about $\underline{w}$.

The difference between test cases 8 and 9 is the size of the random track input as compared to the deterministic track input. Case 9 has a larger random track input than case 8, which signifies that case 9 has more uncertainty in the deterministic track input than case 8. As Table 3.3 shows, for case 8 the likelihood function had a maximum but for case 9 the likelihood function did not have a maximum. This would suggest that the maximum likelihood method needs a well-defined deterministic track input if it is to identify a peak in the likelihood function. However, maximum likelihood theory has no restrictions.

This suggests that the maximum likelihood processor may need more measurements--a longer data record--when the deterministic track input has a large amount of uncertainty. In reference (6) an increase in the number of observations is shown to cause a more pronounced maximum in the likelihood function. Although there are no quantitative results which prove more measurements will produce a peak in the likelihood function for the dynamically scaled wheelset case, this is an area in which further research should be done.

For all of the test cases in group 1, the Kalman Filter in the maximum likelihood program was given the following values for Q and R, as shown in Table 3.1 and Table 3.2.

$$Q = 1.0$$

$$R = \begin{bmatrix} .01 & 0 & 0 \\ 0 & .01 & 0 \\ 0 & 0 & .01 \end{bmatrix}$$

These five test cases were run for diagnostic purposes but have been included here because they exhibit some interesting trends.

Case 2 is called the perfect observation case in the literature. In reference (7) the likelihood function is shown to reduce to equation 3.9 for this case.

$$L = \sum_{i=1}^N T(t_i) Q^{-1}(t_i) \quad (3.9)$$

where
$$v(t_i) = z(t_i) - \phi z(t_{i-1}) \quad (3.10)$$

when Q is known. Essentially equation 3.9 says that the likelihood function simplifies to the observation term for the perfect measurement case. The maximum likelihood processor did not analyze this limiting case because equations 3.9 and 3.10 were not incorporated into the maximum likelihood

algorithm for test case number 2. The results of test case 2 do provide insight into the performance of maximum likelihood processor when it has incorrect or imprecise information about $\underline{w}$ or $\underline{v}$. Figures 3.3, 3.4 and 3.5 contain plots of the output of the maximum likelihood program for test case 2.

Test case 2 has a random track input only which makes it a limiting case for the effect of $\underline{u}$ versus $\underline{w}$ that was examined in cases 8 and 9. Test case 2 suggests that the more randomness there is in the track input, the more difficulty the maximum likelihood processor has in identifying maximum for the likelihood function. There are no theoretical limitations of this type for maximum likelihood. As in test cases 8 and 9 one practical aspect of implementing the maximum likelihood technique that should be investigated is data record length. Although several assumptions were made in the generation of the simulated data as discussed in section 1.4 it is believed that these assumptions will not seriously affect the performance of the maximum likelihood processor.

Case number 3 is a special case of maximum likelihood identification because there is no random track input. The likelihood function reduces to the observation term for this case also; however, the weighting matrix is different than in the no measurement noise case.

$$L = \sum_{i=1}^N \underline{v}^T(t_i) R^{-1} \underline{v}(t_i) \quad (3.11)(7)$$

where $\underline{v}(t_i) = \underline{z}(t_i) - H\hat{\underline{x}}(t_i) \quad (3.12)(7)$

when R is known. In this limiting case a Kalman Filter is used by the maximum likelihood processor as shown in equation 3.12, however the weighting matrix is not dependent on P but only on R . The observation term calculated by the maximum likelihood program for test case 3 was

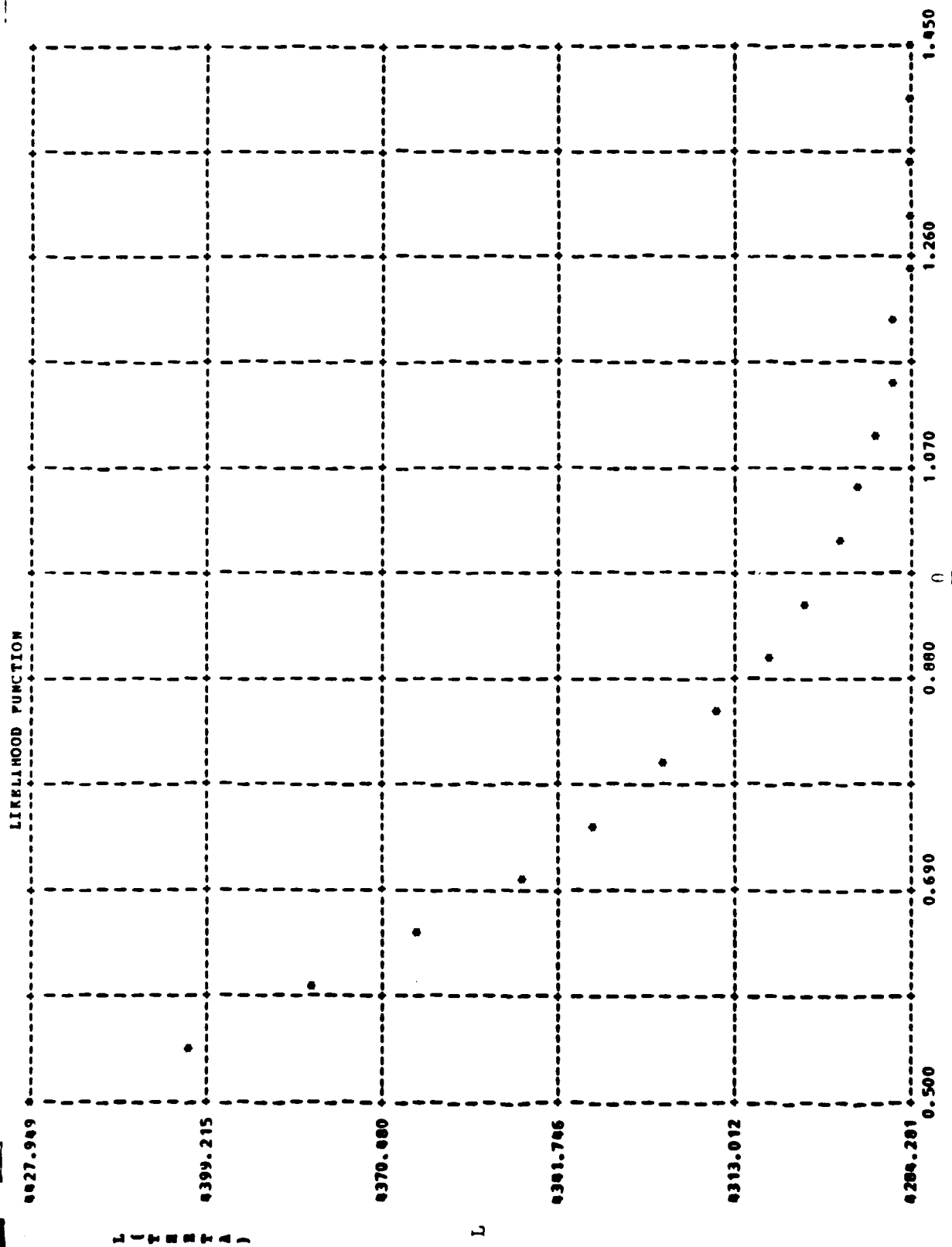


Figure 3.3 Plot of the likelihood function for test case 2.

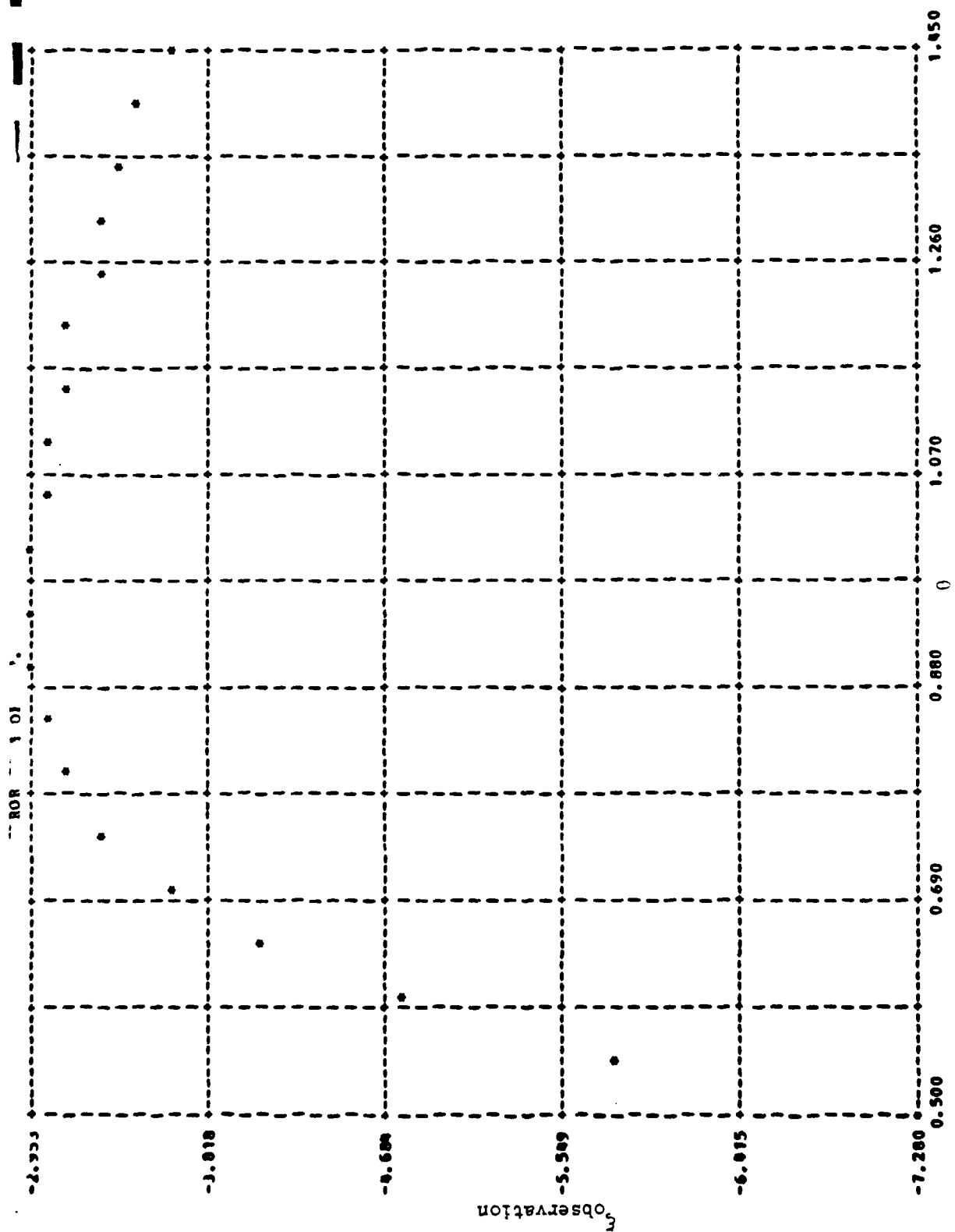


Figure 3.4 Plot of the observation term for test case 2.

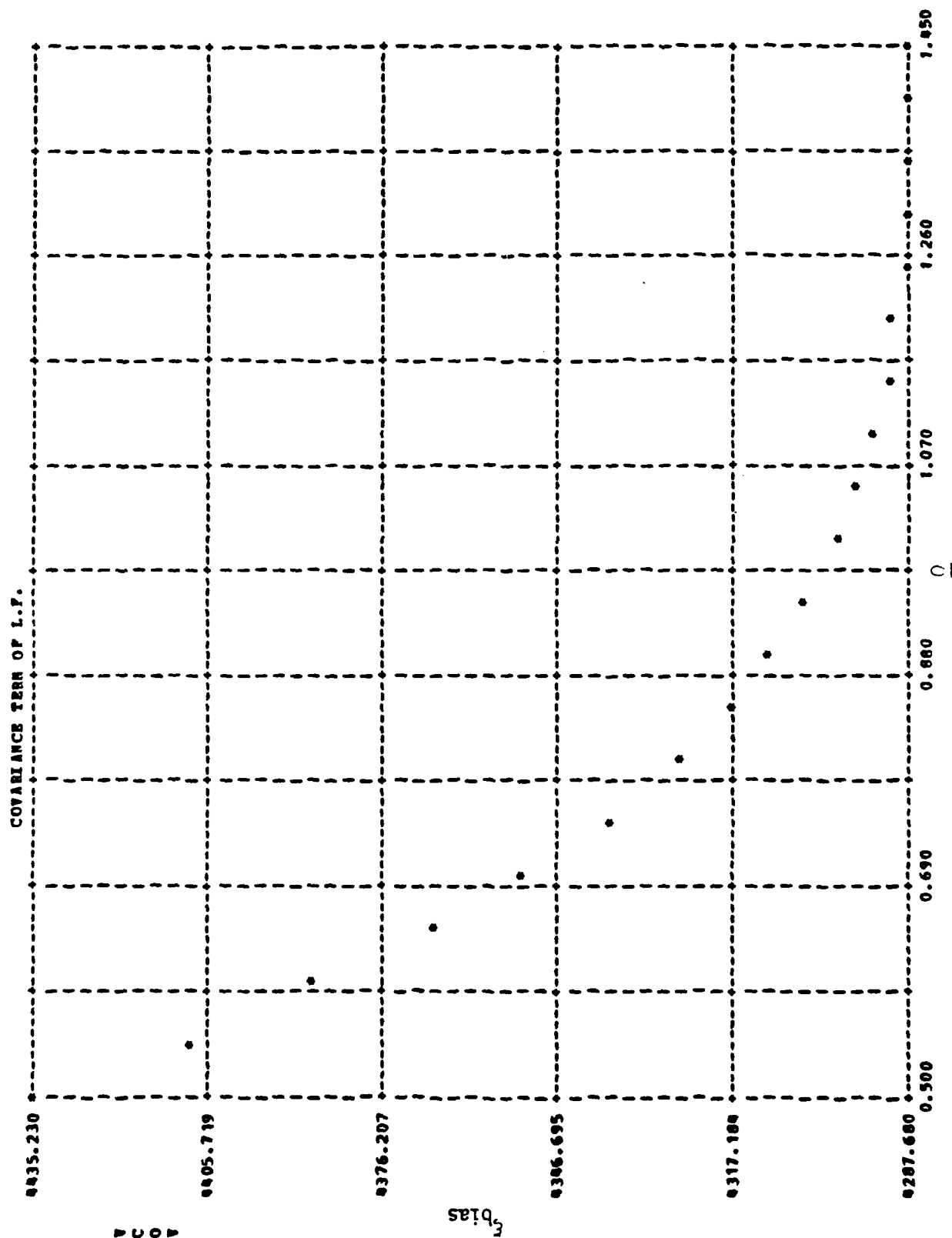


Figure 3.5 Plot of the bias term for test case 2.

$$\xi = -\frac{1}{2} \sum_{i=1}^N v^T(t_i) B^{-1} v(t_i) \quad (3.13)$$

where $B = HPH^T + R \quad (3.14)$

and $v(t_i) = z(t_i) - H\hat{x}(t_i) \quad (3.15)$

The state estimates as calculated by the maximum likelihood program are not as accurate as they could be because the Kalman Filter was given an incorrect value for Q (Table 3.2). Also the weighting matrix used in equation 3.13 is very different from the weighting matrix used in equation 3.11. Because the maximum likelihood algorithm for test case 3 did not incorporate equation 3.11 and because Q was incorrect, the limiting case of no process noise was not analyzed correctly.

Test case 1 is a combination of the two limiting cases presented above. The likelihood function reduces to:

$$L = \sum_{i=1}^N v^T(t_i) v(t_i) \quad (3.16)$$

where $v(t_i) = z(t_i) - \phi z(t_{i-1}) \quad (3.17)$

As equation 3.17 shows a Kalman Filter would not be used in this case because there is no random track input and no measurement noise. As for the other test cases in group 1 the values for Q and R used by the maximum likelihood program for test case 1 were:

$$Q = 1.0$$

$$R = \begin{bmatrix} .01 & 0 & 0 \\ 0 & .01 & 0 \\ 0 & 0 & .01 \end{bmatrix}$$

Because of this discrepancy, the results obtained from test case 1 are not necessarily those that would be obtained if the no random track input-no

measurement noise case were implemented correctly using equations 3.16 and 3.17.

Test case 4 and test case 5 are the only members of group 1 for which the maximum likelihood program was given correct values for Q and R . The results of test cases 4 and 5 given. As demonstrated in other test cases the maximum likelihood processor did not identify a peak in the likelihood function when the only input driving the wheelset system is a random track input. From the test cases performed as part of this research program no conclusive explanations can be given for the above problem. As suggested before, a longer data record in these cases may be necessary however there is no direct evidence to support this conclusion.

Test cases 6 and 7 were run to see if the ratio of Q to R had any affect on the performance of the maximum likelihood processor. For test case 6 the ratio of Q to R is the same as that for test cases 4 and 5. According to maximum likelihood if the values of Q and R are changed but their ratio remains constant then the observation term will remain the same but the bias term will change. Graphs of the observation term for test case 4 and test case 6 are presented in Figure 3.6 and 3.7 respectively. Comparing these two plots shows that the observation term did not change between these two cases. Comparing Figures 3.8 and 3.9 it can be seen that the bias term did change its range of values, but not its general shape. The likelihood functions for case 4 and case 6 are given in Figures 3.10 and 3.11. The likelihood function has the same general shape for each case, however, the location of the local maximum does change. This agreement between test cases 4 and 6 further supports the validity of the maximum likelihood algorithm given in Chapter 1, and the computer program which implements it.

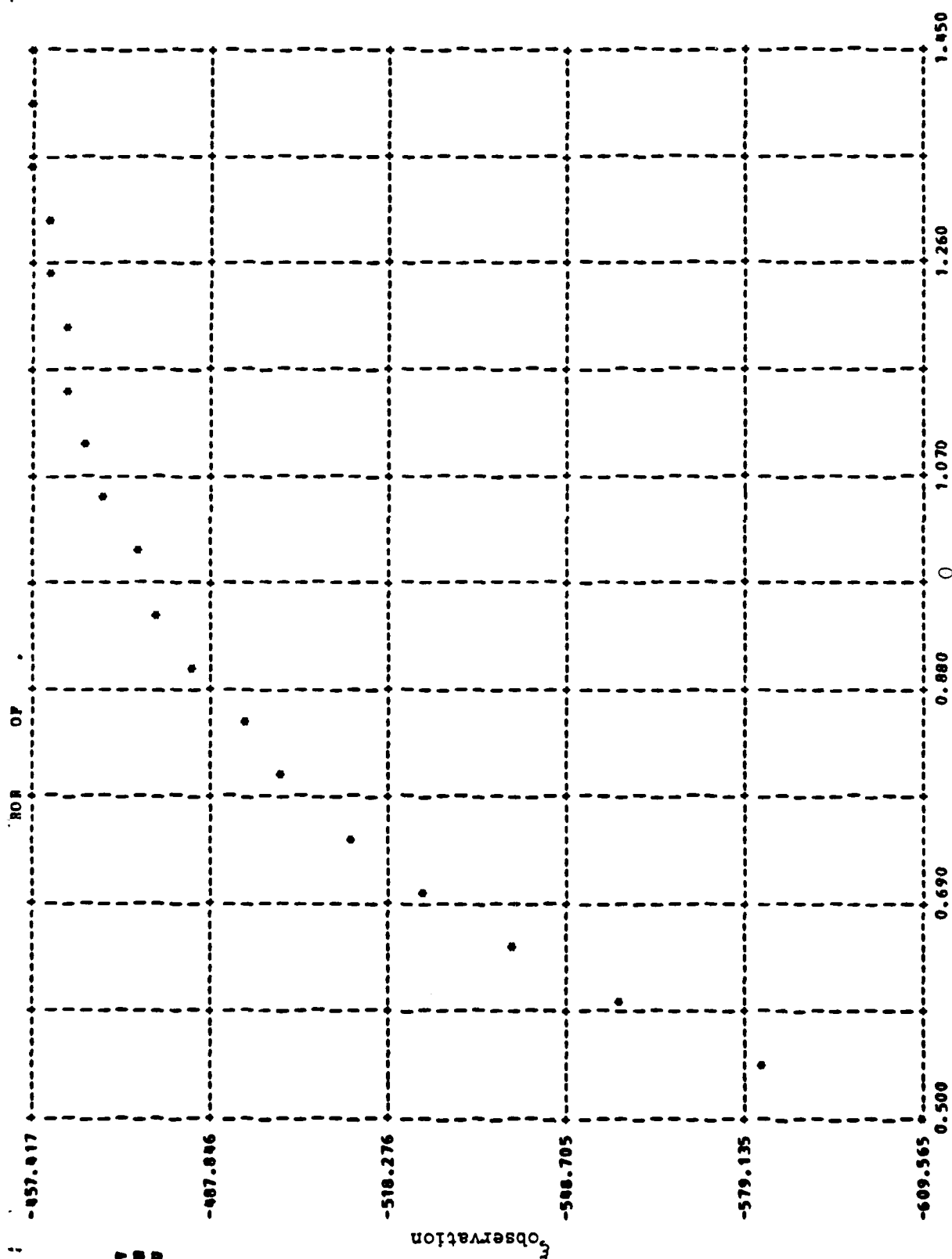


Figure 3.6 Plot of the observation term for test case 1.

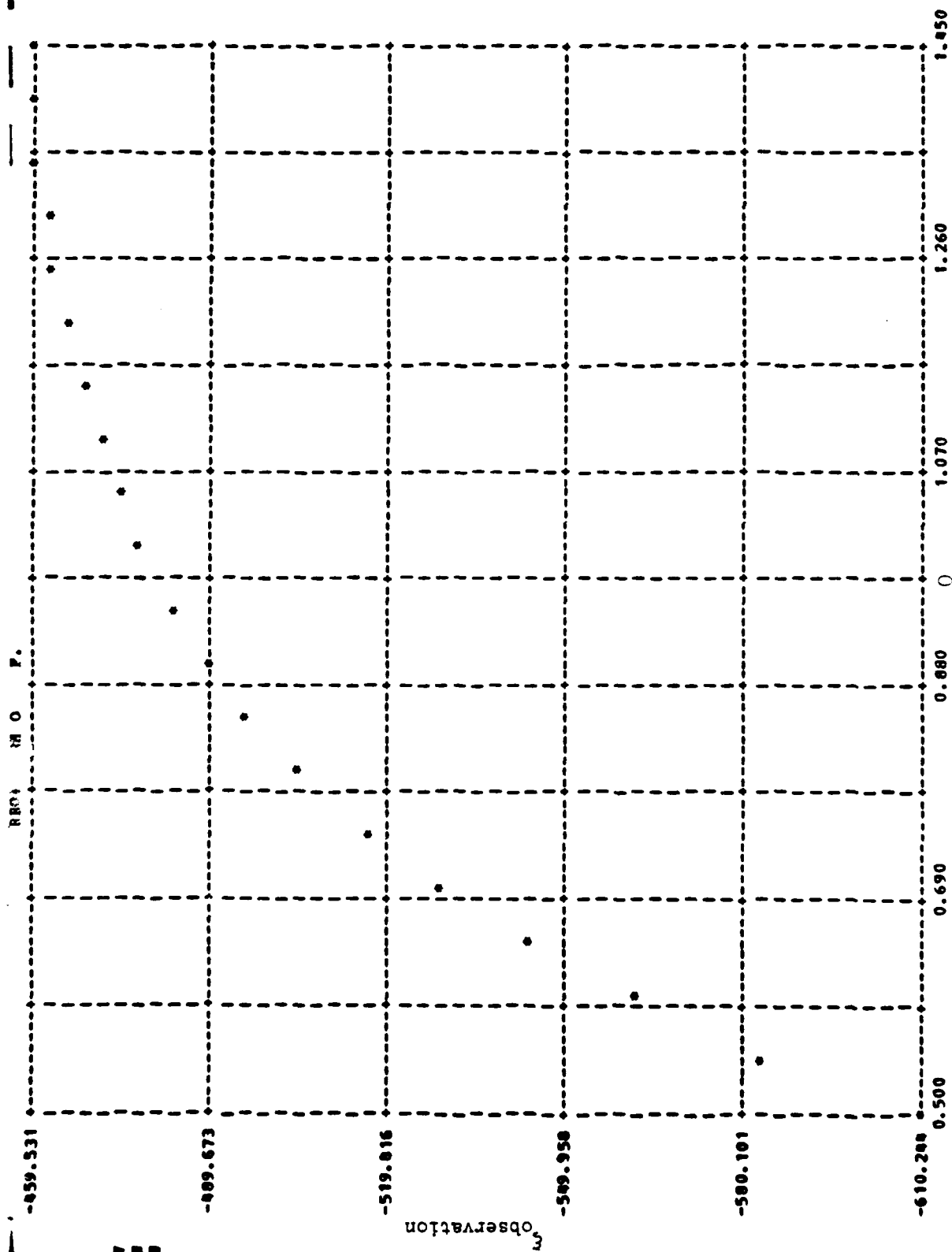


Figure 3.7 Plot of the observation term for test case 6.

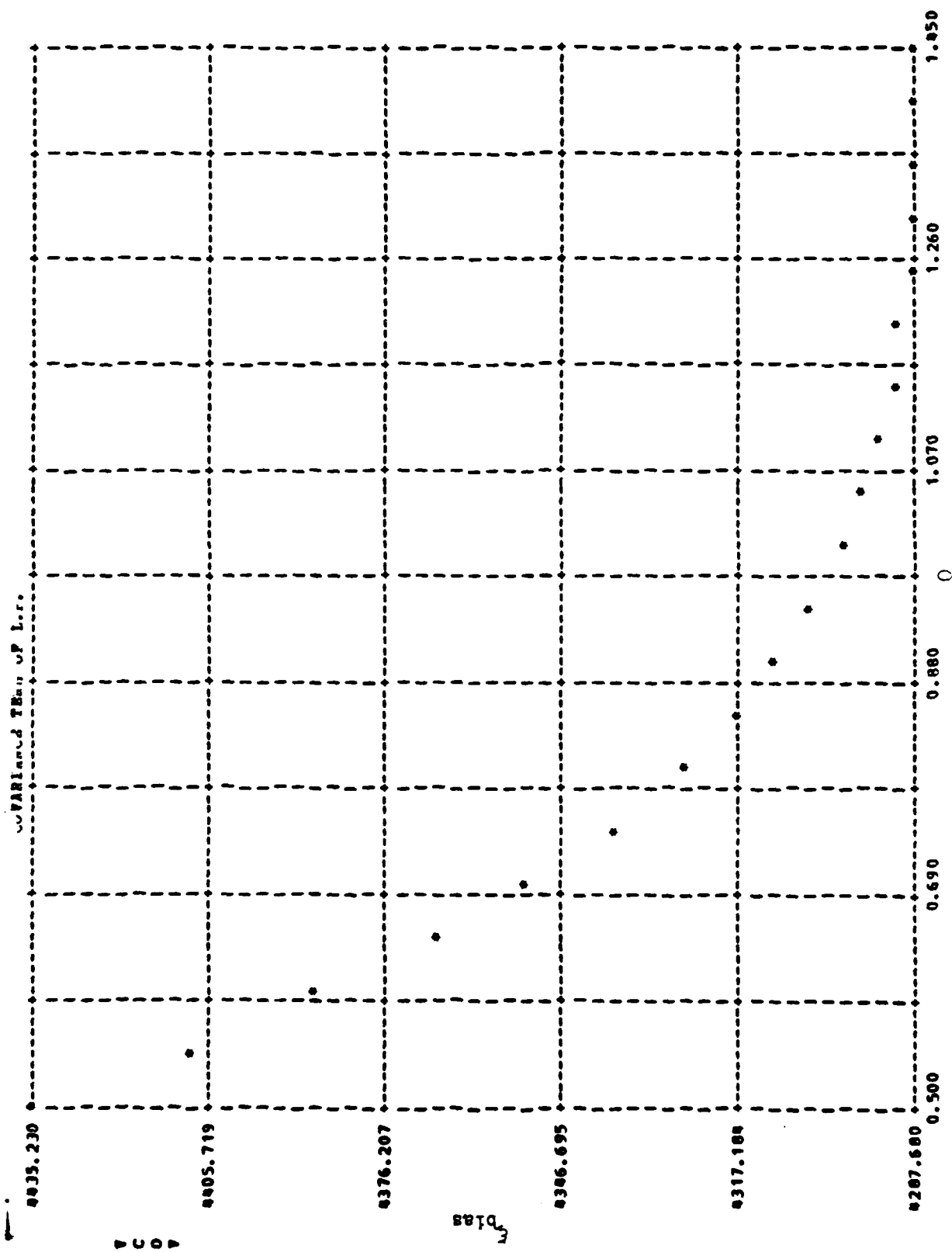


Figure 3.8 Plot of the bias term for test case 4.

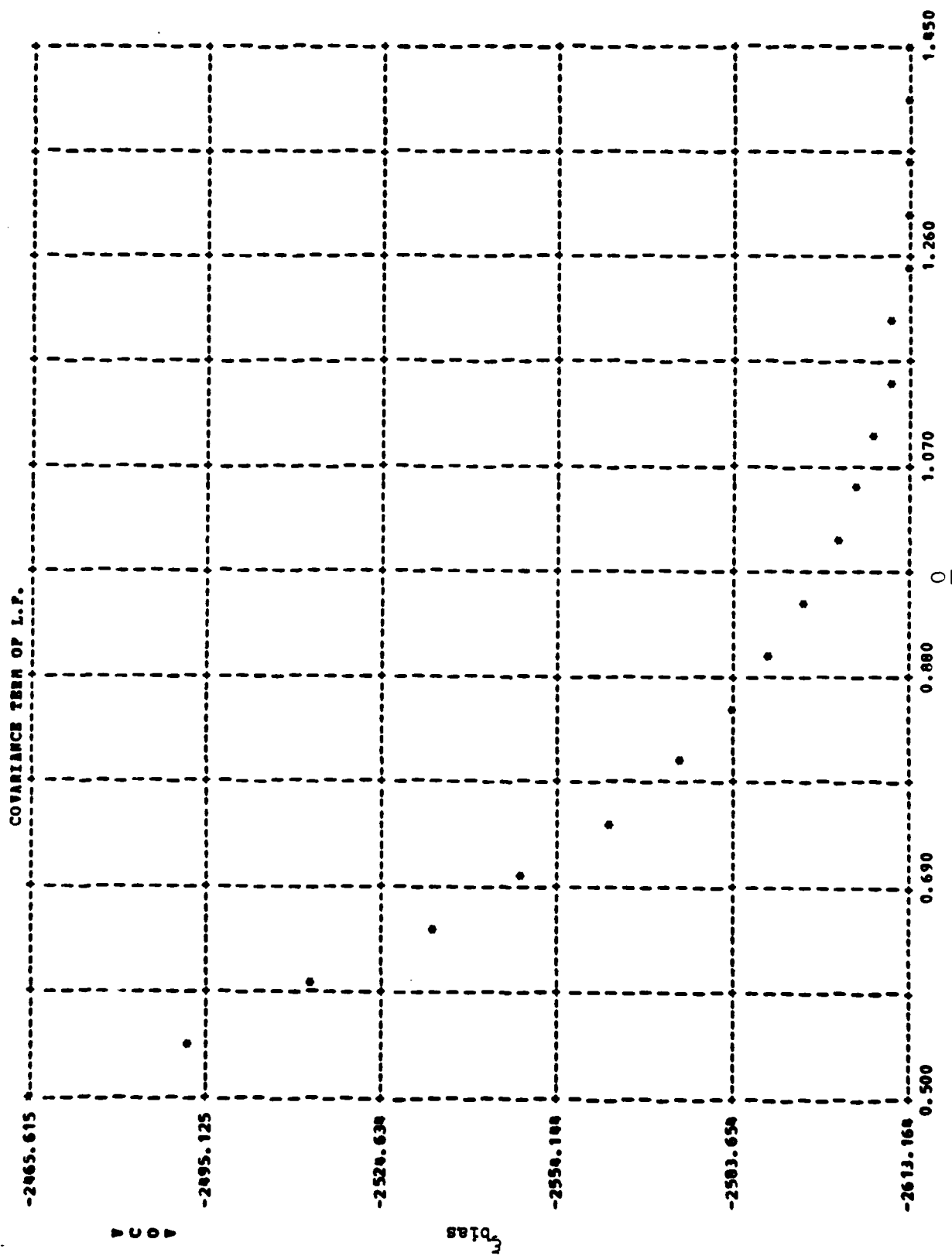


Figure 3.9 Plot of the bias term for test case 6.

