

DTIC FILE COPY

1

AD-A202 711



JOINT PROBABILITY DATA ASSOCIATION
 (JPDA) ON TRACKING MULTIPLE
 MUNITIONS FRAGMENTS

THESIS

Ronald J. Beyers
 Captain, USAF

AFIT/GE/ENG/88D-2

DTIC
 ELE
 JAN 1 8 1989
 S D

DISTRIBUTION STATEMENT A
 Approved for public release
 Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
 AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

89 1 17 094

AFIT/GE/ENG/88D-2

①

DTIC
ELECTE
S JAN 18 1989 D
Dcy

JOINT PROBABILITY DATA ASSOCIATION
(JPDA) ON TRACKING MULTIPLE
MUNITIONS FRAGMENTS

THESIS

Ronald J. Beyers
Captain, USAF

AFIT/GE/ENG/88D-2

Approved for public release; distribution unlimited

Preface

The purpose of this study was to research and apply current technology electronic data acquisition and tracking techniques to conventional munitions live-fire testing. The choice to apply Joint Probability Data Association (JPDA) in a Kalman filter estimator/smoothener permitted tracking multiple fragments simultaneously.

Extensive Monte Carlo simulation modeling and subsequent analysis demonstrated good tracking performance of position and velocity for non-curving fragments and good rejection of clutter. However, the simplified Kalman filter first-order dynamics model did indicate difficulty in tracking curving fragments, and the overall results from all simulations indicated poor acceleration estimation performance.

In performing this research, I am greatly indebted to my entire thesis committee; Capt Rob Williams, thesis advisor; Maj William H. Worsley, committee member; and Prof Peter S. Maybeck, committee member, for their helpful encouragement and concerned feedback when it was needed the most. A thank you also goes to Dan Zambon of the Information Sciences Laboratory for tolerating my simulation runs' bogging the VAX computers. My last thank you goes to my wife, [REDACTED] for "putting up with it all" over this past year.

Ronald J. Beyers

Table of Contents

	Page
Preface	ii
List of Figures	v
List of Tables	viii
Abstract	ix
I. Introduction	1
Problem Overview	1
Scope	6
Assumptions	6
Chapter Overviews	8
II. Background	10
Past Testing Methods	10
Review of Candidate Sensor Technologies	14
Final Sensor Selection	21
Kalman Filters	21
Joint Probability Data Association (JPDA) History	27
JPDA Specifics	29
Rauch-Tung-Striebel (RTS) Fixed-Interval Optimal Smoother	38
Summary	39
III. Development	41
Reference Frames	42
Fragment Dynamics	43
Filter Dynamics	45
Measurement Model	50
Filter Initialization Procedure	58
Summary	65
IV. Simulation	66
Fragment Trajectory Generator	66
EKF/JPDA/RTS Tracker/Smoother Program	69
Summary	73
V. Analysis	75
Simulation Runs	75
Performance Summaries	78
Summary	99

	Page
VI. Conclusions and Recommendations	100
General Conclusions	100
Recommendations	102
Appendix A: Measurement Geometry	105
Appendix B: Event Algebra	113
Appendix C: Simulation Output Plots	118
Bibliography	443
Vita	448

List of Figures

Figure	Page
1.1 Master Data Acquisition Block Diagram	4
1.2 Post-Mission Data Processing Block Diagram	5
1.3 Zero-lift Drag Coefficients for Various Bodies	7
2.1 Exploded View of Typical Dahlgren Velocity Screen	12
2.2 Example Arena Set-up, General Overhead View	13
2.3 Detail of Arena Set-up, with Example Dimensions.	14
2.4 Characteristic Plots of Beam-Discharge Lag	17
2.5 Target Weighted Tracking Method	28
2.6 Measurement Weighted Tracking Method	28
2.7 JPDA System Block Diagram	30
2.8 Example Validation Gates with Associated Ω Validation Matrix	33
3.1 Free-body Diagram of Typical Fragment	42
3.2 Fragment Trajectory State Generation	44
3.3 One Fragment Image Example	54
3.4 Multiple Fragment Image Example	55
3.5 Full Test System Block and Signal Flow Diagram	57
4.1 Measurement Noise Corruption Process	69
4.2 Example Performance Plot	73
5.1 EKF Plots for 20 Versus 100 Monte Carlo Runs	75
5.2 RTS Plots for 20 Versus 100 Monte Carlo Runs	76
5.3 True Standard Deviation Measurement Noise Levels for 1000 Monte Carlo Runs	79
5.4 Resulting Standard Deviation Noise Levels for the X_I , Y_I , and Z_I Position States	80

Figure	Page
5.5 "Close" Versus "Distanced" Dual Fragment Run at 100 Monte Carlo Runs	86
5.6 RTS Smoother Output for the Top Plot of Figure 5.5	87
5.7 Clutter Fragment Trajectory Locations	94
A.1 Camera Object Plane Coordinate Frame	104
A.2 Camera Field of View Geometry	105
A.3 Overhead View of Arena Geometry	106
A.4 Radial (Inward) View of Arena Geometry	106
A.5 Side View of Arena Geometry	107
C.1 to C.18 Category 1 Error Performance Plots for Fragment Initial Speed of 3000 ft/sec	119-136
C.19 to C.36 Category 1 Error Performance Plots for Fragment Initial Speed of 6000 ft/sec	137-154
C.37 to C.54 Category 1 Error Performance Plots for Fragment Initial Speed of 10,000 ft/sec	155-172
C.55 to C.72 Category 2, Fragment 'A', Error Performance Plots for Fragment Initial Speed of 3000 ft/sec	173-190
C.73 to C.90 Category 2, Fragment 'B', Error Performance Plots for Fragment Initial Speed of 3000 ft/sec	191-208
C.91 to C.108 Category 2, Fragment 'C', Error Performance Plots for Fragment Initial Speed of 3000 ft/sec	209-226
C.109 to C.126 Category 2, Fragment 'A', Error Performance Plots for Fragment Initial Speed of 6000 ft/sec	227-244
C.127 to C.144 Category 2, Fragment 'B', Error Performance Plots for Fragment Initial Speed of 6000 ft/sec	245-262

Figure	Page
C.145 to C.162	Category 2, Fragment 'C', Error Performance Plots for Fragment Initial Speed of 6000 ft/sec 263-280
C.163 to C.180	Category 2, Fragment 'A', Error Performance Plots for Fragment Initial Speed of 10,000 ft/sec 281-298
C.181 to C.198	Category 2, Fragment 'B', Error Performance Plots for Fragment Initial Speed of 10,000 ft/sec 299-316
C.199 to C.216	Category 2, Fragment 'C', Error Performance Plots for Fragment Initial Speed of 10,000 ft/sec 317-334
C.217 to C.234	Category 3, Fragment 'A', Error Performance Plots for Fragment Initial Speed of 3000 ft/sec 335-352
C.235 to C.252	Category 3, Fragment 'B', Error Performance Plots for Fragment Initial Speed of 3000 ft/sec 353-370
C.253 to C.270	Category 3, Fragment 'A', Error Performance Plots for Fragment Initial Speed of 6000 ft/sec 371-388
C.271 to C.288	Category 3, Fragment 'B', Error Performance Plots for Fragment Initial Speed of 6000 ft/sec 389-406
C.289 to C.306	Category 3, Fragment 'A', Error Performance Plots for Fragment Initial Speed of 10,000 ft/sec 407-424
C.307 to C.324	Category 3, Fragment 'B', Error Performance Plots for Fragment Initial Speed of 10,000 ft/sec 425-442

List of Tables

Table		Page
V.I	Category 1, (single fragment) Worse-Case Error Performances	83
V.IIA	Category 2, Fragment 'A' Worse-Case Error Performances	89
V.IIB	Category 2, Fragment 'B' Worse-Case Error Performances	90
V.IIC	Category 2, Fragment 'C' Worse-Case Error Performances	91
V.IIIA	Category 3, Fragment 'A' Worse-Case Error Performances	96
V.IIIB	Category 3, Fragment 'B' Worse-Case Error Performances	97

Abstract

The purpose of this study was to research and apply current technology electronic data acquisition and tracking techniques to conventional munitions live-fire testing. Previously applied high-speed film cameras, celotex bundles, and associated technologies for munitions testing have proven themselves expensive in materials, labor, and time. Such previous test methods cost upwards to \$250,000 per test blast and require from days to weeks to manually compile and reduce collected blast data to an analytical format. The specific scope of this study was to research methods to electronically acquire and track the position, velocity, and acceleration of multiple warhead fragments as they dispersed from the test-blast center. A design specification for a maximum trackable fragment speed was set at 10,000 ft/sec. The theoretical application of xenon strobe illuminated (2.0 microsecond flash duration), orthogonally oriented Charge-Coupled Device (CCD) cameras (in sets of two) provides three dimensional image measurements at a 2.0 microsecond exposure, 5000 frame/sec rate. The acquired and assumed noisy fragment position measurements (recorded digitally) are post-mission processed through an Extended Kalman Filter (EKF) based Joint Probability Data Association (JPDA) multiple target/tracker state estimator followed by a backward time Rauch-Tung-Striebel (RTS) fixed-interval optimal smoother. Strong emphasis was placed on Monte-Carlo computer simulation

testing of this EKF/JPDA/RTS tracker-smoother algorithm. Representative trajectories of straight, curving, and crossing spherical fragments at 3000, 6000 and 10000 ft/sec were modeled and tracked with promising accuracies in position and velocity. Information gained by these fragment state variables allows the estimation of each fragment's kinetic energy and therefore lethality. The presented fragment data acquisition system was deemed realizable and practical with existing technologies, although the CCD camera 5000 frame/sec requirement was found difficult to obtain reliably. The initial proposed system hardware cost will be high; however, critical system components (such as cameras) survive the test blast and are continuously reusable to keep overall long-term costs down. In addition, the entire data reduction process is reduced from days or weeks to several hours (overnight) on an autonomous EKF/JPDA/RTS computer program.

JOINT PROBABILITY DATA ASSOCIATION ON TRACKING MULTIPLE MUNITIONS FRAGMENTS

I. INTRODUCTION

I.1 Problem Overview

This document presents research results for an all-electronic data acquisition and processing system for the tracking of multiple airborne munitions fragments for arena munitions test-blast applications. The problem is primarily approached as a stochastic process, and employs recently developed Joint Probability Data Association (JPDA) concepts. This research is the first effort of such a system design and therefore has a primarily theoretical foundation combined with numerous problem simplifications to keep the design reasonably tractable.

I.1.a Problem Statement. The thorough test and evaluation of conventional munitions (bombs, mines, air-to-air missile warheads, etc.) requires experimental detonations in a test environment to collect pertinent functional and lethality performance data. The 3246 Test Wing (TW), Eglin AFB, FL performs such test detonations as "arena tests" where the test munition is partially or fully surrounded by both electronic and non-electronic data collection devices. Upon detonation, the dissemination of pertinent performance data occurs in the micro and millisecond time-frame and imposes a very hostile environment toward the survivability of nearby

instrumentation devices. Past and present arena test data collection techniques have primarily employed passive, pre-positioned, soft target arrays to catch a representative portion of the munition fragment cloud, combined with minimal photo-optical and electronic instrumentation devices. Such a data collection scheme is both expensive (in expended materials and manhours) and of limited data collection versatility and accuracy (21). This research presents an alternative, all-electronic, video/optical instrumentation technology to supplement or replace present data collection systems. The intent is to increase data quantity and quality at lower total cost per test in both expended materials and manhours.

I.1.b Basic Approach. The general approach to the overall system design consists of two parts:

- 1) finding a suitable data collection/sensing technology for the arena test mission, and
- 2) deriving a suitable data processing package that applies stochastic estimation techniques to extract pertinent fragment data from the raw (assumed noisy) data measurements.

I.1.b.1 Data Collection. The developed data acquisition system employs multiple pairs of orthogonally oriented Charge Coupled Device (CCD) video cameras to collect three dimensional (3D) position measurements of the radially dispersing fragments following detonation (see Appendix A for geometry). For mathematical convenience, this study models

the CCD camera geometry such that, each camera has a 10.0 x 10.0 ft object plane at a focal distance of $20 \times 2^{1/2} \approx 28.28$ ft in front of the camera. Appendix A gives a full description of the camera and arena geometry model. The use of CCD cameras is derived from a thorough study of sensor alternatives that is discussed in chapter II. Based on arena test reports that indicate maximum fragment speeds near 8000 ft/sec (16; 18), and an additional 20% design margin, this system is designed for a maximum expected fragment speed of 10,000 ft/sec. This maximum speed specification defined the need for a high measurement (frame) rate with a corresponding short exposure time for each frame to "freeze" motion. These characteristics of high frame rate combined with the short exposure time per frame are deemed feasible through the use of synchronous xenon stroboscopic illumination with the test performed at night. Several CCD camera pairs and associated strobe-lamp arrays are placed throughout the test arena to provide the necessary volume coverage necessary to collect sufficient fragment data, as determined by munitions analysts. The collected image data is recorded (magnetic tape or optical disk) during the test blast for post-mission reduction and analysis. Figure 1.1 presents this data acquisition scheme. In this figure, the timing and control computer controls the timing and synchronization of camera frame rate, strobe lamp triggers, and other instrumentation timing. The collected analog

camera images are digitized (A/D converters) and routed to an applicable high-speed digital recorder (several magnetic tape drives or laser optical disk system). All data is simultaneously recorded with the master clock and appropriate control status flags.