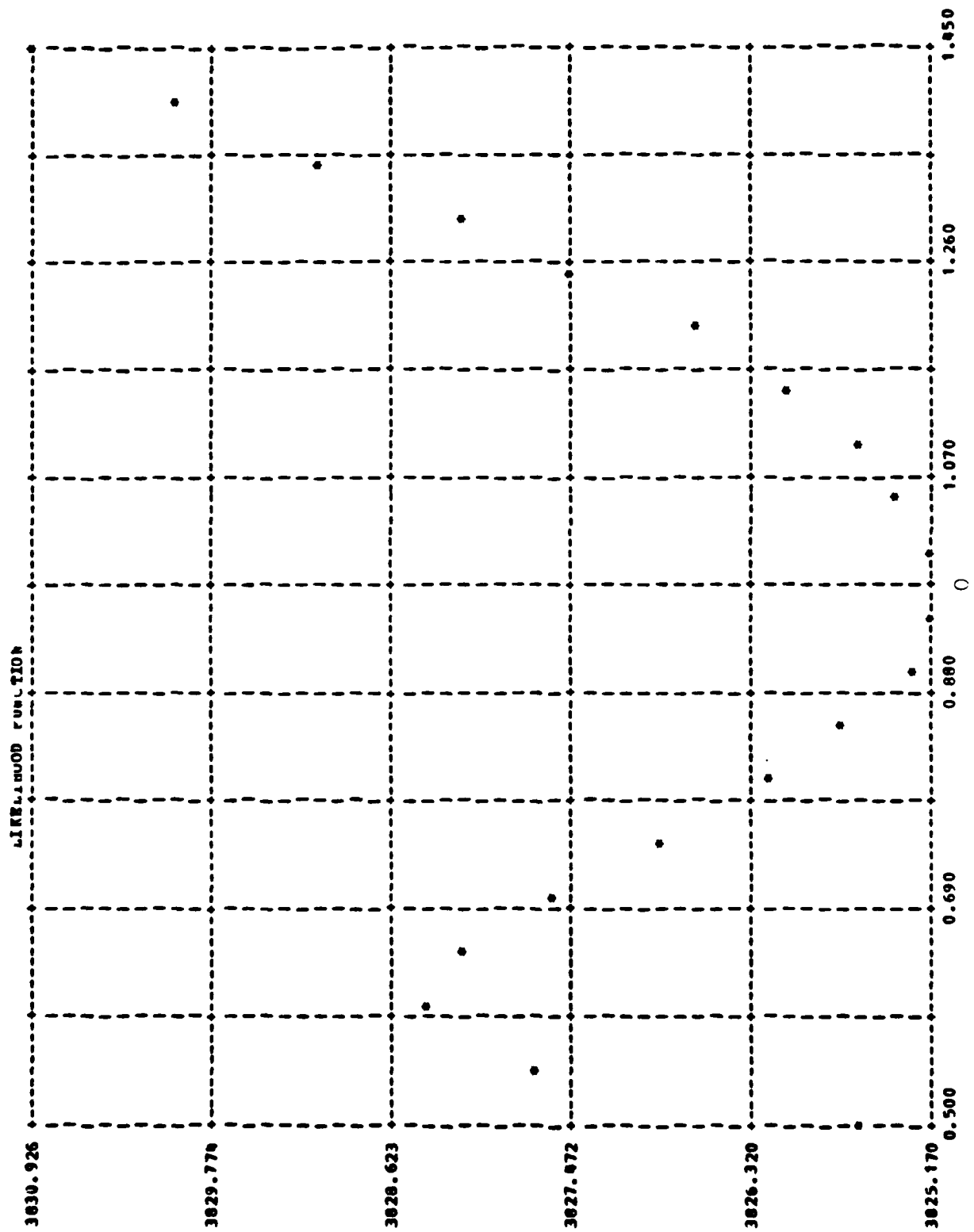


Figure 3.10 Plot of the likelihood function for test case 4.

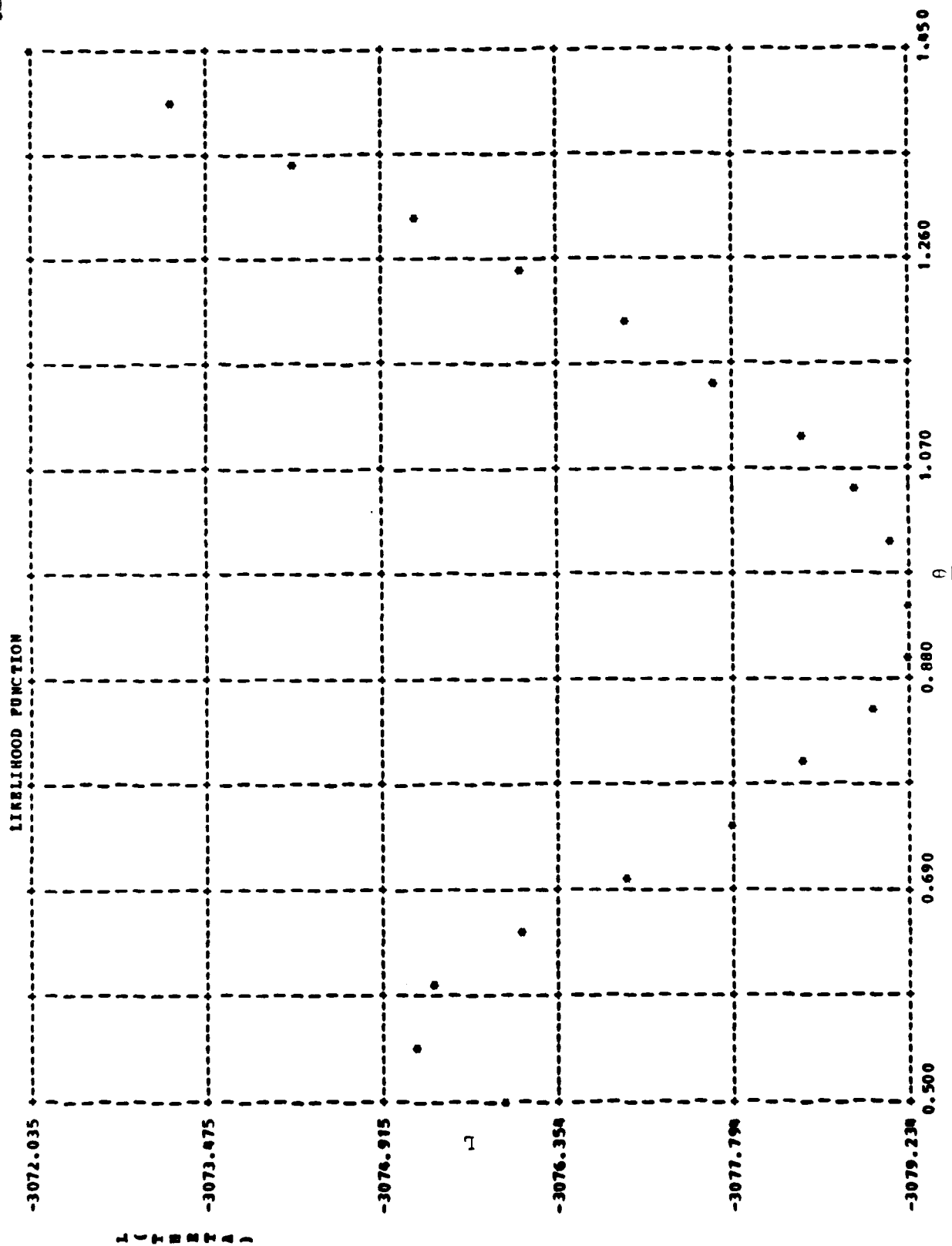


Figure 3.11 Plot of the likelihood function for test case 6.

Test case 6 has as its only input a random input disturbance and in accordance with the results of other similar test cases, its likelihood function does not have a maximum. Test case 7 also has a random disturbance as the only input to the wheelset model; however, the likelihood function in the case 7 results does have a maximum at $\theta = .65$. A plot of this likelihood function is given in Figure 3.12. The results of test case 7 indicate that the test data record was long enough for the maximum likelihood processor to identify a peak in the likelihood function, although the peak occurred at $\theta = .65$ instead of $\theta = 1.00$. There is no indication that a longer record will shift the peak from $\theta = .65$ to $\theta = 1.00$, although this is an area which needs further research. The fact that the measurement noise in case 7 is much smaller than in case 6 may be the reason the maximum likelihood processor was able to identify a peak in the likelihood function.

There is another possible explanation for the problems the likelihood function had in identifying a peak in the likelihood function. As discussed in section 1.4 a $\Delta t = .005$ sec was used to generate the test data. This time step resulted in 6 data points per cycle of the highest frequency response of the wheelset. The maximum likelihood processor may need more information about this particular mode of response of the wheelset. This could be accomplished by using a smaller Δt . However, because the present method used to generate the simulated data is restricted to 1000 data points per state variable, a smaller Δt would produce data with fewer cycles of the low frequency response of the wheelset. In order to investigate the effects of a smaller Δt a new method for generating the simulated data will have to be developed.

Since all of the simulated wheelset data was generated with the same Δt , and for several cases a peak in the likelihood function was identified there

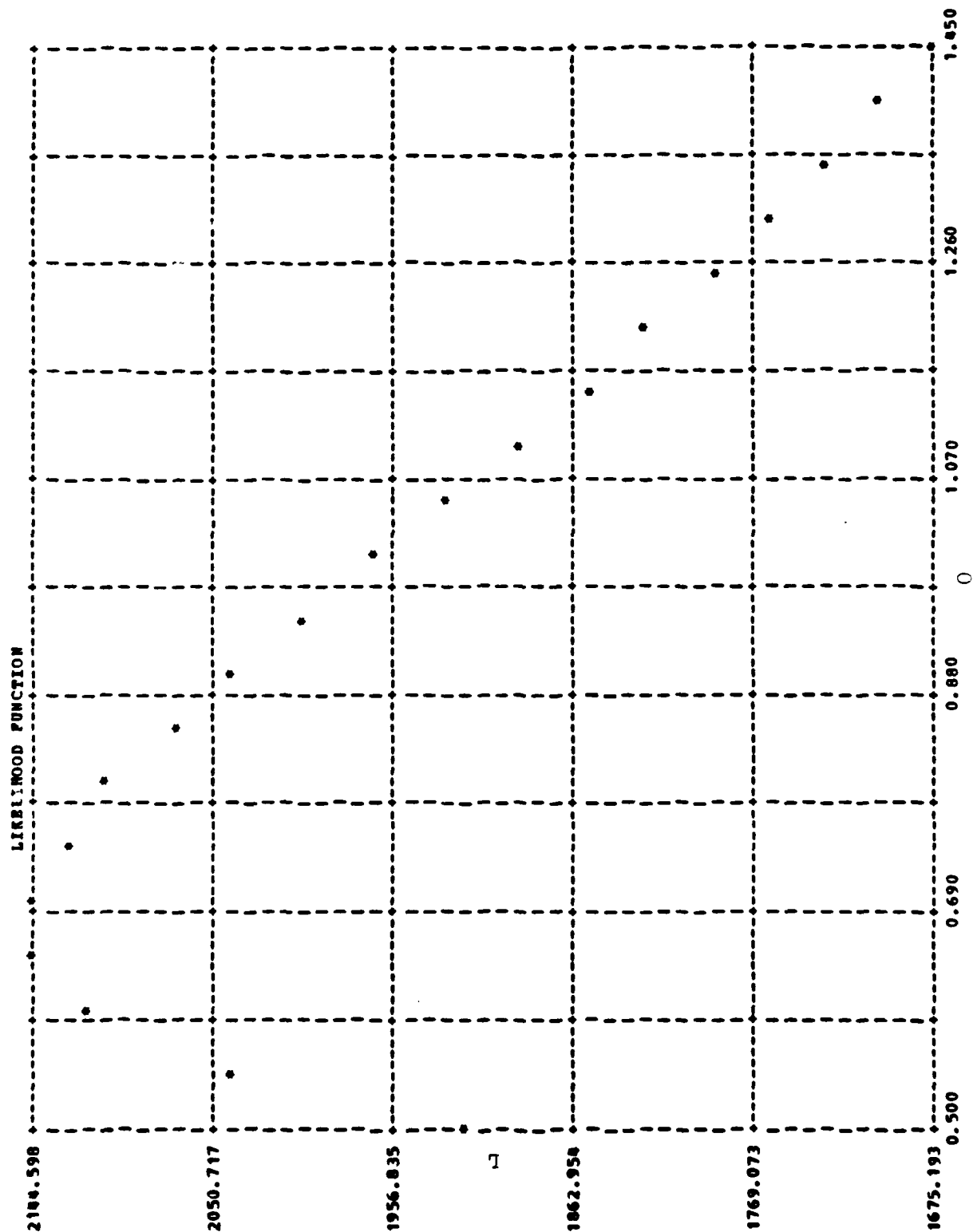


Figure 3.12 Plot of the likelihood function for test case 7.

does not seem to be a need for a smaller Δt . However, in those cases where the simulated data was generated using a random track input, the maximum likelihood processor was given minimal information about the input into the wheelset model that generated the data--only the covariance matrix for the random track input. To make up for this limited information about the input the maximum likelihood processor may need better information in other areas. This "better" information includes more data points per cycle of the high frequency response mode of the wheelset--obtained by a smaller Δt --and more observations of the low frequency response of the wheelset--obtained with a larger data record.

3.3.2 Gradient and Fisher Information Matrix

The task of evaluating the performance of the maximum likelihood processor in calculating $\frac{\partial L}{\partial \theta}$ and $\frac{\partial^2 L}{\partial \theta^2}$ is considerably more difficult than evaluating the calculation of L itself. There are two methods that can be used to check the gradient and second partial. The first method involves curve fitting likelihood function and gradient data points with n^{th} order polynomials. The derivatives can then be calculated analytically using the polynomial equation. The problem experienced with this method is that the data points do not represent a smooth function and inflection points which do not exist in the data are created. These inflection points cause large errors in the calculation of the derivative. The second method, and the one used to analyze the test data results, is explained in Figure 3.13. From Figure 3.13

$$m_{32} = \frac{y_3 - y_2}{x_3 - x_2} \quad (3.18)$$

$$m_{43} = \frac{y_4 - y_3}{x_4 - x_3} \quad (3.19)$$

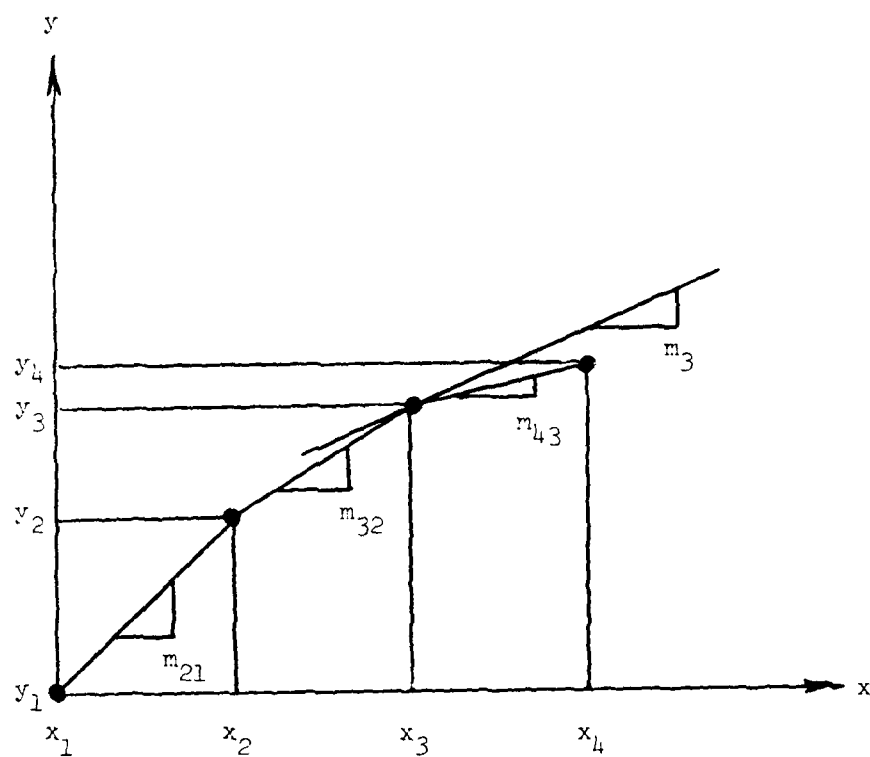


Figure 3.13 Average slope method for checking the gradient and second partial of the likelihood function.

$$m_3 = \frac{m_{43} + m_{32}}{2} \quad (3.20)$$

$$m_1 = m_{21} \quad (3.21)$$

$$m_4 = m_{43} \quad (3.22)$$

Equations 3.18 through 3.22 were used to calculate the derivatives of the likelihood function and the gradient. The slope at the first data point was set equal to the slope between the first two data points. The slope at the last data point was set equal to the slope between the last two data points. This method of averaging slopes works very well when the data points form a smooth function. In many cases the likelihood function data points and the gradient data points do not form smooth functions and the derivatives calculated using this second method can have large errors. Since the only objective was to check the reasonableness of the gradient and second partial data, the second method of averaging slopes was used. In general the difference between the gradient and the derivative of L as determined by the average slope method is on the order of 5%. The difference between the second partial and the slope of the gradient as determined by the averaging method was considerably larger--on the order of 35%. This difference for the second partial was considerably larger for test cases 4 and 6. The reason the difference between the second partial and the slope of the gradient may be so large for test cases 4 and 6 is that determining derivatives by averaging slopes produces large errors for data points which do not form smooth functions. The data points for the test cases 4 and 6 are not as smooth as in other cases. See Figures 1 and 2. The large discrepancy is most probably due to errors in the calculation of the Information Matrix rather than in the calculation of the gradient and the second partial was

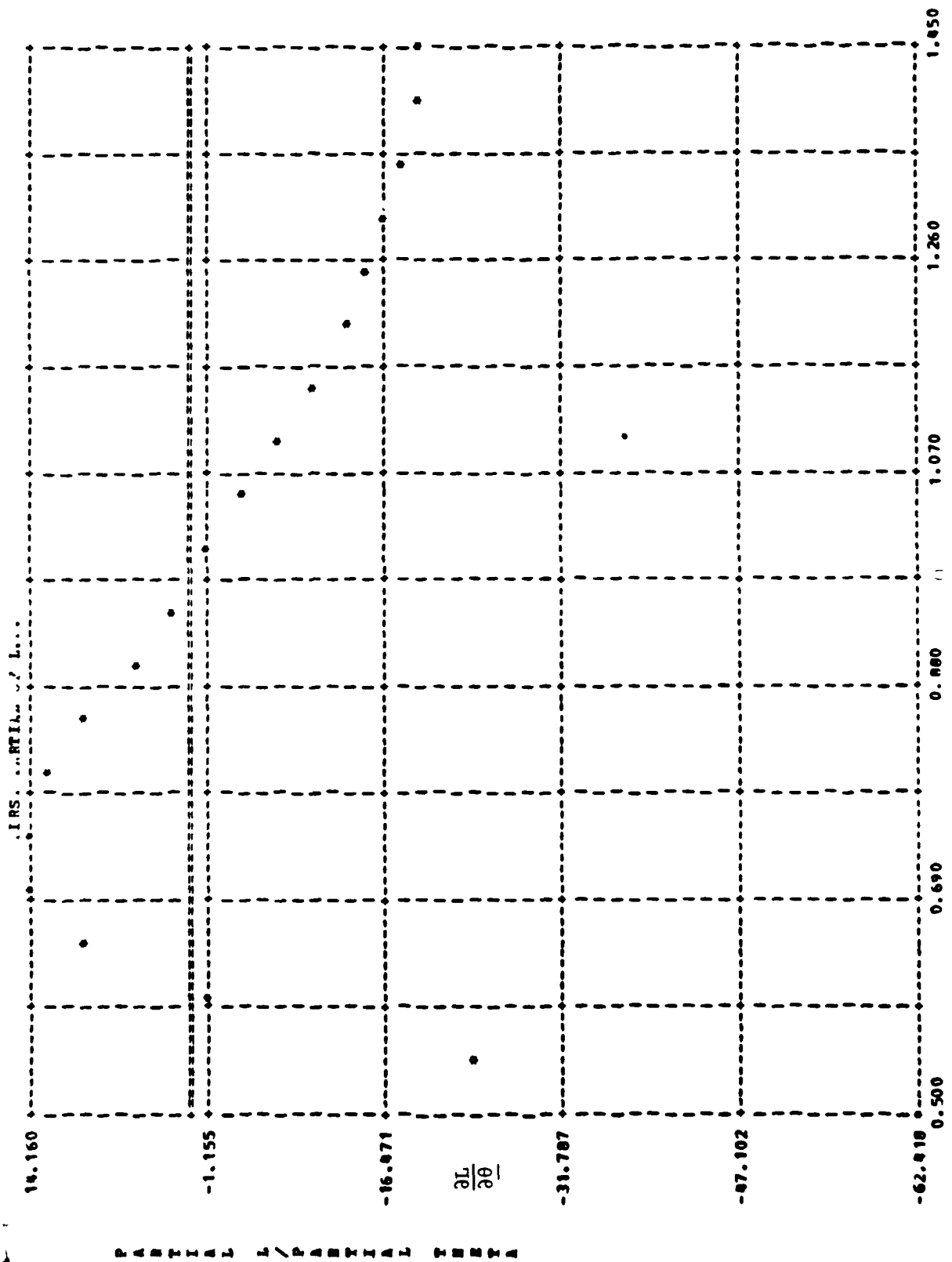


Figure 3.14 Plot of the gradient for the log-likelihood

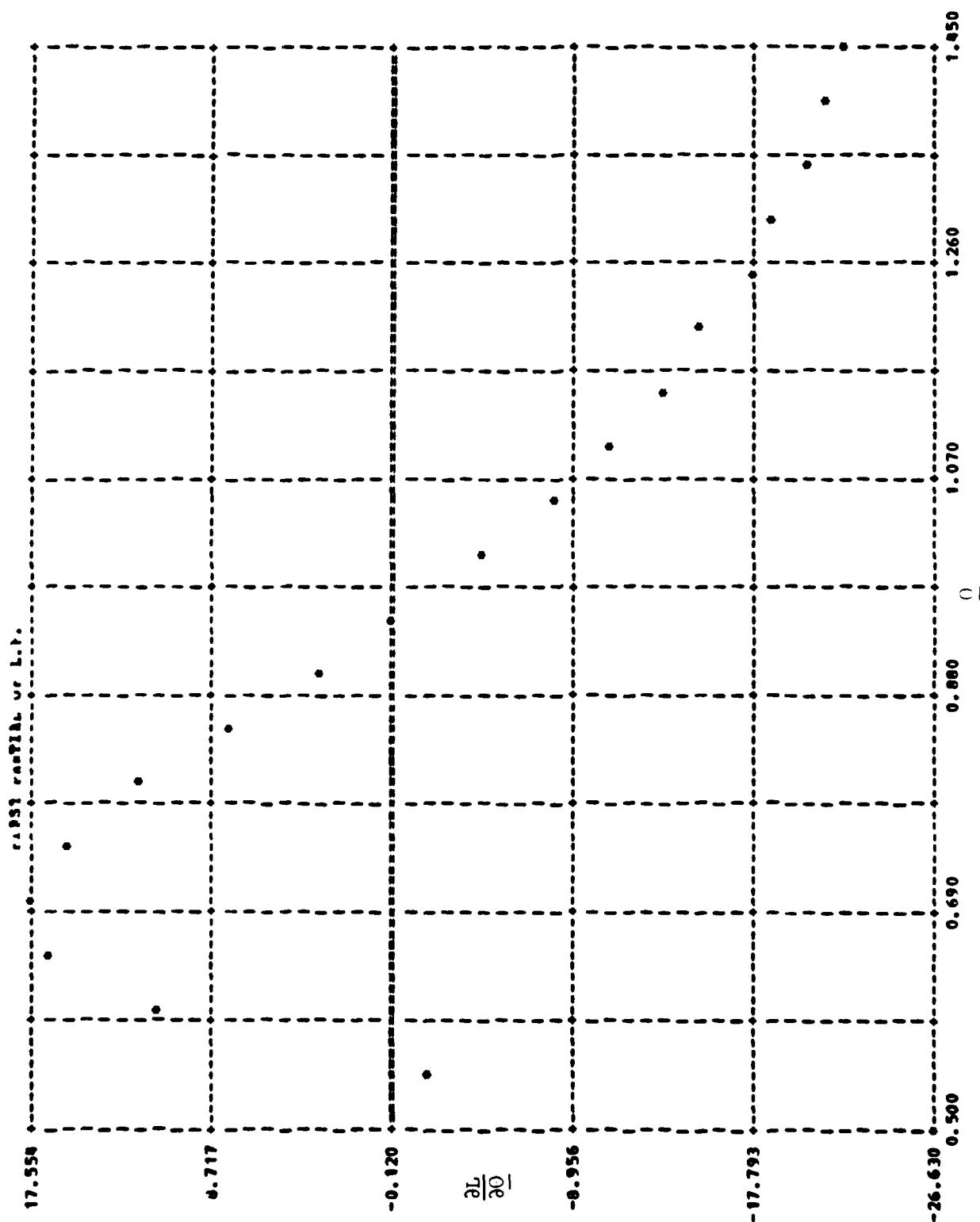


Figure 3.15 Plot of the gradient for test case 6.

not carried out for test cases 8 and 9 because based on the results for test cases 1 through 7 it was concluded that the quasilinearization section of the maximum likelihood algorithm was implemented correctly in the maximum likelihood computer program.

3.4 Application of Results to Actual Wheelset Data

The test data cases were analyzed by the maximum likelihood program for the following reasons:

- (1) to ensure that the computer program was working properly.
- (2) to identify any potential problems that may arise when the actual wheelset data is analyzed.
- (3) to make recommendations for further research, based on the conclusions reached in (2) above.

The first objective was attained as documented in the previous sections. The purpose of this section is to discuss items (2) and (3).

In every case except one where the only input into the wheelset model was the random track input the likelihood function did not have a maximum. Several times in the previous section the point has been stressed that the likelihood function failed to have a maximum because the data record needed to be longer. The test cases run as part of this research program do not form a large enough experimental base to state the above point with any certainty. It is suggested that further research in this area be directed to find out how the length of the data record affects the performance of the maximum likelihood processor. As mentioned in Chapter 1, the present method of generating simulated data is limited to 1000 observations because of limitations in computer storage, therefore another method of generating simulated data will

have to be designed.

Another area of research which should be examined is the effect of a smaller Δt in generating the test data. A smaller Δt would provide more information about the high frequency response mode of the wheelset. In order not to lose information about the low frequency mode, data records longer than 1000 points per state variable will have to be used. As pointed out above this requires a new method of generating simulated data.

Appendix H contains plots of the likelihood function, observation time and bias term for test cases 1 through 9.

CONCLUSION

As stated in the Introduction, the objective of this research program was the development and implementation of a maximum likelihood parameter identification algorithm applicable to a dynamically scaled wheelset model. The following is a summary of the conclusions that can be drawn from the results of this research program.

- 1) The maximum likelihood parameter identification equations can be tailored to the problem of identifying creep coefficients for the dynamically scaled wheelset model.
- 2) The reasonable results obtained for many of the test cases proves that the implementation of the maximum likelihood algorithm in Fortran IV was performed correctly.
- 3) More research involving simulated wheelset data is necessary before actual wheelset data can be processed. It is recommended that the following areas be investigated.
 - a) the effect of data record length on the performance of the maximum likelihood processor
 - b) advantages in representing the track input as a deterministic input with uncertainty as compared to representing the track input as a random track input
 - c) the effect of a smaller Δt on the performance of the maximum likelihood processor.

The primary motivation for the above three recommendations is that it seems reasonable to assume the maximum likelihood processor works better

when it has more information about the system and about the inputs into the system. If the track input is a random track input then the covariance matrix is the only information the maximum likelihood processor receives about this random track input. In order to make up for this limited information about the input into the wheelset system, the maximum likelihood processor should receive better information in other areas. A smaller Δt and a longer data record provide more information about the wheelset high frequency response mode and low frequency model respectively. This increase in the quantity of information could be considered better from the maximum likelihood processor point of view.

LIST OF REFERENCES

- (1) Bendat, Julius S., and Piersol, Allan G., Random Data: Analysis and Measurement Procedures, John Wiley and Sons, Inc., 1971.
- (2) Brogan, William L., Modern Control Theory, Quantum Publishers, Inc., New York, 1974.
- (3) Gelb, Arthur, ed., Applied Optimal Estimation, M.I.T. Press, Cambridge, Mass., 1974.
- (4) Kalker, J.J., "On the Rolling Contact of Two Elastic Bodies in the Presence of Dry Friction." Doctoral Dissertation, 1967, Technische Hogeschool, Delft, The Netherlands.
- (5) Kwakernaak, Huibert, and Sivan, Raphael, Linear Optimal Control Systems, John Wiley and Sons, Inc., New York, 1972.
- (6) Sage, Andrew P., and Melsa, James L., System Identification, Academic Press, New York, 1971.
- (7) Schweppe, Fred C., Uncertain Dynamic Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
- (8) Stepner, David E., and Mehra, Raman K., "Maximum Likelihood Identification and Optimal Input Design for Identifying Aircraft Stability and Control Derivatives," Systems Control, Inc., Palo Alto, California, NASA CR-2800, March 1973.
- (9) Sweet, Larry M., and Sivak, Joseph A., "Measurements of Nonlinear Wheelset Forces in Flange Contact Using Dynamically Scaled Models," Princeton University, Department of Mechanical and Aerospace Engineering, Report No. 1406, September 1978.
- (10) Sweet, Larry M., and Sivak, Joseph A., "Analysis of Nonlinear Wheelset Forces in Flange Contact," Princeton University, Department of Mechanical and Aerospace Engineering, Report No. 1407, September 1978.
- (11) Weinstock, H., "Analysis of Rail Vehicle Dynamics in Support of Development of the Wheel Rail Dynamics Research Facility." DOT Report DOT-MA-06-0025-73, June 1973.

Appendix A

DERIVATION OF $\frac{\partial \hat{x}(t)}{\partial \theta}$

$$\dot{\hat{x}}(t) = F(t)\hat{x}(t) + L(t)u(t) + K(t)[z(t) - H(t)\hat{x}(t)] \quad (A.1)$$

$$\begin{aligned} \frac{\partial \hat{x}(t)}{\partial \theta} = & G(t) \frac{\partial \hat{x}(t)}{\partial \theta} + \frac{\partial F(t)}{\partial \theta} \hat{x}(t) + L(t) \frac{\partial u(t)}{\partial \theta} + \frac{\partial L(t)}{\partial \theta} u(t) \\ & + K(t) \frac{\partial z(t)}{\partial \theta} + \frac{\partial K(t)}{\partial \theta} z(t) - K(t) \left[H(t) \frac{\partial \hat{x}(t)}{\partial \theta} + \frac{\partial H(t)}{\partial \theta} \hat{x}(t) \right] \\ & - \frac{\partial K(t)}{\partial \theta} H(t) \hat{x}(t) \end{aligned} \quad (A.2)$$

Since the input disturbance is deterministic $\frac{\partial u(t)}{\partial \theta} = 0$. The measurements z are independent of θ therefore $\frac{\partial z}{\partial \theta} = 0$. Therefore equation A-3 reduces to:

$$\begin{aligned} \frac{\partial \hat{x}(t)}{\partial \theta} = & F(t) \frac{\partial \hat{x}(t)}{\partial \theta} + \frac{\partial F(t)}{\partial \theta} \hat{x}(t) + \frac{\partial L(t)}{\partial \theta} u(t) + \frac{\partial K(t)}{\partial \theta} z(t) \\ & - K(t)H(t) \frac{\partial \hat{x}(t)}{\partial \theta} - K(t) \frac{\partial H(t)}{\partial \theta} \hat{x}(t) - \frac{\partial K(t)}{\partial \theta} H(t) \hat{x}(t) \end{aligned} \quad (A.3)$$

Appendix B
SOLUTION TO $\frac{\partial P(t_i)}{\partial \theta}$

The matrix Riccati equation is

$$\dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - P(t)H^T(t)R^{-1}(t)H(t)P(t) \quad (B.1)$$

Taking the partial derivative of both sides with respect to θ :

$$\begin{aligned} \frac{\partial P(t)}{\partial \theta} &= F(t) \frac{\partial P(t)}{\partial \theta} + \frac{\partial F(t)}{\partial \theta} P \\ &+ P(t) \frac{\partial (F^T(t))}{\partial \theta} + \frac{\partial P(t)}{\partial \theta} F^T(t) \\ &+ G(t)Q(t) \frac{\partial (G^T(t))}{\partial \theta} + G(t) \frac{\partial Q(t)}{\partial \theta} G^T(t) + \frac{\partial G(t)}{\partial \theta} Q(t)G^T(t) \\ &- P(t)H^T(t)R^{-1}(t)H(t) \frac{\partial P(t)}{\partial \theta} - P(t)H^T(t)R^{-1}(t) \frac{\partial H(t)}{\partial \theta} P(t) \\ &- P(t)H^T(t) \frac{\partial (R^{-1}(t))}{\partial \theta} H(t)P(t) - P(t) \frac{\partial (H^T(t))}{\partial \theta} R^{-1}(t)H(t)P(t) \\ &- \frac{\partial P(t)}{\partial \theta} H^T(t)R^{-1}(t)H(t)P(t) \end{aligned} \quad (B.2)$$

Post-multiply both sides of (B.2) by $\underline{y}$. The time notation will be dropped in further equations however all matrices are still time dependent.

$$\begin{aligned} \frac{\partial \dot{P}}{\partial \theta} \underline{y} &= F \frac{\partial P}{\partial \theta} \underline{y} + \frac{\partial F}{\partial \theta} P \underline{y} + P \frac{\partial F^T}{\partial \theta} \underline{y} + \frac{\partial P}{\partial \theta} F^T \underline{y} + GQ \frac{\partial G^T}{\partial \theta} \underline{y} + G \frac{\partial Q}{\partial \theta} G^T \underline{y} \\ &+ \frac{\partial G}{\partial \theta} QG^T \underline{y} - PH^T R^{-1} H \frac{\partial P}{\partial \theta} \underline{y} - PH^T R^{-1} \frac{\partial H}{\partial \theta} P \underline{y} - PH^T \frac{\partial R^{-1}}{\partial \theta} H P \underline{y} \\ &- P \frac{\partial H^T}{\partial \theta} R^{-1} H P \underline{y} - \frac{\partial P}{\partial \theta} H^T R^{-1} H P \underline{y} \end{aligned} \quad (B.3)$$

Apply the transformations:

$$\underline{y} = \frac{\partial P}{\partial \theta} \underline{y} \quad (B.4)(3)$$

$$\dot{\underline{Y}} = \frac{\partial P}{\partial \theta} \dot{\underline{Y}} + \left(\frac{\partial P}{\partial \theta} \right) \underline{Y} \quad (\text{B.5})$$

By reviewing the order of differentiation for the term $\frac{\partial P}{\partial \theta} \underline{Y}$ in equation (B.3) the equation (B.5) can be substituted in equation (B.3).

$$\begin{aligned} \dot{\underline{Y}} - \frac{\partial P}{\partial \theta} \dot{\underline{Y}} &= F\underline{Y} + \frac{\partial F}{\partial \theta} P\underline{Y} \\ &+ P \frac{\partial F^T}{\partial \theta} \underline{Y} + \frac{\partial P}{\partial \theta} F^T \underline{Y} + GQ \frac{\partial G^T}{\partial \theta} \underline{Y} + G \frac{\partial Q}{\partial \theta} G^T \underline{Y} + \frac{\partial G}{\partial \theta} QG^T \underline{Y} - PH^T R^{-1} H\underline{Y} \\ &- PH^T R^{-1} \frac{\partial H}{\partial \theta} P\underline{Y} - PH^T \frac{\partial R^{-1}}{\partial \theta} HP\underline{Y} - P \frac{\partial H^T}{\partial \theta} R^{-1} HP\underline{Y} - \frac{\partial P}{\partial \theta} H^T R^{-1} HP\underline{Y} \end{aligned} \quad (\text{B.6})$$

Apply the transformation

$$\dot{\underline{Y}} = -F^T \underline{Y} + H^T R^{-1} HP\underline{Y} \quad (\text{B.7})(\underline{3})$$

$$\begin{aligned} \dot{\underline{Y}} &= F\underline{Y} + \frac{\partial F}{\partial \theta} P\underline{Y} + P \frac{\partial F^T}{\partial \theta} \underline{Y} + GQ \frac{\partial G^T}{\partial \theta} \underline{Y} + G \frac{\partial Q}{\partial \theta} G^T \underline{Y} + \frac{\partial G}{\partial \theta} QG^T \underline{Y} \\ &- PH^T R^{-1} H\underline{Y} - PH^T R^{-1} \frac{\partial H}{\partial \theta} P\underline{Y} - PH^T \frac{\partial R^{-1}}{\partial \theta} HP\underline{Y} - P \frac{\partial H^T}{\partial \theta} R^{-1} HP\underline{Y} \end{aligned} \quad (\text{B.8})$$

Equation (B.7) and (B.8) are a linear system and are written in matrix form in

B.9.

$$\begin{bmatrix} \dot{\underline{Y}} \\ \dot{\underline{Y}} \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix} \begin{bmatrix} \underline{Y} \\ \underline{Y} \end{bmatrix} \quad (\text{B.9})$$

where

$$\omega_{11} = -F^T + H^T R^{-1} HP \quad (\text{B.10})$$

$$\omega_{12} = 0 \quad (\text{B.11})$$

$$\begin{aligned} \omega_{21} &= \frac{\partial F}{\partial \theta} P + P \frac{\partial F^T}{\partial \theta} + GQ \frac{\partial G^T}{\partial \theta} + G \frac{\partial Q}{\partial \theta} G^T + \frac{\partial G}{\partial \theta} QG^T \\ &- PH^T R^{-1} \frac{\partial H}{\partial \theta} P - PH^T \frac{\partial R^{-1}}{\partial \theta} HP - P \frac{\partial H^T}{\partial \theta} R^{-1} HP \end{aligned} \quad (\text{B.12})$$

$$\omega_{22} = F - PH^T R^{-1} H \quad (\text{B.13})$$

Equation (B.9) can be solved using a state transition matrix approach.

$$\begin{bmatrix} \underline{y}(t_o + \Delta t) \\ \underline{y}(t_o + \Delta t) \end{bmatrix} = \begin{bmatrix} \phi_{yy}(t_o, \Delta t) & \phi_{yY}(t_o, \Delta t) \\ \phi_{Yy}(t_o, \Delta t) & \phi_{YY}(t_o, \Delta t) \end{bmatrix} \begin{bmatrix} \underline{y}(t_o) \\ \underline{Y}(t_o) \end{bmatrix} \quad (\text{B.14})$$

where

$$\phi(t_o, \Delta t) = \exp \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \Delta t \quad (\text{B.15})$$

From equation (B.14)

$$\underline{y}(t_o + \Delta t) = \phi_{yy}(t_o, \Delta t) \underline{y}(t_o) + \phi_{yY}(t_o, \Delta t) \underline{Y}(t_o) \quad (\text{B.16})$$

$$\underline{Y}(t_o + \Delta t) = \phi_{Yy}(t_o, \Delta t) \underline{y}(t_o) + \phi_{YY}(t_o, \Delta t) \underline{Y}(t_o) \quad (\text{B.17})$$

Applying equation (B.4) and combining (1.16) and (1.17):

$$\begin{aligned} \frac{\partial P(t_o + \Delta t)}{\partial \theta} &= [\phi_{yy}(t_o, \Delta t) + \phi_{Yy}(t_o, \Delta t) \frac{\partial P(t_o)}{\partial \theta}^{-1}] \\ &\quad [\phi_{yy}(t_o, \Delta t) + \phi_{yY}(t_o, \Delta t) \frac{\partial P(t_o)}{\partial \theta}^{-1}] \quad (\text{B.18}) \end{aligned}$$

Appendix C

DEFINITIONS OF VARIABLES USED IN F AND G MATRICES

<u>Parameter</u>	<u>Symbol</u>	<u>Value</u>	<u>Units</u>
Wheelset mass	M	12.045	kg
Nominal longitudinal creep coefficient	f_{11}	8705.5	n
Nominal lateral creep coefficient	f_{22}	2549.0	n
Lateral spring constant	K_y	1490.0	n/m
Yaw spring constant	K_ψ	77.3	n-m/rad
One-half distance between contact points	l	0.147	m
Wheel conicity	α	0.05	rad
Wheelset rolling radius	r_o	0.083	m
Lateral damping constant	c	96.5	n-m-s/rad
Forward velocity	V	3.021	m/s

Appendix D
SIMPLIFICATION OF $\frac{\partial P(t_i)}{\partial \theta}$ FOR WHEELSET PROBLEM

Simplifying equations (B.11) through (B.14) from Appendix B, equation (B.10) reduces to:

$$\begin{bmatrix} \dot{\underline{Y}} \\ \dot{\underline{Y}} \end{bmatrix} = \begin{bmatrix} -F^T + H^T R^{-1} H P & 0 \\ \frac{\partial F}{\partial \theta} P + P \frac{\partial F^T}{\partial \theta} & F - P H^T R^{-1} H \end{bmatrix} \begin{bmatrix} \underline{Y} \\ \underline{Y} \end{bmatrix} \quad (D.1)$$

Equation (D.1) is solved using the state transition matrix approach.

$$\begin{bmatrix} \underline{Y}(t_o + \Delta t) \\ \underline{Y}(t_o + \Delta t) \end{bmatrix} = \begin{bmatrix} \phi_{YY}(\Delta t) & \phi_{Y\dot{Y}}(\Delta t) \\ \phi_{\dot{Y}Y}(\Delta t) & \phi_{\dot{Y}\dot{Y}}(\Delta t) \end{bmatrix} \begin{bmatrix} \underline{Y}(t_o) \\ \underline{Y}(t_o) \end{bmatrix} \quad (D.2)$$

$$\phi(\Delta t) = \exp \begin{bmatrix} -F^T + H^T R^{-1} H P & 0 \\ \frac{\partial F}{\partial \theta} P + P \frac{\partial F^T}{\partial \theta} & F - P H^T R^{-1} H \end{bmatrix} \Delta t \quad (D.3)$$

Using the same methodology as in Appendix B

$$\begin{aligned} \frac{\partial P(t_o + \Delta t)}{\partial \theta} &= \left[\phi_{\dot{Y}Y}(\Delta t) + \phi_{Y\dot{Y}}(\Delta t) \frac{\partial P(t_o)}{\partial \theta} \right] \\ &\quad \left[\phi_{YY}(\Delta t) + \phi_{Y\dot{Y}}(\Delta t) \frac{\partial P(t_o)}{\partial \theta} \right]^{-1} \end{aligned} \quad (D.4)$$

Equation (D.4) is an iterative equation and is solved for $\left. \frac{\partial P}{\partial \theta} \right|_{ss}$.
Because the wheelset is a time invariant system, $\left. \frac{\partial P}{\partial \theta} \right|_{ss}$ is solved
for only once for each iteration of the maximum likelihood equations.

F/0 12/1

UNCLASSIFIED

AFIT-CI-79-254T

ML

212

2000

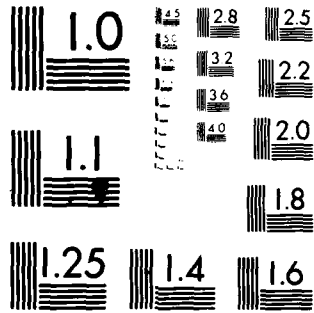
END

DATE _____

FILED



DIIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Appendix E

LISTING OF THE MAXIMUM LIKELIHOOD PROGRAM

CALL INDUMP

C
C
C

THIS PROGRAM IS THE CONTROLLING PROGRAM FOR A SERIES OF
SUBROUTINES WHICH FORM THE MAXIMUM LIKELIHOOD PARAMETER
IDENTIFICATION PROCESSOR FOR THE SIMULATED WHEELSET DATA.

COMMON P(3,3), G(3,1), H(3,3), R(3,3), Q(1,1), K1P, K1G, K2G, K1H, K2H, K1Q,
\$K1R, RKGM(3,3), K1PHIX, K2PHIX, K1PHIZ, K2PEIZ,
\$K1RKGM, K2RKGM, RMASS, P11, P22, RKY, RKSI, RIE, ALPHA, RZERO, DT, NRECF,
\$NRECD, IROSTA, IROEND, IR1STA, IR1END, ILATSA, IDATEN, NDATA, NRCAL,
\$ATTEN(4,3), RVCAI, TVEL, GRCAI, NFOINT, SDT, P(3,3), B(3,3), K1B, K1P,
\$RLIKE, K1DPH, K2DPH, K1DRKG, K2DRKG, NPT1, XIC(3,1), BZERO, K1DLP,
\$K2DLP, K1DELK, K2DELK, K1DP, K2DP, IFLAG

DIMENSION DELTAP(3,3)

DIMENSION DELTAK(3,3)

DIMENSION DPHIX(3,3)

DIMENSION DPMIX(3,3)

DIMENSION ZMEAS(3,1001)

DIMENSION ZMEAS1(3,1001)

DIMENSION XHAT(3,1001)

DIMENSION RNU(3,1001)

DIMENSION ZHAT(3,1001)

DIMENSION DXHAT(3,1001)

DIMENSION XMEAS1(3,1001)

DIMENSION XMEAS(3,1001)

DIMENSION XERROR(3,1001)

DIMENSION YVECT1(1001)

DIMENSION YVECT1(1001)

DIMENSION YVECT2(1001)

DIMENSION U(1,1001)

READ(5,10) K1F, K1G, K2G, K1H, K2H, K1Q, K1R

10 FORMAT(I3)

WRITE(6,15) K1F, K1G, K2G, K1H, K2H, K1Q, K1R

15 FORMAT(/,/, ' ', 5X, 'K1F=', I3, 5X, 'K1G=', I3, 5X, 'K2G=', I3, 5X, 'K1H=',
\$I3, 5X, 'K2H=', I3, 5X, 'K1Q=', I3, 5X, 'K1R=', I3)

K1DELK=K1F

K2DELK=K1F

K1DLP=K1F

K2DLP=K1F

K1DPH=K1F

K2DPH=K1F

K1DP=K1F

K2DP=K1F

READ(5,25) IDATSA, IDATEN

25 FORMAT(I4)

NPOINT= (IDATEN-IDATSA) + 1

NPT1=NPOINT+1

IFLAG=0

WRITE(6,45) IDATSA, IDATEN, NPOINT, NPT1

45 FORMAT(/,/, ' ', 5X, 'IDATSA=', I4, 5X, 'IDATEN=', I4, 5X, 'NPOINT=',
\$I4, 5X, 'NPT1=', I4)

CALL DRIVER(ZMEAS, ZMEAS1, XHAT, RNU, ZHAT, DELTAP, DELTAK, DXHAT,
\$DPHIX, DPMIX, XERROR, XMEAS1, XMEAS, YVECT1, YVECT1, YVECT2, U)

STOP

END

C
C

SUBROUTINE DRIVER(ZMEAS, ZMEAS1, XHAT, RNU, ZHAT, DELTAP, DELTAK, DXHAT,
\$DPHIX, DPMIX, XERROR, XMEAS1, XMEAS, YVECT1, YVECT1, YVECT2, U)
COMMON P(3,3), G(3,1), H(3,3), R(3,3), Q(1,1), K1P, K1G, K2G, K1H, K2H, K1Q,
\$K1R, RKGM(3,3), K1PHIX, K2PHIX, K1PHIZ, K2PEIZ,
\$K1RKGM, K2RKGM, RMASS, P11, P22, RKY, RKSI, RIE, ALPHA, RZERO, DT, NRECF,

```

$NRACD, IROSTA, IROEND, IR1STA, IR1END, IDATSA, IDATEN, NDATA, NRCAL,
$ATTEN(4,3), RVCA1, TVEL, GRCAL, NPOINT, SET, P(3,3), B(3,3), K1B, K1P,
$RLIKE, K1DPH, K2DPH, K1DRKG, K2DRKG, NPT1, XIC(3,1), BZERO, K1DELP,
$K2DELP, K1DELK, K2DELK, K1DF, K2DF, IFIAG
DOUBLE PRECISION PHIZ(6,6)
DIMENSION PHIX(3,3)
DOUBLE PRECISION PHIXDP(3,3)
DIMENSION ZMEAS(3,NPOINT)
DIMENSION ZMEAS1(3,NPT1)
DIMENSION DXHAT(3,NPOINT)
DIMENSION RNU(3,NPOINT)
DIMENSION ZHAT(3,NPOINT)
DIMENSION DELTAP(K1DELP,K2DELP)
DIMENSION DELTAK(K1DELK,K2DELK)
DIMENSION DPHIX(K1DPH,K2DPH)
DIMENSION DFMTX(K1DF,K2DF)
DIMENSION Z(6,6)
DIMENSION XHAT(3,NPOINT)
DIMENSION XMEAS1(3,NPT1)
DIMENSION XMEAS(3,NPOINT)
DIMENSION XERRCR(3,NPOINT)
DIMENSION XVECT1(NPOINT)
DIMENSION YVECT1(NPOINT)
DIMENSION YVECT2(NPOINT)
DOUBLE PRECISION PHIK(3,3)
DIMENSION ITIT1(10)
DIMENSION ITIT2(10)
DIMENSION JLAB1(10)
DIMENSION JLAB2(10)
DIMENSION JLAB3(10)
DIMENSION ITIT3(10)
DIMENSION VDEIJ(20)
DIMENSION VDELJ2(20)
DIMENSION VRLIKE(20)
DIMENSION VTHETA(20)
DIMENSION ITIT5(10)
DIMENSION ITIT4(10)
DIMENSION JLAB4(10)
DIMENSION JLAB5(10)
DIMENSION ITIT6(10)
DIMENSION JLAB6(10)
DIMENSION JLAB7(10)
DIMENSION ITIT7(10)
DIMENSION VCONST(20)
DIMENSION VDET(20)
DIMENSION U(1,NPT1)
TVEL=3.021
CALL PARAM(THETA)
WRITE(6,20) RMASS,P11,P22,RKY,RKSI,RLE,ALPHA,RZERO,THETA,BZERO,
$DT,SDT
20  FORMAT(/, ' ', 2X, 'RMASS=', F12.5, 2X, 'P11=', F12.5, 2X, 'P22=', F12.5,
$2X, 'RKY=', F12.5, 2X, 'RKSI=', F12.5, 2X, 'RLE=', F12.5, //,
$' ', 2X, 'ALPHA=', F12.5, 2X, 'RZERO=', F12.5, 2X, 'THETA=',
$F12.5, 2X, 'BZERO=', F12.5, 2X, 'DT=', F12.5, 2X, 'SDT=', F12.5)
WRITE(6,40)
40  FORMAT(/, ' ', 12X, 'G MATRIX')
CALL OUTPUT(G,K1G,K2G)
WRITE(6,60)
60  FORMAT(/, ' ', 12X, 'H MATRIX')
CALL OUTPUT(H,K1H,K2H)

```

```

      WRITE (6,80)
80  FORMAT(/,' ',12X,'Q MATRIX')
      CALL OUTPUT(Q,K1Q,K1Q)
      WRITE (6,100)
100  FORMAT(/,' ',12X,'R MATRIX')
      CALL OUTPUT(R,K1R,K1R)
      DO 120 I=1,NPT1
      READ (5,140) (ZMEAS1(J,I),J=1,3)
120  CONTINUE
140  FORMAT(3F15.8)
      DO 160 J=1,3
      I=IDATSA-1
      XIC(J,I)=ZMEAS1(J,I)
160  CONTINUE
      WRITE (6,180)
180  FORMAT(/,' ',12X,'XIC')
      CALL OUTPUT(XIC,3,1)
      DO 200 J=1,3
      DO 200 I=IDATSA,IDATEN
      M=I-1
      ZMEAS(J,M)=ZMEAS1(J,I)
200  CONTINUE
      READ (5,210) ITIT1
210  FORMAT(10A4)
      READ (5,210) JLA E1
      READ (5,210) ITIT2
      READ (5,210) JLA E2
      READ (5,210) ITIT3
      READ (5,210) JLA B3
      READ (5,210) ITIT4
      READ (5,210) JLA B4
      READ (5,210) ITIT5
      READ (5,210) JLA B5
      READ (5,210) ITIT6
      READ (5,210) JLA B6
      READ (5,210) ITIT7
      READ (5,210) JLA E7
      DO 212 J=1,NPT1
212  READ (5,213) U(1,J)
213  FORMAT(E15.8)
      DO 800 ISTEP=1,20
      WRITE (6,215) ISTEP
215  FORMAT(////,' ',1X,'ISTEP=',I3)
      WRITE (6,216) THETA
216  FORMAT(/,' ',2X,'THETA=',F15.8)
      CALL SYSMTX(THETA)
      WRITE (6,220)
220  FORMAT(/,' ',12X,'F MATRIX')
      CALL OUTPUT(F,K1F,K1F)
      CALL ZNTRX(Z,K1Z)
      WRITE (6,240)
240  FORMAT(/,' ',12X,'Z MATRIX')
      CALL OUTPUT(Z,K1Z,K1Z)
      CALL STMZ(Z,DT,PHIZ,K1Z,K1PHIZ,K2PHIZ)
      IF(IPLAG.EQ. 1)GO TO 1000
      WRITE (6,260)
260  FORMAT(/,' ',12X,'KALMAN FILTER STM (PHIZ)')
      CALL OUTDP(PHIZ,6,6)
      CALL RICATI(PHIZ,K1PHIZ,P,K1P)
      IF(IPLAG.EQ. 1)GO TO 1000

```

```

WRITE(6,280)
280  FORMAT(/,' ',12X,'STATE COVARIANCE MATRIX (P)')
    CALL OUTPUT(F,K1P,K1P)
    CALL KGMTRX
    WRITE(6,300)
300  FORMAT(/,' ',12X,'KALMAN GAIN MATRIX (RKGM)')
    CALL OUTPUT(RKGM,K1RKGM,K2RKGM)
    CALL STMF(F,DT,PHIXDP,K1P,K1PHIX,K2PHIX)
    CALL DPTOSP(PHIXDP,K1PHIX,K2PHIX,PHIX)
    WRITE(6,320)
320  FORMAT(/,' ',12X,'SYSTEM STATE TRANSITION MATRIX (PHIX)')
    CALL OUTPUT(PHIX,K1PHIX,K2PHIX)
    CALL STATE(ZMEAS,XHAT,PHIX,K1PHIX,U)
339  CALL RINOVA(XHAT,ZMEAS,ZHAT,RNU)
    CALL ZCOV
    WRITE(6,340)
340  FORMAT(/,' ',12X,'MEASUREMENT COVARIANCE MATRIX (B)')
    CALL OUTPUT(B,3,3)
    CALL RMLPI(RNU,SC,SD)
    VCONST(ISTEP)=SC
    VDET(B(ISTEP))=SD
    WRITE(6,360)
360  FORMAT(/,' ',12X,'VALUE OF THE LIKELIHOOD FUNCTION (RLIKE)')
    WRITE(6,380) RLIKE
380  FORMAT(' ',12X,'RLIKE=',E15.8)
    CALL PARF(THETA,DFMTX)
    WRITE(6,400)
400  FORMAT(/,' ',12X,'PARTIAL OF F WRT THETA (DFMTX)')
    CALL OUTPUT(DFMTX,K1DF,K2DF)
    CALL PDP(DFMTX,DELTAP,DELTAK)
    WRITE(6,440)
440  FORMAT(/,' ',12X,'PARTIAL OF P WRT THETA (DELTAP)')
    CALL OUTPUT(DELTAP,K1DELP,K2DELP)
    WRITE(6,460)
460  FORMAT(/,' ',12X,'PARTIAL OF RKGM WRT THETA (DELTAK)')
    CALL OUTPUT(DELTAK,K1DELK,K2DELK)
    CALL XSEN(PHIX,K1PHIX,DFMTX,DELTAK,XHAT,ZMEAS,DXHAT)
    CALL GRADE(DXHAT,RNU,DELTAP,DELTAJ)
    WRITE(6,480)
480  FORMAT(/,' ',12X,'GRADIENT - PARTIAL OF THE LIKELIHOOD FUNCTION',
$' WRT THETA')
    WRITE(6,500) DELTAJ
500  FORMAT(' ',12X,'DELTAJ=',E15.8)
    CALL FISHER(RNU,DXHAT,DELTAP,DELJ2)
    WRITE(6,520)
520  FORMAT(/,' ',12X,'FISHER INFORMATION MATRIX')
    WRITE(6,540) DELJ2
540  FORMAT(' ',12X,'DELJ2=',E15.8)
    CALL STEP(THETA,DELTAJ,DELJ2,DTHETA,THETAN)
    WRITE(6,550) DTHETA
550  FORMAT(/,' ',12X,'DTHETA=',E15.8)
    WRITE(6,560) THETAN
560  FORMAT(/,' ',12X,'THETAN=',E15.8)
    VTHETA(ISTEP)=THETA
    VDELJ(ISTEP)=DELTAJ
    VDELJ2(ISTEP)=DELJ2
    VRLIKE(ISTEP)=RLIKE
    THETA=THETA+0.05
800  CONTINUE
805  CONTINUE

```

```

      WRITE (6,810) ITIT3
810  FORMAT('1',40X,10A4)
      CALL WPLOT1(VTHETA,VRLIKE,ISTEP,40,JIAE3)
      WRITE (6,810) ITIT1
      CALL WPLOT1(VTHETA,VCONST,ISTEP,40,JIAE1)
      WRITE (6,810) ITIT6
      CALL WPLOT1(VTHETA,VDETB,ISTEP,40,JIAE6)
      WRITE (6,810) ITIT4
      CALL WPLOT1(VTHETA,VDELJ,ISTEP,40,JLAB4)
      WRITE (6,810) ITIT5
      CALL WPLOT1(VTHETA,VDELJ2,ISTEP,40,JIAE5)
900  CONTINUE
1000 RETURN
      END

```

```

C
C
C      SUBROUTINE PARAM(THETA)
C          SUBROUTINE PARAM READS IN THE PARAMETERS NECESSARY TO FORM
C          THE SYSTEM MATRIX
      COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
      $K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
      $K1RKGM,K2RKGM,RMASS,F11,F22,RKY,RKSI,RLE,ALPHA,BZERO,DT,NPECE,
      $NRECD,IROSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NPCAL,
      $ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SDT,P(3,3),B(3,3),K1B,K1P,
      $RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
      $K2DELP,K1DELK,K2DELK,K1DP,K2DP,IFLAG
10  FORMAT(F15.8)
      READ(5,10) RMASS,F11,F22,RKY,RKSI,RLE,ALPHA,BZERO,THETA,BZERO,DT,
      $SDT
      DO 20 I=1,K1G
      DO 20 J=1,K2G
20  READ(5,10) G(I,J)
      DO 30 I=1,K1H
      DO 30 J=1,K2H
30  READ(5,10) H(I,J)
      DO 40 I=1,K1Q
      DO 40 J=1,K1Q
40  READ(5,10) Q(I,J)
      DO 50 I=1,K1R
      DO 50 J=1,K1R
50  READ(5,10) P(I,J)
      DO 70 I=1,4
      DO 70 J=1,3
      READ(5,10) ATTEN(I,J)
70  CONTINUE
      RETURN
      END

```

```

C
C
C      SUBROUTINE SYSMTX(THETA)
C          SUBROUTINE SYSMTX FORMS THE SYSTEM MATRIX - P MATRIX
      COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
      $K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
      $K1RKGM,K2RKGM,RMASS,F11,F22,RKY,RKSI,RLE,ALPHA,BZERO,DT,NRECD,
      $NRECD,IROSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
      $ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SDT,P(3,3),B(3,3),K1B,K1P,
      $RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
      $K2DELP,K1DELK,K2DELK,K1DP,K2DP,IFLAG
      F(1,1)=-(((2.0)*F22*THETA)+(BZERO*TVEL))/(RMASS*TVEL)
      F(1,2)=2.0*F22*THETA/RMASS

```

```

P(1,3)=(-RKY)/RMASS
P(2,1)=0.0
P(2,2)=(-RKSI)*TVEL/(2.0*P11*THETA*(RIE**2.0))
P(2,3)=(-ALPHA)*TVEL/(RIE*BZERO)
P(3,1)=1.0
P(3,2)=0.0
P(3,3)=0.0
RETURN
END

```

C
C

SUBROUTINE ZMTRX(Z,K1Z)

C
C SUBROUTINE ZMTRX FORMS THE Z MATRIX FOR THE KALMAN ENGLER
C SOLUTION TO THE RICCATI EQUATION

```

COMMON P(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
$K1A,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1AKGM,K2RKGM,RMASS,P11,P22,RKY,RKSI,RIE,ALPHA,BZERO,DT,NRECF,
$NRECD,IPOSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GFCAL,NPOINT,SLT,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NFT1,XIC(3,1),BZERO,K1DELP,
$K2DELP,K1DELK,K2DELK,K1DF,K2DF,IFLAG
DIMENSION Z(6,6)
DIMENSION FT(3,3)
DIMENSION Z11(3,3)
DIMENSION RI(3,3)
DIMENSION HT(3,3)
DIMENSION FIH(3,3)
DIMENSION Z12(3,3)
DIMENSION GT(1,3)
DIMENSION OGT(3,3)
DIMENSION Z21(3,3)
DIMENSION Z22(3,3)
DOUBLE PRECISION WKAREA(9)
DOUBLE PRECISION RIDP(3,3)
DOUBLE PRECISION RDP(3,3)
01 CALL TRANS(P,FT,K1F,K1F,K1FT,K2FT)
CALL NEG(FT,Z11,K1FT,K2FT,K1Z11,K2Z11)
IDGT=0
CALL SPTODP(R,K1R,K1R,REP)
CALL LINVIF(RDP,K1R,K1R,RIDP,IDGT,WKAREA,IEP)
CALL DPTOSP(RIDF,K1R,K1F,RI)
CALL TRANS(H,HT,K1H,K2H,K1HT,K2HT)
CALL MULT(RI,H,RIH,K1R,K1R,K1H,K2H,K1RIH,K2RIH)
CALL MULT(HT,RIH,Z12,K1HT,K2HT,K1RIH,K2RIH,K1Z12,K2Z12)
CALL TRANS(G,GT,K1G,K2G,K1GT,K2GT)
CALL MULT(Q,GT,OGT,K1Q,K1Q,K1GT,K2GT,K1OGT,K2OGT)
CALL MULT(G,OGT,Z21,K1G,K2G,K1OGT,K2OGT,K1Z21,K2Z21)
CALL SCALAR(F,1.0,Z22,K1F,K1F,K1Z22,K2Z22)
DO 10 I=1,K1Z11
DO 10 J=1,K2Z11
M=I
N=J
10 Z(M,N)=Z11(I,J)
DO 20 I=1,K1Z12
DO 20 J=1,K2Z12
M=I
N=J+K2Z11
20 Z(M,N)=Z12(I,J)
DO 30 I=1,K1Z21
DO 30 J=1,K2Z21

```

```

      M=I+K1Z11
      N=J
30    Z(4,N)=Z21(I,J)
      DO 40 I=1,K1Z22
      DO 40 J=1,K2Z22
      M=I+K1Z21
      N=J+K2Z12
40    Z(M,N)=Z22(I,J)
      K1Z=K1Z11+K1Z21
      RETURN
      END

```

C
C

```

      SUBROUTINE STMF(F,DT,PHI,K1F,K1PHI,K2PHI)
      SUBROUTINE STMF DIMENSIONS THE NECESSARY ARRAYS FOR
      SUBROUTINE STM. IT ACTS AS A DRIVER FOR SUBROUTINE STM.
      DOUBLE PRECISION FDP(3,3)
      DIMENSION F(K1F,K1F)
      DOUBLE PRECISION PHI(K1F,K1F)
      DOUBLE PRECISION FDT(3,3)
      DOUBLE PRECISION PRD(3,3)
      DOUBLE PRECISION PRD1(3,3)
      DOUBLE PRECISION TERM(3,3)
      DOUBLE PRECISION PHI1(3,3)
      DOUBLE PRECISION DIFF(3,3)
      DOUBLE PRECISION RM(3,3)
01    CALL SPTODP(F,K1F,K1F,FDP)
      CALL STM(FDP,PHI,FDT,PRD,PRD1,TERM,PHI1,DIFF,RM,DT,K1F,
      $K1PHI,K2PHI,IFLAG)
      RETURN
      END

```

C
C

```

      SUBROUTINE STMZ(Z,DT,PHI,K1Z,K1PHIZ,K2PHIZ)
      SUBROUTINE STMZ DIMENSIONS THE NECESSARY ARRAYS FOR SUB-
      ROUTINE STM. IT ACTS AS A DRIVER FOR SUBROUTINE STM
      DIMENSION Z(K1Z,K1Z)
      DOUBLE PRECISION PHI(6,6)
      DOUBLE PRECISION FDT(6,6)
      DOUBLE PRECISION PRD(6,6)
      DOUBLE PRECISION PRD1(6,6)
      DOUBLE PRECISION TERM(6,6)
      DOUBLE PRECISION ZDP(6,6)
      DOUBLE PRECISION PHI1(6,6)
      DOUBLE PRECISION DIFF(6,6)
      DOUBLE PRECISION RM(6,6)
01    CALL SPTODP(Z,K1Z,K1Z,ZDP)
      CALL STM(ZDP,PHI,FDT,PRD,PRD1,TERM,PHI1,DIFF,RM,DT,K1Z,
      $K1PHIZ,K2PHIZ,IFLAG)
      RETURN
      END

```

C
C

```

      SUBROUTINE STM(F,PHI,FDT,PRD,PRD1,TERM,PHI1,DIFF,RM,DT,K1F,
      $K1PHI,K2PHI,IFLAG)
      SUBROUTINE STM CALCULATES THE STATE TRANSITION MATRIX.
      DOUBLE PRECISION F(K1F,K1F)
      DOUBLE PRECISION FDT(K1F,K1F)
      DOUBLE PRECISION PHI(K1F,K1F)
      DOUBLE PRECISION PRD(K1F,K1F)