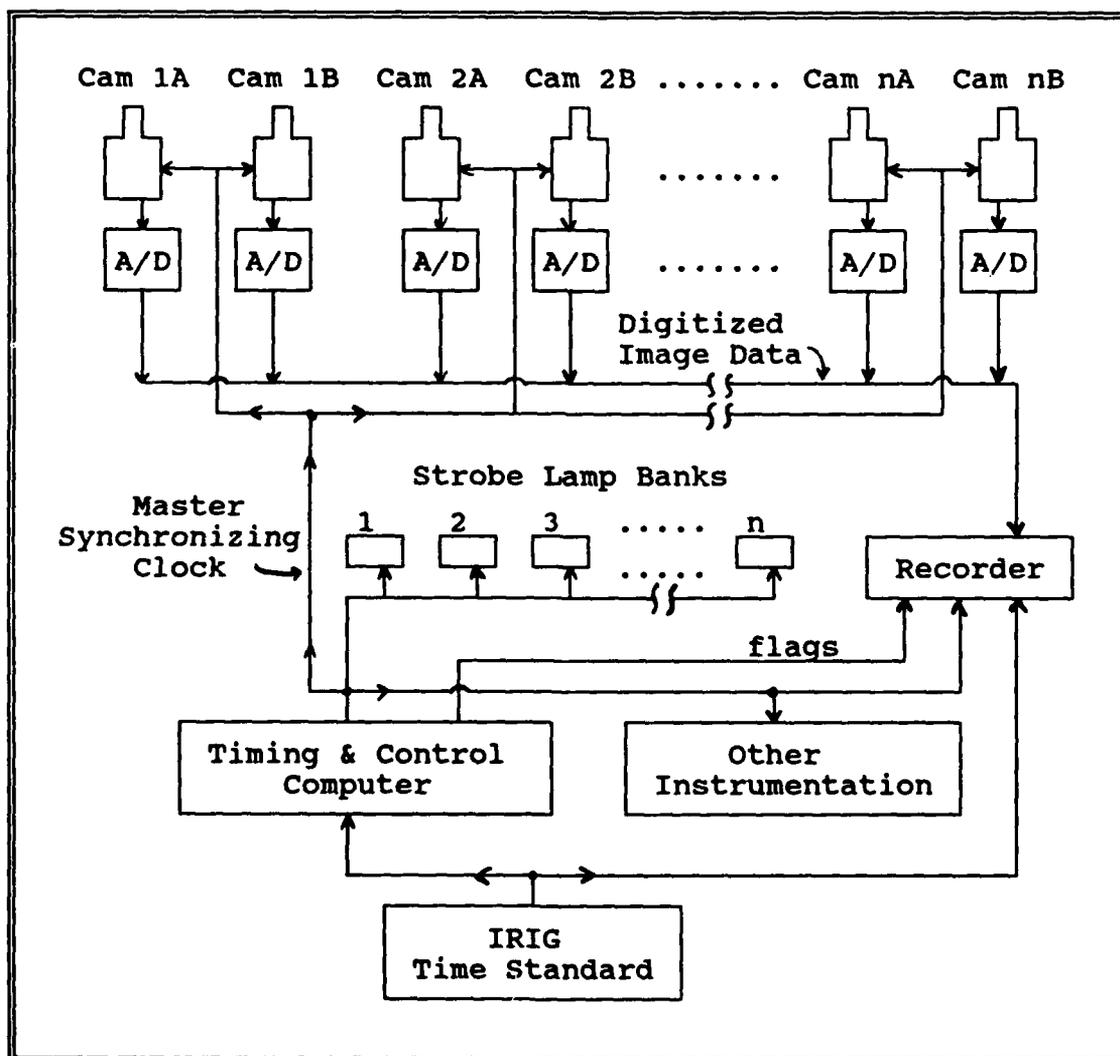


Figure 1.1. Master Data Acquisition Block Diagram

I.1.b.2 Data Reduction/Processing. The recorded image data is played back and processed through a fragment state estimator package presented in this document consisting of an Extended Kalman Filter (EKF) operating together with a Joint Probability Data Association (JPDA) algorithm (see Figure 1.2). The derived forward-time state estimates are then reverse-time processed through a Rauch-Tung-Striebel (RTS) fixed-interval smoother to optimize the fragment state estimates for all time from all information available within the recorded measurement data associated with the test-blast event. Estimated states include the time varying three-dimensional (3D) position, velocity and acceleration for each fragment observed by the cameras.

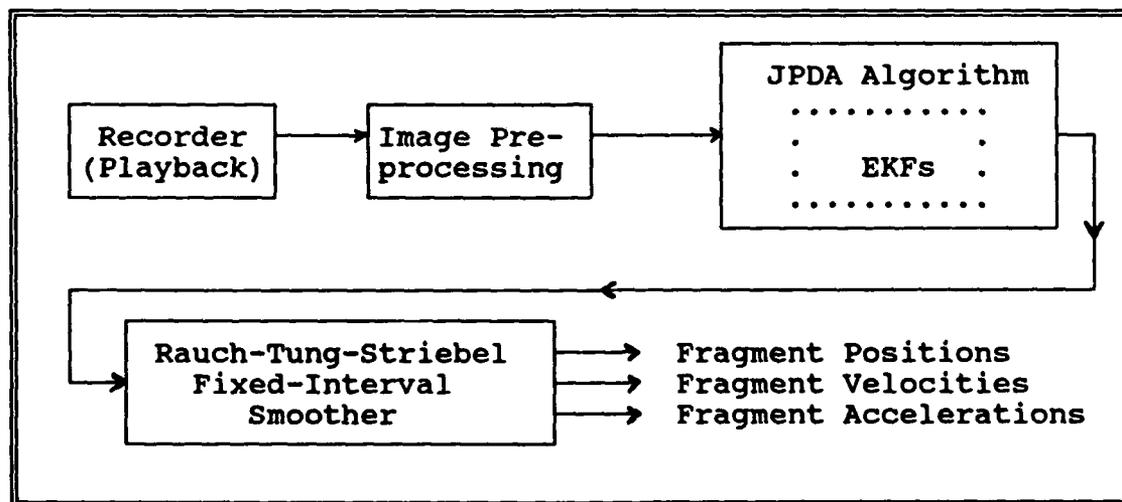


Figure 1.2. Post-Mission Data Processing Block Diagram

Knowledge of the above states allows further analysis to determine each fragment's time varying kinetic energy and

associated lethality. Such lethality determination is not the purpose of this research, only the initial design and simulation of the instrumentation system.

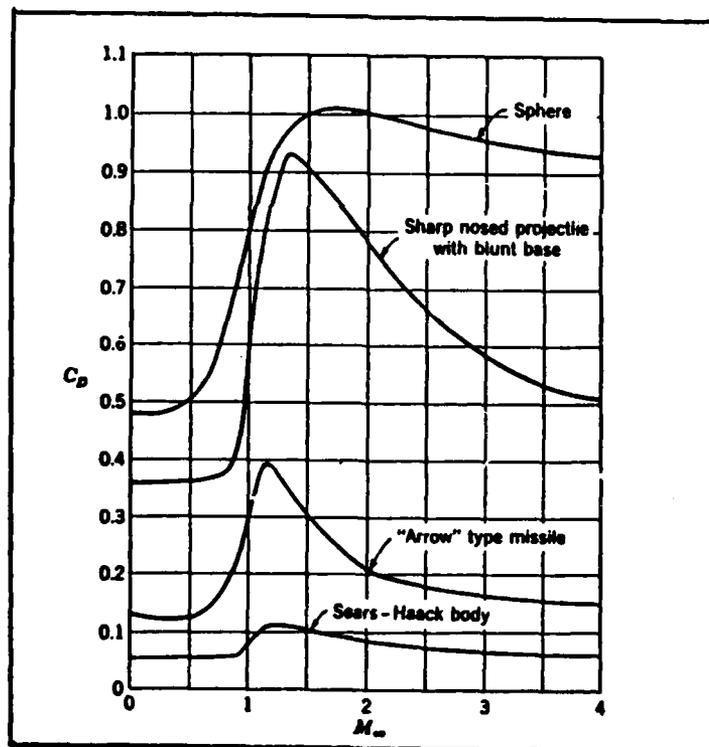
I.2 Scope

This research addresses a number of the important aspects regarding the data acquisition and tracking of multiple, high-speed targets (fragments). It is emphasized that, at the time of this research, no previous work had been found or reported regarding this task. Based on information extracted from previous arena munitions test reports and correspondence with the 3246 Test Wing, elementary level design specifications are developed to define the overall technical problem (e.g. the 10,000 ft/sec spec.). From these specifications, a suitable acquisition/sensing technology is selected, followed by the development of the previously mentioned EKF/JPDA/RTS Tracker/Smother.

I.3 Assumptions

The following assumptions are made, primarily to simplify this initial investigation of the proposed approach and to illustrate the important basic concepts involved. Another reason for making these assumptions is the lack of data needed to support assuming otherwise. Many of the assumptions are presented during the development portion of this document; however, some of the main primary simplifying assumptions are stated here.

- 1) Fragments generally follow a straight-line trajectory from the arena center (location of test munition) radially outward with minimal lateral accelerations normal to the fragment's velocity vector.
- 2) The only accelerations acting on a fragment (post detonation) are those due to aerodynamic drag and gravity.
- 3) Fragments are modelled as spheres consisting of uniform material density.
- 4) Each fragment is traveling in the transonic to hypersonic (0.9 to 4.0 Mach number) speed range, and their coefficients of drag, C_D , are set to 1.0 based on explicit test data for spherical projectiles presented in Figure 1.3, below:



(22:742)

Figure 1.3. Zero-Lift Drag Coefficient for Various Bodies; where M_∞ is the relative speed in Mach.

- 5.) All sensors (cameras, etc) are perfectly aligned and calibrated with no vibration induced from the munition detonation. This significantly simplifies the measurement noise modeling requirements.
- 6.) Each frame of collected image data from the CCD cameras has been pre-processed to determine:
 - a) Each observed fragment's image centroid coordinates, (x_c, y_c) .
 - b) Each observed fragment's diameter, d .
 - c) The total integer number of fragments present in each camera's image.

These three measurement variables are then supplied to the EKF/JPDA/RTS package for processing.

- 7.) All noises, unless otherwise stated, are considered white, Gaussian, stationary, and mutually independent.

From the above set of assumptions, a simplified yet reasonably realistic set of models and solutions may be derived.

I.4 Chapter Overviews

I.4.a Background. The background covers a general review of present arena test technologies and their relation to the problem statement. A research review for various sensor technologies is given to describe the CCD camera selection as the applicable sensor. This is followed by a brief review of Kalman filter basics and the EKF structure by applicable equations. This leads into a background and structural description of the JPDA algorithm, including a general example. The last section gives a brief overview of the Rauch-Tung-Striebel fixed-interval optimal smoother structure.

I.4.b Development. This chapter explains the applied system reference frames, followed by the derived fragment dynamics truth model description. A detailed equation layout follows of the applied Singer approach used in the EKF structure. A breakout of the EKF nonlinear measurement model is then given, accompanied by a measurement ambiguity example. The last section describes in detail the applied EKF initialization procedure developed in this study.

I.4.c Simulation. This chapter gives a general description of the developed simulation programs with their respective features. In addition, this chapter lists the actual numerical values and constants used in the equations developed in previous chapters. A detailed description is included of the applied pseudo-random noise corruption process. The chapter ends with a representative example output plot generated by the simulation package.

I.4.d Analysis. This chapter explains in detail the specific simulation details, nuances, and findings. This chapter also contains the analysis of special "extra runs" used to investigate peculiarities noted in the EKF/JPDA/RTS algorithm.

I.4.e Conclusions and Recommendations. Gives an overall listing of good and bad performance characteristics gathered in the research. This is followed by a list of interest items that need further attention in regards to more simulation runs or system design modifications.

II. Background

This background chapter contains three main parts. The first presents the present arena test data acquisition system in use by the 3246 Test Wing. The second deals with a research and applicability study of various sensor technologies for the arena test application. The final part discusses Kalman filters; putting emphasis on past work regarding the JPDA algorithm.

II.1 Past Testing Methods

II.1.a Test Profile. Isolated portions of the Eglin AFB test range are reserved for the conduct of arena blast test missions (14; 15:225). Munitions tested include omnidirectional and shaped charge munitions that range in size from small antipersonnel (AP) mines and shells to 2000 pound general purpose (GP) bombs. Additionally, for a single given munition type, several tests may require differing sets of data collection under differing test conditions. As discussed in Chapter I, fragment velocities commonly approach 8000 ft/sec. Therefore, the overall conventional munitions arena test mission requires an adaptable data collection scheme that will accommodate a large variation of munitions type, test conditions, and range of fragment velocities.

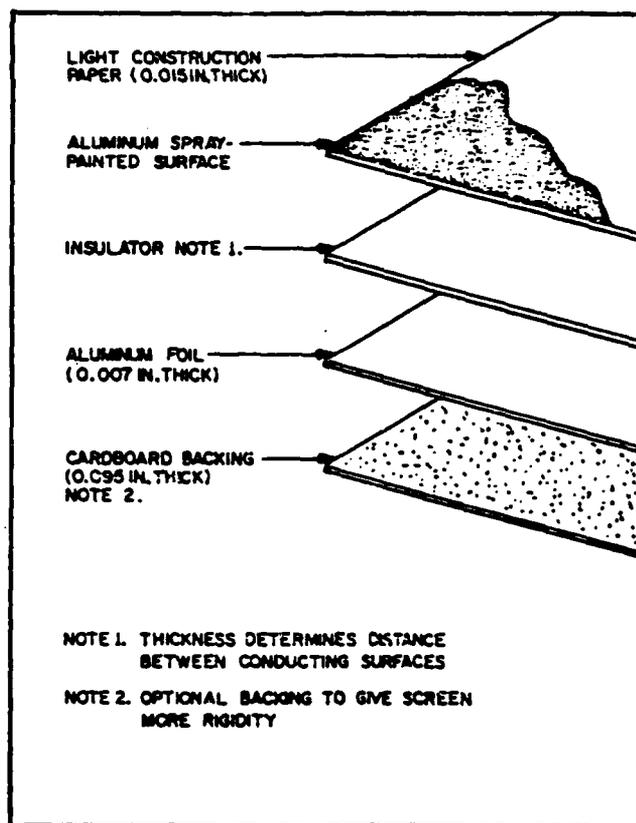
II.1.b Present System. The present primary data acquisition system in use by the 3246 Test Wing for its arena testing is the Vulnerability and Lethality Testing System

(VALTS), a computerized system that employs light sensors, velocity screens, and pressure sensors (21). Supplementing VALTS is the use of fire-resistant mineral fiberboard target bundles, high-speed color film motion picture cameras, and a standard closed circuit TV camera for safety purposes.

Light sensors of the VALTS are used to detect the first light that defines the initiation of case breakup (time t_0). This initial t_0 becomes starting time, to which, all subsequent measurements and observations are referenced to. Timing information is maintained by generating and recording an Inter-Range Instrumentation Group (IRIG) time-code with all VALTS measurements.

A VALTS velocity (Dahlgren) screen is normally placed immediately in front of each fiberboard bundle to detect each fragment's exact arrival time at the bundle (see Figure 2.1). The Dahlgren screen senses the fragment arrival when the fragment provides an electrically conductive path between the sprayed aluminum and aluminum foil layers as it passes through the structure. (21:Sec III). The resulting "pulse" is recorded with the IRIG time-code of the pulse, t_p , on magnetic tape for post-mission analysis. The average velocity for each detected fragment may then be determined by taking the known distance from the test munition to the velocity screen, r_s , and dividing it by the elapsed time since t_0 (21:9):

$$v_{avg} = r_s / (t_p - t_0) \quad (1.1)$$



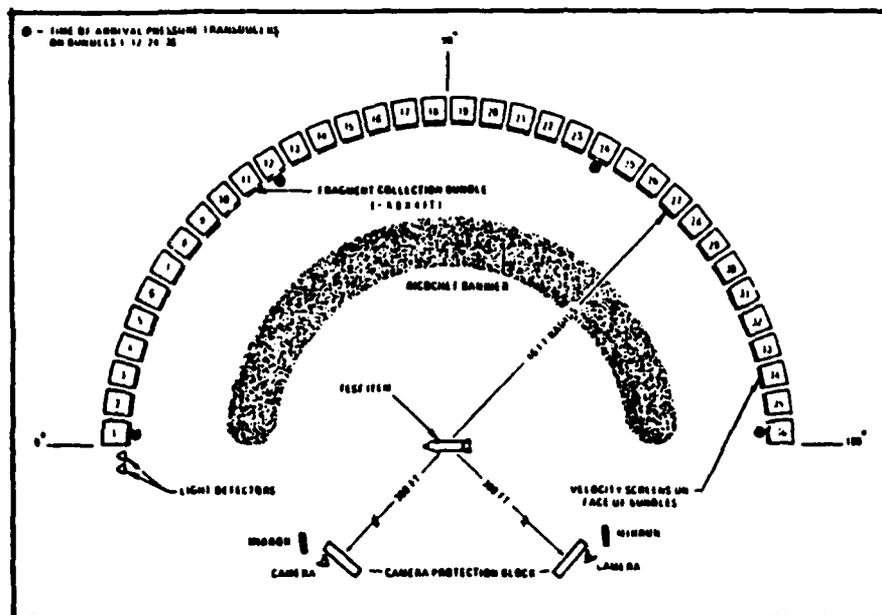
(21:15)

Figure 2.1. Exploded View of Typical Dahlgren Velocity Screen

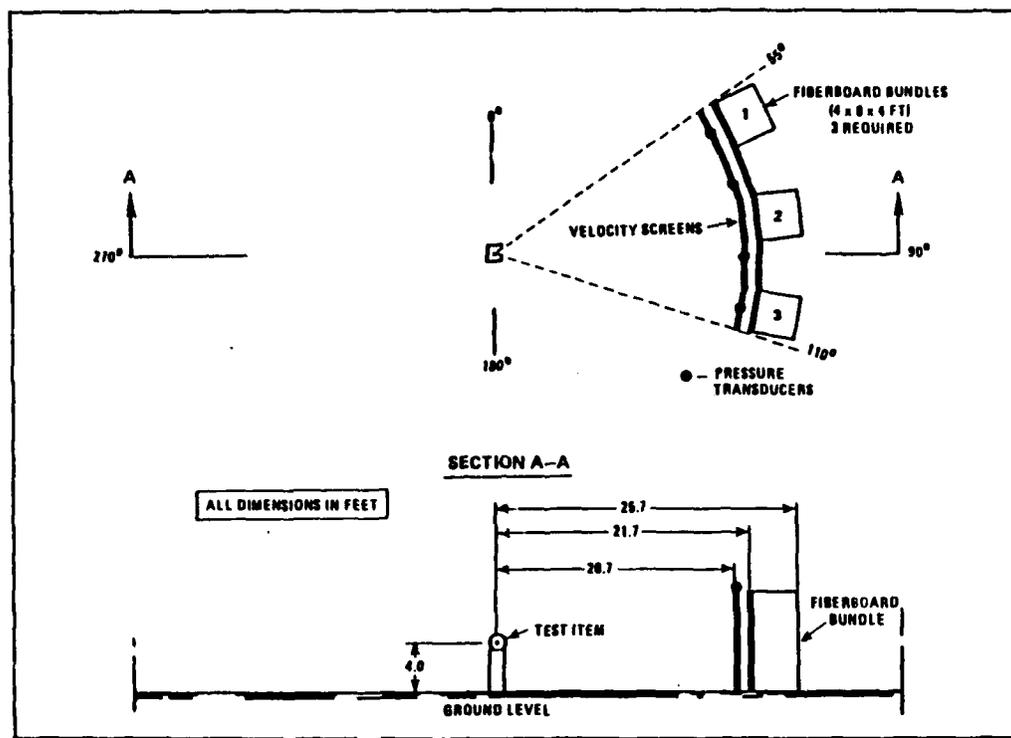
Piezoelectric pressure sensors of VALTS are normally placed in various locations throughout the arena to detect relative blast pressures and to record their time of occurrence with the IRIG time-code (21:24). The fiberboard target bundles may vary in size for each test. Typically, the bundles are eight feet high, four feet wide, and stacked four feet thick. They are positioned to intercept a representative segment of the fragmentation cloud as it disperses from the detonated munition. The bundles are then post-mission disassembled one layer at a time and the coordinates, weight, size, and composition of each embedded

fragment are tabulated for later analysis (16:79-202; 18:21-28; 20:APPX B).

Lastly, each arena test normally includes at least one high-speed motion picture camera operating at up to 8000 frames per second optically recording the test munition detonation and its effect on the surrounding arena environment (16:4; 17:18; 18:6; 19:16; 20:3). Example arena set-ups are shown in Figures 2.2 and 2.3:



(20:5)
Figure 2.2. Example Arena Set-up, General Overhead View



(18:10)

Figure 2.3. Detail of Arena Set-up, with Example Dimensions

II.2 Review of Candidate Sensor Technologies

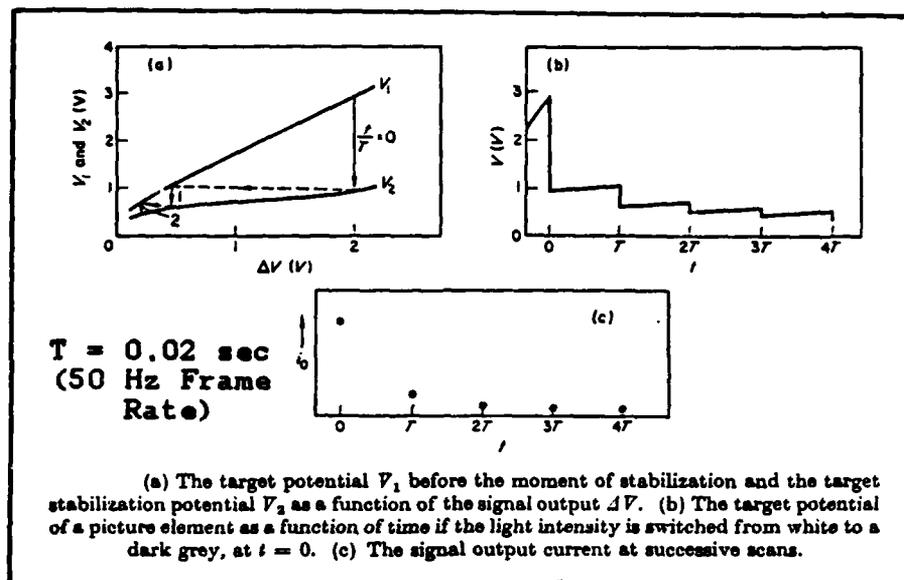
II.2.a Radar/Lidar Imaging. Various radar technologies, including standard wavelength (less than 35 GHz) and millimeter wavelength (40-300 GHz), did not meet the required spatial resolution required to individually track such a large number of closely co-located, high-speed fragments. This resolution shortfall is primarily due to oversized beamwidth caused by limited hardware capabilities. Laser radar can theoretically satisfy the spatial resolution requirement with its inherently narrow beamwidth; however, the required number of illumination lasers and their

associated pointing and scanning slew-rates, based on the 10,000 ft/sec tracking requirement, rendered such a system unrealistically complicated and expensive (7; 40:553-566). Overall, any sort of radar fails to meet the design criteria in either spatial resolution or dynamic tracking capability (bandwidth).

II.2.b Optical Imaging. The implementation of optical imaging type sensing systems can theoretically satisfy the necessary three dimensional spatial resolution and tracking requirements as long as the frame rate is fast enough to prevent any appreciable motion-induced smearing of the fragment image (3; 25:313-318). Such imaging devices may be classified into two general classifications: Photo-electronic image devices (camera tubes), and solid-state (chip) imaging devices.

II.2.b.1 Electronic Image Devices (Tubes). The most common photo-electronic image device applied to instrumentation applications is the vidicon tube, although other tube type imagers exist such as the image isocon, image orthicon, plumbicon, and secondary emission conduction (SEC). All these camera tubes have existed for decades and basically operate similarly, in that an optical image is focused on a photosensitive layered photocathode (image target) which generates a corresponding video signal by scanning a relatively low current, low velocity electron beam across the

image target (3; 26; 27). The dominant drawbacks of any direct optical input camera tube used for recording high-speed events are beam-discharge and photoconductive lag of the target's photosensitive layer -- causing image smearing (36; 41:24-29). The more dominant beam-discharge lag is due to the incomplete neutralization of the entire image target surface by a single scan of the electron beam, as graphically presented in plot (c), of Figure 2.4 (36; 41). Photoconductive lag is defined where the layer's charge does not change instantaneously with a corresponding instantaneous change in light input, as shown graphically in plot (b), of Figure 2.4. The important facts to realize from these plots in Figure 2.4 are that they represent a 0.02 second frame period (50 frames/sec) and that the arena test application requires a much shorter frame period of 200 μ sec (5,000 frames/sec). Therefore, the slow, exponential decay rates displayed in this figure would dominate the camera tube's operation and render its output a uselessly smeared image in such a high frame-rate application. A commonly applied technique that is actually a spill-over from high-speed photography is to apply high-speed shuttering between the image and the target to "gate" the incoming light (12). Such a technique shortens the target exposure time: effectively "freezing" the moving image, with the only additional requirement being increased illumination levels to compensate for the shortened photo-optical integration period of the



(36:242)

Figure 2.4. Characteristic Plots of Beam-Discharge Lag

image target. Mechanical rotating disk type shutters can effectively shorten the target exposure time down to one millisecond (msec). Looking at the worse case tracking velocity requirement of 10,000 ft/sec combined with a 10 ft wide field of view (FOV), a 1.0 msec gate period would result in a 10 ft streak image of that fragment which is the full FOV. Therefore, the gate period must be shortened significantly for effective "freezing" of the 10,000 ft/sec fragment. Shortening the gate period to 2.0 μ sec results in a streak of 0.02 ft (≈ 0.25 inches), which is acceptable. This 2.0 μ sec gate period can be achieved by using electronically gated image tubes or stroboscopic illumination techniques (3; 12; 33). However, electronically gated image

tubes are undesirably bulky, expensive, and fragile. They also require awkward support circuits and voltages; an undesirable choice for the arena test environment.

Stroboscopic lighting techniques perfected for high-speed photography can be directly applied to electronic imaging. The only unusual requirement would be that the arena tests would have to be conducted at night since there must be no ambient light between strobe flashes (12:1; 34; 42).

Once the gate period problem is resolved and the exposure time is restricted to the desired 2.0 μsec , the induced image must be read off the image tube's target within the remaining 198 μsec before the next strobe flash.

Recalling the previous discussion regarding target lag, to read and neutralize the target and then to be ready for the next flash within 198 μsec is virtually impossible even with the fastest responding image tube (plumbicon) (12:5). An analogy to this dilemma would be taking a snap-shot with a standard film camera but not being able to remove the film before another exposure is taken on top of it.

This leaves the final conclusion that tube type imaging devices cannot satisfy the required 200 μsec frame rate. This is true even though their exposure gate period can be reduced to the desired 2.0 μsec period.

II.2.b.2 Solid-State Imagers. Solid-state imaging devices include charge-coupled devices (CCD's) and charge-injection devices (CID's) (4; 10; 30; 38). Sensitivity and

resolution characteristics have steadily improved over the years (30). Both solid-state technologies are similar in that the input image is focused on a silicon planar array (chip) of photosites. The photon bombardment on each of these photosites induces an accumulation of minority carriers proportional to the light intensity on each photosite. These accumulated charges are periodically transferred out of each photosite and eventually off the chip by appropriate logic switching signals. The off-loaded signals are then amplified and mixed with appropriate synchronization signals to produce a usable video signal.

Charge coupled devices differ from CIDs in the actual mechanism used to transfer the accumulated photocharges. In CCDs, the individual photocharges are transferred ("coupled") from one charge well to another towards the perimeter of the chip (similar to a "bucket brigade" of firemen passing buckets of water), where an output circuit then converts the charges to minute currents for off-chip processing (30). The CID differs in that the photocharges are directly transferred ("injected") to the chip's substrate for direct output (10). Both devices and their associated charge readout mechanisms have their tradeoffs in operational flexibility and performance, dependent on the proposed imaging application.

In a study to compare the relative spectral sensitivities and resolution characteristics of the vidicon

tube, CCD, and CID, the solid-state devices outperformed the vidicon in all categories except where the vidicon resolution (28.4 lines/mm) outperformed the CCD resolution (23.0 lines/mm). The CID performed better under low light conditions, while the CCD performed better with scene contrast values of 50 per cent or greater (29:44). The arena testing environment will unlikely be a low light classification, since additional illumination may be provided if warranted by the system design. This fact favors the CCD over the CID. Another important drawback with the CID is its larger inherent readout capacitance (10:192) which limits the device readout speed to well below the required 198 μ S specification. This effectively eliminates the CID as a candidate technology.

Current applications of CCDs are widespread, especially in military applications as surveillance, guidance, and target identification (6; 9; 28; 31). In addition, CCDs have a strong potential commercial market, which has motivated commercial research and development (38). Present CCD resolution densities of 1280 x 970 pixels (photosites) are available on the commercial market at production cost (2:122).

Using the CCD instead of an imaging tube leaves the same problems of requiring a 2.0 μ sec optical gating period and 198 μ sec to read out the acquired image from the chip before the next image must be taken. CCDs can and have been

coupled with electronically gated image intensifier tubes as standard image tubes have. However, as previously pointed out, gated image tubes are undesirable for the arena test application. Again, using stroboscopic lighting as the gating mechanism appears the most technically and economically feasible. Although the majority of CCD imager chips have been designed for slow (0.02 sec) frame readout, there have been chip architectures and techniques developed to allow full frame readout well within the 198 μ sec period (1). Thermocouple cooling of the CCD chip to approximately -40° C significantly improves the CCD's signal-to-noise ratio, thereby raising its optical sensitivity (4:142). Many of the above described characteristics of CCDs make them the desirable choice for the arena test application.

II.3 Final Sensor Selection

Radar, lidar, tube imaging, and solid-state imaging technologies were investigated as possible solutions to the problem of tracking 10,000 ft/sec fragments. After analyzing each technology, the following conclusions were made:

1. Use thermocouple cooled CCD cameras as the image sensing device operating at 5000 frames/sec.
2. Use stroboscopic illumination to facilitate 2.0 μ sec optical gating to "freeze" fragment motion.

II.4 Kalman Filters

The reader is assumed to have a thorough knowledge of Kalman filtering concepts. This section is only a review of

the basic concepts and equations. For a more in-depth review, see (33).