```

```

DOUBLE PRECISION PHI1(K1F,K1F)
DOUBLE PRECISION PRD1(K1F,K1F)
DOUBLE PRECISION DIFF(K1F,K1F)
DOUBLE PRECISION TERM(K1F,K1F)
DOUBLE PRECISION RM(K1F,K1F)
DOUBLE PRECISION B
DOUBLE PRECISION COEF
DOUBLE PRECISION CT
DOUBLE PRECISION CHK
DOUBLE PRECISION DTD
DOUBLE PRECISION FACTDP
01 DTD=DBLE(DT)
   CT=DBLE(1.0E-16)
   IFLAG=0
   B=DBLE(0.0)
   CALL SCALDP(F,B,RM,K1F,K1F,K1RM,K2RM)
   DO 20 I=1,K1RM
20  RM(I,I)=DBLE(1.0)
   DO 320 I=1,200
   IF(I.GT. 1)GO TO 120
   CALL SCALDP(F,DTD,FDT,K1F,K1F,K1FDT,K2FDT)
   CALL ADDDP(RM,FDT,PHI,K1RM,K2RM,K1PHI,K2PHI)
   B=DBLE(1.0)
   CALL SCALDP(F,B,PRD,K1F,K1F,K1PRD,K2PRI)
   GO TO 300
120 CALL MULTDP(F,PRD,PRD1,K1F,K1F,K1PRD,K2PRD,K1PRD1,K2PRD1)
   CHK=DELE(1.0E 71)
   CALL OVERDP(PRD1,CHK,K1PRD1,K2PRD1,IFLAG)
   IF(IFLAG.EQ. 0)GO TO 160
   WRITE(6,125)
125  FORMAT(/,' ',2X,'PRD1 EXCEEDS EXPONENT OVERFLOW.',
$1X,'CURRENT VALUE OF PHI USED.  EXECUTION CONTINUES.')
```

```

   GO TO 420
160  B=DBLE(1.0)
   CALL SCALDP(PRI1,B,PRD,K1PRD1,K2PRD1,K1PRD,K2PRD)
   COEF=(DT*I)/(FACTDP(I))
   CHK=DBLE(1.0E-76)
   IF(COEF.LE. CHK)GO TO 380
   CALL SCALDP(PRI,COEF,TERM,K1PRD,K2PRI,K1TERM,K2TERM)
   CALL ADDDP(PHI,TERM,PHI,K1PHI,K2PHI,K1PHI,K2PHI)
   CALL SUBDP(PHI,PHI1,DIFF,K1PHI,K2PHI,K1DIFF,K2DIFF)
   DO 260 J=1,K1PHI
   DO 260 K=1,K2PHI
   IF(DABS(DIFF(J,K)).GT. CT)GO TO 280
260  CONTINUE
   GO TO 420
280  IF(I-200)300,340,340
300  B=DBLE(1.0)
   CALL SCALDP(PHI,B,PHI1,K1PHI,K2PHI,K1PHI1,K2PHI1)
320  CONTINUE
340  WRITE(6,360)
360  FORMAT(/,' ',5X,'200 ITERATIONS. NO CONVERGENCE')
   GO TO 450
380  WRITE(6,400)
400  FORMAT(/,' ',2X,'COEF .LE. 1.0E-76. ITERATION STOPPED.',
$1X,'CURRENT VALUE OF PHI USED.  EXECUTION CONTINUES.')
```

```

420  WRITE(6,440)I
440  FORMAT(/,' ',2X,I3,1X,'ITERATIONS FOR STM.')
```

```

450  WRITE(6,460)
460  FORMAT(/,' ',12X,'STM DIFF MATRIX')
```

```

480 CALL OUTDP(DIFF,K1DIFF,K2DIFF)
RETURN
END

```

C
C

```

SUBROUTINE RICATI(PHIZ,K1PHI,PSP,K1P)
SUBROUTINE FICCATI CALCULATES THE STATE COVARIANCE MATRIX I.E.
THE P MATRIX.
C

```

```

DOUBBLE PRECISION PHI11(3,3)
DOUBBLE PRECISION PHI12(3,3)
DOUBBLE PRECISION PHI21(3,3)
DOUBBLE PRECISION PHI22(3,3)
DOUBBLE PRECISION W(3,3)
DOUBBLE PRECISION X(3,3)
DOUBBLE PRECISION Y(3,3)
DOUBBLE PRECISION Z(3,3)
DOUBBLE PRECISION DIFF(3,3)
DOUBBLE PRECISION F1(3,3)
DOUBBLE PRECISION P2(3,3)
DOUBBLE PRECISION F2T(3,3)
DOUBBLE PRECISION ZI(3,3)
DOUBBLE PRECISION PHIZ(K1PHI,K1PHI)
DOUBBLE PRECISION PSUM(3,3)
DOUBBLE PRECISION F(3,3)
DIMENSION PSP(3,3)
DOUBBLE PRECISION CONST
DOUBBLE PRECISION ERROR
DOUBBLE PRECISION WKAREA(36)
DOUBBLE PRECISION A,B
01 ERROR=DELE(1.0E-16)
K11=K1PHI/2.0
DO 20 I=1,K11
DO 20 J=1,K11
20 PHI11(I,J)=PHIZ(I,J)
DO 40 I=1,K11
DO 40 J=1,K11
M=I
N=K11+J
40 PHI12(I,J)=PHIZ(M,N)
DO 60 I=1,K11
DO 60 J=1,K11
M=I+K11
N=J
60 PHI21(I,J)=PHIZ(M,N)
DO 80 I=1,K11
DO 80 J=1,K11
M=K11+I
N=K11+J
80 PHI22(I,J)=PHIZ(M,N)
A=DELE(0.0)
CALL SCALDP(PHI11,A,P2,K11,K11,K1P2,K2P2)
ICOUNT=1
85 B=DELE(1.0)
CALL SCALDP(P2,B,P1,K1P2,K2P2,K1P1,K2P1)
CALL MULTDP(PHI22,P2,W,K11,K11,K1P2,K2F2,K1W,K2W)
CALL ADDDP(PHI21,W,X,K11,K11,K1X,K2X)
CALL MULTDP(PHI12,P2,Y,K11,K11,K1P2,K2P2,K1Y,K2Y)
CALL ADDDP(PHI11,Y,Z,K11,K11,K1Z,K2Z)
IDGT=0
CALL LINVIP(Z,K1Z,K11,ZI,IDGT,WKAREA,IER)

```

```

CALL MULTDP(X,ZI,F2,K1X,K2X,K1Z,K2Z,K1P2,K2P2)
CALL TRANSD(P2,P2T,K1P2,K2P2,K1P2T,K2P2T)
CALL ADDDP(P2,P2T,PSUM,K1P2,K2P2,K1PSUM,K2PSUM)
CONST=DELE(.500)
CALL SCALDP(PSUM,CONST,F2,K1PSUM,K2PSUM,K1P2,K2P2)
CALL SUBDP(P2,P1,DIFF,K1P2,K2P2,K1DIFF,K2DIFF)
DO 100 I=1,K1DIFF
DO 100 J=1,K1DIFF
90 IF(DABS(DIFF(I,J))-ERROR) 100,100,110
100 CONTINUE
GO TO 150
110 ICOUNT=ICOUNT+1
IF(ICOUNT.GT. 200) GO TO 115
GO TO 85
115 WRITE(6,120)
120 FORMAT(//,' ',5X,'200 ITERATIONS. RICCATI SOLUTION DOES NOT',
$' CONVERGE.')
```

```

B=DELE(1.0)
150 CALL SCALDP(P2,B,P,K1P2,K2P2,K1P,K2P)
CALL DPTOSP(P,K1P,K1P,PSP)
IF(ICOUNT.GE. 200) GO TO 165
WRITE(6,160)ICCOUNT
160 FORMAT(//,' ',5X,I3,1X,'ITERATIONS FOR RICCATI SOLUTION TO',
$' CONVERGE.')
```

```

165 WRITE(6,170)
170 FORMAT(//,' ',12X,'RICCATI DIFFERENCE MATRIX.')
```

```

CALL OUTDP(DIFF,K1DIFF,K2DIFF)
RETURN
END
```

C
C

SUBROUTINE KGMTRX

C
C
C
SUBROUTINE KGMTRX CALCULATES THE KALMAN GAIN MATRIX.

```

COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
$K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1RKGM,K2RKGM,RMASS,F11,P22,RKY,PKST,RIE,ALPHA,RZERO,DT,NRECR,
$NRECD,IROSTA,IROEND,IR1STA,IR1END,IDATSA,IDAT3N,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SET,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NP11,XIC(3,1),BZERO,K1DELP,
$K2DELP,K1DEIK,K2DEIK,K1DF,K2DF,IFLAG
DIMENSION HTRI(3,3)
DIMENSION HT(3,3)
DIMENSION RI(3,3)
DOUBLE PRECISION WKAREA(3)
DOUBLE PRECISION RIDP(3,3)
DOUBLE PRECISION RDP(3,3)
01 IDGT=0
CALL SPTODP(R,K1R,K1R,FDP)
CALL LINV1F(RDF,K1R,K1R,RIDP,IDGT,WKAREA,IER)
CALL DPTOSP(RIDF,K1R,K1R,RI)
CALL TRANS(H,HT,K1H,K2H,K1HT,K2HT)
CALL MULT(HT,RI,HTRI,K1HT,K2HT,K1R,K1R,K1HTRI,K2HTRI)
CALL MULT(P,HTRI,RKGM,K1P,K1P,K1H1RI,K2HTRI,K1RKGM,K2RKGM)
RETURN
END
```

C
C

SUBROUTINE STABLE

```

COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
$K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
```

```

SK1RKGM,K2RKGM,REASS,P11,P22,BKY,RKSI,RIE,ALPHA,RZERO,DT,NRECR,
SNRECD,IROSTA,IROEND,IR1STA,IR1END,ICATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SDT,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
$K2DELP,K1DEIK,K2DEIK,K1DF,K2DF,IPLAG
  DIMENSION RH(3,3)
  DIMENSION PKMTX(3,3)
  DOUBLE PRECISION WKAREA(6)
  COMPLEX*16 EIGVEC(3,3)
  COMPLEX*16 WDF(3)
  DOUBLE PRECISION PKDP(3,3)
  CALL MULT(RKGM,H,RH,K1RKGM,K2RKGM,K1H,K2H,K1RH,K2RH)
  CALL SUB(P,RH,PKMTX,K1P,K1P,K1PK,K2PK)
  WRITE(6,10)
10  FORMAT(/,/, ' ',12X,'KALMAN FILTER SYSTEM MATRIX - (F-KH) ')
  CALL OUTPUT(PKMTX,K1PK,K2PK)
  IJOB=0
  CALL SPTODP(PKMTX,K1P,K1P,PKDP)
  WRITE(6,15) IER
15  FORMAT(/,/, ' ',7X,'IER=',I4)
  WRITE(6,20)
20  FORMAT(/,/, ' ',12X,'EIGENVALUES OF (F-KH) ')
  DO 40 I=1,K1PK
40  WRITE(6,60) WDF(I)
60  FORMAT(' ',14X,2D15.8)
  DO 80 I=1,K1PK
  IF (REAL(WDF(I)) .LE. 0) GO TO 120
80  CONTINUE
  WRITE(6,100)
100 FORMAT(/,/, ' ', '***** FILTERED SYSTEM IS UNSTABLE *****')
  IFLAG=1
120 RETURN
  END

```

C
C

C
C
C
C
SUBROUTINE STATE(ZMEAS,XHAT,PHIK2,K1PHIK,U)
SUBROUTINE STATE CALCULATES THE STATE ESTIMATES OF THE
KALMAN FILTER ACCORDING TO THE CONTINUOUS FORM OF THE
KALMAN FILTER.

```

COMMON P(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1P,K1G,K2G,K1H,K2H,K1Q,
$K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1RKGM,K2RKGM,REASS,P11,P22,BKY,RKSI,RIE,ALPHA,RZERO,DT,NRECR,
$SNRECD,IROSTA,IROEND,IR1STA,IR1END,ICATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SDT,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
$K2DELP,K1DEIK,K2DEIK,K1DF,K2DF,IPLAG
  DIMENSION ZMEAS(3,NPOINT)
  DIMENSION XHAT(3,NPOINT)
  DIMENSION RH(3,3)
  DOUBLE PRECISION WKAREA(36)
  DIMENSION PHIK1(3,3)
  DIMENSION HNEG(3,3)
  DOUBLE PRECISION PHIK(3,3)
  DIMENSION PHIKSP(3,3)
  DIMENSION PKF(3,3)
  DIMENSION PKFI(3,3)
  DIMENSION RM(3,3)
  DIMENSION T2(3,3)
  DOUBLE PRECISION PHIK2(3,3)
  DOUBLE PRECISION FACTOR

```

```

DIMENSION PRD1(3,3)
DIMENSION PRD2(3,3)
DIMENSION XM1(3,1)
DIMENSION XE1(3,1)
DIMENSION XE2(3,1)
DIMENSION GAMMA(3,3)
DIMENSION X1(3,1)
DIMENSION X2(3,1)
DOUBLE PRECISION PKFDP(3,3)
DOUBLE PRECISION PKFIDP(3,3)
DOUBLE PRECISION PHIKID(3,3)
DIMENSION U(1,NPT1)
DIMENSION U1(1,1)
DIMENSION A1(3,1)
DIMENSION A2(3,1)
DIMENSION LAMEDA(3,1)
DIMENSION X3(3,1)
DIMENSION XEP(3,1)
CALL NEG(H,HNEG,K1H,K2H,K1HNEG,K2HNEG)
CALL MULT(RKGM,HNEG,RH,K1RKGM,K2RKGM,K1HNEG,K2HNEG,K1RH,K2RH)
CALL ADD(F,RH,PKF,K1F,K1F,K1FKF,K2FKF)
WRITE(6,5)
5  FORMAT(//,' ',12X,'KALMAN FILTER SYSTEM MATRIX (PKF)')
CALL OUTPUT(PKF,K1FKF,K2FKF)
CALL STMF(PKF,DI,PHIK,K1FKF,K1PHIK,K2PHIK)
FACTOR=1.0D 00
CALL SCALDP(PHIK,FACTOR,PHIK2,K1PHIK,K1PHIK,K1PHK2,K2PHK2)
CALL DPTOSP(PHIK,K1PHIK,K2PHIK,PHIKSP)
WRITE(6,10)
10  FORMAT(//,' ',12X,'STM FOR KALMAN FILTER SYSTEM (PHIK)')
CALL OUTDP(PHIK,K1PHIK,K2PHIK)
IDGT=0
CALL LINV1F(PHIK,K1PHIK,K1PHIK,PHIKID,IDGT,WKAREA,IER)
CALL DPTOSP(PHIKID,K1PHIK,K2PHIK,PHIKI)
IDGT=0
CALL SPTODP(PKF,K1FKF,K2FKF,PKFDP)
CALL LINV1F(PKFDP,K1FKF,K1FKF,PKFIDP,IDGT,WKAREA,IER)
CALL DPTOSP(PKFIDP,K1FKF,K2FKF,PKFI)
CALL SCALAR(F,0.0,RM,K1F,K1F,K1RM,K2RM)
DO 20 I=1,K1RM
RM(I,I)=1.0
20  CONTINUE
CALL SUB(RM,PHIKI,T2,K1RM,K2RM,K1T2,K2T2)
CALL MULT(T2,RKGM,PRD1,K1T2,K2T2,K1RKGM,K2RKGM,K1PRD1,K2PRD1)
CALL MULT(PKFI,PRD1,PRD2,K1FKF,K2FKF,K1PRD1,K2PRD1,K1PRD2,K2PRD2)
CALL MULT(PHIKSP,PRD2,GAMMA,K1PHIK,K2PHIK,K1PRD2,K2PRD2,K1GAM,
$K2GAM)
CALL MULT(T2,G,A1,K1T2,K2T2,K1G,K2G,K1A1,K2A1)
CALL MULT(PKFI,A1,A2,K1FKF,K2FKF,K1A1,K2A1,K1A2,K2A2)
CALL MULT(PHIKSP,A2,LAMBDA,K1PHIK,K1PHIK,K1A2,K2A2,K1LAMB,K2LAMB)
CALL SCALAR(XIC,1.0,XE1,3,1,K1XE1,K2XE1)
DO 1000 I=1,NPOINT
DO 100 J=1,3
XM1(J,1)=ZMEAS(J,I)
100  CONTINUE
U1(1,1)=U(1,I)
CALL MULT(PHIKSP,XE1,X1,K1PHIK,K2PHIK,K1XE1,K2XE1,K1X1,K2X1)
CALL MULT(GAMMA,XM1,X2,K1GAM,K2GAM,3,1,K1X2,K2X2)
CALL MULT(LAMEDA,U1,X3,K1LAMB,K2LAMB,1,1,K1X3,K2X3)
CALL ADD(X1,X2,XEP,K1X1,K2X1,K1XEP,K2XEP)

```

```

CALL ADD(XEP,X3,XE2,K1XEP,K2XEP,K1XE2,K2XE2)
DO 200 J=1,3
  XHAT(J,1)=XE2(J,1)
200 CONTINUE
CALL SCALAP(XE2,1.0,XE1,K1XE2,K2XE2,K1XE1,K2XE1)
1000 CONTINUE
RETURN
END

```

C
C

```

SUBROUTINE ZCOV
  SUBROUTINE ZCOV CALCULATES THE F MATRIX FOR THE MAXIMUM
  LIKELIHOOD PARAMETER IDENTIFICATION PROCESSOR.
  COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
  $K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
  $K1RKGM,K2RKGM,RMASS,F11,P22,RKY,RKSI,R1E,ALPHA,RZERO,DT,NRECR,
  $NRECD,IPOSTA,IPOEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
  $ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SIT,P(3,3),B(3,3),K1E,K1P,
  $RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
  $K2DELP,K1DEIK,K2DEIK,K1DF,K2DF,IFLAG
  DIMENSION HT(3,3),PHT(3,3),HPHT(3,3)
01 CALL TRANS(H,HT,K1H,K2H,K1HT,K2HT)
  CALL MULT(P,HT,PHT,K1P,K1P,K1HT,K2HT,K1PHT,K2PHT)
  CALL MULT(H,PHT,HPHT,K1H,K2H,K1PHT,K2PHT,K1HPHT,K2HPHT)
  CALL ADD(HPHT,R,B,K1HPHT,K2HPHT,K1E,K2E)
  RETURN
END

```

C
C

```

SUBROUTINE RINCVA(XHAT,ZMEAS,ZHAT,RNU)
  SUBROUTINE RINCVA CALCULATES THE INNOVATION FOR THE MLPI
  PROCESSOR.
  COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
  $K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
  $K1RKGM,K2RKGM,RMASS,F11,P22,RKY,RKSI,R1E,ALPHA,RZERO,DT,NRECR,
  $NRECD,IPOSTA,IPOEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
  $ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SIT,P(3,3),B(3,3),K1E,K1P,
  $RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
  $K2DELP,K1DEIK,K2DEIK,K1DF,K2DF,IFLAG
  DIMENSION XHAT(3,NPOINT)
  DIMENSION ZMEAS(3,NPOINT)
  DIMENSION RNU(3,NPOINT)
  DIMENSION ZHAT(3,NPOINT)
01 CALL MULT(H,XHAT,ZHAT,K1H,K2H,3,NPOINT,K1ZHAT,K2ZHAT)
  CALL SUB(ZMEAS,ZHAT,RNU,3,NPOINT,K1RNU,K2RNU)
  RETURN
END

```

C
C

```

SUBROUTINE RMLPI(RNU,SC,SD)
  SUBROUTINE RMLPI CALCULATES THE VALUE OF THE LIKELIHOOD FUNC-
  TION.
  COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
  $K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
  $K1RKGM,K2RKGM,RMASS,F11,P22,RKY,RKSI,R1E,ALPHA,RZERO,DT,NRECR,
  $NRECD,IPOSTA,IPOEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
  $ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SIT,P(3,3),B(3,3),K1E,K1P,
  $RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
  $K2DELP,K1DEIK,K2DEIK,K1DF,K2DF,IFLAG
  DIMENSION RNU(3,NPOINT)

```

```

DIMENSION C(3,1)
DIMENSION ATC(1,1)
DIMENSION AT(1,3)
DIMENSION A(3,1)
DIMENSION BI(3,3)
DOUBLE PRECISION EX
DOUBLE PRECISION WKAREA(9)
DOUBLE PRECISION BIDP(3,3)
DOUBLE PRECISION DETBDP
DOUBLE PRECISION BEP(3,3)
DOUBLE PRECISION VECTOR(3)
DOUBLE PRECISION D1DP
DOUBLE PRECISION D2DP
DOUBLE PRECISION BDP1(3,3)
DOUBLE PRECISION FACTOR
01 SUM=0.0
SC=0.0
SD=0.0
IDGT=0
CALL SPTODP(B,K1B,K1B,BDP)
FACTOR=1.00D 00
CALL SCALDP(BDP,FACTOR,BDP1,K1B,K1B,K1BDP1,K2BDP1)
CALL LINV1P(BDP,K1B,K1B,BIDP,IDGT,WKAREA,IER)
CALL DPTOSP(BIDP,K1B,K1B,BI)
IJOB=4
D1DP=0.00D 00
CALL LINV3P(BDP1,VECTOR,IJOB,K1B,K1B,D1DP,D2DP,WKAREA,IER)
IF(D2DP .GE. C.CCI 00) GO TO 50
D2DP=-D2DP
EX=2.00D 00**D2DP
DETBDP=D1DP/EX
GO TO 60
50 EX=2.00D 00**D2DP
DETBDP=D1DP*EX
60 DETB=DETBDP
S1=ALOG(DETB)
WRITE(6,65) S1
65 FORMAT(//,' ',12X,'NATURAL LOG OF THE DETERMINANT OF B= ',
$1X,E15.8)
DO 100 I=1,NPCINT
DO 80 J=1,3
A(J,1)=RNU(J,I)
80 CONTINUE
CALL MULT(BI,A,C,3,3,3,1,K1C,K2C)
CALL TRANS(A,AT,3,1,K1AT,K2AT)
CALL MULT(AT,C,ATC,K1AT,K2AT,K1C,K2C,K1ATC,K2ATC)
CONST=ATC(1,1)
SC=SC+CONST
SD=SD+ALOG(DETB)
SUM=SUM+(CONST+ALOG(DETB))
100 CONTINUE
SD=(-1.0/2.0)*SD
SC=(-1.0/2.0)*SC
RLIKE=(-1.0/2.0)*SUM
RETURN
END

```

C
C
C

SUBROUTINE PARPHI(DPMIX,DPHIX)

SUBROUTINE PARPHI CALCULATES THE PARTIAL DERIVATIVE OF THE

```

C      STATE TRANSITION MATRIX (PHI) WITH RESPECT TO THETA.
COMMON P(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1P,K1G,K2G,K1H,K2H,K1Q,
$K1A,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1RKGM,K2RKGM,RMASS,P11,P22,FKY,RKSI,RLE,ALPHA,RZERO,DT,NRECR,
$NRCD,IROSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPCINT,SDT,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
$K2JELP,K1DELK,K2DELK,K1DF,K2DF,IFLAG
DATA ERROR/.000001/
DIMENSION PRD(3,3)
DIMENSION PRD1(3,3)
DIMENSION TERM(3,3)
DIMENSION SUM(3,3)
DIMENSION DFMTX(K1DF,K2DF)
DIMENSION DPHIX(K1DPH,K2DPH)
DIMENSION PRD2(3,3)
DIMENSION DIFF(3,3)
01 CALL SCALAR(F,C.0,DPHIX,K1P,K1P,K1DPH,K2DPH)
CALL SCALAR(F,1.0,PRD,K1P,K1P,K1PRD,K2PRD)
DO 100 I=1,200
COEF=(1/FACT(I))*(DT**I)*I
CALL MULT(PRD,DFMTX,PRD2,K1PRD,K2PRD,K1DF,K2DF,K1PRD2,K2PRD2)
CALL SCALAR(PRD2,COEF,TERM,K1PRD2,K2PRD2,K1TERM,K2TERM)
CALL SCALAR(DPHIX,1.0,SUM,K1DPH,K2DPH,K1SUM,K2SUM)
CALL ADD(TERM,DPHIX,DPHIX,K1TERM,K2TERM,K1DPH,K2DPH)
CALL SUB(DPHIX,SUM,DIFF,K1DPH,K2DPH,K1DIFF,K2DIFF)
CALL MULT(F,FFD,PRD1,K1F,K1P,K1PRD,K2PRD,K1PRD1,K2PRD1)
CALL SCALAR(PRD1,1.0,PRD,K1PRD1,K2PRD1,K1PRD,K2PRD)
DO 50 J=1,K1DIFF
DO 50 K=1,K2DIFF
IF(ABS(DIFF(J,K))-ERROR)50,50,100
50 CONTINUE
GO TO 200
100 CONTINUE
WRITE(6,110)
110 FORMAT(//,' ',10X,'200 ITERATIONS. NO CONVERGENCE.')
200 RETURN
END

```

```

C
C      SUBROUTINE PARF(THETA,DFMTX)
C      SUBROUTINE PARF CALCULATES THE PARTIAL DERIVATIVE OF THE SYSTEM
C      MATRIX WITH RESPECT TO THETA.
COMMON P(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1P,K1G,K2G,K1H,K2H,K1Q,
$K1A,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1RKGM,K2RKGM,RMASS,P11,P22,FKY,RKSI,RLE,ALPHA,RZERO,DT,NRECR,
$NRCD,IROSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SDT,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
$K2JELP,K1DELK,K2DELK,K1DF,K2DF,IFLAG
DIMENSION DFMTX(K1DF,K2DF)
DIMENSION A(3,3)
A(1,1)=((-2.0)*P22)/(RMASS*TVEL)
A(1,2)=(2.0*P22)/RMASS
A(1,3)=0.0
A(2,1)=0.0
A(2,2)=(RKSI*TVEL)/(2.0*(THETA**2.0)*P11*(RLE**2.0))
A(2,3)=0.0
A(3,1)=0.0
A(3,2)=0.0

```

```

A(3,3)=0.0
CALL SCALAR(A,1.0,DFMTX,3,3,K1DF,K2DF)
RETURN
END

```

C
C

```

SUBROUTINE PDP(DFMTX,DELTAP,DELTAK)
COMMON F(3,3),G(3,1),H(3,3),R(3,3),C(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
$K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1RKGM,K2RKGM,RMASS,F11,F22,RKY,RKSI,RIE,ALPHA,RZERO,DT,NPCCR,
$NRECD,IROSTA,IFEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SET,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
$K2DELP,K1DELK,K2DELK,K1DF,K2DF,IPIAG
DIMENSION RI(3,3)
DIMENSION HTRI(3,3)
DIMENSION DELTAK(3,3)
DIMENSION DFMTX(3,3)
DIMENSION HT(3,3)
DIMENSION DFMTXT(3,3)
DIMENSION HP(3,3)
DIMENSION RIHP(3,3)
DIMENSION PRD1(3,3)
DIMENSION FT(3,3)
DIMENSION A11(3,3)
DIMENSION PRD2(3,3)
DIMENSION PRD3(3,3)
DIMENSION A21(3,3)
DIMENSION RIH(3,3)
DIMENSION HRIH(3,3)
DIMENSION PRD4(3,3)
DIMENSION A22(3,3)
DIMENSION A12(3,3)
DIMENSION DELTAP(3,3)
DOUBLE PRECISION PHI(6,6)
DIMENSION ZDELP(6,6)
DOUBLE PRECISION RIDP(3,3)
DOUBLE PRECISION RDP(3,3)
DOUBLE PRECISION WKAREA(9)
NAMELIST/DIMEN/M
C1 CALL TRANS(F,FT,K1F,K1F,K1FT,K2FT)
CALL SPTODP(R,K1R,K1R,RI)
IDGT=0
CALL LINV1P(RDP,K1P,K1R,RIDP,IDGT,WKAREA,IFR)
CALL DPTOSP(RIDP,K1F,K1F,RI)
CALL TRANS(H,HT,K1H,K2H,K1HT,K2HT)
CALL TRANS(DFMTX,DFMTXT,K1F,K1F,K1DFM,K2DFM)
CALL MULT(H,P,HP,K1H,K2H,K1F,K1P,K1HP,K2HP)
CALL MULT(RI,HP,RIHP,K1R,K1R,K1HP,K2HP,K1RIHP,K2RIHP)
CALL MULT(HT,RIHP,PRD1,K1HT,K2HT,K1RIHP,K2RIHP,K1PRD1,K2PRD1)
CALL SUB(PR1,FT,A11,K1PRD1,K2PRD1,K1A11,K2A11)
CALL MULT(P,DFMTXT,PRD2,K1P,K1P,K1DFM,K2DFM,K1PRD2,K2PRD2)
CALL MULT(DFMTX,P,PRD3,K1DFM,K2DFM,K1P,K1P,K1PRD3,K2PRD3)
CALL ADD(PR2,PRD3,A21,K1PRD2,K2PRD2,K1A21,K2A21)
CALL MULT(RI,H,RIH,K1R,K1R,K1H,K2H,K1RIH,K2RIH)
CALL MULT(HT,RIH,HRIH,K1HT,K2HT,K1RIH,K2RIH,K1HRIH,K2HRIH)
CALL MULT(P,HRIH,PRD4,K1P,K1P,K1HRIH,K2HRIH,K1PRD4,K2PRD4)
CALL SUB(P,PRD4,A22,K1P,K1P,K1A22,K2A22)
CALL SCALAR(P,0.0,A12,K1P,K1P,K1A12,K2A12)
DO 20 I=1,3

```

```

DO 20 J=1,3
ZDEL P(I,J)=A11(I,J)
20 CONTINUE
DO 40 I=1,3
DO 40 J=1,3
M=I
N=J+K2A11
ZDEL P(M,N)=A12(I,J)
40 CONTINUE
DO 60 I=1,3
DO 60 J=1,3
M=I+K1A11
N=J
ZDEL P(M,N)=A21(I,J)
60 CONTINUE
DO 80 I=1,3
DO 80 J=1,3
M=I+K1A11
N=J+K2A11
ZDEL P(M,N)=A22(I,J)
80 CONTINUE
WRITE(6,100)
100 FORMAT(//,' ',12X,'Z MATRIX FOR SOLUTION TO PARTIAL(P) /',
$'PARTIAL(THETA)')
CALL OUTPUT(ZDEL P,M,M)
WRITE(6,DIMEN)
CALL STMZ(ZDEL P,DT,PHI,M,K1PHI,K2PHI)
WRITE(6,120)
120 FORMAT(//,' ',12X,'STM FOR SOLUTION TO PARTIAL(P) /',
$'PARTIAL(THETA)')
CALL OUTDP(PHI,K1PHI,K2PHI)
CALL RICATI(PHI,K1PHI,DELTAP,K1DEL P)
CALL TRANS(H,HT,K1H,K2H,K1HT,K2HT)
CALL MULT(HT,FI,HTRI,K1HT,K2HT,K1R,K1R,K1HTRI,K2HTRI)
CALL MULT(DELTAP,HTPI,DELTAK,K1DEL P,K2DEL P,K1HTRI,K2HTRI,K1DEL K,
$K2DEL K)
RETURN
END

```

C
C
C
C

```

SUBROUTINE XSEN(PHIK,K1PHIK,DPMTX,DELTAK,XHAT,ZMEAS,DXHAT)
SUBROUTINE XSEN CALCULATES THE PARTIAL DERIVATIVE OF XHAT
WITH RESPECT TO THETA(THE VECTOR OF UNKNOWN).
COMMON P(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1F,K1G,K2G,K1H,K2H,K1Q,
$K1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
$K1RKGM,K2RKGM,RMASS,P11,P22,PKY,RKSI,RIE,ALPHA,RZERO,DT,NRECR,
$NRECD,IROSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SIT,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DEL P,
$K2DEL P,K1DEL K,K2DEL K,K1DF,K2DF,IPIAG
DOUBLE PRECISION PHIK(3,3)
DOUBLE PRECISION PHIKI(3,3)
DOUBLE PRECISION WKAREA(9)
DOUBLE PRECISION DPKH(3,3)
DOUBLE PRECISION DPKHI(3,3)
DIMENSION SPHIKI(3,3)
DIMENSION DPMTX(3,3)
DIMENSION DELTAK(3,3)
DIMENSION XHAT(3,NPOINT)
DIMENSION ZMEAS(3,NPOINT)