The original Kalman filter formulation can only be used for linear systems, while the arena test application of interest is partially nonlinear. The linear form has been revised through perturbation techniques to handle nonlinear systems. This revision brought about two formats: 1) the linearized Kalman filter and 2) the extended Kalman filter. The linearized filter assumes a nominal trajectory over the entire estimation space of interest, while the extended filter revises the trajectory declaration based on the latest state estimate. The linearized filter is simpler to implement, making it attractive for on-line computer application. However, its largest drawback is its tendency to break track if the measurements deviate significantly from the precalculated nominal. The extended filter overcomes this off-nominal sensitivity (within reasonable limits) at the cost of being more computationally burdensome. For the application at hand, on-line computational capability is not required. In addition, although a series of nominal trajectories for the arena test environment (i.e. there are more than one) may be pre-generated and stored, the large quantity of "typical nominals" would yield a linearized filter that would quickly surpass the computational load of the extended filter's implementation. The result is to apply the extended filter format including a linear dynamics,

single-input (gravity), and nonlinear measurement model as the following equation outline illustrates.

For a given linear stochastic single-input system, its dynamics model (in continuous form) may be defined as:

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{b}(t)u(t) + \underline{G}(t)\underline{w}(t) \quad (2.1)$$

where

$\underline{x}(t)$ = the n-dimensional vector of time-varying system state variables,

$\dot{\underline{x}}(t)$ = the time derivative of $\underline{x}(t)$,

$\underline{F}(t)$ = the n-by-n dimensional system dynamics or "plant" matrix,

$\underline{b}(t)$ = the n-dimensional control input gain vector,

$u(t)$ = the scalar control input (gravity),

$\underline{G}(t)$ = the n-by-s dimensional noise distribution and gain matrix,

$\underline{w}(t)$ = the s-dimensional vector of mutually independent white Gaussian noises of expected value:

$$E\{ \underline{w}(t) \} = \underline{0} \quad (\text{zero-mean}), \quad (2.2)$$

and of strength $\underline{Q}(t)$, where

$\underline{Q}(t)$ = the s-by-s dimensional $\underline{w}(t)$ strength matrix defined by:

$$E\{ \underline{w}(t) \underline{w}(t') \} = \underline{Q}(t) \delta(t - t'), \quad (2.3)$$

and

$E\{ \cdot \} =$ the expected value of $\{ \cdot \}$

$\delta(\cdot) =$ the delta function at time (\cdot) .

The previous system, Eq.(2.1), rewritten into discrete propagation equation form becomes:

$$\begin{aligned} \underline{x}(t_k^-) = & \underline{\Phi}(t_k, t_{k-1}) \underline{x}(t_{k-1}^+) + \left[\int_{t_{k-1}}^{t_k} \underline{\Phi}(t, \tau) \underline{b}(\tau) d\tau \right] u(t_{k-1}) \\ & + \underline{G}_d(t_{k-1}) \underline{w}_d(t_{k-1}) \end{aligned} \quad (2.4)$$

where $\underline{\Phi}(t_k, t_{k-1})$ is the state transition matrix associated with $\underline{F}(t)$: the solution to which is described by (32:40):

$$\begin{aligned} \dot{\underline{\Phi}}(t_k, t_{k-1}) &= \underline{F}(t) \underline{\Phi}(t_k, t_{k-1}) \\ \underline{\Phi}(t_{k-1}, t_{k-1}) &= \underline{I} \end{aligned}$$

In Eq.(2.4), $\underline{G}_d(\cdot)$ may be dropped by setting it equal to the identity matrix without loss of generality:

$$\underline{G}_d(\cdot) = \underline{I}$$

allowing Eq.(2.4) to simplify to a final state propagation equation as:

$$\underline{x}(t_k^-) = \underline{\Phi}(t_k, t_{k-1}) \underline{x}(t_{k-1}^+) + \underline{b}_d(t_{k-1}) u(t_{k-1}) + \underline{w}_d(t_{k-1}) \quad (2.5)$$

where

$\underline{w}_d(\cdot, \cdot)$ = discrete-time, zero-mean, white, Gaussian noise sequence independent of $\underline{x}(t_k)$, and has the following covariance kernel:

$$E(\underline{w}_d(t_k) \underline{w}_d^T(t_j)) = \begin{cases} \underline{Q}_d(t_k) & , \quad t_k = t_j \\ \underline{0} & , \quad t_k \neq t_j \end{cases} ; \text{ for } k \neq j$$

where

$$\mathbf{Q}_d(t_k) = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau) \mathbf{G}(\tau) \mathbf{Q}(\tau) \mathbf{G}^T(\tau) \mathbf{\Phi}^T(t_k, \tau) d\tau \quad (2.6)$$

The Kalman filter state estimate propagation equation based on this model is:

$$\hat{\mathbf{x}}(t_k^-) = \mathbf{\Phi}(t_k, t_{k-1}) \hat{\mathbf{x}}(t_{k-1}^+) + \mathbf{b}_d(t_{k-1}) u(t_{k-1}) \quad (2.7)$$

where

t_k^- denotes the propagated time just prior to update.
 t_{k-1}^+ denotes the post-update time of the previous recursion cycle.

Meanwhile, the discrete-time propagation form for the filter's error covariance becomes:

$$\begin{aligned} \mathbf{P}(t_k^-) &= \mathbf{\Phi}(t_k, t_{k-1}) \mathbf{P}(t_{k-1}^+) \mathbf{\Phi}^T(t_k, t_{k-1}) + \\ &+ \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau) \mathbf{G}(\tau) \mathbf{Q}(\tau) \mathbf{G}^T(\tau) \mathbf{\Phi}^T(t_k, \tau) d\tau \quad (2.8) \end{aligned}$$

or:

$$\mathbf{P}(t_k^-) = \mathbf{\Phi}(t_k, t_{k-1}) \mathbf{P}(t_{k-1}^+) \mathbf{\Phi}^T(t_k, t_{k-1}) + \mathbf{Q}_d(t_{k-1}) \quad (2.9)$$

For the stochastic nonlinear discrete measurement model:

$$\mathbf{z}(t_k) = \mathbf{h}[\mathbf{x}(t_k), t_k] + \mathbf{v}(t_k) \quad (2.10)$$

where

$\mathbf{h}[\cdot, \cdot]$ = an m -dimension vector of functions of $\mathbf{x}(t_k)$ and time,

and

$\mathbf{v}(\cdot, \cdot)$ = an m -dimensional vector of mutually independent, zero-mean, white Gaussian discrete noise sequences, independent of $\mathbf{x}(t_k)$ and $\mathbf{w}(\cdot, \cdot)$, that has the following covariance kernel:

$$E\{\underline{v}(t_k) \underline{v}^T(t_j)\} = \begin{cases} \underline{R}(t_k) & , t_k = t_j \\ \underline{0} & , t_k \neq t_j \end{cases} ; \text{ for } k \neq j \quad (2.11)$$

The time-variant state-space to measurement-space transformation matrix, $\underline{H}[\underline{x}(t_k), t_k]$, that is needed within the extended Kalman filter is defined as:

$$\underline{H}[\underline{x}(t_k^-), t_k] = \left. \frac{\partial \underline{h}[\underline{x}_n, t_k]}{\partial \underline{x}} \right|_{\underline{x}_n = \hat{\underline{x}}(t_k^-)} \quad (2.12)$$

The update equations for the residual, $\underline{r}(t_k)$, appear as:

$$\underline{r}(t_k) = \underline{z}_k - \underline{h}[\hat{\underline{x}}(t_k^-), t_k] \quad (2.13)$$

where

\underline{z}_k is the actual measurement data (a realization of Eq. (2.10))

$\underline{h}[\hat{\underline{x}}(t_k^-), t_k]$ is the best prediction of the measurement value before it arrives at sample time t_k .

The residual covariance, $\underline{S}(t_k)$; Kalman gain, $\underline{K}(t_k)$; updated state estimate vector, $\hat{\underline{x}}(t_k^+)$; and updated estimate covariance matrix, $\underline{P}(t_k^+)$ are now evaluated as:

$$\underline{S}(t_k) = \underline{H}[\hat{\underline{x}}(t_k^-), t_k] \underline{P}(t_k^-) \underline{H}^T[\hat{\underline{x}}(t_k^-), t_k] + \underline{R}(t_k) \quad (2.14)$$

$$\underline{K}(t_k) = \underline{P}(t_k^-) \underline{H}^T[\hat{\underline{x}}(t_k^-), t_k] \underline{S}^{-1}(t_k) \quad (2.15)$$

$$\hat{\underline{x}}(t_k^+) = \hat{\underline{x}}(t_k^-) + \underline{K}(t_k) \underline{r}(t_k) \quad (2.16)$$

$$\underline{P}(t_k^+) = \underline{P}(t_k^-) - \underline{K}(t_k) \underline{H}[\hat{\underline{x}}(t_k^-), t_k] \underline{P}(t_k^-) \quad (2.17)$$

where

- superscript denotes pre-update values

and

+ superscript denotes post-update values

Further details of the KF development including actual numerical assignment methods are given in Chapter III.

II.5 Joint Probability Data Association (JPDA) History

The JPDA algorithm is actually an outgrowth from the Probabilistic Data Association (PDA) algorithm developed by Bar-Shalom and Tse (5). The PDA and JPDA algorithms are intended for applications involving one or a combination of:

- 1) Multiple simultaneous measurements
- 2) Multiple targets (including false targets or "clutter")
- 3) Multiple sensors

Such situations define a track/data/measurement association problem that PDA and JPDA are intended to solve.

There are two ways to approach the track association problem, from a measurement oriented approach or a target oriented approach. The measurement oriented approach, represented in Figure 2.5, associates each measurement with all possible prior target tracks, new tracks, or clutter. From this association, a hypothesis tree is made to determine the most likely association of the new measurements with previously determined tracks. As the tracks are updated, appropriate weights are assigned to compute the conditional probabilities (37).

The target oriented approach, represented in Figure 2.6, uses the established target tracks to gain statistical information about the propagated expected fragment position.

This information is then combined "a posteriori" with the newly acquired measurement set, to form a set of joint probabilities for each measurement to each established track (5; 11; 24).

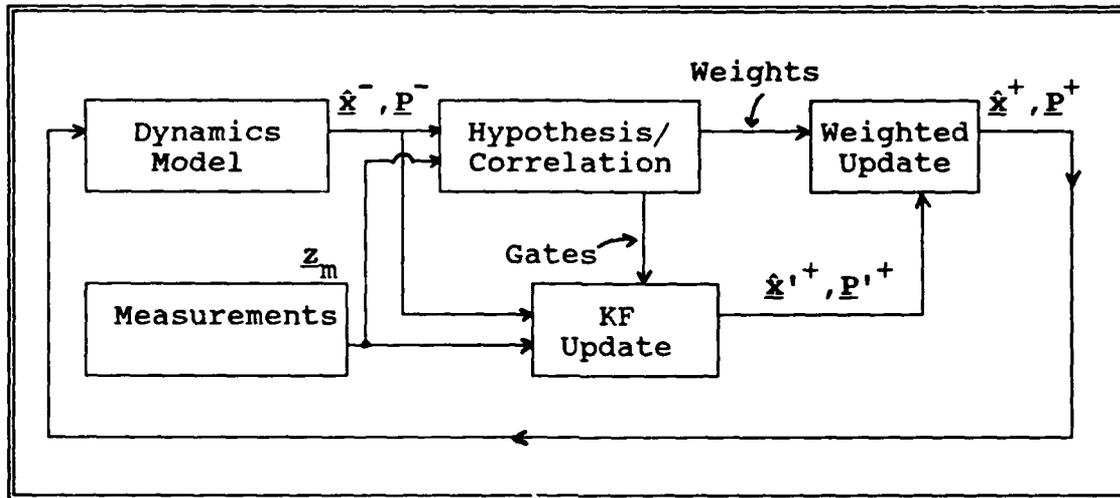


Figure 2.5. Target Weighted Tracking Method

(35)

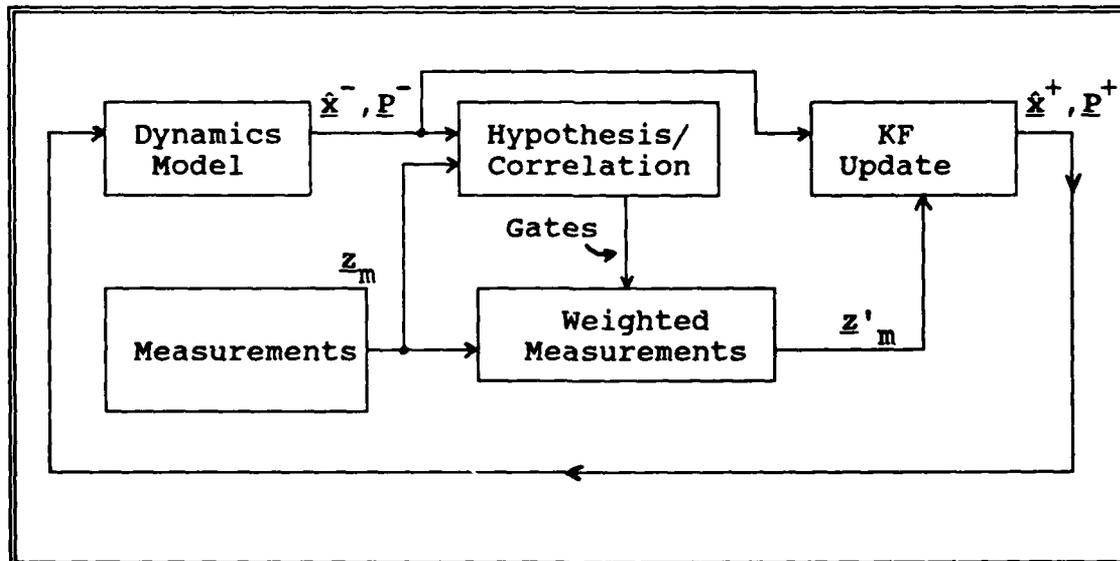


Figure 2.6. Measurement Weighted Tracking Method

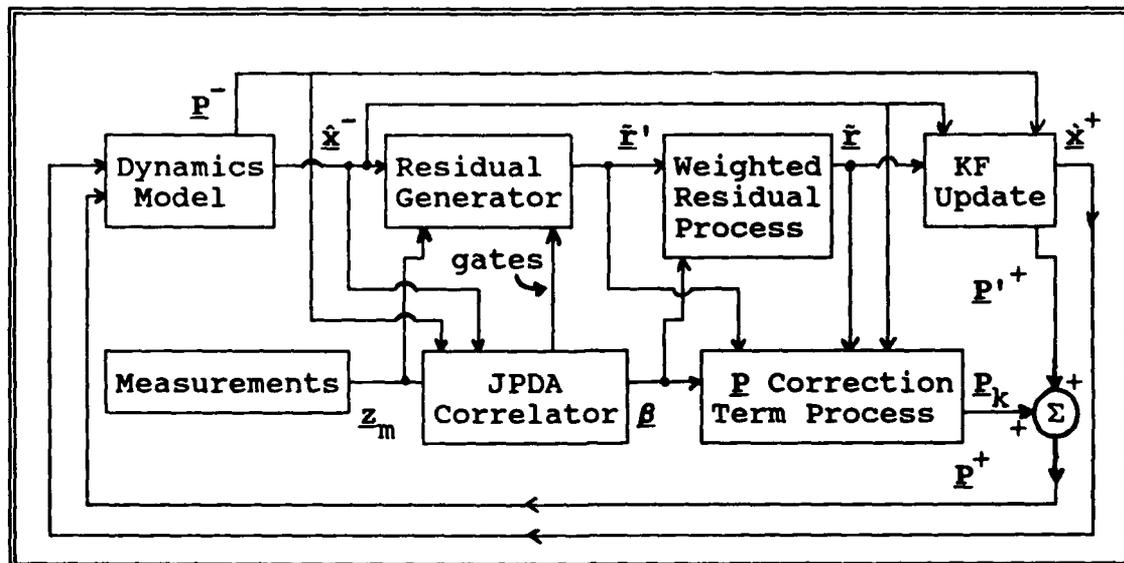
(35)

The previous articles explained various applications for measurement oriented and target oriented approaches. In addition, the work accomplished by Purvis (35), which used the target oriented approach, was studied for comparison to the arena test application. The arena test application, with its multiple target and potentially high clutter environment was compared to these published examples. The target oriented approach was deemed most applicable to the fragment tracking application since the literature indicated it to be better at rejecting clutter. Therefore, the target oriented JPDA algorithm was chosen for this design. The combination of work by Bar-Shalom, Chang, Fortmann, and Tse (5; 11; 24) set the foundation for the EKF/JPDA/RTS reduction package developed in this research.

II.6 Joint Probability Data Association (JPDA) Specifics

The originally devised PDA scheme is primarily intended to track a single target in a cluttered environment. Although PDA has been applied to tracking scattered multiple targets, its reliability becomes questionable for target clustering or for the worse case of track crossing (5). The original PDA discarded measurements as clutter if they did not associate with a target. Therefore the PDA scheme was modified to include probability terms for any measurement not associated with a certain target, then that measurement might be associated with a different target. The result is a set of joint probabilities across the entire measurement and track

event space including clutter. Referring to Figure 2.7, the resulting JPDA probabilities, β , are incorporated as weighting terms in the update process. This is accomplished by the generation of modified residuals, $\tilde{\mathbf{z}}$; residual covariances, \mathbf{S} ; and filter covariances, \mathbf{P} ; based on all feasibly validated permutations and combinations (events) of measurement to target/track association including clutter. The result is a weighted residual sum of all residuals that might correspond to each target. The residual covariance becomes the key player in measurement-to-target association. The larger the residual covariance for a particular measurement/target pair the lower its probability term and contribution to the weighted sum of residuals. This process will become clear in the following section that details the equation steps.



(35)

Figure 2.7. JPDA System Block Diagram

The implementation of the JPDA follows; a detailed derivation is found in (24). Key definitions include:

n = the number of targets

M = the number of measurements

m = the dimension of the measurement vector

i = the target index ($i = 0, 1, 2 \dots n$)

j = the measurement index ($j = 1, 2, 3 \dots M$)

X = index of each feasible event; defined as no more than one measurement originates from each target and each measurement has only one origin

$\tau_j(X)$ = measurement association indicator;

$$\tau_j(X) = \begin{cases} 1 & \text{if measurement } j \text{ is associated} \\ & \text{with any target for event } X \\ 0 & \text{otherwise} \end{cases}$$

$\delta_i(X)$ = target detection indicator;

$$\delta_i(X) = \begin{cases} 1 & \text{if any measurement is associated} \\ & \text{with target } i \text{ for event } X \\ 0 & \text{otherwise} \end{cases}$$

$\underline{\Omega}$ = validation matrix, made up of binary elements

w_{ji}

$$w_{ji} = \begin{cases} 1 & \text{if measurement } j \text{ lies within} \\ & \text{the validation gate (detailed} \\ & \text{below) for target } i. \\ 0 & \text{otherwise} \end{cases}$$

β_j^i = the probability that measurement j belongs to target i

$P(X|Z^k)$ = the probability that event X is true given all measurements Z where $Z = \underline{z}[t_0, t_k]$.

Note that the previous target index counter, i , begins at "0" to indicate the "no target" or clutter condition. The JPDA algorithm begins by generating the above defined $\underline{\Omega}$ validation matrix. At update time t_k , the $\underline{\Omega}$ is generated using a "g-sigma" ellipsoid test (24) beginning with the filter residual similar to Eq.(2.13):

$$\underline{\hat{r}}_j^i(t_k) = \underline{z}_j(t_k) - \underline{h}^i[\underline{\hat{x}}(t_k^-), t_k^-] \quad (2.17)$$

where in this case

$\underline{\hat{r}}_j^i(t_k)$ = the residual corresponding to the association of measurement j and the predicted measurement for target i at time t_k^- .

and the g-sigma test now equates as:

$$\left[\left[\underline{\hat{r}}_j^i(t_k) \right]^T \left[\underline{S}^i(t_k) \right]^{-1} \underline{\hat{r}}_j^i(t_k) \right] \leq g^2 \quad (2.18)$$

where

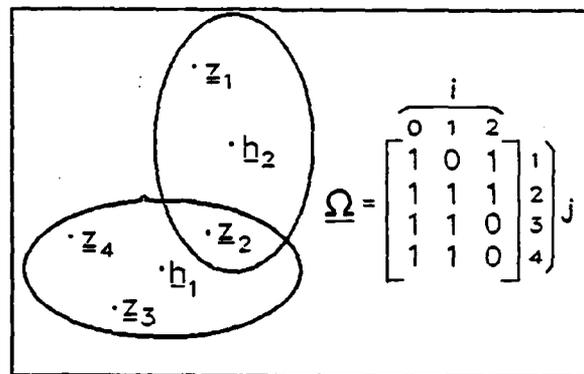
$\left[\underline{S}^i(t_k) \right]^{-1}$ = the inverse of the residual covariance matrix for target i as defined by eq.(2.13)

and

g = the m -dimensional validation gate volume defined to have a chi-squared probability distribution.

An example pair of two-dimension ($m = 2$) validation gates that would be generated by Eq.(2.18) and the corresponding $\underline{\Omega}$ matrix generated from the pictured case of four candidate measurements ($M = 4$), \underline{z}_1 through \underline{z}_4 and

two current targets ($n = 2$), h_1 and h_2 , appears in Figure 2.8. For each element ω_{ji} of Ω , the g -sigma test, Eq.(2.18) is performed to determine which measurement/target pairs are valid (set that ω_{ji} element equal to one if true; zero if false). A "good" measurement, j , will form a small residual when tested against the correct target, i ; producing a small product in Eq.(2.18) and "falling inside" the volume limit defined by g^2 . Also note in the pictured Ω validation matrix that the first column, $i = 0$, corresponds to each measurement, z_j not associated to any target, $h_{i>0}$ therefore making it clutter. This first column is always filled with ones since there is always some probability that each measurement may be clutter. Based on the previously stated definition of a feasible event,



(24:176)

Figure 2.8. Example Validation Gates with Associated Ω Validation Matrix

the Ω matrix must be broken down into individual $\hat{\Omega}(X)$ event matrices made up of elements $\hat{\omega}_{ji}(X)$. This event generation process is governed (24:176) by the following

rules:

- 1) Scan Ω by rows and pick one unit per row for $\hat{\Omega}(X)$ (i.e. there can be only one origin for a measurement).
- 2) Only one unit from each column $i \geq 1$ can be taken (i.e. at most one measurement could have originated from a target). The number of units from column $i = 0$ is not restricted because any measurement can be clutter.

Although at first glance the previous rules for generating the individual $\hat{\Omega}(X)$ matrices may appear trivial, much thought is required to develop a permutation/combination event algorithm. The devised algorithm must ensure all measurement-to-target permutations as well as all possible clutter combinations are obtained for any given Ω matrix without inadvertent repeats of events. Appendix B gives the mathematical formulation for determining the maximum number (worse-case) of possible events given M measurements, n targets, and a fully populated (no zero elements) Ω matrix.

For the previous example Ω matrix (Figure 2.8), if the Ω matrix were fully populated with ones, then Eq. (B.7) of Appendix B, indicates there would be 21 possible events. However, due to the validation gating process setting some Ω elements to zero, this number has been reduced to 12 possible $\hat{\Omega}(X)$ event matrices, thereby "trimming" the size of the hypothesis tree. The resulting twelve $\hat{\Omega}(X)$ event matrices defined by the above rules appear as:

$$\hat{\Omega}(1) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}(2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}(3) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\hat{\Omega}(4) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}(5) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}(6) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\hat{\Omega}(7) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \hat{\Omega}(8) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{\Omega}(9) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\hat{\Omega}(10) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \hat{\Omega}(11) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \hat{\Omega}(12) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

From each event matrix $\hat{\Omega}(X)$, the measurement association indicator, $\tau_j(X)$, is defined equal to one if any element (other than the first column) in measurement row j of $\hat{\Omega}(X)$ equals one; or it is set to zero otherwise. Likewise, the target detection indicator, $\delta_i(X)$, is set to one if any element of target column i (except for $i = 0$) of

$\hat{\Omega}(X)$ equals one; or it is set to zero otherwise.

The next step is to calculate the event probabilities, $P(X|Z^k)$, for each event where:

$$P(X|Z^k) = \frac{c^{\phi(X)}}{c} \prod_{j:\tau_j=1} \frac{\exp\left[-0.5 [\hat{x}_j^i(t_k)]^T [S^i(t_k)]^{-1} \hat{x}_j^i(t_k)\right]}{(2\pi)^{m/2} |S^i(t_k)|^{1/2}} \cdot \prod_{i:\delta_i=1} P_D^i \cdot \prod_{i:\delta_i=0} (1 - P_D^i) \quad (2.19)$$

where

P_D^i = the probability of detection for target i
(assumed equal for all targets in this study to simplify application),

C = the density of false measurements, assumed Poisson distributed by (24),

c = normalization constant,

$\phi(X)$ = the number of false measurements for event X found by:

$$\phi(X) = \sum_{j=1}^M [1 - \tau_j(X)] \quad (2.20)$$

Next calculate the association probabilities, β_j^i , for each event from:

$$\beta_j^i = \sum_X P(X|Z^k) \hat{w}_{ji}(X) \quad \begin{cases} j = 1, 2, 3, \dots, M \\ i = 0, 1, 2, \dots, n \end{cases} \quad (2.21)$$

while the associated probability that target i goes undetected, β_0^i , is:

$$\beta_0^i = 1 - \sum_{j=1}^M \beta_j^i \quad i = 0, 1, \dots, n \quad (2.22)$$

From the preceding terms the combined weighted residual is:

$$\hat{\mathbf{x}}^i(t_k) = \sum_{j=1}^M \beta_j^i(t_k) \hat{\mathbf{x}}_j^i(t_k) \quad (2.23)$$

which is then applied into target i 's account of Eq.(2.16) as the state estimate update. Meanwhile, for target i 's filter covariance update, Eq.(2.17) is modified to account for clutter and measurement origin uncertainty to re-equate as:

$$\mathbf{P}^i(t_k^+) = \mathbf{P}^i(t_k^-) - (1 - \beta_0^i) \mathbf{K}^i(t_k) \mathbf{S}^i(t_k) [\mathbf{K}^i(t_k)]^T + \mathbf{P}'^i(t_k) \quad (2.24)$$

where the superscript i indicates target i 's account of Eqs.(2.14) and (2.15). For the correction matrix, $\mathbf{P}'^i(t_k)$:

$$\mathbf{P}'^i(t_k) = \mathbf{K}^i(t_k) \left[\sum_{j=1}^M \beta_j^i(t_k) \hat{\mathbf{x}}_j^i(t_k) [\hat{\mathbf{x}}_j^i(t_k)]^T - \hat{\mathbf{x}}^i(t_k) [\hat{\mathbf{x}}^i(t_k)]^T \right] [\mathbf{K}^i(t_k)]^T \quad (2.25)$$

This concludes the JPDA structure.

II.7 Rauch-Tung-Striebel (RTS) Fixed-Interval Optimal Smoother

This section gives a brief summary of the Rauch-Tung-

Striebel (RTS) fixed-interval optimal smoother structure. For a detailed derivation and analysis refer to (33). The RTS smoother has a continuous-time and a discrete-time format. The latter of the two formats was the applicable choice for the presented EKF/JPDA/RTS tracker/smoothing system.

Computationally, the RTS algorithm entails performing the forward EKF computations and storing the $\hat{\mathbf{x}}(t_k^-)$, $\mathbf{P}(t_k^-)$, $\hat{\mathbf{x}}(t_k^+)$ and $\mathbf{P}(t_k^+)$ values for all time over the interval $[t_0, t_f]$.

The final forward-time updated estimate, $\hat{\mathbf{x}}(t_f^+)$ is applied as the starting boundary condition and the smoothed estimate is generated backward in time as (33):

$$\hat{\mathbf{x}}(t_f | t_f) = \hat{\mathbf{x}}(t_f^+) \quad (2.26)$$

$$\hat{\mathbf{x}}(t_k | t_f) = \hat{\mathbf{x}}(t_k^+) + \mathbf{A}(t_k) \left[\hat{\mathbf{x}}(t_{k+1} | t_f) - \hat{\mathbf{x}}(t_{k+1}^-) \right] \quad (2.27)$$

where the smoother gain matrix is given by:

$$\mathbf{A}(t_k) = \mathbf{P}(t_k^+) \Phi^T(t_{k+1}, t_k) \mathbf{P}^{-1}(t_{k+1}^-) \quad (2.28)$$

Note that in Eq.(2.28), the time indices for the $\Phi(\cdot, \cdot)$ matrix seem to indicate a forward-time transition matrix. However, because of the adjoint nature for this system, the transpose, $\Phi^T(\cdot, \cdot)$ reverses the propagation direction (8:276). Also note special care must be exercised in software implementations that $\mathbf{P}(\cdot)$ is checked to be non-zero before the matrix inversion is attempted.

For the reverse propagation covariance generation, the final time updated forward filter covariance, $\underline{P}(t_f^+)$, becomes the initial value so:

$$\underline{P}(t_f | t_f) = \underline{P}(t_f^+) \quad (2.29)$$

and

$$\underline{P}(t_k | t_f) = \underline{P}(t_k^+) + \underline{A}(t_k) \left[\underline{P}(t_{k+1} | t_f) - \underline{P}(t_{k+1}^-) \right] \underline{A}^T(t_k) \quad (2.30)$$

The implementation of the RTS smoother is quite simple because of its reverse-time recursive structure as the preceding equations indicate.