```

```

DIMENSION DXHAT (3,NPOINT)
DIMENSION RM (3,3)
DIMENSION PRD1 (3,3)
DIMENSION XINPT (3,3)
DIMENSION SPHIK (3,3)
DIMENSION RKGH (3,3)
DIMENSION PKH (3,3)
DIMENSION PRD2 (3,3)
DIMENSION PRD3 (3,3)
DIMENSION PRD4 (3,3)
DIMENSION PRD5 (3,3)
DIMENSION PRD6 (3,3)
DIMENSION PKHI (3,3)
DIMENSION VXHAT (3,1)
DIMENSION VDXHAT (3,1)
DIMENSION VZ (3,1)
DIMENSION TERM1 (3,1)
DIMENSION TERM2 (3,1)
DIMENSION TERM3 (3,1)
DIMENSION SUM1 (3,1)
DIMENSION VXSEN (3,1)
DIMENSION GAMMA (3,3)
DIMENSION LAMBDA (3,3)
CALL MULT (DELTAK,H,PRD1,K1DELK,K2DELK,K1H,K2H,K1PRD1,K2PRD1)
CALL SUB (DPMTX,FRD1,XINPT,K1F,K1F,K1XINP,K2XINP)
CALL DPTOSP (PHIK,3,3,SPHIK)
FACTOR=0.0
CALL SCALAR (F,FACTOR,RM,K1F,K1F,K1RM,K2RM)
DO 40 I=1,K1F
  RM (I,I)=1.0
40 CONTINUE
IDGT=0
CALL LINV1F (PHIK,K1PHIK,K1PHIK,PHIKI,IDGT,WKAREA,IER)
CALL DPTOSP (PHIKI,K1PHIK,K1PHIK,SPHIKI)
CALL MULT (RKGM,H,RKGH,K1RKGM,K2RKGM,K1E,K2H,K1RKGH,K2RKGH)
CALL SUB (F,RKGH,PKH,K1F,K1F,K1FKH,K2FKH)
CALL SPTODP (PKH,K1FKH,K2FKH,DFKH)
IDGT=0
CALL LINV1F (DFKH,K1FKH,K2FKH,DFKHI,IDGT,WKAREA,IER)
CALL SUB (RM,SPHIKI,PRD2,K1RM,K2RM,K1PRD2,K2PRD2)
CALL MULT (PRD2,XINPT,FRD3,K1PRD2,K2PRD2,K1XINP,K2XINP,K1PRD3,
$K2PRD3)
CALL DPTOSP (DFKHI,K1FKH,K2FKH,PKHI)
CALL MULT (FKHI,PRD3,PRD4,K1FKH,K2FKH,K1PRD3,K2PRD3,K1PRD4,K2PRD4)
CALL MULT (SPHIK,PRD4,GAMMA,K1PHIK,K1PHIK,K1PRD4,K2PRD4,K1GAMA,
$K2GAMA)
CALL MULT (PRD2,DELTAK,PRD5,K1PRD2,K2PRD2,K1DELK,K2DELK,K1PRD5,
$K2PRD5)
CALL MULT (PKHI,PRD5,PRD6,K1PKH,K2PKH,K1PRD5,K2PRD5,K1PRD6,K2PRD6)
CALL MULT (SPHIK,PRD6,LAMBDA,K1PHIK,K1PHIK,K1PRD6,K2PRD6,
$K1LAMB,K2LAMB)
DO 200 J=1,3
  VDXHAT (J,1)=0.0
200 CONTINUE
DO 500 I=1,NPCINT
  DO 220 J=1,3
    VXHAT (J,1)=XHAT (J,I)
    VZ (J,1)=ZMEAS (J,I)
220 CONTINUE
CALL MULT (SFBIK,VDXHAT,TERM1,K1PHIK,K1PHIK,3,1,K1TER1,K2TER1)

```

```

CALL MULT (GAMMA,VXHAT,TERM2,K1GAMA,K2GAMA,3,1,K1TER2,K2TER2)
CALL MULT (LAMBDA,VZ,TERM3,K1LAMB,K2LAMB,3,1,K1TER3,K2TER3)
CALL ADD (TERM1,TERM2,SUM1,K1TER1,K2TER1,K1SUM1,K2SUM1)
CALL ADD (SUM1,TERM3,VXSEN,K1SUM1,K2SUM1,K1VXSN,K2VXSN)
DO 400 J=1,3
  DXHAT (J,I)=VXSEN (J,1)
400 CONTINUE
CALL SCALAR (VXSEN,1.0,VXHAT,K1VXSN,K2VXSN,K1VDHT,K2VDHT)
500 CONTINUE
RETURN
END

```

C
C

C SUBROUTINE GRADE (DXHAT,RNU,DELTAP,DELTAJ)
C SUBROUTINE GRADE COMPUTES THE GRACIENT OF THE LIKELIHOOD
C FUNCTION.

```

COMMON F(3,3),G(3,1),H(3,3),R(3,3),Q(1,1),K1P,K1G,K2G,K1H,K2H,K1Q,
SK1R,RKGM(3,3),K1PHIX,K2PHIX,K1PHIZ,K2PHIZ,
SK1RKM,K2RKM,RMASS,P11,P22,RKY,RKSI,R1E,ALPHA,RZERO,DT,NRECR,
$NRACD,IPOSTA,IROEND,IR1STA,IR1END,IDATSA,IDATEN,NDATA,NRCAL,
$ATTEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SET,P(3,3),B(3,3),K1B,K1P,
$RLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
$K2JELP,K1DELK,K2DELK,K1DP,K2DP,IFLAG

```

```

DIMENSION DXHAT(3,NPOINT)
DIMENSION DELTAP(K1DELP,K2DELP)
DIMENSION RNU(3,NPOINT)
DIMENSION A1(3,1)
DIMENSION A1T(1,3)
DIMENSION BI(3,3)
DIMENSION HT(3,3)
DIMENSION A2(3,3)
DIMENSION A3(3,3)
DIMENSION DELTAE(3,3)
DIMENSION A4(3,3)
DIMENSION A5(3,3)
DIMENSION A6(3,1)
DIMENSION HA6(3,1)
DIMENSION A7(3,1)
DIMENSION A8(3,1)
DIMENSION TERM1(1,1)
DIMENSION TERM2(1,1)
DOUBLE PRECISION WKAREA(9)
DOUBLE PRECISION BDP(3,3)
DOUBLE PRECISION BIDP(3,3)
DIMENSION DELTAV(3,1)

```

01

```

SUM=0.0
IDGT=0
CALL SPTODP(B,K1B,K1B,BDP)
CALL LINV1P(BDP,K1B,K1B,BIDP,IDGT,WKAREA,IER)
CALL DPTOSP(BIDP,K1B,K1B,BI)
CALL TRANS(H,HT,K1H,K2H,K1dT,K2HT)
CALL MULT (DELTAP,HT,A2,K1DELP,K2DELP,K1HT,K2HT,K1A2,K2A2)
CALL MULT (H,A2,DELTAB,K1H,K2H,K1A2,K2A2,K1DELB,K2DELB)
CALL MULT (DELTAB,BI,A3,K1DELB,K2DELB,K1B,K1B,K1A3,K2A3)
CALL MULT (BI,A3,A4,K1B,K1B,K1A3,K2A3,K1A4,K2A4)
CALL MULT (BI,DELTAB,A5,K1B,K1B,K1DELB,K2DELB,K1A5,K2A5)
CALL TRACE(A5,T3,K1A5)
DO 100 I=1,NPOINT
DO 20 J=1,3
  A1(J,1)=RNU(J,I)

```

```

20      A6(J,1)=DXHAT(J,I)
      CONTINUE
      CALL TRANS(A1,A1T,3,1,K1A1T,K2A1T)
      CALL MULT(H,A6,HA6,K1H,K2H,3,1,K1HA6,K2HA6)
      CALL NEG(HA6,DELTAV,K1HA6,K2HA6,K1DELV,K2DELV)
      CALL MULT(BI,DELTAV,A7,K1B,K1B,K1DELV,K2DELV,K1A7,K2A7)
      CALL MULT(A1T,A7,TERM1,K1A1T,K2A1T,K1A7,K2A7,K1TER1,K2TER1)
      CALL MULT(A4,A1,A8,K1A4,K2A4,3,1,K1A8,K2A8)
      CALL MULT(A1T,A8,TERM2,K1A1T,K2A1T,K1A8,K2A8,K1TER2,K2TER2)
      T1=TERM1(1,1)
      T2=TERM2(1,1)
      SUM=SUM+T1+((1.C/2.0)*(T1-T2))
100     CONTINUE
      DELTAJ=SUM
      RETURN
      END

C
C
C      SUBROUTINE FISHER(RNU,DXHAT,DELTAP,DELJ2)
C      SUBROUTINE FISHER COMPUTES THE FISHER INFORMATION MATRIX.
C      THE FISHER INFORMATION MATRIX IS THE SECOND PARTIAL DERIVATIVE
C      OF THE LIKELIHOOD FUNCTION WITH RESPECT TO THETA.
      COMMON P(3,3),G(3,1),H(3,3),R(3,3),C(1,1),K1P,K1G,K2G,K1H,K2H,K1Q,
      K1A,RKGM(3,3),K1PHIX,K2PHIX,K1PHI2,K2PHI2,
      K1LKG,K2RKGM,RMAS,P11,P22,RKY,RKSI,RLE,ALPHA,PZERO,DT,NRECR,
      NRECD,IROSTA,IROEND,IR1STA,IR1END,ICATSA,IDATEN,NDATA,NRCAL,
      SATEN(4,3),RVCAL,TVEL,GRCAL,NPOINT,SIT,P(3,3),B(3,3),K1B,K1P,
      $BLIKE,K1DPH,K2DPH,K1DRKG,K2DRKG,NPT1,XIC(3,1),BZERO,K1DELP,
      $K2DELP,K1DELK,K2DELK,K1DF,K2DF,IFLAG
      DIMENSION RNU(3,NPOINT)
      DIMENSION DXHAT(3,NPOINT)
      DIMENSION DELTAP(K1DELP,K2DELP)
      DIMENSION DELTAE(3,3)
      DIMENSION NEG(3,3)
      DIMENSION HT(3,3)
      DIMENSION DELNU(3,1)
      DIMENSION DELNUT(1,3)
      DIMENSION DXHAT1(3,1)
      DIMENSION RNU1(3,1)
      DIMENSION RN1T(1,3)
      DIMENSION TERM1(1,1)
      DIMENSION TERM2(1,1)
      DIMENSION TERM3(1,1)
      DIMENSION A1(3,3)
      DIMENSION PRD11(3,1)
      DIMENSION PRD21(3,1)
      DIMENSION PRD22(3,1)
      DIMENSION PRD23(3,1)
      DIMENSION PRD31(3,1)
      DIMENSION PRD32(3,1)
      DIMENSION PRD33(3,1)
      DIMENSION PRD41(3,3)
      DIMENSION PRD42(3,3)
      DIMENSION PRD43(3,3)
      DIMENSION BI(3,3)
      DOUBLE PRECISION WKAREA(9)
      DOUBLE PRECISION BDP(3,3)
      DOUBLE PRECISION BIDP(3,3)
01      SUM=0.0
      IDJT=0

```

```

CALL SPTODP (E, K1B, K1B, BDP)
CALL LINVIF (BCF, K1B, K1B, BIDP, IDGT, WKAP1A, IER)
CALL DPTOSP (BILF, K1B, K1B, BI)
CALL TRANS (H, HT, K1H, K2H, K1HT, K2HT)
CALL NEG (H, HNEG, K1H, K2H, K1HNEG, K2HNEG)
CALL MULT (DELTA E, HT, A1, K1DELP, K2DELP, K1HT, K2HT, K1A1, K2A1)
CALL MULT (H, A1, DELTAB, K1H, K2H, K1A1, K2A1, K1DELB, K2DELB)
DO 100 I=1, NPCINT
DO 20 J=1, 3
DXHAT1 (J, 1) = DXHAT (J, I)
RNU1 (J, 1) = RNU (J, I)
20 CONTINUE
C COMPUTE THE 1ST TERM OF THE INFORMATION MATRIX
CALL MULT (HNEG, DXHAT1, DELNU, K1HNEG, K2HNEG, 3, 1, K1DNU, K2DNU)
CALL MULT (BI, DELNU, PRD11, K1B, K1E, 3, 1, K1PD11, K2PD11)
CALL TRANS (DELNU, DELNUT, K1DNU, K2DNU, K1DNUT, K2DNUT)
CALL MULT (DELNUT, PRD11, TERM1, K1DNUT, K2DNUT, K1PD11, K2PD11, K1TER1,
$K2TER1)
C COMPUTE THE 2ND TERM OF THE INFORMATION MATRIX
CALL MULT (BI, DELNU, PRD21, K1B, K1B, K1DNU, K2DNU, K1PD21, K2PD21)
CALL MULT (DELTA E, PRD21, PRD22, K1DELB, K2DELB, K1PD21, K2PD21, K1PD22,
$K2PD22)
CALL MULT (BI, PRD22, PRD23, K1B, K1E, K1PD22, K2PD22, K1PD23, K2PD23)
CALL TRANS (RNU1, RN1T, 3, 1, K1RN1T, K2RN1T)
CALL MULT (RN1T, PRD23, TERM2, K1RN1T, K2RN1T, K1PD23, K2PD23, K1TER2,
$K2TER2)
C COMPUTE THE 3RD TERM OF THE INFORMATION MATRIX
CALL MULT (BI, DELNU, PRD31, K1B, K1B, K1DNU, K2DNU, K1PD31, K2PD31)
CALL MULT (DELTA E, PRD31, PRD32, K1DELB, K2DELB, K1PD31, K2PD31, K1PD32,
$K2PD32)
CALL MULT (BI, PRD32, PRD33, K1E, K1E, K1PD32, K2PD32, K1PD33, K2PD33)
CALL MULT (RN1T, PRD33, TERM3, K1RN1T, K2RN1T, K1PD33, K2PD33, K1TER3,
$K2TER3)
C COMPUTE THE 4TH TERM OF THE INFORMATION MATRIX
CALL MULT (BI, DELTAB, PRD41, K1E, K1B, K1DELB, K2DELB, K1PD41, K2PD41)
CALL MULT (DELTAB, PRD41, PRD42, K1DELB, K2DELB, K1PD41, K2PD41, K1PD42,
$K2PD42)
CALL MULT (BI, PRD42, PRD43, K1B, K1B, K1PD42, K2PD42, K1PD43, K2PD43)
CALL TRACE (PRD43, T4, K1PD43)
T1 = TERM1 (1, 1)
T2 = TERM2 (1, 1)
T3 = TERM3 (1, 1)
PSUM = T1 - T2 - T3 - ((1.0/2.0) * T4)
SUM = SUM + PSUM
100 CONTINUE
DELJ2 = SUM
RETURN
END

C
C
C SUBROUTINE STEP (THETA, DELTAJ, DELJ2, LTHETA, THETAN)
C SUBROUTINE STEP CALCULATES THE NEW VALUE OF THETA (THETAN).
DTHETA = (-1.0/DELJ2) * DELTAJ
THETAN = DTHETA + THETA
RETURN
END

C
C SUBROUTINE MULT (A, B, D, K1A, K2A, K1B, K2B, F1D, K2D)
C DIMENSION A (K1A, K2A), B (K1B, K2B), D (K1A, K2B)

```

```

      IF(K2A .NE. K1B)GC TO 40
      DO 20 I=1,K1A
      DO 20 J=1,K2B
      L=J
      DO 20 K=1,K2A
      L=L+1
      IF(L .NE. 1)GO TO 20
      D(I,J)=0
20    D(I,J)=D(I,J) + (A(I,K)*B(K,J))
      K1D=K1A
      K2D=K2B
      GO TO 80
40    WRITE(6,60)
60    FORMAT(1H0,'MATRICES A AND B ARE NOT CCNFORMABLE')
80    RETURN
      END

```

C
C

```

      SUBROUTINE SCALAR(A,B,C,K1A,K2A,K1C,K2C)
      DIMENSION A(K1A,K2A),C(K1A,K2A)
      DO 10 I=1,K1A
      DO 10 J=1,K2A
10    C(I,J)=B*A(I,J)
      K1C=K1A
      K2C=K2A
      RETURN
      END

```

C
C

```

      SUBROUTINE TRANS(A,B,K1A,K2A,K1B,K2B)
      DIMENSION A(K1A,K2A),B(K2A,K1A)
      DO 20 I=1,K1A
      DO 20 J=1,K2A
20    B(J,I)=A(I,J)
      K1B=K2A
      K2B=K1A
      RETURN
      END

```

C
C

```

      SUBROUTINE NEG(A,B,K1A,K2A,K1B,K2B)
      DIMENSION A(K1A,K2A),B(K1A,K2A)
      DO 10 I=1,K1A
      DO 10 J=1,K2A
10    B(I,J)=0-A(I,J)
      K1B=K1A
      K2B=K2A
      RETURN
      END

```

C
C

```

      SUBROUTINE ADD(A,B,C,K1A,K2A,K1C,K2C)
      DIMENSION A(K1A,K2A),B(K1A,K2A),C(K1A,K2A)
      DO 20 I=1,K1A
      DO 20 J=1,K2A
20    C(I,J)=A(I,J)+B(I,J)
      K1C=K1A
      K2C=K2A
      RETURN
      END

```

C
C

```

SUBROUTINE SUB (A,B,C,K1A,K2A,K1C,K2C)
DIMENSION A(K1A,K2A),B(K1A,K2A),C(K1A,K2A)
DO 20 I=1,K1A
DO 20 J=1,K2A
20 C(I,J)=A(I,J)-E(I,J)
K1C=K1A
K2C=K2A
RETURN
END

```

C
C

```

FUNCTION FACT(K)
FACT=1.0
DO 20 I=1,K
RI=I
20 FACT=FACT*RI
RETURN
END

```

C
C

```

SUBROUTINE TRACE(A,B,K1A)
DIMENSION A(K1A,K1A)
SUM=0.0
DO 20 I=1,K1A
SUM=SUM+A(I,I)
20 CONTINUE
B=SUM
RETURN
END

```

C
C

```

SUBROUTINE OUTPUT(A,K1A,K2A)
DIMENSION A(K1A,K2A)
DO 20 I=1,K1A
WRITE(6,30) (A(I,J),J=1,K2A)
20 CONTINUE
30 FORMAT(' ',12X,6(2X,E15.8))
RETURN
END

```

C
C

```

SUBROUTINE OVER(RMATRX,CHK,K1,K2,IPLAGO)
DIMENSION RMATRX(K1,K2)
DO 20 I=1,K1
DO 20 J=1,K2
IF(RMATRX(I,J).GE. CHK) IPLAGO=1
20 CONTINUE
RETURN
END

```

C

```

SUBROUTINE SFTCEP(X,K1X,K2X,Y)
DIMENSION X(K1X,K2X)
DOUBLE PRECISION Y(K1X,K2X)
DO 20 I=1,K1X
DO 20 J=1,K2X
Y(I,J)=DBLE(X(I,J))
20 CONTINUE
RETURN

```

END

C
C

```

SUBROUTINE DPTOSP (X,K1X,K2X,Y)
DOUBLE PRECISION X (K1X,K2X)
DIMENSION Y (K1X,K2X)
DO 20 I=1,K1X
DO 20 J=1,K2X
Y(I,J)=X(I,J)
20 CONTINUE
RETURN
END

```

C
C

```

SUBROUTINE SCALDP (A,B,C,K1A,K2A,K1C,K2C)
DOUBLE PRECISION A (K1A,K2A) ,C (K1A,K2A)
DOUBLE PRECISION B
DO 10 I=1,K1A
DO 10 J=1,K2A
10 C(I,J)=B*A(I,J)
K1C=K1A
K2C=K2A
RETURN
END

```

C
C
C

```

SUBROUTINE TRANSD (A,B,K1A,K2A,K1B,K2B)
DOUBLE PRECISION A (K1A,K2A) ,B (K2A,K1A)
DO 20 I=1,K1A
DO 20 J=1,K2A
20 B(J,I)=A(I,J)
K1B=K2A
K2B=K1A
RETURN
END

```

C
C

```

SUBROUTINE NEGDP (A,B,K1A,K2A,K1B,K2B)
DOUBLE PRECISION A (K1A,K2A) ,B (K1A,K2A)
DO 10 I=1,K1A
DO 10 J=1,K2A
10 B(I,J)=DBLE (0.0) -A (I,J)
K1B=K1A
K2B=K2A
RETURN
END

```

C
C

```

SUBROUTINE ADDEP (A,B,C,K1A,K2A,K1C,K2C)
DOUBLE PRECISION A (K1A,K2A) ,B (K1A,K2A) ,C (K1A,K2A)
DO 20 I=1,K1A
DO 20 J=1,K2A
20 C(I,J)=A(I,J) +B(I,J)
K1C=K1A
K2C=K2A
RETURN
END

```

C
C

```

SUBROUTINE MULTDP(A,B,D,K1A,K2A,K1B,K2B,K1D,K2D)
DOUBLE PRECISION A(K1A,K2A),B(K1B,K2B),D(K1A,K2B)
IF(K2A.NE.K1B)GO TO 40
DO 20 I=1,K1A
DO 20 J=1,K2B
L=0
DO 20 K=1,K2A
L=L+1
IF(L.NE.1)GO TO 20
D(I,J)=DBLE(0.0)
20 D(I,J)=D(I,J)+(A(I,K)*B(K,J))
K1D=K1A
K2D=K2B
GO TO 80
40 WRITE(6,60)
60 FORMAT(1H0,'MATRICES A AND B ARE NOT CONFORMABLE')
80 RETURN
END

```

C
C

```

SUBROUTINE SUBDP(A,B,C,K1A,K2A,K1C,K2C)
DOUBLE PRECISION A(K1A,K2A),B(K1A,K2A),C(K1A,K2A)
DO 20 I=1,K1A
DO 20 J=1,K2A
20 C(I,J)=A(I,J)-B(I,J)
K1C=K1A
K2C=K2A
RETURN
END

```

C
C

```

SUBROUTINE TRACED(A,B,K1A)
DOUBLE PRECISION A(K1A,K1A)
DOJBLE PRECISION SUM,B
SUM=DBLE(0.0)
DO 20 I=1,K1A
SUM=SUM+A(I,I)
20 CONTINUE
B=SUM
RETURN
END

```

C
C
C
C

```

SUBROUTINE OUTDBL(A,K1A,K2A)
DOUBLE PRECISION A(K1A,K2A)
DO 20 I=1,K1A
DO 20 J=1,K2A
WRITE(6,30) A(I,J)
20 CONTINUE
30 FORMAT(' ',12X,D15.8)
RETURN
END

```

C
C

```

SUBROUTINE OVERDP(RMATRIX,CHK,K1,K2,IFLAGO)
DOUBLE PRECISION RMATRIX(K1,K2)
DOJBLE PRECISION CHK
DO 20 I=1,K1

```

```

DO 20 J=1,K2
IF(RMATRIX(I,J) .GE. CHK) IFLAGO=1
20 CONTINUE
RETURN
END

```

C
C

```

DOUBLE PRECISION FUNCTION FACTDP(K)
DOUBLE PRECISION DREALI
FACTDP=DBLE(1.0)
DO 20 I=1,K
REALI=I
DREALI=DBLE(REALI)
20 FACTDP=FACTDP*DREALI
RETURN
END

```

C
C

```

SUBROUTINE OUTDP(A,K1A,K2A)
DOUBLE PRECISION A(K1A,K2A)
DO 20 I=1,K1A
WRITE(6,30) (A(I,J),J=1,K2A)
20 CONTINUE
30 FORMAT(' ',12X,6(2X,D15.9))
RETURN
END

```

C
C

```

SUBROUTINE PLOT(X,Y,IPOINT)
DIMENSION X(IPOINT),Y(IPOINT)
WRITE(6,20)
20 FORMAT(/,' ',40X,'STATE ESTIMATES')
CALL WPLOT1(X,Y,IPOINT,4,XHAT)
RETURN
END

```

C
C

```

SUBROUTINE VMNEX(X,KX,RMIN,RMAX)
DIMENSION X(KX)
RMIN=0.0
RMAX=0.0
DO 100 I=1,KX
IF(X(I) .GT. RMAX) RMAX=X(I)
IF(X(I) .LT. RMIN) RMIN=X(I)
100 CONTINUE
RETURN
END

```

C
C

```

SUBROUTINE PLXHAT(Y1,Y2,K1Y1,K2Y1,K1Y2,K2Y2,XVECT1,YVECT1,
SYVECT2,DT,ITIT1,JLAB1)
DIMENSION ITIT1(10)
DIMENSION JLAB1(10)
DIMENSION Y1(K1Y1,K2Y1)
DIMENSION Y2(K1Y2,K2Y2)
DIMENSION XVECT1(K2Y2)
DIMENSION YVECT1(K2Y2)
DIMENSION YVECT2(K2Y2)
DATA ICHARA,ICHARE/'A','E'/
DO 20 I=1,K2Y2

```

```

      XVECT1(I)=DT*I
20    CONTINUE
      DO 200 J=1,K1Y2
      DO 50 I=1,K2Y2
      YVECT1(I)=Y1(J,I)
      YVECT2(I)=Y2(J,I)
50    CONTINUE
      CALL VMNMX(YVECT1,K2Y2,Y1MIN,Y1MAX)
      CALL VMNMX(YVECT2,K2Y2,Y2MIN,Y2MAX)
      IF(Y1MIN.LT.Y2MIN)YMIN=Y1MIN
      IF(Y2MIN.LT.Y1MIN)YMIN=Y2MIN
      IF(Y2MIN.EQ.Y1MIN)YMIN=Y2MIN
      IF(Y1MAX.GT.Y2MAX)YMAX=Y1MAX
      IF(Y2MAX.GT.Y1MAX)YMAX=Y2MAX
      IF(Y2MAX.EQ.Y1MAX)YMAX=Y2MAX
      CALL WPLOT2(XVECT1(K2Y2),XVECT1(1),YMAX,YMIN)
      CALL WPLOT3(ICHARA,XVECT1,YVECT1,K2Y2)
      CALL WPLOT3(ICHARE,XVECT1,YVECT2,K2Y2)
      WRITE(6,90)ITIT1
90    FORMAT('1',40X,10A4)
      CALL WPLOT4(40,JLAB1)
200   CONTINUE
      RETURN
      END

```

C
C

```

SUBROUTINE PLXSEN(DXHAT,K1X,K2X,XVECT1,YVECT1,DT,ITIT2,JLAB2)
DIMENSION DXHAT(K1X,K2X)
DIMENSION XVECT1(K2X)
DIMENSION YVECT1(K2X)
DIMENSION ITIT2(10)
DIMENSION JLAB2(10)
DO 20 I=1,K2X
XVECT1(I)=DT*I
20    CONTINUE
      DO 100 J=1,K1X
      DO 50 I=1,K2X
      YVECT1(I)=DXHAT(J,I)
50    CONTINUE
      WRITE(6,60)ITIT2
60    FORMAT('1',40X,10A4)
      CALL WPLOT1(XVECT1,YVECT1,K2X,40,JLAB2)
100   CONTINUE
      RETURN
      END

```

C
C

```

SUBROUTINE CPLCT(X,K1X,K2X,XVECT1,YVECT1,YMIN,YMAX,DT,JLABEL,
$ITITLE)
DIMENSION X(K1X,K2X)
DIMENSION YVECT1(K2X)
DIMENSION XVECT1(K2X)
DIMENSION JLABEL(10)
DIMENSION ITITLE(10)
DATA ICHAR/'*'/
DO 20 I=1,K2X
XVECT1(I)=DT*I
20    CONTINUE
      DO 200 J=1,K1X
      DO 40 I=1,K2X

```

```
      YVECT1(I)=X(J,I)
40    CONTINUE
      WRITE(6,100) ITITLE
100   FORMAT('1',40X,10A4)
      CALL WPLLOT2(XVECT1(K2X),XVECT1(1),YMAX,YMIN)
      CALL WPLLOT3(ICHAR,XVECT1,YVECT1,K2X)
      CALL WPLLOT4(40,JLABEL)
200   CONTINUE
      RETURN
      END
```

C

Appendix F

LIST OF APL FUNCTIONS USED TO GENERATE WHEELSET SIMULATED DATA

```

▽ WHLSIM
[1] A←SYSMTX
[2] B←INFMTX
[3] C← 3 3 P0
[4] D← 3 1 P0
[5] H← 3 3 P 1 0 0 0 1 0 0 0 1
[6] 'RANDOM INPUT STATISTICS'
[7] Q←'ENTER SUM LENGTH MEAN MNSQVALUE'
[8] STATRI←Q
[9] W←WHTNOISE STATRI
[10] 'DETERMINISTIC INPUT STATISTICS'
[11] Q←'ENTER SUM LENGTH MEAN MNSQVALUE'
[12] STATDI←Q
[13] WT←WHTNOISE STATDI
[14] U←(STATRI[2],1,1)PW+WT
[15] DYN
[16] XMEAS←X
[17] 'MEASUREMENT NOISE STATISTICS'
[18] Q←'ENTER SUM LENGTH MEAN MNSQVALUE'
[19] STATMN←Q
[20] V1←WHTNOISE STATMN
[21] V2←WHTNOISE STATMN
[22] V3←WHTNOISE STATMN
[23] V←(3,STATMN[2])P V1,V2,V3
[24] Z←(H+,X XMEAS)+V

```

```

▽ F←SYSMTX
[1] F11←-1X(((2XTHETAXCOEF22)+(B0XVEL))÷MASSXVEL)
[2] F12←2XCOEF22XTHETA÷MASS
[3] F13←-1XKT÷MASS
[4] F21←0
[5] F22←-1XKSI XVEL÷2XTHETAXCOEF11XLXL
[6] F23←-1XALPHAXVEL÷LXR0
[7] F31←1
[8] F32←0
[9] F33←0
[10] F← 3 3 P F11,F12,F13,F21,F22,F23,F31,F32,F33

```

```

▽ G←INFMTX
[1] G11←B0÷MASS
[2] G21←0
[3] G31←-1
[4] G← 3 1 P G11,G21,G31

```

```

▽ Y←WHTNOISE X;A;B;MN;MS
[1] A←?(X[1]X[2])f100000
[2] A←(X[1],X[2])fA
[3] B←(1,X[2])f+A
[4] MN←(+/B)÷(X/fB)
[5] B←(-1X(MN-X[3]))+B
[6] MS←MNSQVAL B
[7] B←((X[4]÷MS)×0.5)XB
[8] Y←B
▽
.
```

```

▽ Y←MNSQVAL X
[1] Y←(+/(X+,XQX))÷(X/(fX))
▽
.
```

```

▽ Y←COV X;MN
[1] MN←(ΦfX)f(+/X)÷-1↑fX
[2] MN←QMN
[3] Y←((X-MN)+,XQ(X-MN))÷-1↑fX
▽
.
```

```

▽ DYN
[1] T←NfDTx\N
[2] INITIALIZE
[3] PHI←DT STM A
[4] DMATRIX
[5] M←0
[6] L0;M←M+1
[7] XX[;;(M+1)]←(PHI+,XXX[;;M])+DD+,XUU[;;M]
[8] YY[;;(M+1)]←(C+,XXX[;;(M+1)])+D+,XUU[;;(M+1)]
[9] +(M(N-1)/L0
[10] X←(NN,N)f,XX
[11] Y←(LL,N)f,YY
▽
.
```

```

▽ INITIALIZE
[1] NN←(f,A)×0.5
[2] MM←[fB[1;]]
[3] LL←[fC[1;]]
[4] XX←(NN,1,N)f0
[5] YY←(LL,1,N)f0
[6] UU←(MM,1,N)f,U
[7] 'ENTER X(0)'
[8] XX[1;1]←0
[9] YY[1;1]←C+.XXX[1;1]
▽
.
```

```

▽ PHI←DT STM A;R;N;FACTOR;INDEX
[1] R←(f,A)×0.5
[2] I←(1/R)×.=1/R
[3] N←0
[4] PHI←I
[5] FACTOR←I
[6] L0:N←N+1
[7] FACTOR←FACTOR+.XAXDT÷N
[8] PHI←PHI+FACTOR
[9] INDEX←(+/,FACTOR)÷R×2
[10] +(INDEX>EPS)/L0
▽
.
```

```

▽ DMATRIX;H
[1] H←DT÷4
[2] DD←(H÷3)×((0 STM A)+(4xH STM A)+(2x(2xH) STM A)+(4x(3xH) STM A)+(DT STM A))+.XB
▽
.
```

Appendix G

LIST OF THE PROGRAM USED TO OPERATE THE A/D CONVERTER

```

0001 FTM4.L
0002 PROGRAM DMAD3(3,40)
0003 C PROGRAM TO DIGITIZE WATER CHANNEL HOT WIRE DATA.
0004 C THE PROGRAM READS NCH CHANNELS, NPPC SAMPLES PER CHANNEL FROM THE
0005 C GMAD/1 A/D CONVERTER. THE DATA READ IS WRITTEN ON TAPE.
0006 C VERSION OF 18 JULY 1978
0007 DIMENSION IBUFA(512),IBUFB(512)
0008 INTEGER PARMS(5)
0009 EQUIVALENCE (PARMS(1),LUCRT),(PARMS(2),IPRINT),(PARMS(3),MAG),
0010 1 (PARMS(4),LU)
0011 C GET LOGICAL UNITS
0012 CALL RPAR(PARMS)
0013 C DEFAULT THEM IF NECESSARY
0014 IF(LUCRT.LE.0) LUCRT=1
0015 IF(MAG.LE.0) MAG=8
0016 IF(LU.LE.0) LU=17
0017 N=0
0018 NCH=2
0019 NPPC=256
0020 2222 FORMAT('ENTER THE NUMBER RECORDS PER FILE <')
0021 ICLAS=0
0022 C AT PRESENT S.A.M. (2048 WORDS), IBUFL MUST EQUAL 512 OR LESS
0023 IBUFL=NCH*NPPC
0024 C GET A CLASS NUMBER
0025 CALL EXEC(18,0,IBUFB,IBUFL,0,0,ICLAS)
0026 C TELL USER WHAT CLASS NUMBER WE GOT
0027 NCH3=IAND(ICLAS,17778)
0028 WRITE(LUCRT,1203) NCH3
0029 1203 FORMAT('CLASS NUMBER ',06,'B')
0030 C SET BIT 15 IN CLASS WORD FOR NO WAIT AND BIT 13 TO KEEP CLASS NR.
0031 ICLAS=IOR(ICLAS,120000B)
0032 WRITE(LUCRT,1201)
0033 1201 FORMAT('ENTER DESIRED SAMPLING RATE IN HZ. <')
0034 READ(LUCRT,*) HERTZ
0035 C COMPUTE CLOCK DIVISOR
0036 ICNV=FIX(1.0E+07/HERTZ)
0037 HERTZ=1.0E+07/FLOAT(ICNV)
0038 WRITE(LUCRT,1202) HERTZ
0039 1202 FORMAT('ACTUAL SAMPLING RATE IS ',G10.4,' HERTZ')
0040 C CONSTRUCT CONTROL WORD FOR SETTING CLOCK DIVIDER
0041 ICNK=LU+3000B
0042 C SET CLOCK DIVIDER
0043 CALL EXEC(3,ICLK,ICNV)
0044 15 CONTINUE
0045 WRITE(LUCRT,2222)
0046 READ(LUCRT,*) MAXREC
0047 IF(MAXREC.LE.0) GO TO 30
0048 C MAXREC IS DIVIDED BY 2 BECAUSE TWO RECORDS