II.8 Summary

This chapter has detailed the arena test background and its current technologies. This was followed by a research review that compared various sensing technologies and resulted in the selection of the CCD camera for this application. The remainder of the chapter reviewed the mathematical structures of the extended Kalman filter, joint probabilistic data association algorithm, and Rauch-Tung-Striebel optimal smoother.

III. DEVELOPMENT

The design and validation of the developed system is determined by the fabrication of a computer simulation for dynamic "exercise" of the applied algorithm. Before the simulation can be accomplished, parameterized models had to be derived that would reasonably represent typical arena test profiles. In addition, the resulting profiles must remain within the limits of the simplifying assumptions in Chapter I. This chapter outlines the applicable reference frames and details the development of 1) the fragment trajectory truth model, and 2) both the EKF dynamics and measurement models.

The real-world applicability of the designed package can only be judged against the fidelity of the truth model that is used as the real-world reference. However, no all-inclusive closed form solution for hypersonic projectiles passing through a fluid could be found other than the standard zero-lift drag equation:

$$D(t) = 0.5 C_D S \rho_a |\underline{v}(t)|^2 \quad (3.1)$$

where

$D(t)$ = the force due to aerodynamic drag (lbf)

C_D = the coefficient of drag for the fragment (unitless)

S = the cross-sectional area of the fragment (ft^2)

Note: $S = 0.25 \pi d^2$ where d = the spherical fragment diameter (ft)

ρ_a = the atmospheric density (lbm/ft^3)

$|\underline{v}(t)|$ = the magnitude of the relative airspeed velocity vector (ft/sec)

t = time (seconds)

This equation is selected as the applied truth model for the fragments.

III.1 Reference Frames

A geometry model consisting of four reference frames, associated coordinate axes, and transformation equations are devised for the arena to facilitate mathematical tractability in the algorithm. For the proposed arena set-up, each camera pair defines three of its own reference frames, a two-dimensional (2D) image frame for each camera, and a three-dimensional (3D) intermediate frame formed at the intersection of the two image frames. The collective set of all 3D intermediate frames are then transformed into the world coordinate frame located at the arena center. A complete description of the arena geometric model is contained in Appendix A.

This study only models one camera pair and therefore includes the following four reference frames:

- 1) Two camera image plane frames, one for each camera as (x_1, y_1) and (x_2, y_2) .
- 2) The intermediate coordinate frame, (X_I, Y_I, Z_I) , located at the orthogonal intersection of the two camera image planes.
- 3) The world coordinate frame, (X_W, Y_W, Z_W) , located at the center of the arena.

The EKF/JPDA/RTS package operates in the intermediate reference frame to reduce computational loading, and the resulting estimates would normally be transformed to the world coordinate frame. To minimize excessive transformations between the world and intermediate frame, this study performed all error analysis in the intermediate frame. Figures A.3, A.4, and A.5 in Appendix A, illustrate the above reference frames.

III.2 Fragment Dynamics

III.2.a Truth Model. Based on the simplifying assumptions outlined in the previous chapter, a free-body representation of a typical fragment, including the forces acting upon it as it travels through the atmosphere in free-flight, appears as:

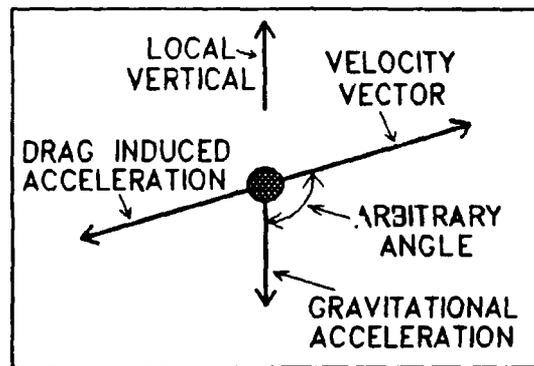


Figure 3.1. Free-body Diagram of Typical Fragment with Arbitrarily Pointed Velocity Vector.

The magnitude of the force applied to the fragment due to aerodynamic drag, Eq.(3.1), is repeated as:

$$D(t) = 0.5 C_D S \rho_a |\underline{v}(t)|^2$$

From this drag force, the associated drag acceleration upon the fragment is found by dividing the force $D(t)$ by the fragment mass. For the case of uniform density fragments (as assumed) the overall mass is expressed as a function of diameter as:

$$m_f = \rho_f \pi d^3 / 6 \quad (3.2)$$

where

m_f = the fragment mass (lbm)

ρ_f = the fragment material density (lbm/ft³)

d = the fragment diameter (ft)

Equations (3.1) and (3.2) may now be combined to form an equation giving the drag-induced acceleration magnitude $a_D(t)$ (ft/sec²) as a function of velocity magnitude (speed), diameter and time.

$$a_D(t) = \frac{-3 \rho_a C_D |\underline{v}(t)|^2}{4 \rho_f d} \quad (3.3)$$

The negative sign indicates deceleration. For notational convenience, we can define

$$K_1 = \frac{-3 \rho_a C_D}{4 \rho_f} \quad (3.3a)$$

yielding Eq.(3.3) more simply as

$$a_D(t) = \frac{K_1 |\underline{v}(t)|^2}{d} \quad (3.3b)$$

From the above nonlinear, speed dependent, fragment

acceleration model, fragment trajectories are generated starting with an initial world frame position $\mathbf{p}(t_0)$ and velocity $\mathbf{v}(t_0)$. This is accomplished by the standard two integration steps of acceleration and velocity to generate new acceleration, velocity, and position vector values as functions of time:

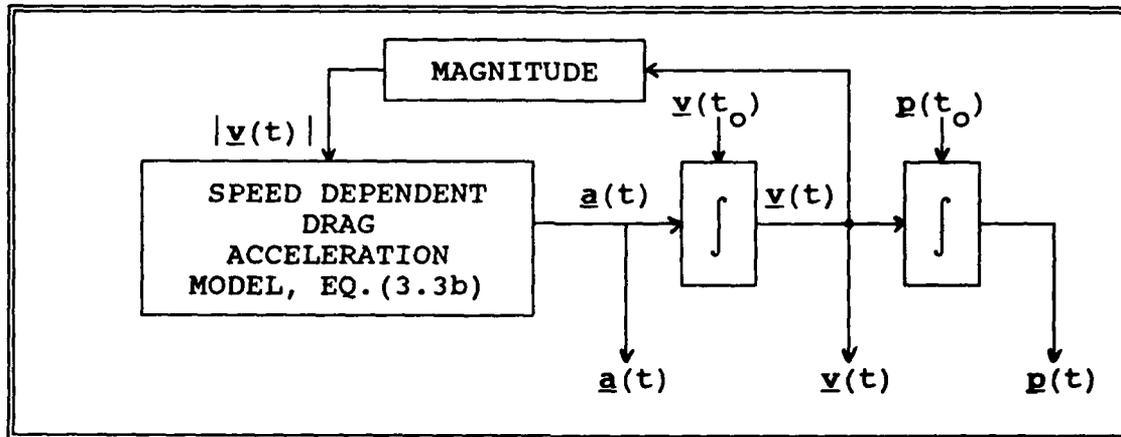


Figure 3.2. Fragment Trajectory State Generation.

III.3 Filter Dynamics Model

The dynamic model must basically follow a free-falling mass (fragment) in the intermediate reference frame where the high initial velocity and its associated drag acceleration clearly dominates any gravitational acceleration component. The EKF must estimate three states, position, velocity and acceleration in each of the X_I , Y_I , and Z_I directions (total of nine states) for the fragment of interest. The filter structure begins as three separate, three-state filters augmented together to form a single nine-state EKF.

Originally, three separate, uncoupled, three-state filters were designed, but had to be coupled together as one large filter because of the JPDA algorithm requires the existence on one residual covariance matrix. The Singer dynamics model (23; 39) is selected because of its good performance in tracking radar algorithms demonstrated in (23) for tracking mild-maneuvering targets. Because the fragments are assumed to be following a nearly straight-line trajectory, they may be classified as mild-maneuvering targets. Such an assumption fits well with the first-order Markov acceleration model included in the Singer approach. The important fact to keep in mind during the following outline of the Singer model is that all the necessary numbers will be provided by the initialization routine discussed later in this chapter. The dynamics equations following Eq.(2.1) with $\underline{G}(t_k) = \underline{I}$ are:

$$\begin{bmatrix} \dot{p}_x \\ \dot{v}_x \\ \dot{a}_x \\ \dots \\ p_y \\ \dot{v}_y \\ \dot{a}_y \\ \dots \\ p_z \\ \dot{v}_z \\ \dot{a}_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \underline{0} & \cdot & \underline{0} \\ 0 & 0 & -1/\tau_x & \cdot & \cdot & \cdot & \cdot \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 & \cdot \\ \underline{0} & \cdot & \cdot & \cdot & 0 & 0 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & 0 & -1/\tau_y & \cdot \\ \dots & \dots \\ \underline{0} & \cdot & \cdot & \cdot & \underline{0} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & -1/\tau_z \end{bmatrix} \cdot \begin{bmatrix} p_x \\ v_x \\ a_x \\ \dots \\ p_y \\ v_y \\ a_y \\ \dots \\ p_z \\ v_z \\ a_z \end{bmatrix} +$$

$$\begin{matrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 1 \\
 0
 \end{matrix}
 \cdot u(t_k) +
 \begin{matrix}
 0 \\
 0 \\
 w_x \\
 0 \\
 0 \\
 w_y \\
 0 \\
 0 \\
 w_z
 \end{matrix}
 \quad (3.4)$$

where

$$u(t_k) = -32.1741 \text{ ft/sec}^2 \text{ (gravity)} \\
 \text{(constant for all } t_k \text{)}$$

τ_x , τ_y , and τ_z = the first-order acceleration time constants, set different for each respective direction as determined by the initialization procedure to yet to be discussed

w_x , w_y and w_z = mutually independent, zero-mean, white, Gaussian noise, each with a covariance kernel as defined in Eq.(2.6) and detailed below in Eqs.(3.8) and (3.9).

Converting to the discrete-time format as outlined in Eq.(2.6) results in the following (nine-by-nine) forward-time state transition matrix (13):

$$\underline{\Phi}(T) = \begin{bmatrix}
 \underline{\Phi}_x & \cdot & \underline{0} & \cdot & \underline{0} \\
 \dots & \dots & \dots & \dots & \dots \\
 \underline{0} & \cdot & \underline{\Phi}_y & \cdot & \underline{0} \\
 \dots & \dots & \dots & \dots & \dots \\
 \underline{0} & \cdot & \underline{0} & \cdot & \underline{\Phi}_z \\
 \cdot & \cdot & \cdot & \cdot & \cdot
 \end{bmatrix} \quad (3.5)$$

where $\underline{\Phi}(T)$ is parameterized by τ_x , τ_y , and τ_z as:

$$\underline{\Phi}_i = \begin{bmatrix} 1 & T & \tau_i^2 [T/\tau_i - 1 + \exp(-T/\tau_i)] \\ 0 & 1 & \tau_i [1 - \exp(-T/\tau_i)] \\ 0 & 0 & \exp(-T/\tau_i) \end{bmatrix} \quad (3.6)$$

where

$i = x, y, \text{ or } z \text{ and}$

$$T = t_k - t_{k-1} \quad (3.7)$$

$= 200.0 \mu\text{Sec (constant sample period)}$

For the discrete $\underline{b}_d(t_k)$ input gain vector, all elements are zero except for elements b_7 and b_8 which are $T^2/2$ and T respectively. This $\underline{b}_d(t_k)$ is constant for all t_k .

The covariance matrix or noise strength $\underline{Q}_d(t_k)$ for $\underline{w}_d(t_k)$ is block diagonal symmetric (nine-by-nine) for the Singer approach and defined as:

$$\underline{Q}_d(t_k) = \begin{bmatrix} \underline{Q}_{dx} & \cdot & \underline{0} & \cdot & \underline{0} \\ \dots & \dots & \dots & \dots & \dots \\ \underline{0} & \cdot & \underline{Q}_{dy} & \cdot & \underline{0} \\ \dots & \dots & \dots & \dots & \dots \\ \underline{0} & \cdot & \underline{0} & \cdot & \underline{Q}_{dz} \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (3.8)$$

$$\underline{Q}_{di} = \frac{2 \sigma_{ai}^2}{\tau_i} \cdot \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \quad (3.9)$$

where

$i = x, y, \text{ or } z$

$\sigma_{ax}^2, \sigma_{ay}^2$ and σ_{az}^2 = the variances or mean-squared values of the fragment accelerations for each respective direction determined normally by "a priori" knowledge or tuning.

The initial filter estimate covariance matrix $\underline{P}(t_0)$ for the Singer approach is also block diagonal symmetric (nine-by-nine) defined as:

$$\underline{P}(t_0) = \begin{bmatrix} \underline{P}_x & \vdots & \underline{0} & \vdots & \underline{0} \\ \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \\ \underline{0} & \vdots & \underline{P}_y & \vdots & \underline{0} \\ \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \\ \underline{0} & \vdots & \underline{0} & \vdots & \underline{P}_z \end{bmatrix} \quad (3.10)$$

where

$$\underline{P}_i = \begin{bmatrix} P_{1,1} & P_{1,2} & 0 \\ P_{2,1} & P_{2,2} & P_{2,3} \\ 0 & P_{3,2} & P_{3,3} \end{bmatrix} \quad (3.11)$$

and each indicated element of Eq.(3.11) is expressed as:

$$P_{1,1} = \sigma_{mm}^2(t_0) \quad (3.12)$$

$$P_{1,2} = P_{2,1} = \sigma_{mm}^2(t_0) / T \quad (3.13)$$

$$P_{2,2} = \frac{2 \sigma_{mm}^2(t_0)}{T^2} + \frac{\sigma_{ai}^2(t_0) r_i^4}{T^2} \left[2 - \frac{T^2}{r_i^2} + \frac{2 T^3}{3 r_i^3} - 2 \text{ETERM} - \frac{2 T}{r_i} \text{ETERM} \right] \quad (3.14)$$

$$P_{2,3} = P_{3,2} = \frac{\sigma_{ai}^2(t_0) r_i^2}{T} \left[\text{ETERM} + \frac{T}{r_i} - 1 \right] \quad (3.15)$$

$$P_{3,3} = \sigma_{ai}^2(t_0) \quad (3.16)$$

where

$$i = x, y, \text{ or } z$$

$\sigma_{mm}^2(t_0)$ = the position measurement corruption noise variance at time t_0 (assumed constant for all t_k)

$$ETERM = \exp (-T / \tau_i) \quad (3.17)$$

This concludes the development of the dynamics portion of the EKF structure and associated parameter definitions. Later in this chapter the method for initializing these parameters with actual numerical values are discussed.

III.4 Measurement Model

III.4.a General Definition. A nonlinear measurement model, Eq.(2.10), results from the nonlinear transform equations that convert the noise-corrupted, orthogonal, 2D camera image plane coordinates to the 3D intermediate reference frame as detailed in Appendix A.

For this study, each camera's object plane size was set to a 10.0 x 10.0 ft square and 512 x 512 pixel resolution level. These numbers were selected purely because they were realistic and nice to work with. These object plane dimensions result in ≈ 0.02 ft per side for each pixel or ≈ 0.0004 ft² per pixel in area. From this resolution definition, the measurement noise is set at a constant three-sigma level of \pm one pixel to give a one-sigma covariance:

$$\sigma_{mm}^2(t_k) = (0.02 / 3.0)^2 = 4.444 \times 10^{-4} \text{ ft}^2 \quad (3.18)$$

Since this design involves two cameras that each give a 2D position pair, (x_1, y_1) and (x_2, y_2) , the 4D measurement vector, $\underline{z}(t_k)$, is now defined as simply augmenting the two camera output pairs as:

$$\underline{z}^T(t_k) = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 \end{bmatrix} \quad (3.19)$$

From Eqs.(A-5) through (A-8) of Appendix A, the resulting 4D vector of functions $\underline{h}[\hat{x}(t_k^-), t_k]$ becomes:

$$\underline{h}[\cdot, \cdot] = \begin{bmatrix} \frac{2 D \hat{x}_1 (R \hat{x}_4 + D)}{2 (\hat{x}_4 \hat{x}_7 + \hat{x}_4^2 + D^2) + R D (3 \hat{x}_4 + \hat{x}_7)} \\ \frac{R D (\hat{x}_7 - \hat{x}_4)}{R (\hat{x}_7 + \hat{x}_4) + 2 D} \\ \left[\frac{R D \hat{x}_1 (\hat{x}_7 + \hat{x}_4)}{R (\hat{x}_7 - \hat{x}_4) + 2 D} \right] - D \hat{x}_1 \\ \frac{D - R \hat{x}_4}{R (\hat{x}_7 - \hat{x}_4) + 2 D} \\ \frac{R D (\hat{x}_7 + \hat{x}_4)}{R (\hat{x}_7 - \hat{x}_4) + 2 D} \end{bmatrix} \quad (3.20)$$

where

$$R = (2.0)^{1/2}$$

D = the camera focal length defined as the distance from the intermediate frame origin to each camera lens (set to $20 \times 2^{1/2}$ per Section I.1.b.1).

This leaves the four-by-nine matrix $\underline{H}[\hat{x}(t_k^-), t_k]$ and each of its elements defined by Eq.(2.12) to be:

$$\underline{H}[\cdot, \cdot] = \begin{bmatrix} H_{1,1} & 0 & 0 & H_{1,4} & 0 & 0 & H_{1,7} & 0 & 0 \\ 0 & 0 & 0 & H_{2,4} & 0 & 0 & H_{2,7} & 0 & 0 \\ H_{3,1} & 0 & 0 & H_{3,4} & 0 & 0 & H_{3,7} & 0 & 0 \\ 0 & 0 & 0 & H_{4,4} & 0 & 0 & H_{4,7} & 0 & 0 \end{bmatrix} \quad (3.21)$$

where, in terms of the following three denominator terms:

$$\text{DENA} = 2 (\hat{x}_4 \hat{x}_7 + \hat{x}_4^2 + D^2) + R D (3 \hat{x}_4 + \hat{x}_7) \quad (3.22)$$

$$\text{DENB} = R (\hat{x}_7 + \hat{x}_4) + 2 D \quad (3.23)$$

$$\text{DENC} = R (\hat{x}_7 - \hat{x}_4) + 2 D \quad (3.24)$$

the entries of Eq.(3.21) may be evaluated as:

$$H_{1,1} = \frac{2 D (R \hat{x}_4 + D)}{\text{DENA}} \quad (3.25)$$

$$H_{3,1} = \frac{\left[\frac{D R (\hat{x}_7 + \hat{x}_4)}{R (\hat{x}_7 - \hat{x}_4) + 2 D} - D \right]}{D - R \hat{x}_4} \quad (3.26)$$

$$H_{1,4} = \left[\frac{2 D R \hat{x}_1}{\text{DENA}} - \frac{2 D \hat{x}_1 (R \hat{x}_4 + D)}{\text{DENA}^2} \right]$$

$$\cdot \left[\frac{(2 \hat{x}_7 + 4 \hat{x}_4 + 3 D R)}{1} \right] \quad (3.27)$$

$$H_{2,4} = -D \cdot \left[\frac{R}{DENB} + \frac{2 (\hat{x}_7 - \hat{x}_4)}{DENB^2} \right] \quad (3.28)$$

$$H_{3,4} = \left[\frac{\left[\frac{2 D \hat{x}_1 (\hat{x}_7 + \hat{x}_4)}{DENC} - D R \hat{x}_1 \right]}{(D - R \hat{x}_4)^2} + \frac{\left[\frac{D R \hat{x}_1}{DENC} + \frac{2 D \hat{x}_1 (\hat{x}_7 + \hat{x}_4)}{DENC^2} \right]}{D - R \hat{x}_4} \right] \quad (3.29)$$

$$H_{4,4} = \frac{D R}{DENC} + \frac{2 D (\hat{x}_7 + \hat{x}_4)}{DENC^2} \quad (3.30)$$

$$H_{1,7} = \frac{-2 D \hat{x}_1 (2 \hat{x}_4 + D R) (R \hat{x}_4 + D)}{DENA^2} \quad (3.31)$$

$$H_{2,7} = \frac{D R}{DENB} - \frac{2 D (\hat{x}_7 - \hat{x}_4)}{DENB^2} \quad (3.32)$$

$$H_{3,7} = \frac{\left[\frac{D R \hat{x}_1}{DENC} - \frac{2 D \hat{x}_1 (\hat{x}_7 + \hat{x}_4)}{DENC^2} \right]}{D - R \hat{x}_4} \quad (3.33)$$

$$H_{4,7} = \frac{D R}{DENC} - \frac{2 D (\hat{x}_7 + \hat{x}_4)}{DENC^2} \quad (3.34)$$

The measurement corruption noise strength matrix, $\underline{R}(t_k)$ has no off-diagonal terms since each measurement direction is mutually decoupled by the orthogonal measurement geometry. Eqs.(2.11) and (3.18) then define $\underline{R}(t_k)$ as:

$$\underline{R}(t_k) = \begin{bmatrix} 4.44 \times 10^{-4} & 0 & 0 & 0 \\ 0 & 4.44 \times 10^{-4} & 0 & 0 \\ 0 & 0 & 4.44 \times 10^{-4} & 0 \\ 0 & 0 & 0 & 4.44 \times 10^{-4} \end{bmatrix} \quad (3.35)$$

where $\underline{R}(t_k)$ has units of ft^2 and is modelled as constant for all t_k .

III.4.b Measurement Ambiguity.

Two CCD cameras used to collect separate orthogonal 2D position measurements may seem, at first thought, to be angle-only position measurements with no range measurement information. However, that is not the case. The 3D position result from the measurement geometry, Eqs.(A-2) through (A-4)

of Appendix A, does include the necessary range information for discrimination between multiple fragment positions. This is true as long as the measurements of the two cameras are properly correlated. Therefore, the problem exists as an observability ambiguity for multiple measurement combinations, such that the EKF/JPDA/RTS package does not know "a priori" which fragment in camera #1 correlates to which fragment in camera #2. Once this ambiguity is overcome, full 3D observability exists. This ambiguity is now illustrated for no clutter, perfect detection situations:

Let: fragment_i represent a measurement coordinate pair (x,y) .

Case 1: Simple one fragment camera images:

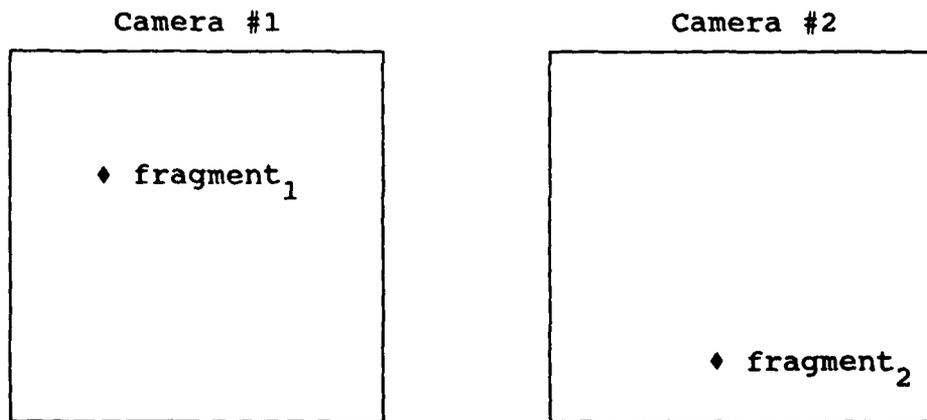


Figure 3.3. One Fragment Image Example

For the above images from cameras #1 and #2 there is no confusion that fragment_1 and fragment_2 are the same fragment (assuming perfect detection and no clutter). The resulting augmented measurement as defined by Eq.(3.19) becomes:

$$\underline{z}_1 = [(\text{fragment}_1), (\text{fragment}_2)] \quad (3.36)$$

Case 2: Given the following multiple fragment camera images:

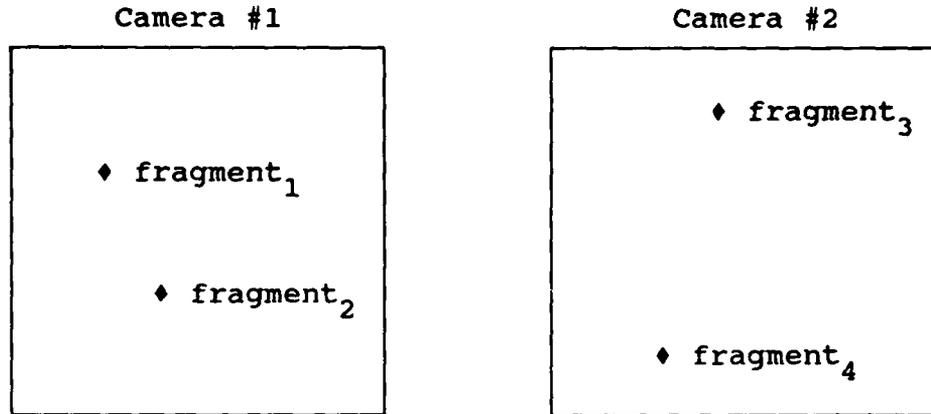


Figure 3.4. Multiple Fragment Image Example

From Figure 3.4, all that can be determined immediately is that there are two fragments. There is no information available, for example, which of the two measurements in camera #2 matches to fragment₁ in camera #1. The result is two possible measurement combinations:

$$\text{or } \underline{z}_1 = [(\text{fragment}_1), (\text{fragment}_3)] \quad (3.37)$$

$$\underline{z}_2 = [(\text{fragment}_1), (\text{fragment}_4)] \quad (3.38)$$

The same problem occurs for the second measurement of camera #1, giving:

$$\text{or } \underline{z}_3 = [(\text{fragment}_2), (\text{fragment}_3)] \quad (3.39)$$

$$\underline{z}_4 = [(\text{fragment}_2), (\text{fragment}_4)] \quad (3.40)$$

The final result is that given p camera #1 measurements and q camera #2 measurements, where $p = q$, there are $p \cdot q$

candidate measurements for a perfect detection/no clutter assumption. However, such an assumption is unrealistic for the arena test environment; therefore, p might not equal q . For such cases, p and q may each be greater than or less than the true number of fragments. This further increases the total possible number of measurement permutations and combinations. For a full description of this event algebra, see Appendix B.

From the previous illustration, and the derivation of Appendix B, one quickly sees that the number of possible candidate fragment measurements grows exponentially as the number of true or clutter fragment measurements increase. The selection of the correct measurement combinations from the candidate set gives the desired 3D position information for each fragment. Fortunately, the JPDA algorithm eliminates or reduces the effect of the probabilistically unlikely candidate measurements primarily through the validation gate process (as in Figure 2.8) and secondarily through the joint probability residual weighting. Figure 3.5 gives a general system block and signal-flow diagram of the arena test EKF/JPDA/RTS processing system.

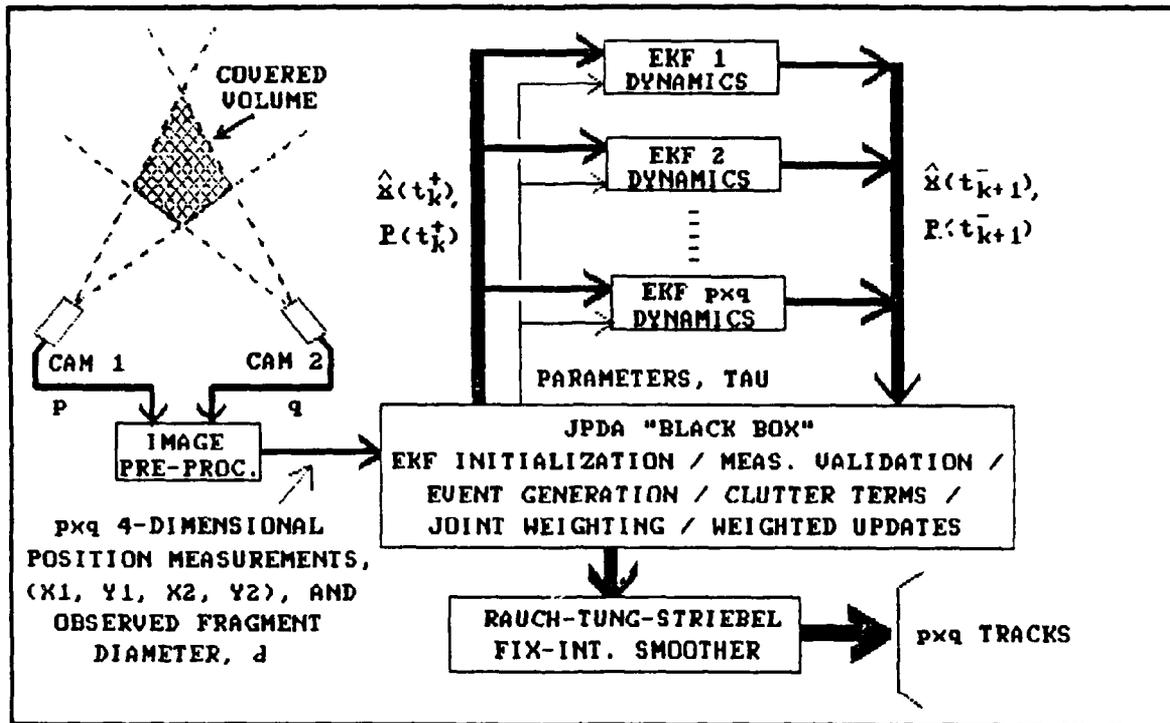


Figure 3.5. Full System Block and Signal-Flow Diagram.