```

```

0049 C ARE WRITTEN EACH TIME THROUGH THE LOOP
0050 MAXREC=MAXREC/2
0051 DO 25 J=1,MAXREC
0052 C READ FROM A/D INTO BUFA AND SUSPEND PROGRAM
0053 CALL EXEC(1,LU,IBUFA,IBUFL)
0054 C DO CLASS GET. INITIALLY IT WILL (I HOPE) CLEAR THE CLASS REQUEST
0055 C THAT GOT US THE CLASS NUMBER. AFTERWARDS IT SHOULD CLEAR ONE OF
0056 C THE CLASS WRITES TO TAPE.
0057 CALL EXEC(21,ICLAS,IBUFB,IBUFL)
0058 C BEGIN WRITING BUFA TO MAG TAPE, THEN
0059 C SIMULTANEOUSLY READ FROM A/D TO BUFB
0060 CALL EXEC(18,MAG,IBUFA,IBUFL,0,0,ICLAS)
0061 CALL ABREG(I RETN,IB)
0062 IF(I RETN .NE. 0) WRITE(LUCRT,1204) I RETN
0063 1204 FORMAT("CLASS WRITE ERROR ",I2)
0064 CALL EXEC(1,LU,IBUFB,IBUFL)
0065 C CHECK IBUFA WRITE FOR COMPLETION
0066 CALL EXEC(21,ICLAS,IBUFA,IBUFL)
0067 C START WRITING IBUFA TO TAPE
0068 CALL EXEC(18,MAG,IBUFB,IBUFL,0,0,ICLAS)
0069 CALL ABREG(I RETN,IB)
0070 IF(I RETN .NE. 0) WRITE(LUCRT,1204) I RETN
0071 25 CONTINUE
0072 ENDFILE MAG
0073 N=N+1
0074 WRITE(LUCRT,1003)N
0075 1003 FORMAT("FILE NUMBER ",I4," HAS BEEN READ")
0076 GO TO 15
0077 30 CONTINUE
0078 C CLEAR BIT 13 IN CLASS WORD, WE ARE DONE WITH THE CLASS NUMBER.
0079 ICLAS=IAND(ICLAS,1577778)
0080 C NOW DO GET TO CLEAR FINAL REQUEST AND RELEASE CLASS NUMBER.
0081 40 CALL EXEC(21,ICLAS,IBUFB,IBUFL)
0082 CALL ABREG(I RETN,IB)
0083 WRITE(LUCRT,1205) I RETN
0084 1205 FORMAT("FINAL GET RETURNED ",06)
0085 IF(I RETN .LT. 0) GO TO 40
0086 ENDFILE MAG
0087 STOP
0088 END

```

FTN4 COMPILER: HP92060-16092 REV. 1805 (780310)

** NO UNFINISHES ** NO ERRORS ** PROGRAM = 01450 COMMON = 00000

Appendix H

PLOTS OF THE LIKELIHOOD FUNCTION, OBSERVATION TERM AND
BIAS TERM FOR TEST CASES 1 THROUGH 9

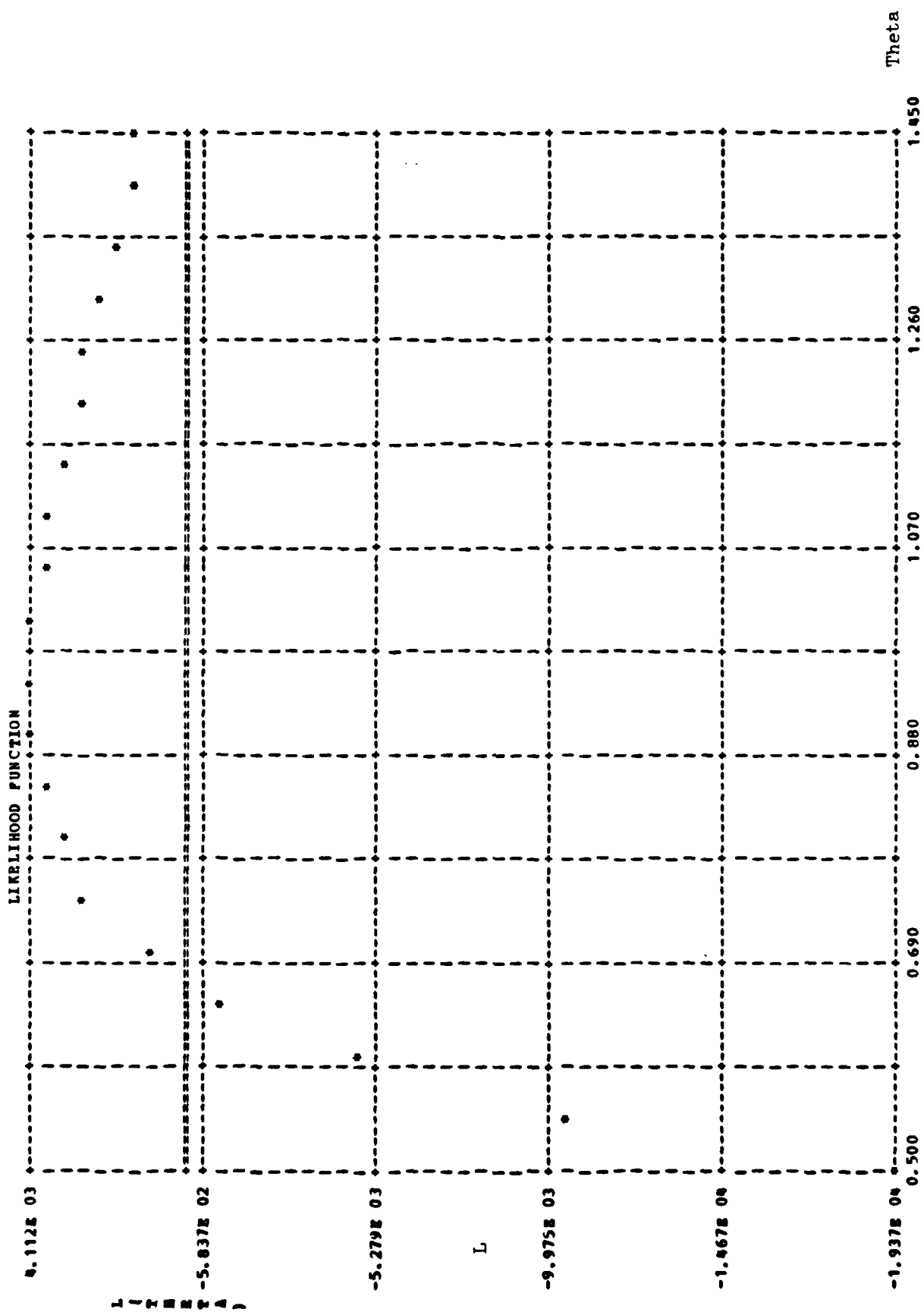


Figure H.1 Plot of the likelihood function for test case 1.

Figure H.2 Plot of the χ^2 deviation norm for test case 1.

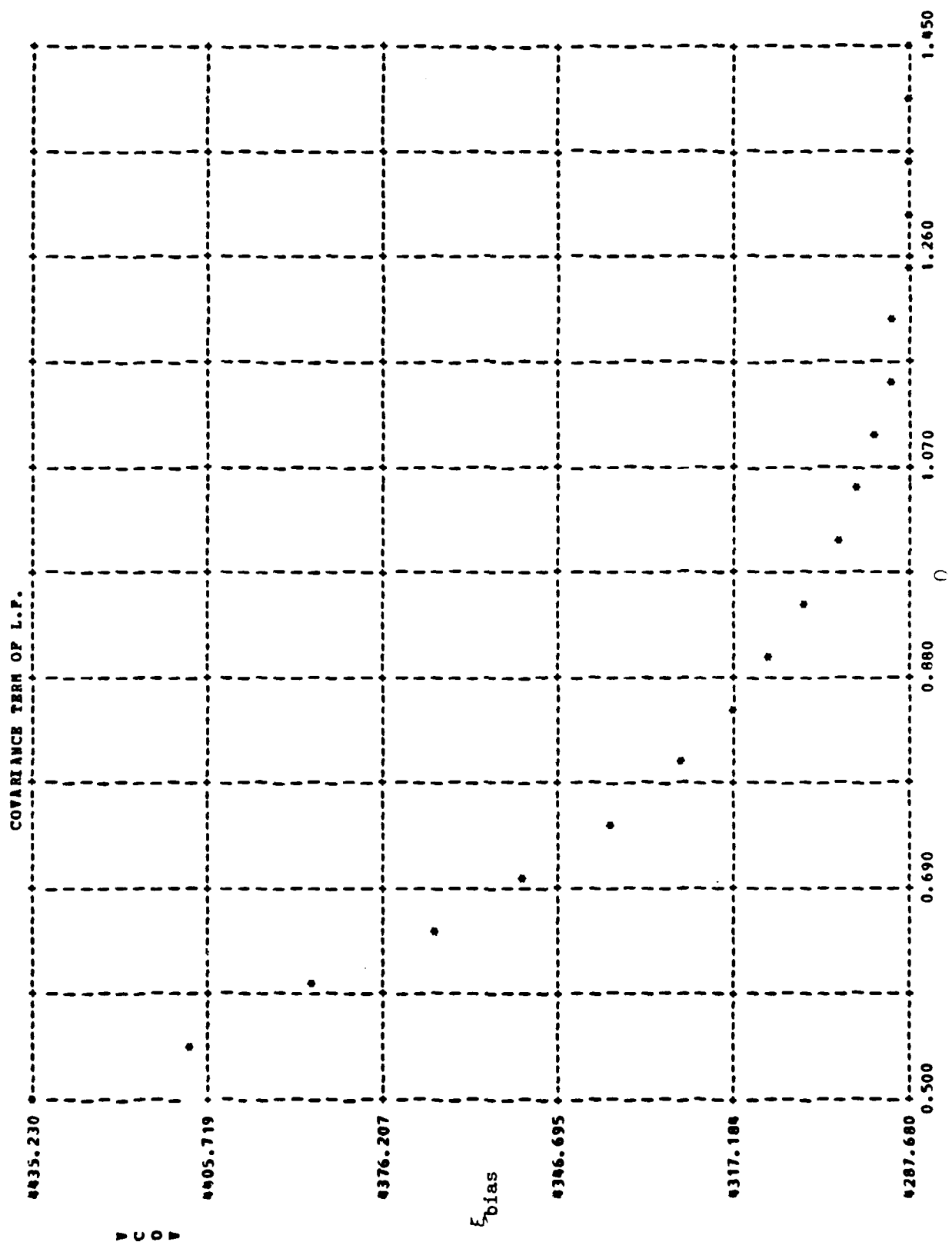


Figure H.3 Plot of the bias term for test case 1.

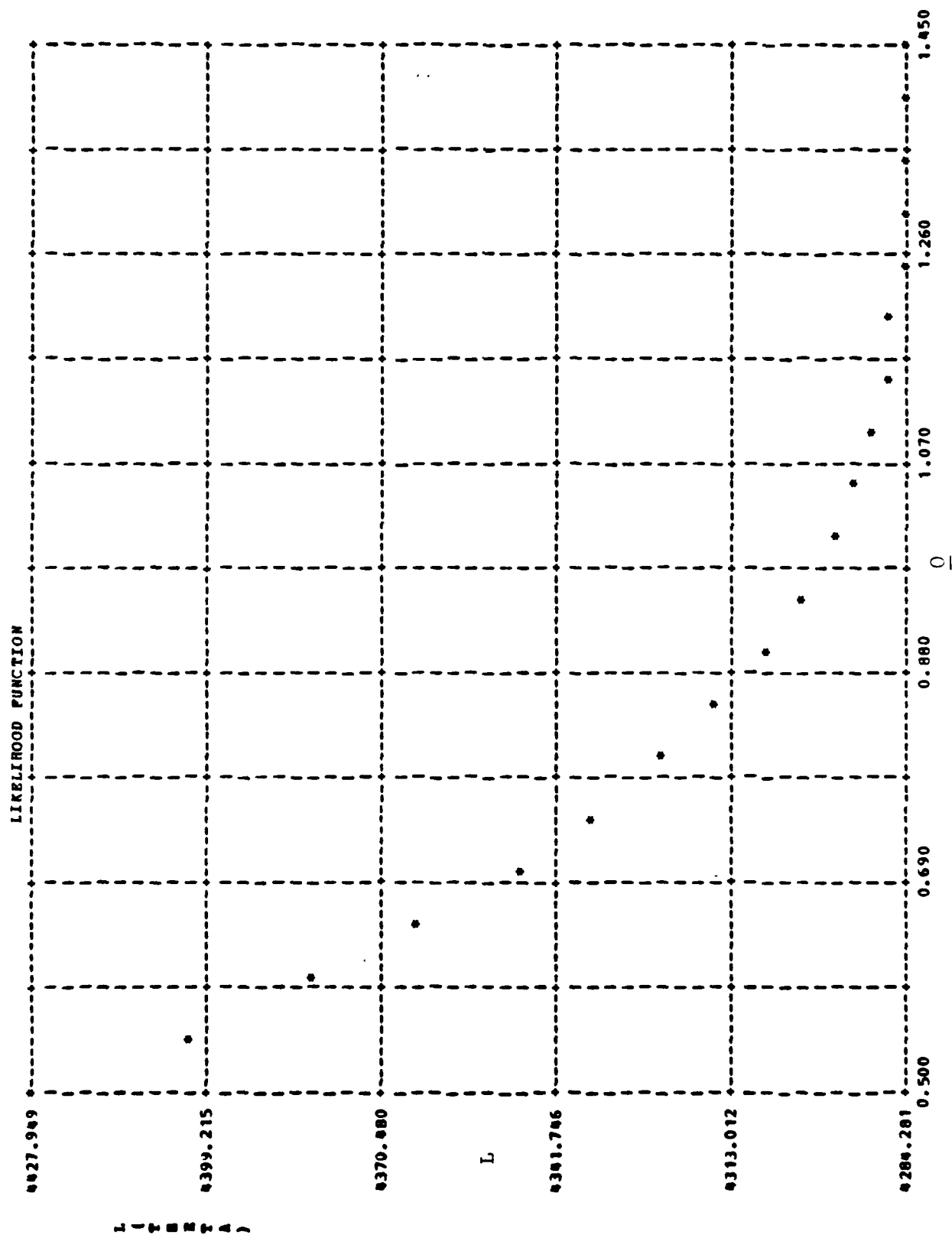


Figure H.4 Plot of the likelihood function for test case 2.

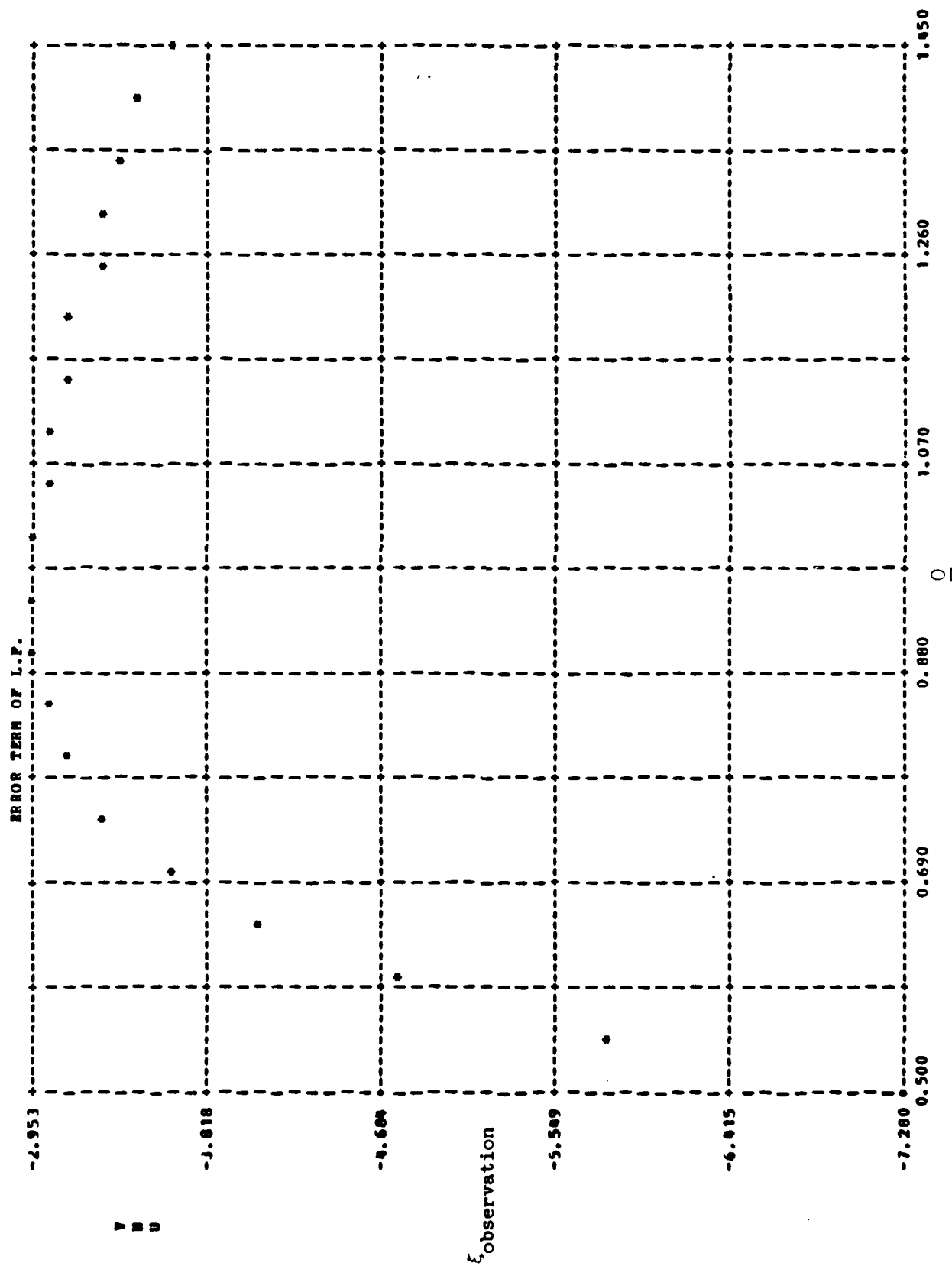


Figure H.5 Plot of the observation term for test case 2.

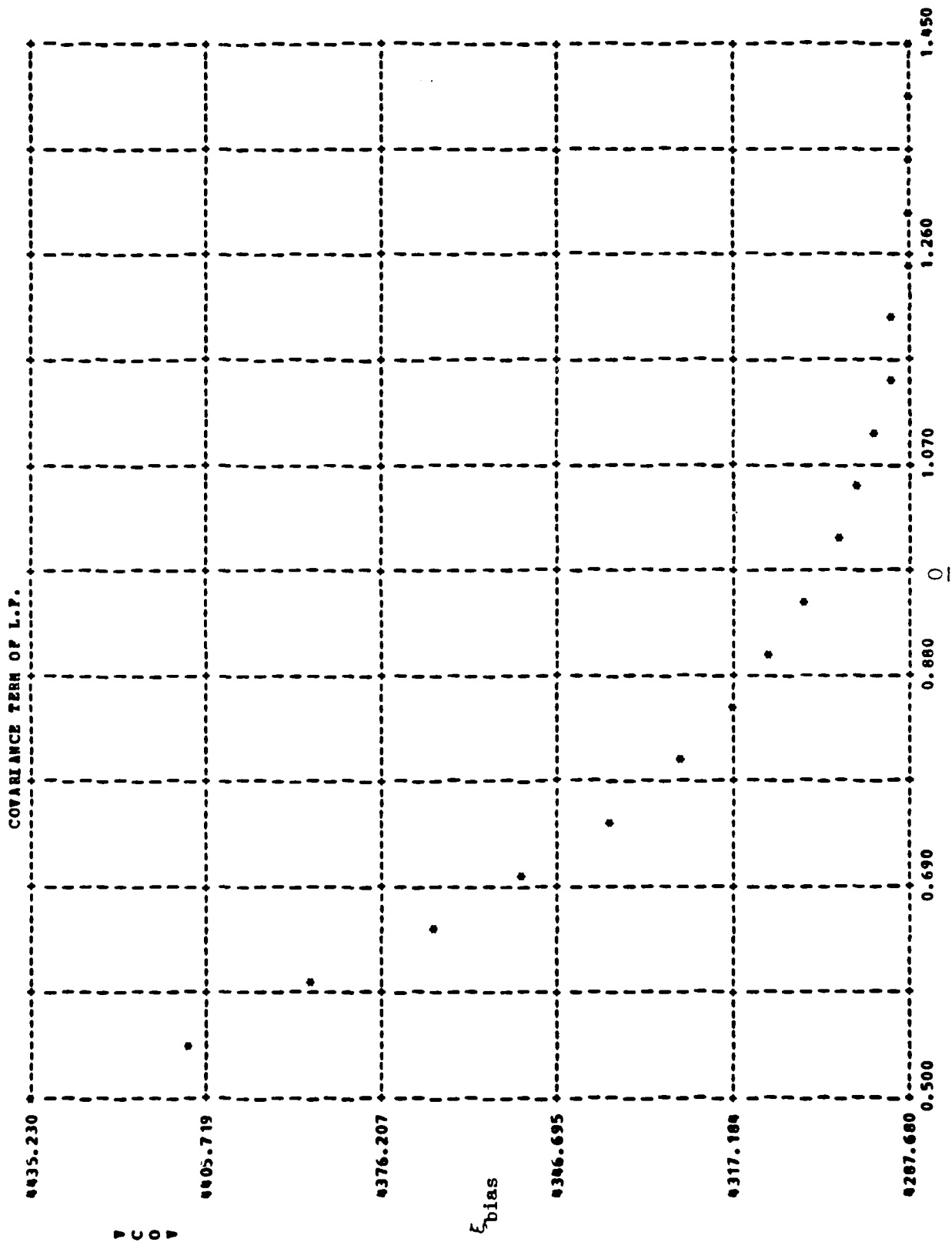


Figure H.6 Plot of the bias term for test case 2.

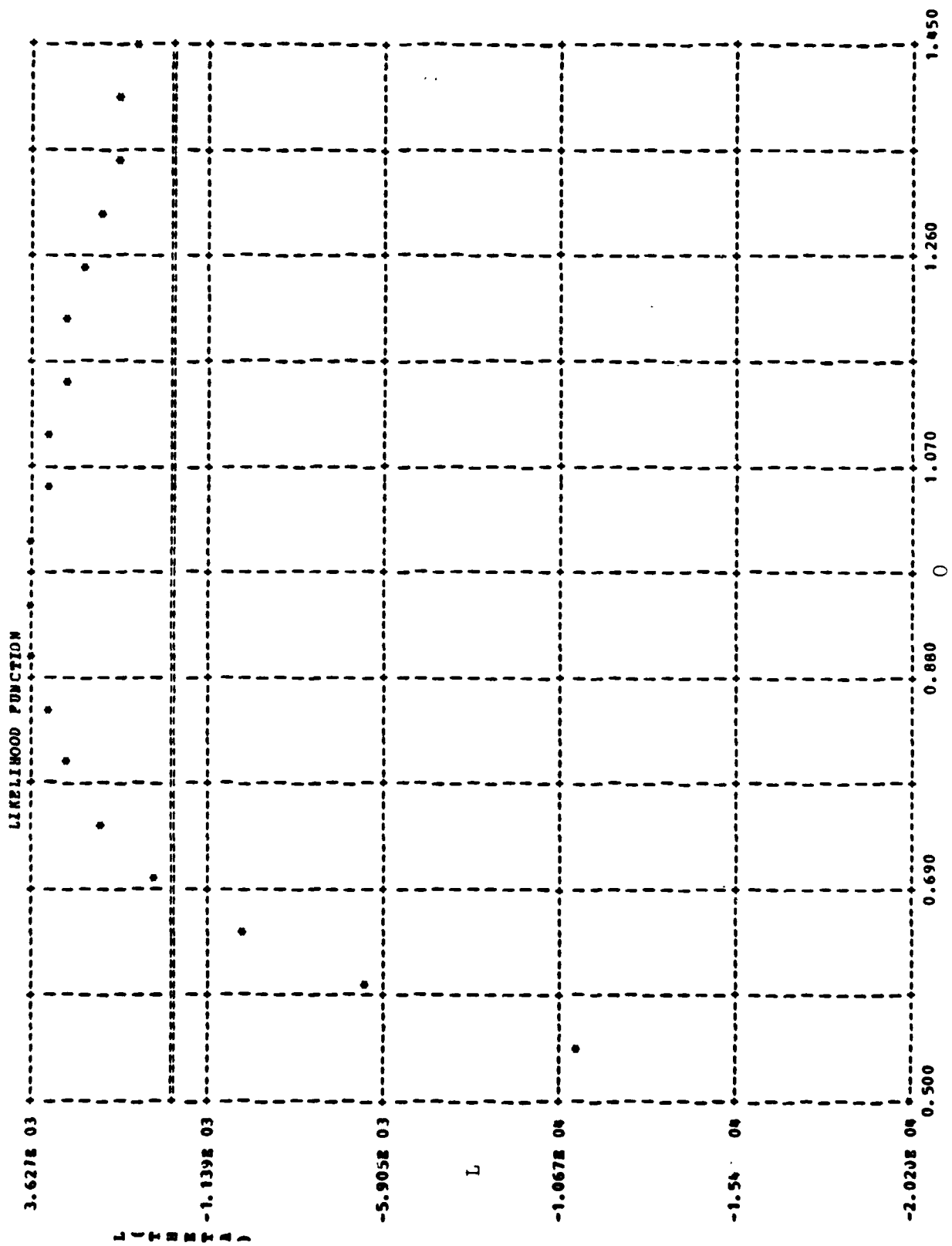


Figure H.7 Plot of the likelihood function for test case 3.

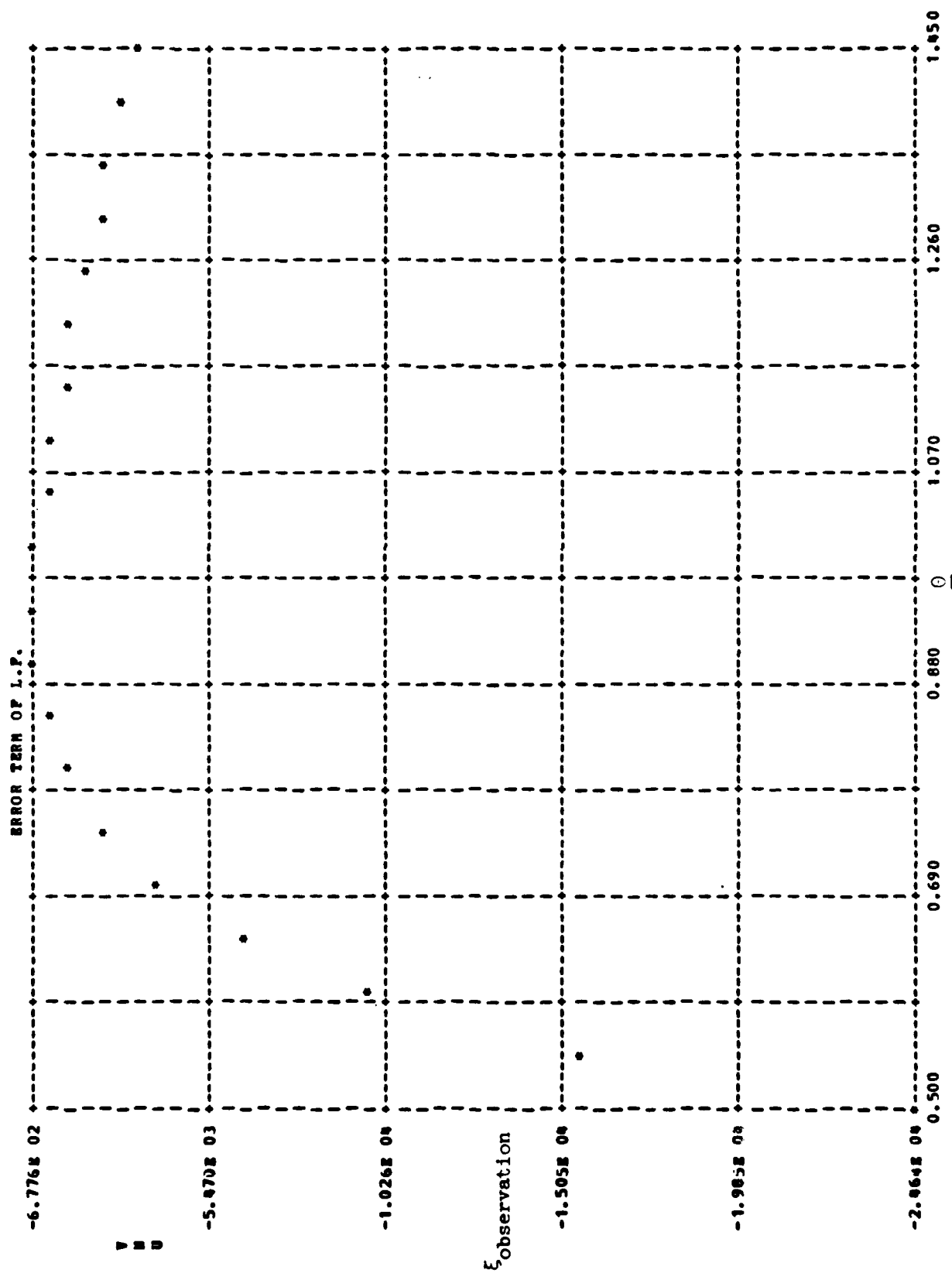


Figure H.8 Plot of the observation term for test case 3.

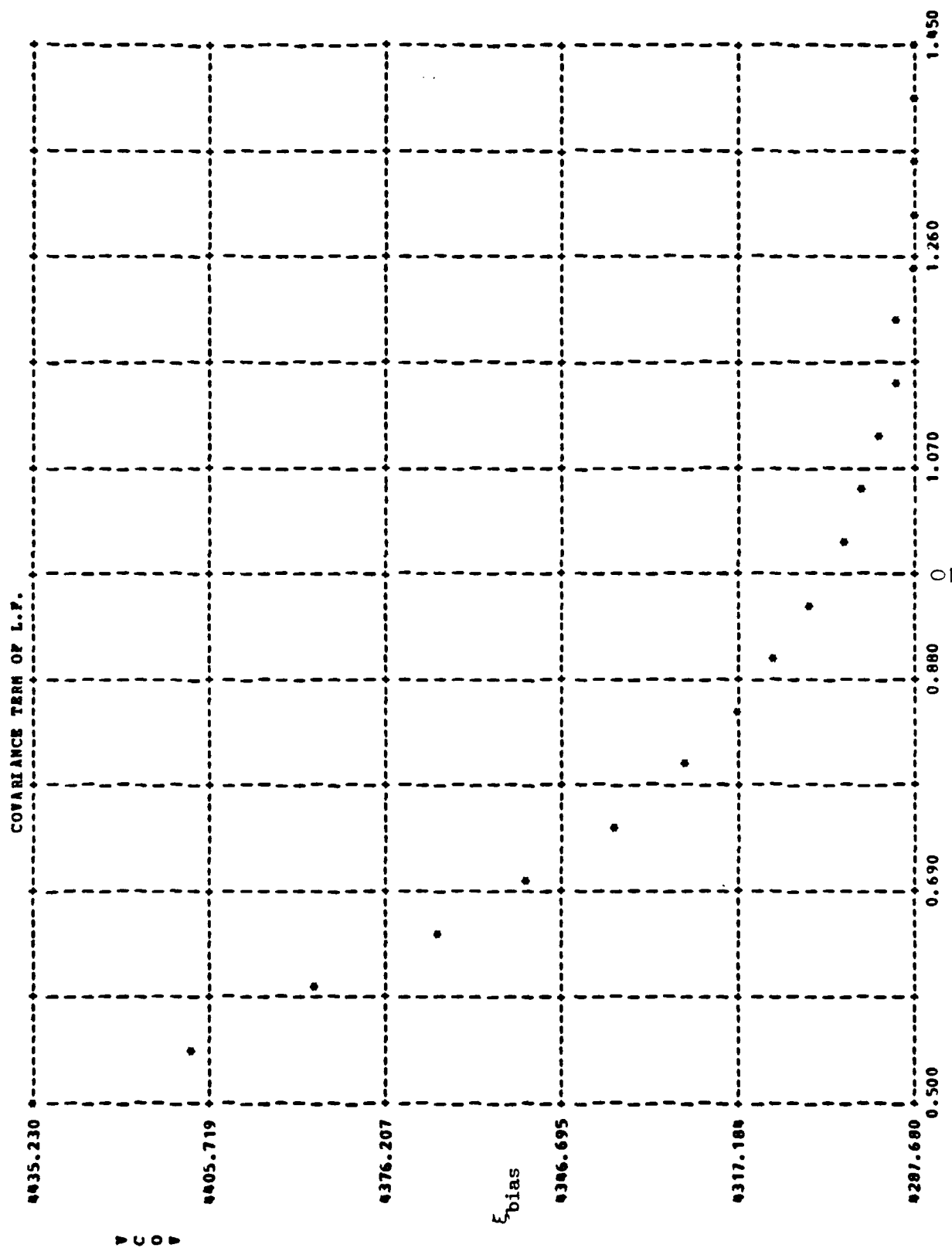


Figure H.9 Plot of the bias term for test case 3.

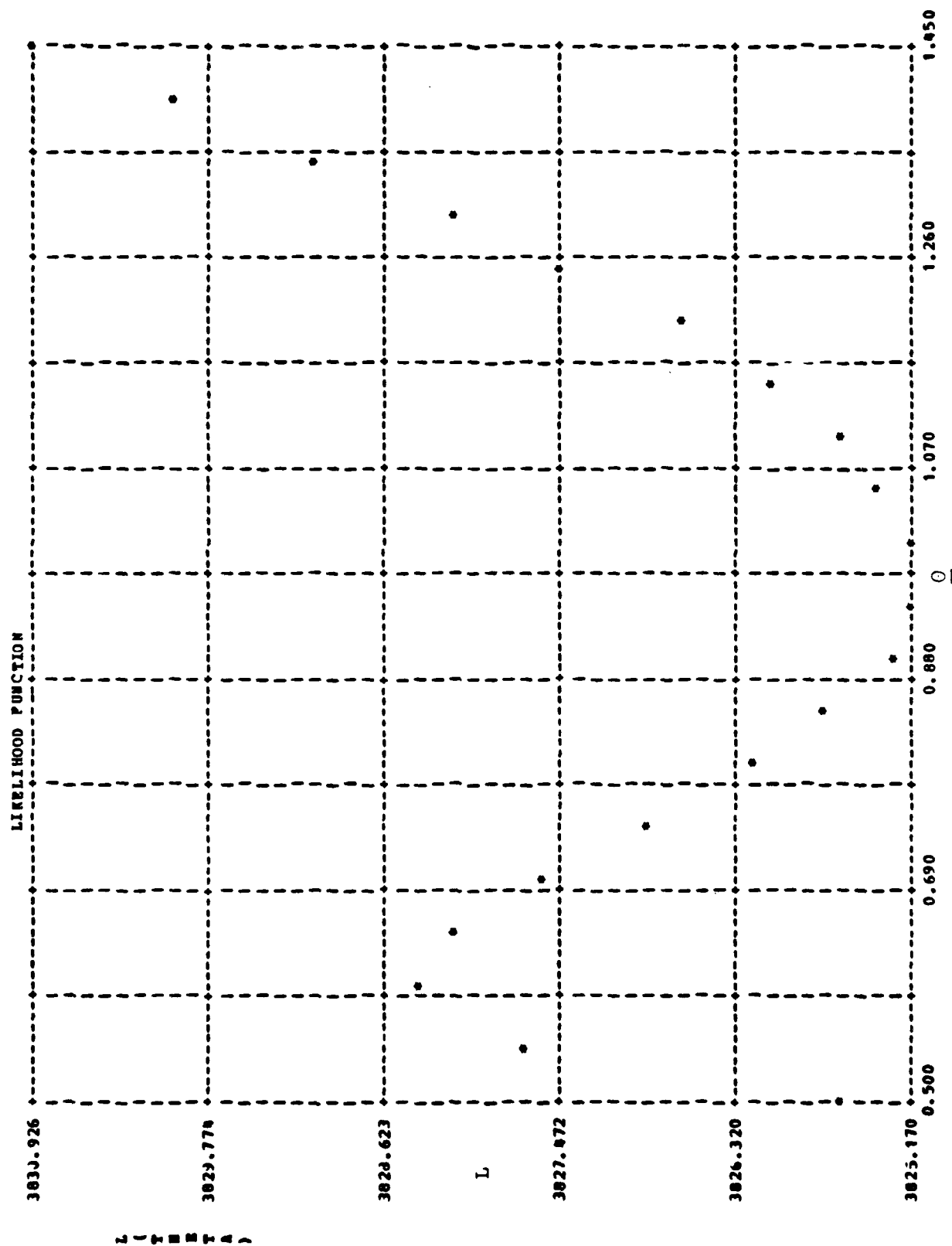


Figure H.10 Plot of the likelihood function for test case 4.

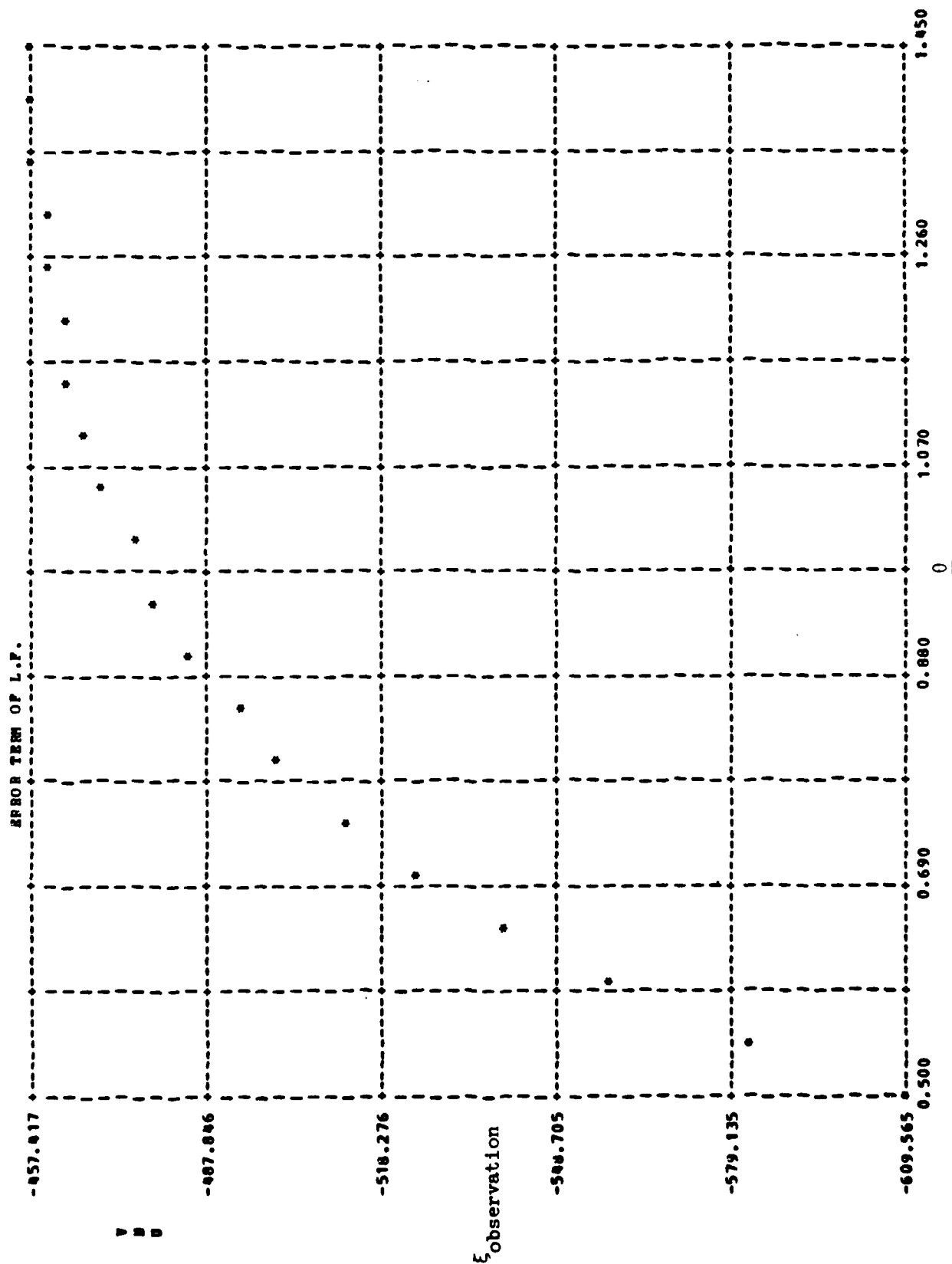


Figure H.11 Plot of the observation term for test case 4.

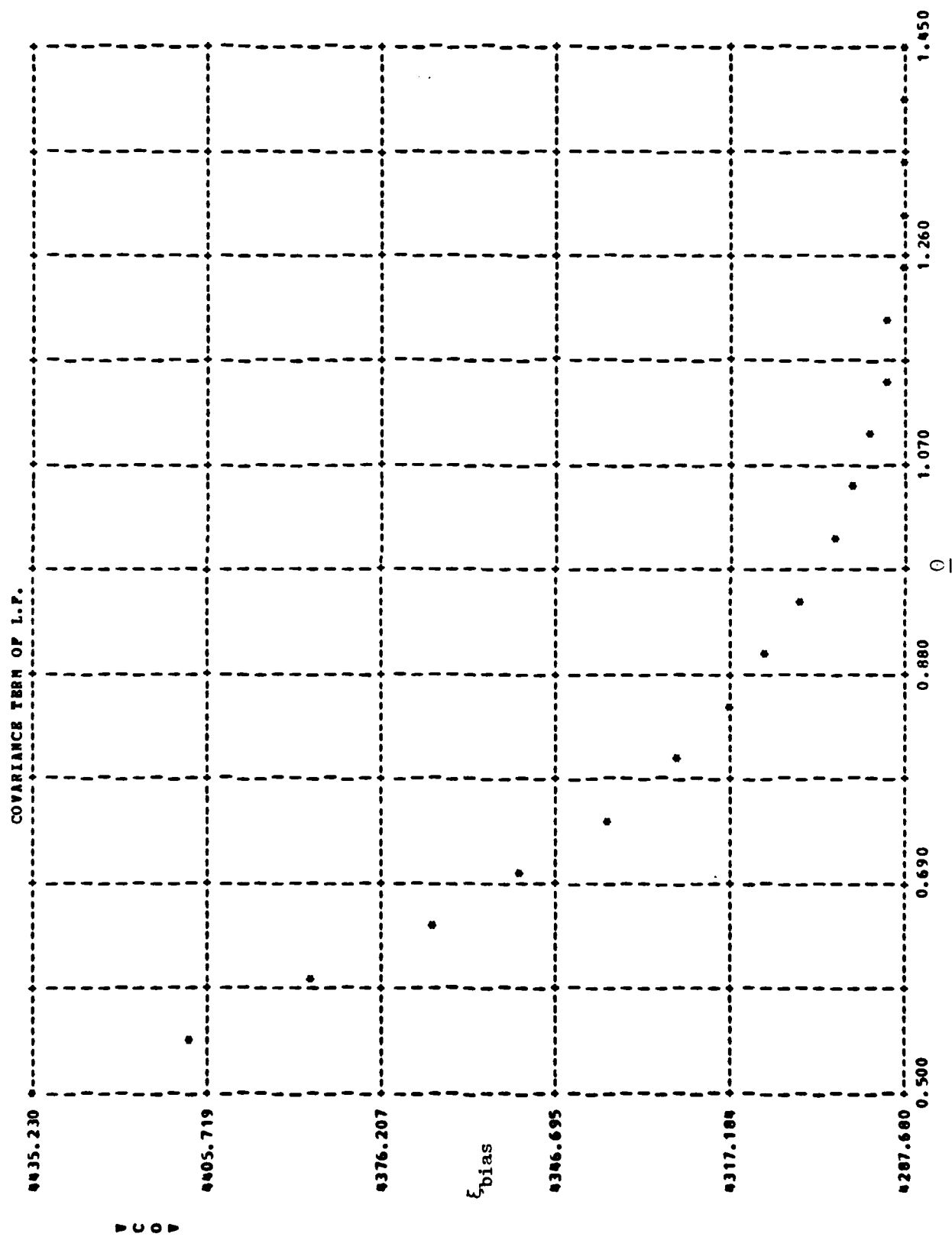


Figure H.12 Plot of the bias term for test case 4.

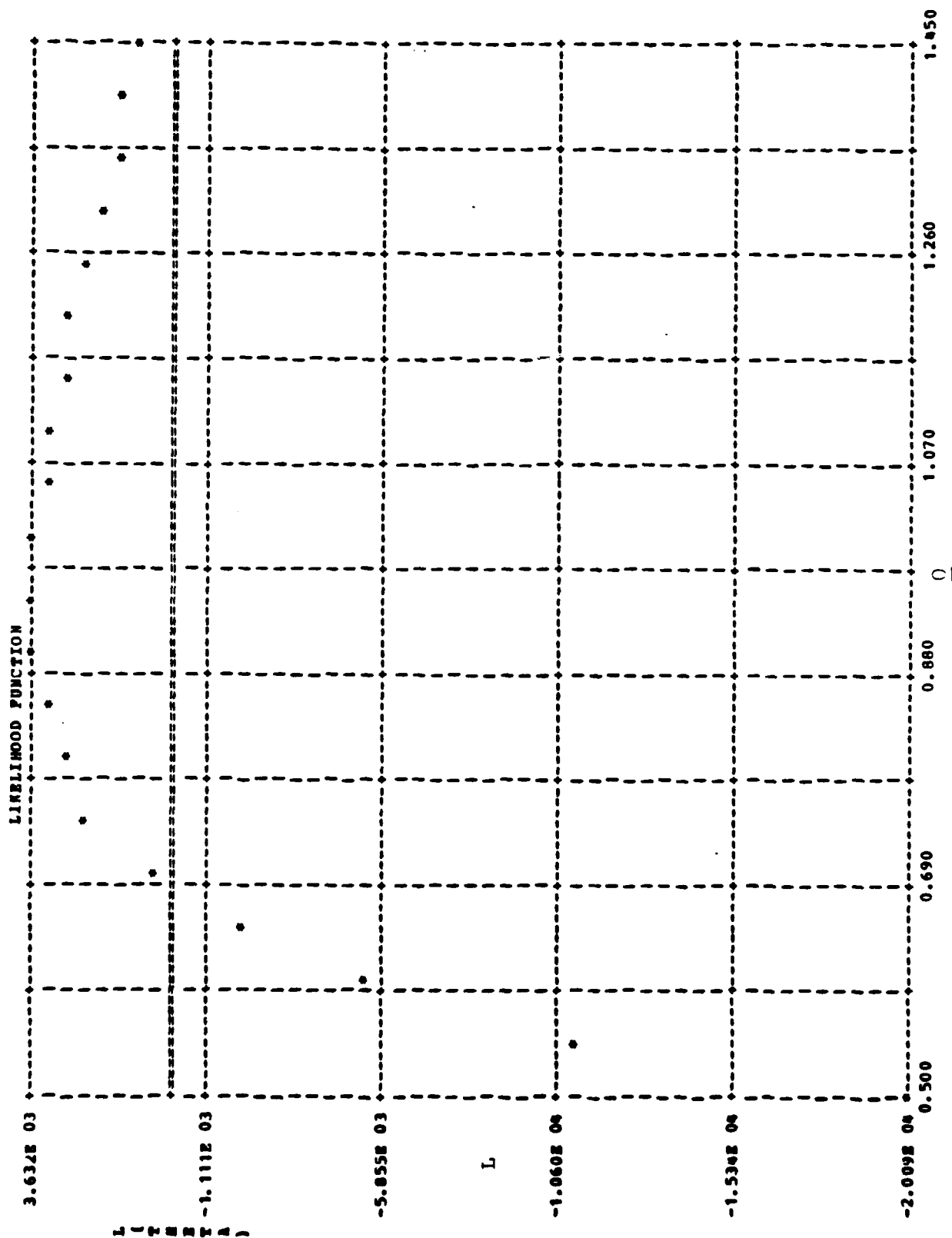


Figure H.13 Plot of the likelihood function for test case 5.

Figure H.14 Plot of the observation term for test case 5.

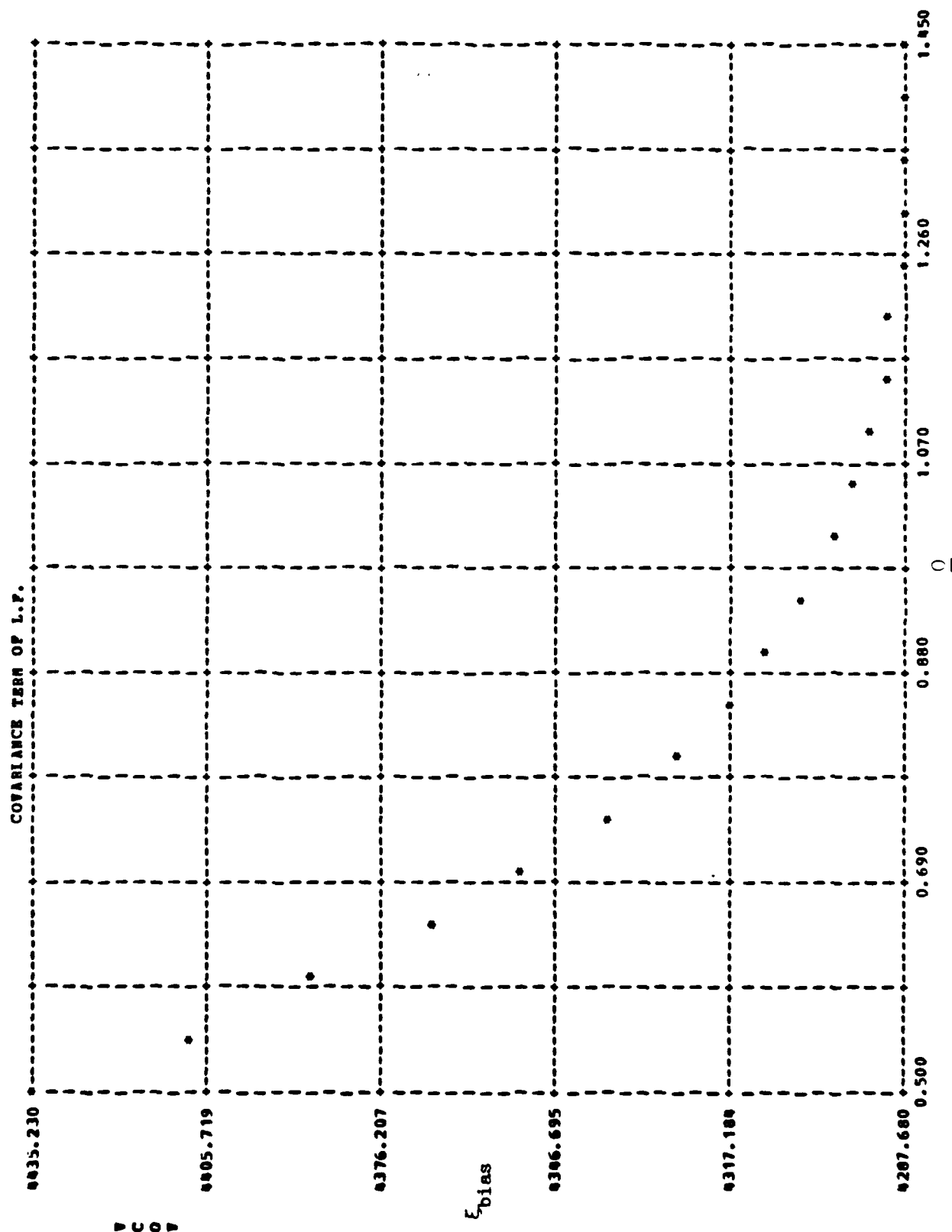


Figure H.15 Plot of the bias term for case 5.

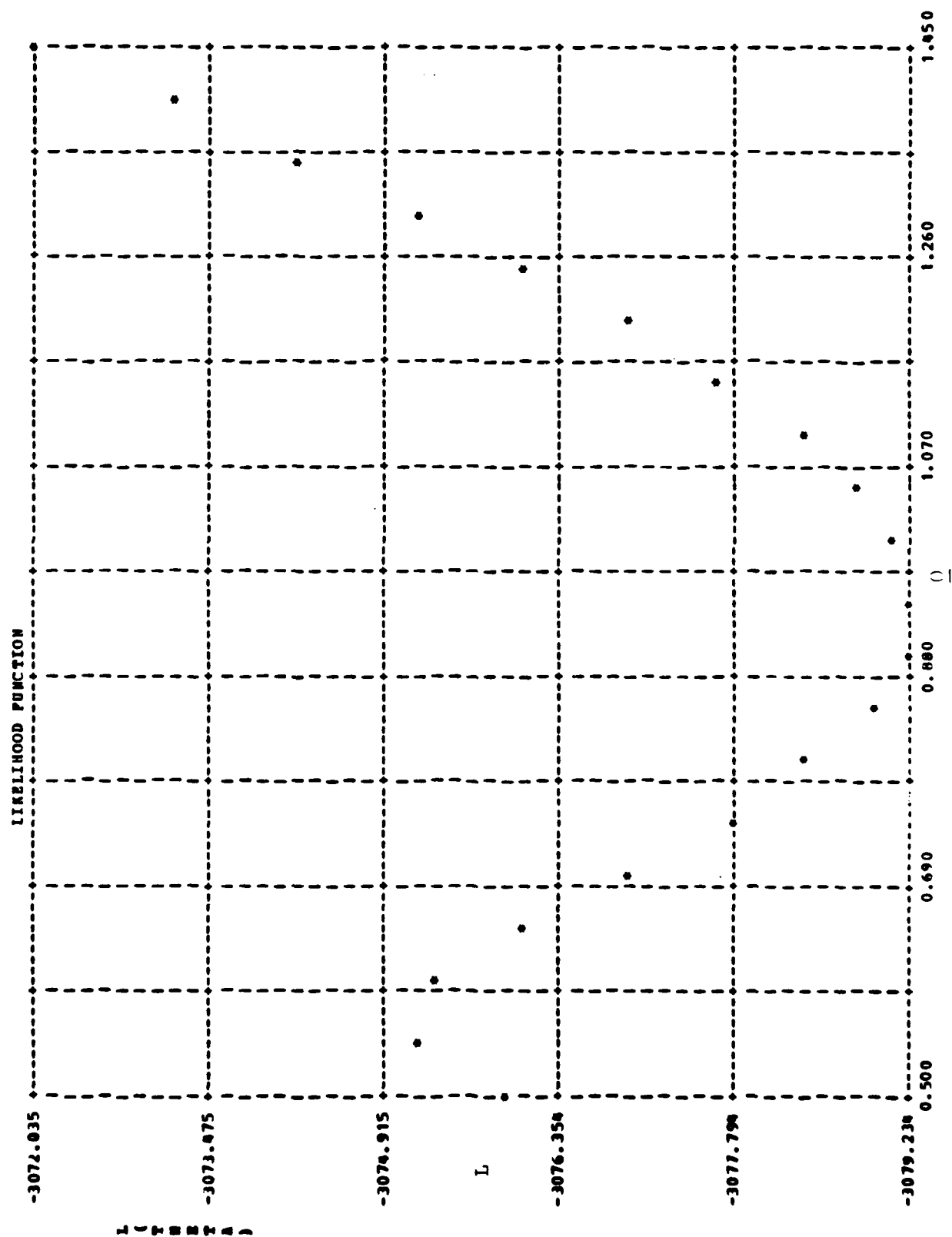


Figure H.16 Plot of the Likelihood function for test case 6.

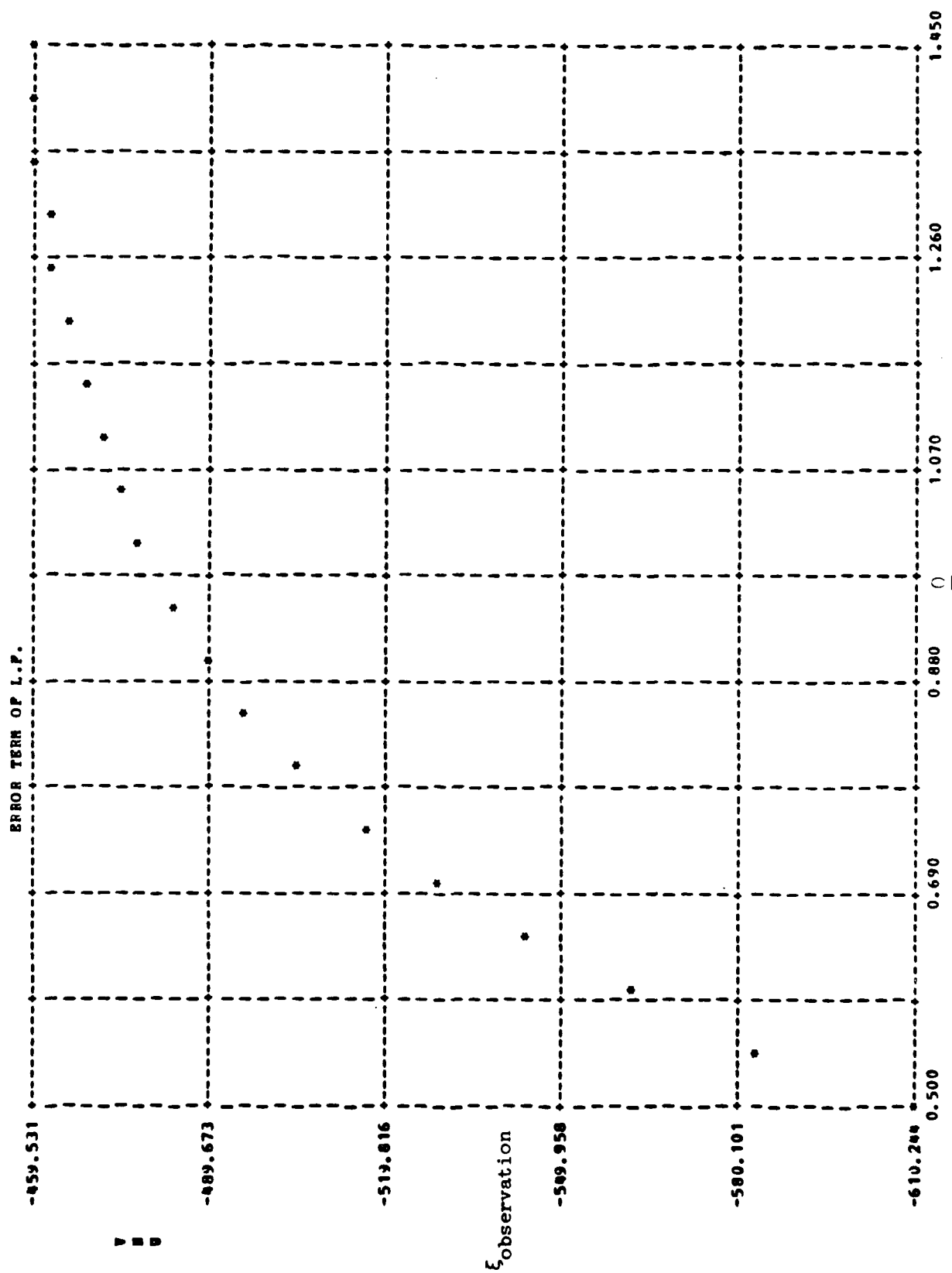


Figure H.17 Plot of the observation term for test case 6.

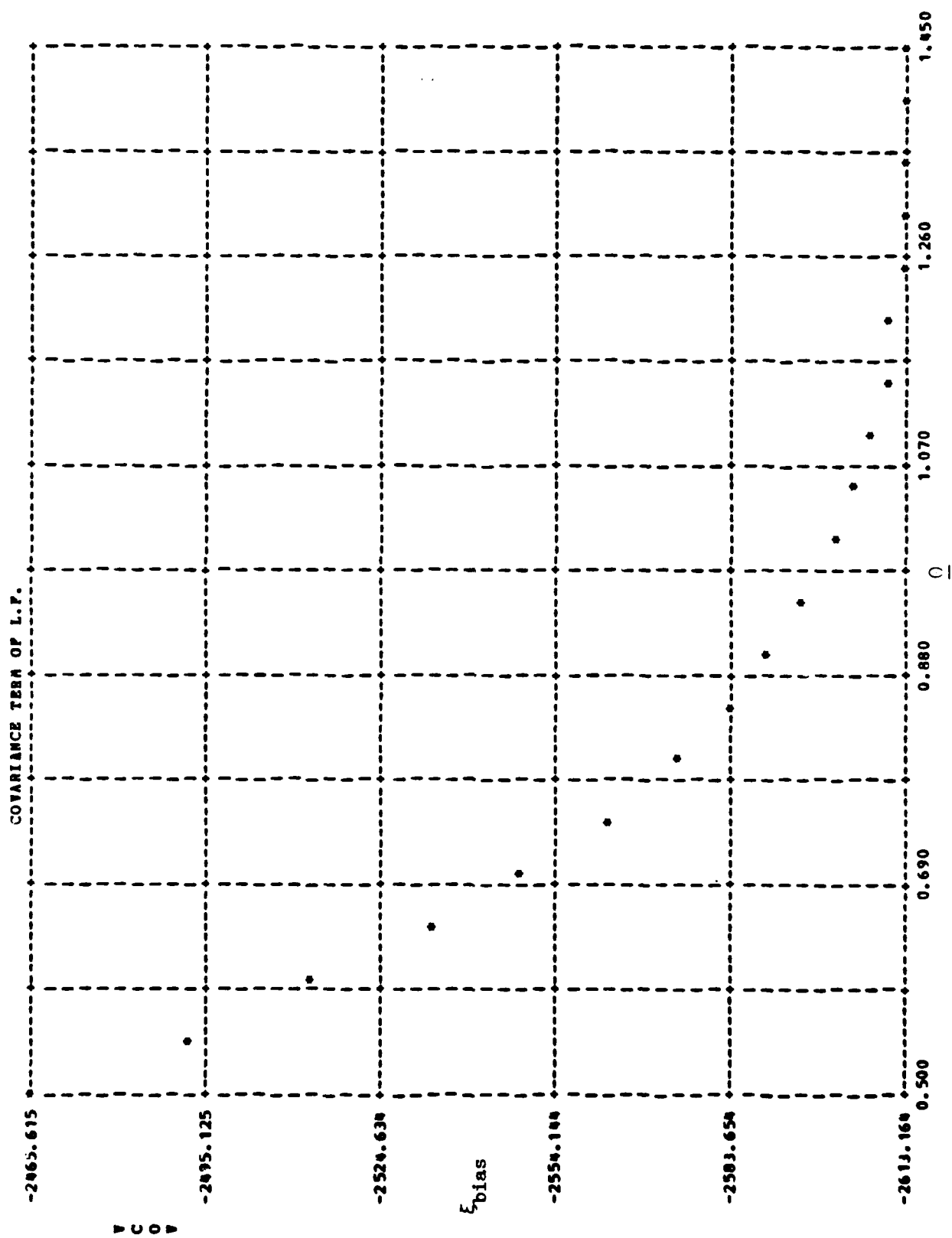


Figure H.18 Plot of the bias term for test case 6.

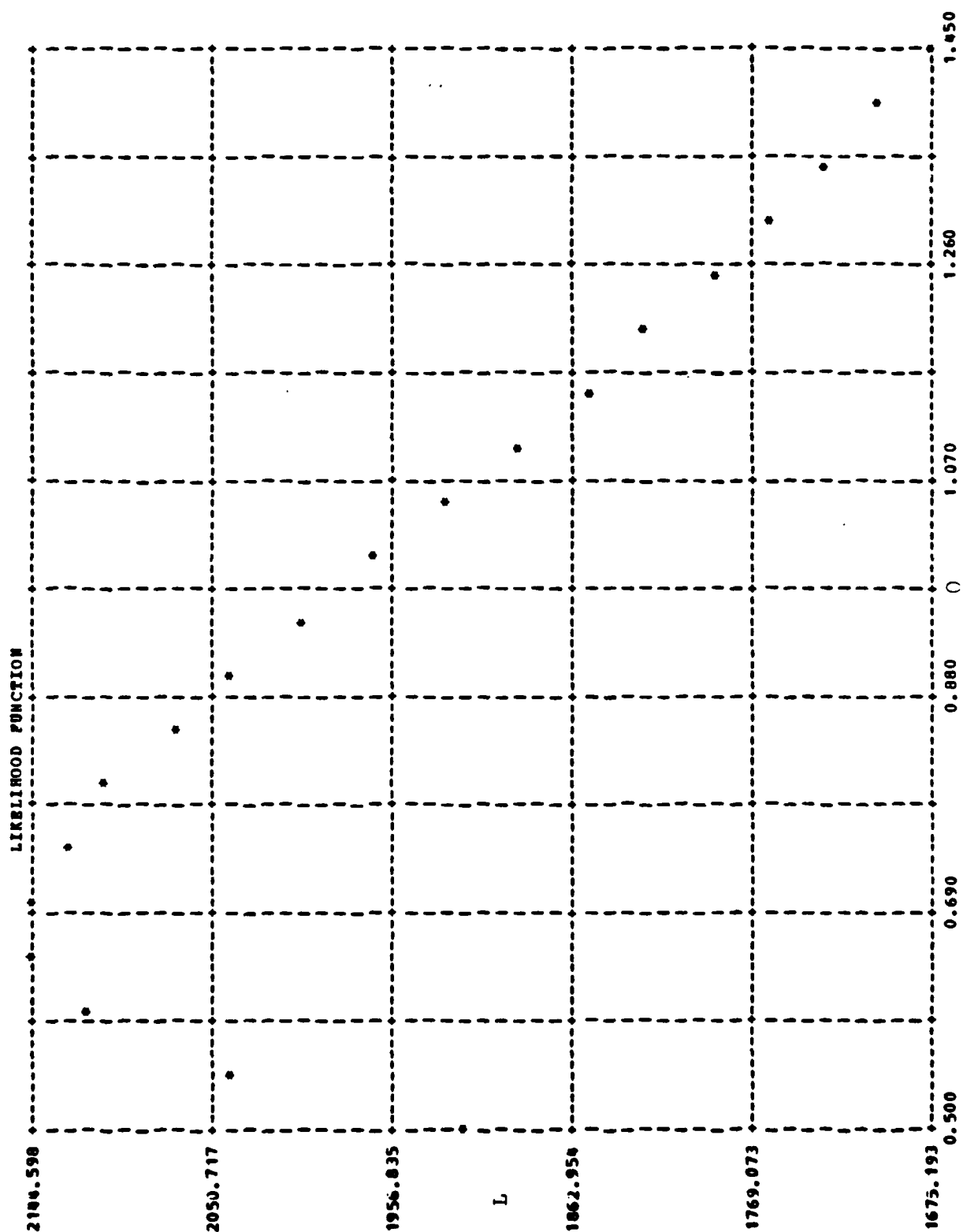


Figure H.19 Plot of the likelihood function for test case 7.

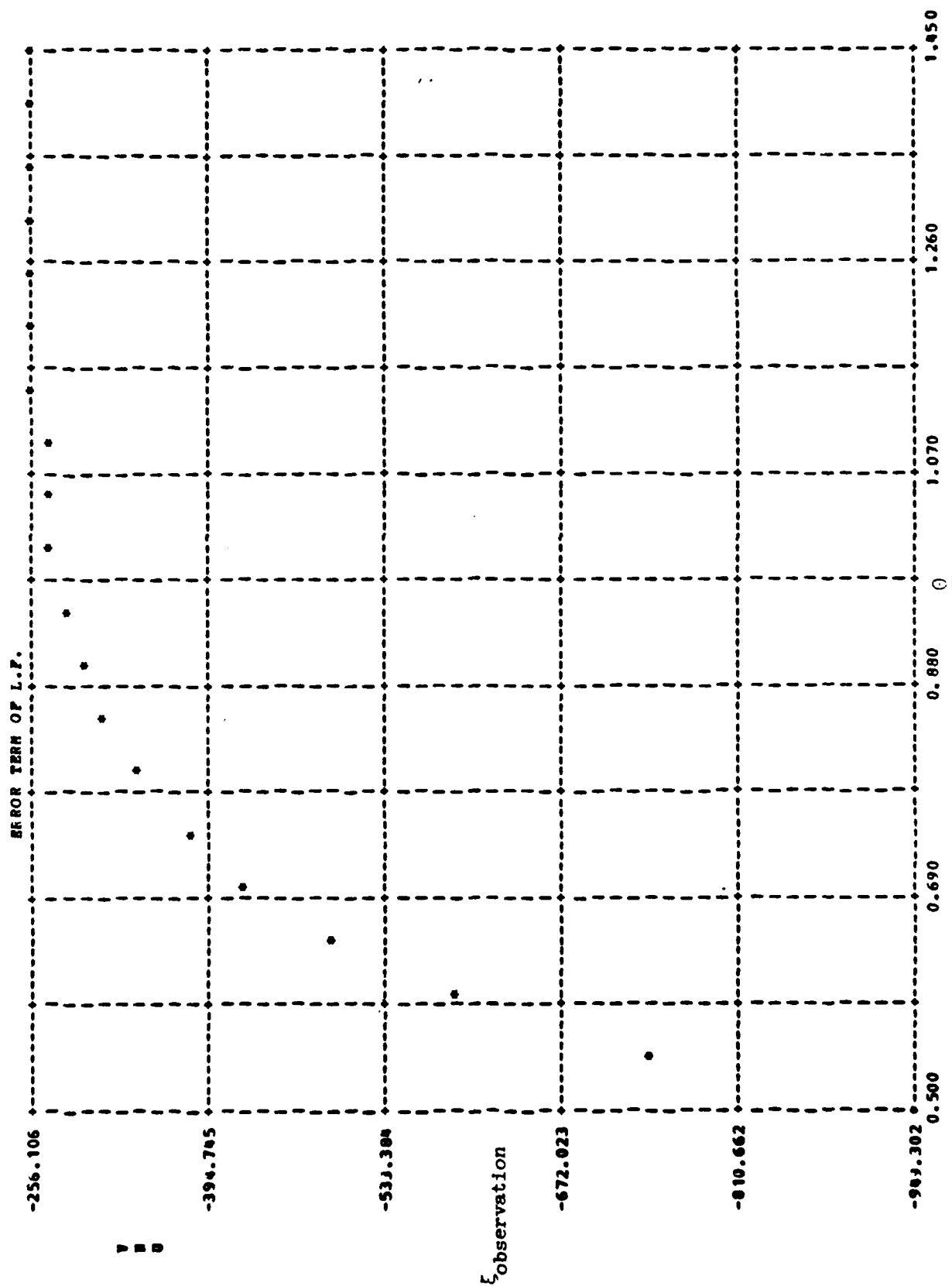


Figure H.20 Plot of the observation term for test case 7.

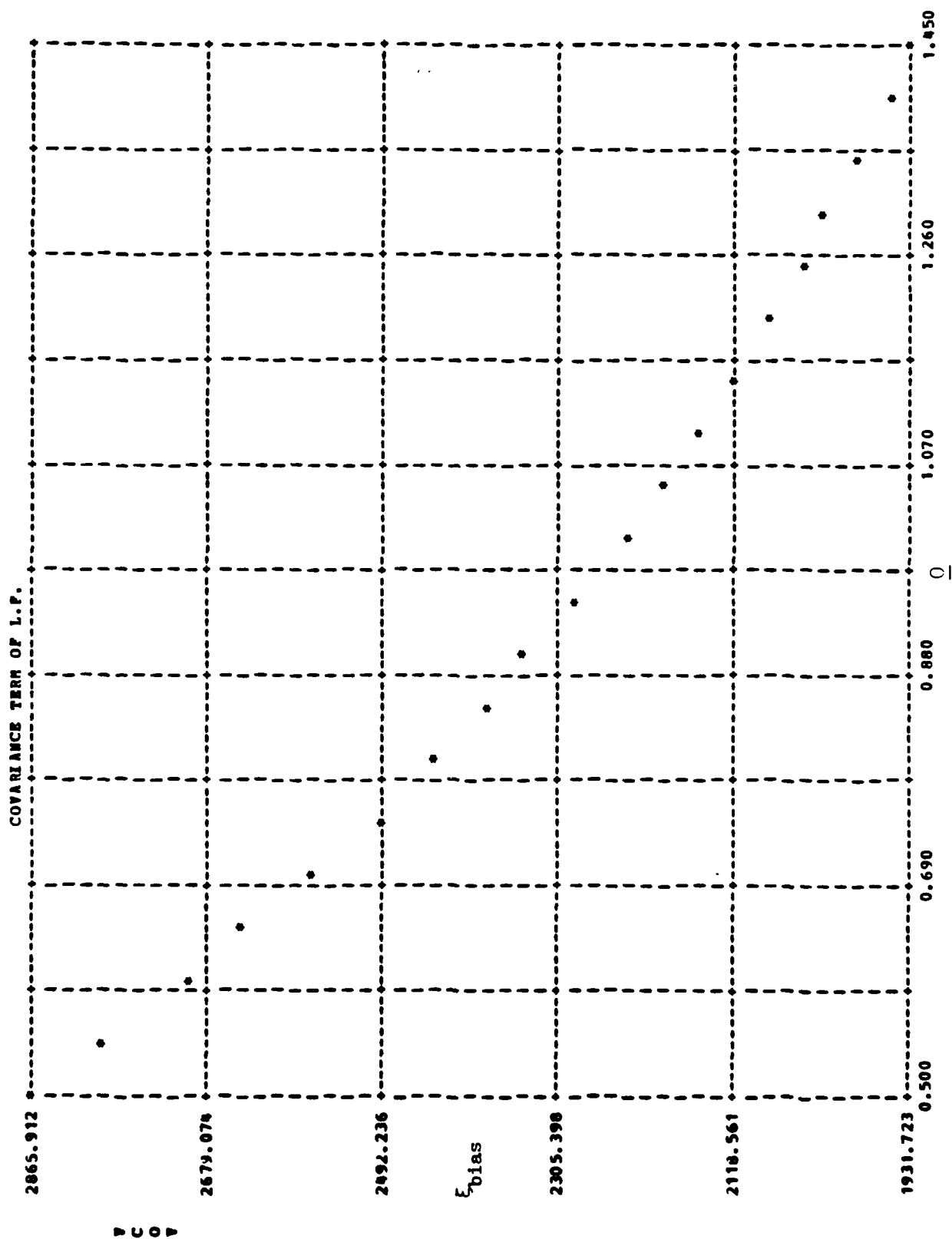


Figure 4.21 Plot of the bias term for test case 7.

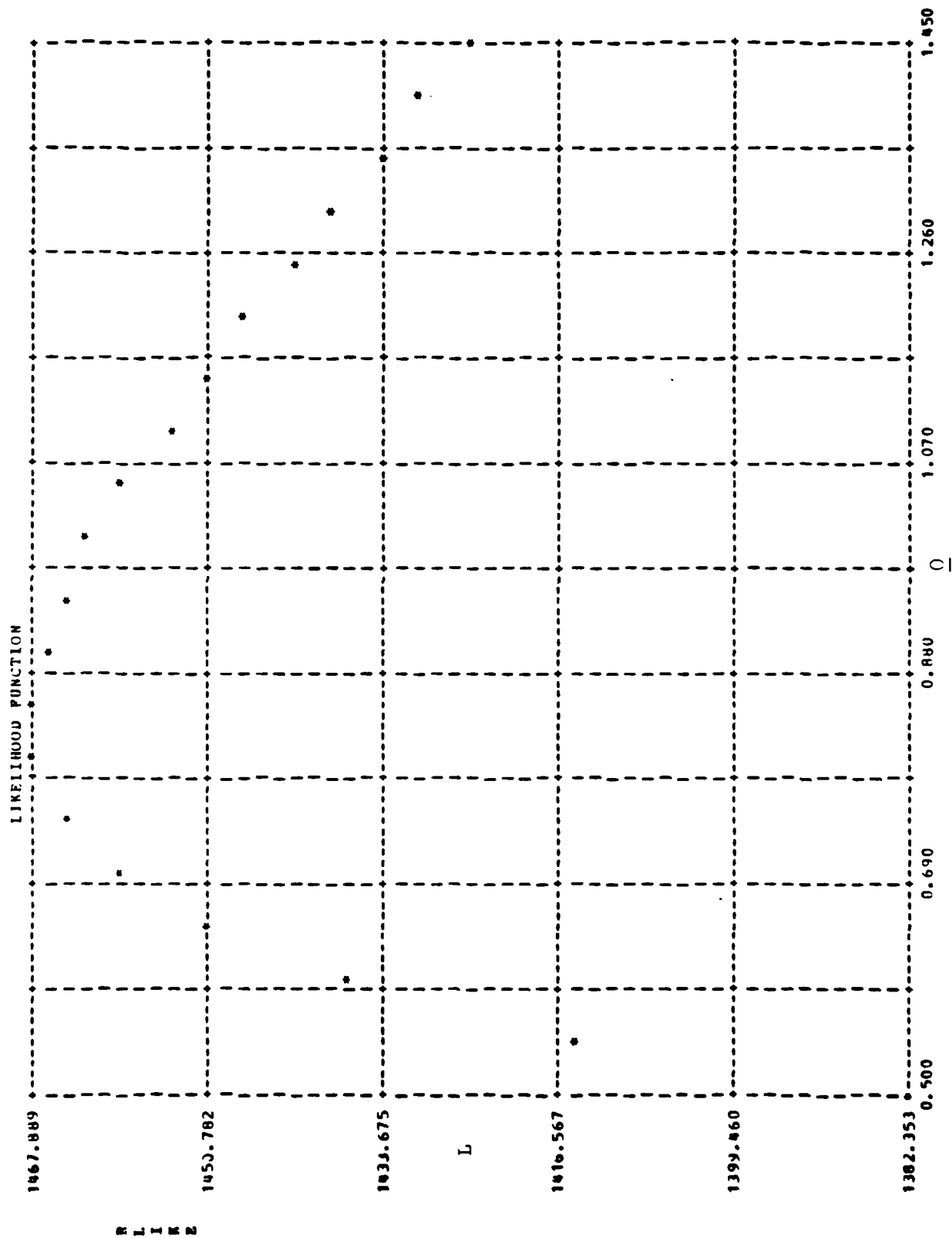


Figure H.22 Plot of the likelihood function for test case 8.

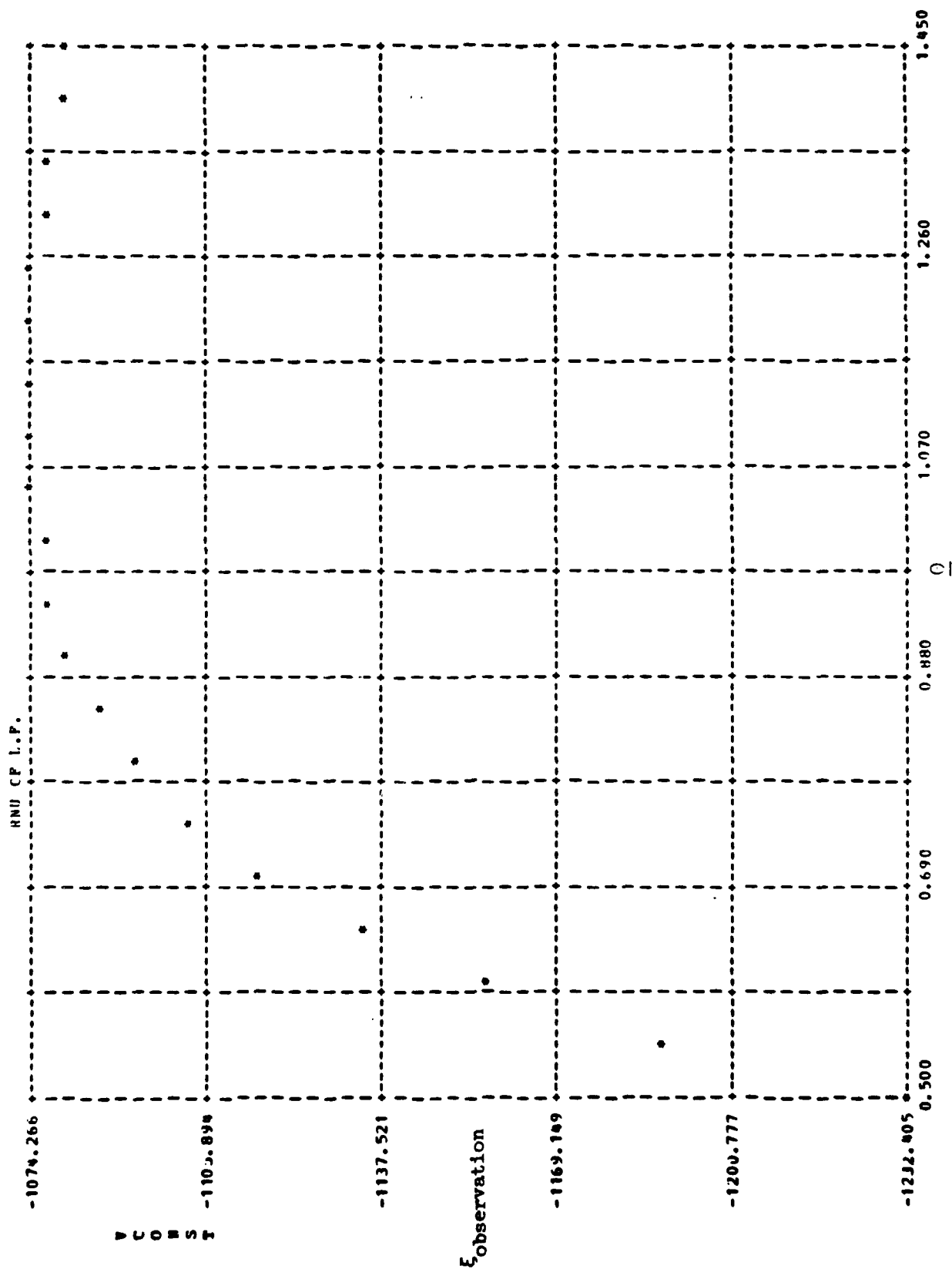


Figure H.23 Plot of the observation term for test case 8.

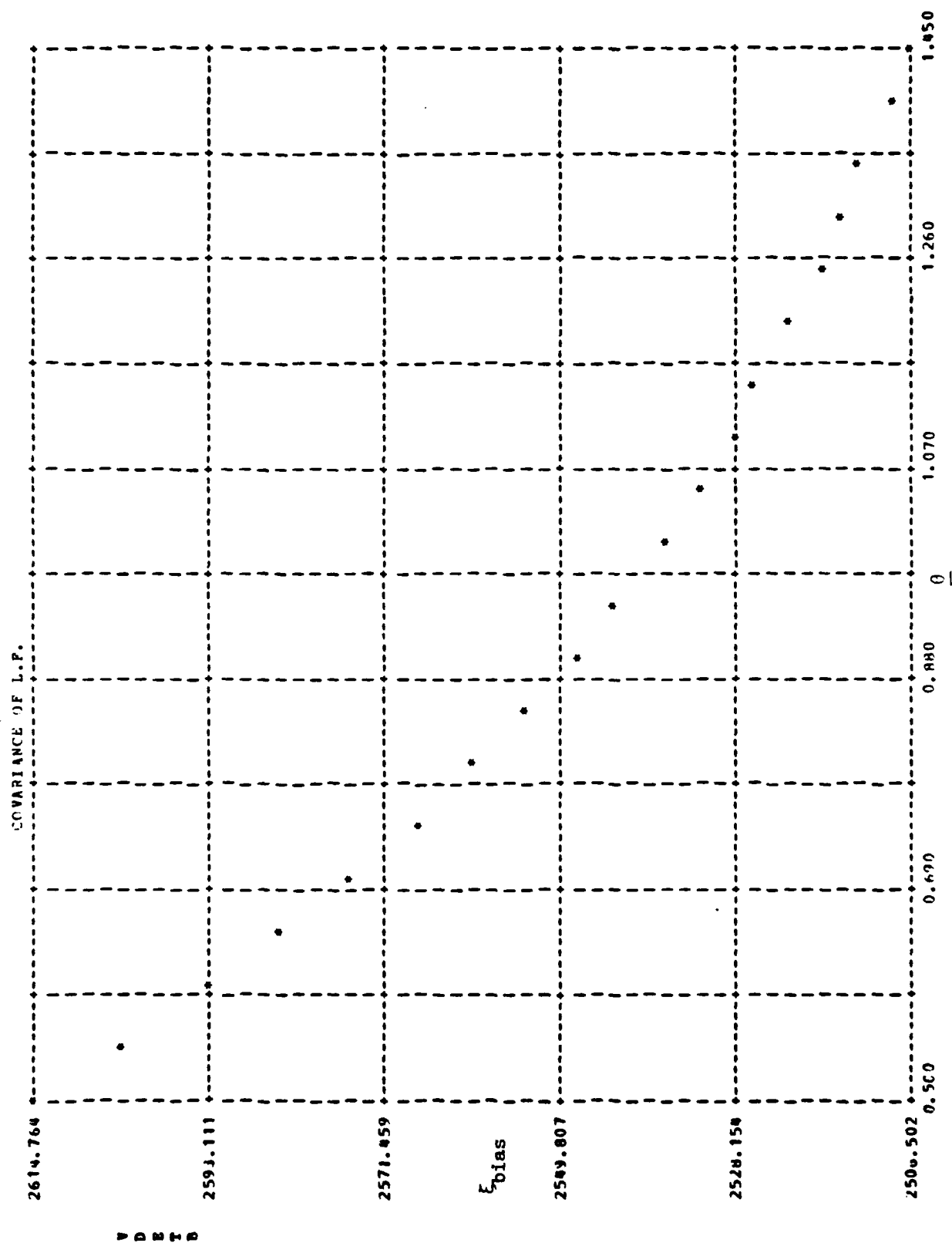


Figure H.24 Plot of the bias term for test case 8.

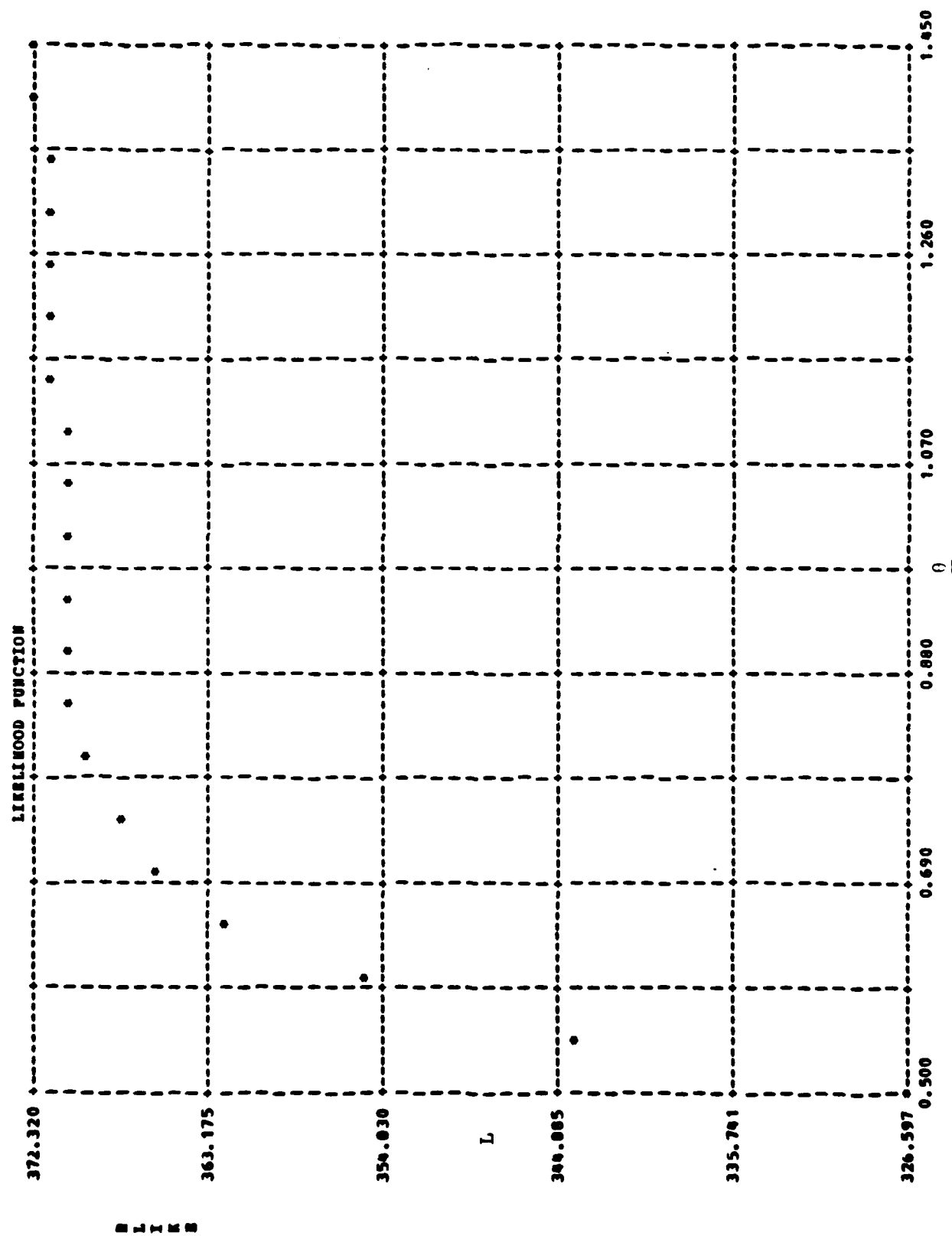


Figure H.25 Plot of the likelihood function for test case 9.

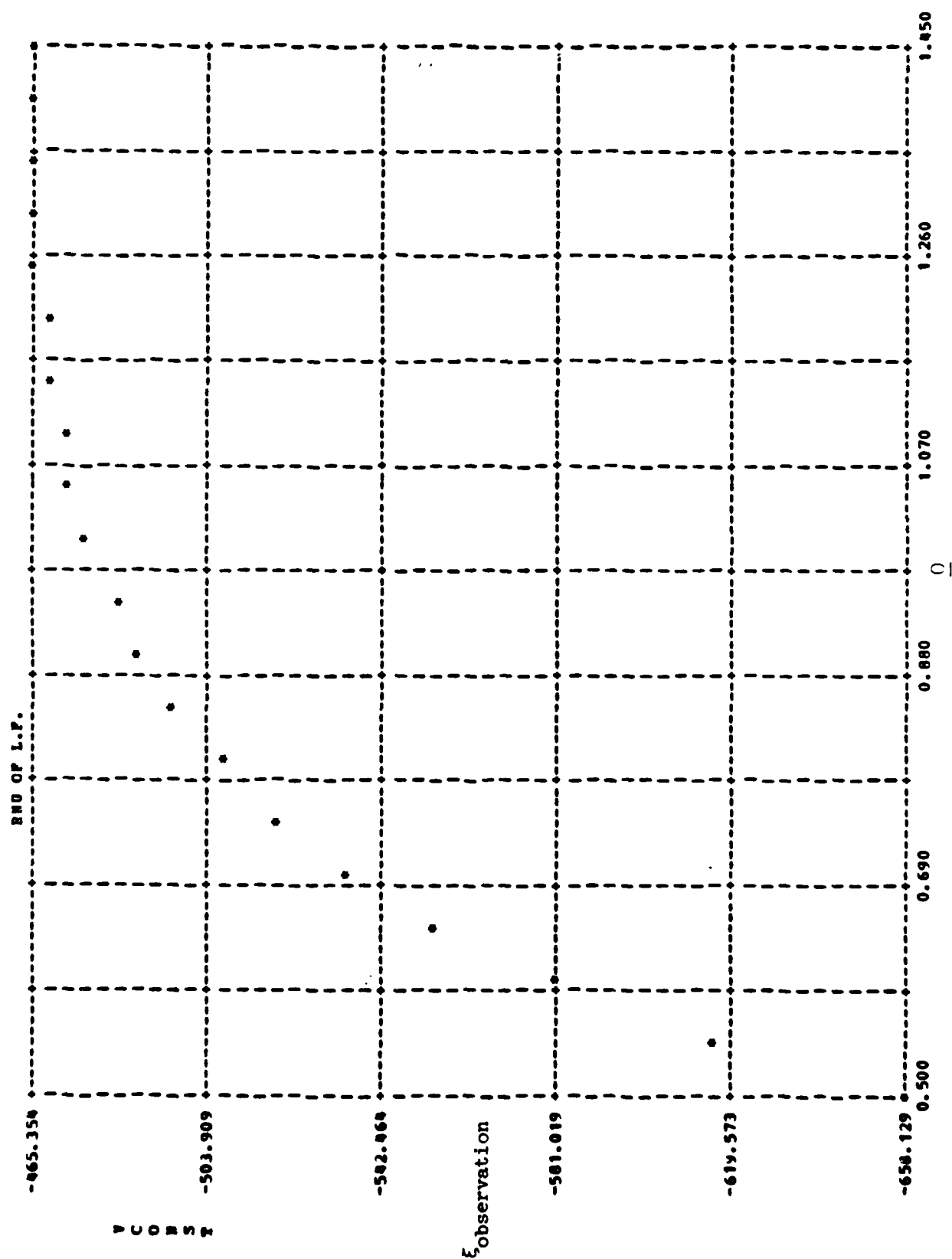


Figure H.26 Plot of the observation term for test case 9.

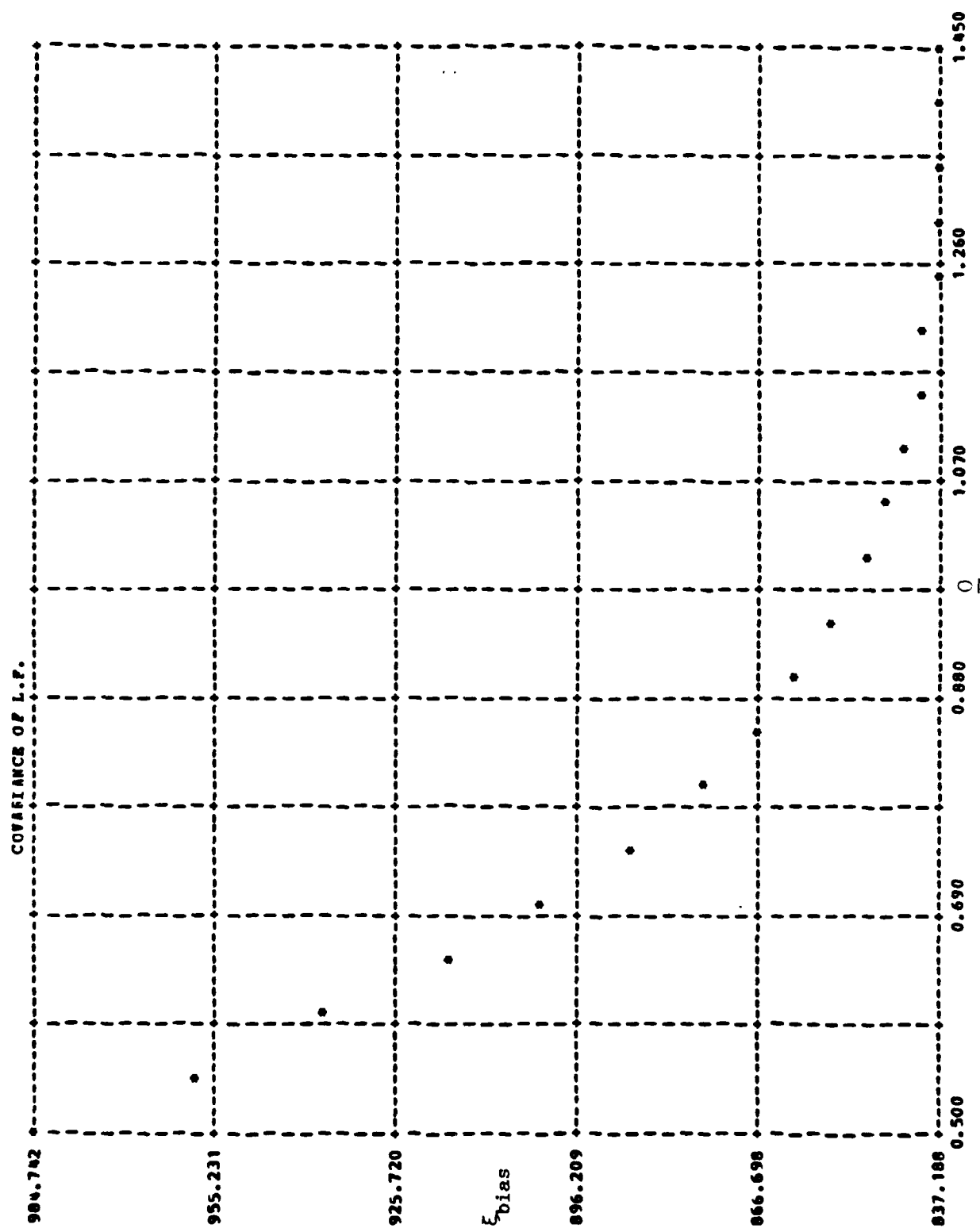


Figure H.27 Plot of the bias term for test case 9.