III.5 Filter Initialization Procedure

The transient performance of the EKF during acquisition is largely determined by the initial estimates of state values, $\hat{x}(t_0)$; initial covariance matrix values, $P(t_0)$; dynamics driving noise, $Q_d(t_0)$; and the dynamics model parameters for the acceleration time constants (τ_x , τ_y and τ_z). The closer these initial values are to the true values, the smaller the transient and associated tracking errors. The following sections present two possible methods to obtain the above initialization values and parameters, 1) by statistical data method, and 2) by exploiting "a priori" arena geometry information. The latter of the two methods being the choice for implementation.

III.5.a Statistical Initialization Approach. The fragment cloud that disperses from a detonated munition can be reasonably assumed to include fragments of various sizes (cross sectional area) and various speeds (velocity magnitude). One approach for selecting initial values is to compute the statistically averaged values for velocities and acceleration time constants (ATCs) from previous test data and selecting abnormally high covariance and dynamics noise values. The high $\underline{P}(t_0)$ and $\underline{Q}_d(t_0)$ values initially increase the Kalman gain such that the increased filter bandwidth quickly (hopefully) dampens out any transients.

However, such a method requires that previous test data exist. In addition, because of the large range of velocities and acceleration time constants that may occur in one test-blast, many of the acquired fragments will have true state values far from the expected averages. This results in large start-up transients or worse--loss of track.

For this research effort, an alternate initialization method is developed and gives excellent results. The deterministic arena set-up geometry with its known dimensions provides significant "a priori" information that the initialization process may exploit.

III.5.b "A Priori" Information. The deterministic geometry model may be exploited by one obvious assumption:

Since the test munition is detonated at the W-frame origin, there is a high probability that any fragment that enters a camera's field-of-view (FOV) initiated its trajectory from the W-frame origin at time t_0 .

Recall in Chapter II that first light sensors detect the breakup of the case of the test munition to start the system clock at t_0 . As a fragment enters a camera's FOV, the fragment's measured diameter, position and the time elapsed since t_0 can be used to calculate that fragment's initial position, velocity, drag induced acceleration, and acceleration time constants. From these values, r_x , r_y , r_z , $\hat{x}(t_0)$, $\underline{p}(t_0)$, and $\underline{Q}(t_0)$ may be generated to initialize the EKFs. This process will now be developed.

The initialization procedure begins by transforming the initial measured position, $\underline{z}(t_m)$, into the W-frame (Appendix A) to become $\underline{p}(t_m)$. The average speed over the time interval, $v_{avg}[t_0, t_m]$, is simply the magnitude of the W-frame position vector, $|\underline{p}(t_m)|$, divided by the elapsed time, $t_m - t_0$.

Begin by setting:

$$\begin{aligned} \underline{p}(t_0) &= \underline{0} \quad (\text{at W-frame origin}) \\ t_0 &= 0 \quad (\text{time of detonation}) \\ t_m &= \text{the time of first measurement} \end{aligned}$$

where

$$|\underline{p}(t_m)| = [p_x^2(t_m) + p_y^2(t_m) + p_z^2(t_m)]^{1/2} \quad (3.41)$$

and

$$v_{avg}[t_o, t_m] = \frac{|p(t_m)|}{t_m} \quad (3.42)$$

To avoid requiring three separate sets of calculations for each axis direction, the following procedures are carried out once with the scalar magnitudes from above. The obtained scalar results will then be projected back into the appropriate axis components. From Eq.(3.4), the deterministic portion of the acceleration model, in scalar form, may be extracted as:

$$\dot{a}(t) = -1/\tau a(t) \quad (3.43)$$

and taking the time-derivative of Eq.(3.3b) for a scalar case where $|\underline{v}(t)|$ is replaced with $v(t)$ gives:

$$\dot{a}_D(t) = \frac{2 K_1 v(t) \dot{v}(t)}{d} \quad (3.44)$$

where

$$\dot{v}(t) = a_D(t) \quad (3.45)$$

Substituting Eqs.(3.45) and (3.44), along with $v_{avg}[t_o, t_m]$ and $d(t_m)$ in place of $v(t)$ and d respectively, into Eq.(3.43) and solving for τ as τ_{avg} gives:

$$\tau_{avg} = \frac{d(t_m)}{2 K_1 v_{avg}[t_o, t_m]} \quad (3.46)$$

Note in Eq.(3.46), that τ_{avg} is dependent upon the measured fragment diameter, $d(t_m)$, and $v_{avg}[t_o, t_m]$. Under the

assumptions that τ_{avg} for a fragment remains near constant and that gravitational acceleration is negligible for all time of interest, this τ_{avg} parameter is now placed into a scalar continuous-time form.

Note that vector notation will be temporarily dropped, where now indicated position, velocity, and acceleration values are scalar magnitudes of the previous vectors:

$$p(t) = a_D(t_0) [-\tau_{avg}^2 + t \tau_{avg} + \tau_{avg}^2 \exp(-t/\tau_{avg})] + t v(t_0) + p(t_0) \quad (3.47)$$

$$v(t) = a_D(t_0) \tau_{avg} [1 - \exp(-t/\tau_{avg})] + v(t_0) \quad (3.48)$$

$$a_D(t) = a_D(t_0) \exp(-t/\tau_{avg}) \quad (3.49)$$

The above equations are all dependant upon $a_D(t_0)$, which may be found through Eq.(3.3b) as:

$$a_D(t_0) = \frac{K_1 v(t_0)^2}{d} \quad (3.50)$$

In this equation, fragment diameter is constant so $d(t_m) = d(t_0) = d$, and $v(t_0)$ is the solution of:

$$p(t_m) = \frac{K_1 [t_m \tau_{avg} - \tau_{avg}^2 + \tau_{avg}^2 \exp(-t_m/\tau_{avg})] v(t_0)^2}{d} + t_m v(t_0) \quad (3.51)$$

where

$$A = \frac{K_1 [t_m \tau_{avg} - \tau_{avg}^2 + \tau_{avg}^2 \exp(-t_m/\tau_{avg})]}{d} \quad (3.52)$$

$$B = t_m \quad (3.53)$$

$$C = -p(t_m) \quad (3.54)$$

and now solving the quadratic below gives $v(t_o)$:

$$v(t_o) = \frac{-B + (B^2 - 4AC)^{1/2}}{2A} \quad (3.55)$$

Note that only the positive root solution for Eq.(3.55) is applicable since scalar magnitudes are involved.

Once $v(t_o)$ and $a_D(t_o)$ have been obtained by solving Eq.(3.55) and substituting back into Eq.(3.50), $v(t_m)$ and $a_D(t_m)$ may be found through Eqs.(3.48) and (3.49) evaluated at t_m :

$$v(t_m) = a_D(t_o) \tau_{avg} [1 - \exp(-t_m / \tau_{avg})] + v(t_o) \quad (3.56)$$

and

$$a_D(t_m) = a_D(t_o) \exp(-t_m / \tau_{avg}) \quad (3.57)$$

The scalar magnitude values of τ_{avg} , $v(t_m)$, and $a_D(t_m)$ are then converted into appropriate X_W , Y_W , and Z_W component vectors by transformation terms, XTRM, YTRM, and ZTRM solved by trigonometric projections to give

$$\tau_x = XTRM \tau_{avg} ; v_x(t_m) = XTRM v(t_m) ; a_{Dx}(t_m) = XTRM a_D(t_m)$$

$$\tau_y = YTRM \tau_{avg} ; v_y(t_m) = YTRM v(t_m) ; a_{Dy}(t_m) = YTRM a_D(t_m)$$

$$\tau_z = ZTRM \tau_{avg} ; v_z(t_m) = ZTRM v(t_m) ; a_{Dz}(t_m) = ZTRM a_D(t_m)$$

where the projection terms onto the X_W , Y_W , and Z_W axes are:

$$XTRM = \cos(\text{ang}_1) \sin(\text{ang}_2) \quad (3.58)$$

$$YTRM = \sin(\text{ang}_1) \sin(\text{ang}_2) \quad (3.59)$$

$$ZTRM = p_z(t_m) / |p(t_m)| \quad (3.60)$$

and

$$\text{ang}_1 = \text{TAN}^{-1}(p_y(t_m) / p_x(t_m)) \quad (3.61)$$

$$\text{ang}_2 = \text{COS}^{-1}[p_z(t_m) / |p(t_m)|] \quad (3.62)$$

Up to this point, all accelerations have been due to drag, so now a gravitational acceleration component is included by summing $-32.1741 \text{ ft/sec}^2$ to the W-frame acceleration vector's W_2 direction. The final set of required transformations takes the derived initial conditions in the W-frame and converts them to the I-frame (Appendix A) where the EKF operates.

To complete the EKF initialization process, the above calculated $a_{Dx}(t_m)$, $a_{Dy}(t_m)$, and $a_{Dz}(t_m)$ values are squared and inserted as the mean-squared accelerations σ_{ax}^2 , σ_{ay}^2 , and σ_{az}^2 into Eqs.(3.9) through (3.16) to generate $\underline{P}(t_0)$ and $\underline{Q}_d(t_0)$, where $\underline{Q}_d(\cdot)$ remains constant with time. Initial EKF test runs indicate not to rely directly on the values of $a_{Dx}(t_m)$, $a_{Dy}(t_m)$, and $a_{Dz}(t_m)$ to set the $\underline{Q}_d(t_0)$. This is because the above acceleration drag terms, $a_D(\cdot)$, may calculate close to zero: resulting in near-zero $\underline{Q}_d(\cdot)$ values. This allows the EKF's gain matrix to decrease too rapidly and possibly breaking track from the true trajectory.

The solution to this initialization shortfall is to incorporate a minimum-level drag acceleration value, a_{Dmin} , that was set (by trial-and-error tuning) to one third the value of $a_{Dx}(t_m)$ since the X_I direction is expected to have the highest $a_D(\cdot)$ component. This minimum-level value is used to calculate $Q_d(t_0)$ and $P(t_0)$ as a default value should the above calculated $a_D(\cdot)$ values fall below the a_{Dmin} threshold.

This completes the entire EKF initialization process for $\hat{x}(t_0)$, $P(t_0)$, and $Q(t_0)$ required by the Singer approach.

III.6 Summary

This chapter has introduced the applicable system reference frames, and given detailed equation structures for the applied truth model, linear EKF Singer dynamics model, and nonlinear EKF measurement model. In addition, a detailed break-down of the applied EKF initialization procedure, based on known arena geometry and the initial fragment measurement, was given.

IV. SIMULATION

This chapter presents the structure and implementation of the fragment tracking digital computer simulation developed in this study. The simulation was written entirely in double precision FORTRAN 77 and is self-supporting (i.e. no IMSL or library system calls) on most large mainframes. The sole exception is the use of MATRIX_x (32) to generate plots. Overall, the simulation is divided into two independent main programs, 1) a fragment trajectory generator (truth model) and 2) a self-contained EKF/JPDA/RTS tracker/smoothing program. The trajectory generator program creates data-files for the tracking program to use as "measurements" for its operation.

IV.1 Fragment Trajectory Generator

The fragment trajectory generator, is a fully self-contained program that calculates the time-varying acceleration, velocity and position of up to six fragments simultaneously from a set of initial conditions. Four "trackable" fragments, numbered one through four, and two "clutter" fragments, numbered five and six may be selectively generated. The resulting data is stored in a formatted datafile to provide "measurements" for subsequent processing by the tracker/smoothing program as well as an easy to read "proof" file that lists each fragment's time varying position, velocity, and acceleration state values in the W-frame, I-frame, and C-frame.

The fragment trajectory dynamics are based on the fragment acceleration truth model, Eqs.(3.1) through (3.3b) presented in Chapter III. Fragment trajectories are modelled in accordance with Appendix A and are initiated from the arena center (W-frame origin), $\mathbf{p}(t_0) = \mathbf{0}$, and an initial velocity vector, $\mathbf{v}(t_0)$. The acceleration, $\mathbf{a}[\mathbf{v}(t), \mathbf{g}]$, on each fragment is defined as the velocity magnitude (speed) dependent atmospheric drag component (initialized with $\mathbf{v}(t_0)$) summed with a constant gravitational component, \mathbf{g} . Using the two integration steps, as illustrated in Figure 3.2, each fragment's acceleration vector, is recursively integrated over one frame period (200 μ sec) to produce the velocity vector, $\mathbf{v}(k+1)$. This new velocity is used to both calculate the new acceleration, $\mathbf{a}[\mathbf{v}(k+1), \mathbf{g}]$, and is integrated to produce the position vector, $\mathbf{p}(k+1)$.

Integration is accomplished by the trapezoidal technique where each integration period (one frame period) is divided into 1000 trapezoidal summations. The 1000 summations quantity was selected assuming the near straight-line trajectory would be adequately smooth for this resolution.

Constant values programmed into the fragment generator program include:

Atmospheric density: $\rho_a = 0.076474 \text{ lbm} / \text{ft}^3$ (sea level)

Fragment density: $\rho_f = 491.0 \text{ lbm} / \text{ft}^3$ (low-carbon steel)

Fragment diameter: $d = 0.5$ inches (0.0416667 ft)

Coefficient of drag: $C_D = 1.0$ (as assumed in Chapter 1)

Camera to W-frame origin vertical displacement: 5.0 ft

The following default values may be interactively changed by the user:

Arena center to camera location radius: $R_d = 50.0$ ft

Camera focal length: $D = 20 \cdot (2.0)^{1/2} = 28.28427712474$ ft

I-frame origin angular displacement from X_w : $\theta = 30.0^\circ$

Camera angular field-of-view (FOV): $\alpha = 20.0499757242^\circ$
(This gave the "nice" 10.0 ft image plane size below.)

The default parameters result in the following dimensions:

W-frame origin to I-frame origin vertical displacement:
 $\delta z = 15.0$ ft

Camera object plane size: $Size = 10.0$ ft

The initial fragment speed (velocity magnitude) is interactively prompted and entered by the user when the program is started. Each fragment's initial direction is pre-determined by a set of polar-coordinate angles such that the trackable fragments, one through four, pass through the cameras' FOV intersection (coverage volume). Clutter fragment number five passes through camera #2's "near-field" FOV while clutter fragment number six passes through camera #1's "far-field" FOV. These result in fragment five being only visible to camera #2 and fragment six only visible to camera #1; creating an extra measurement in each camera that cannot be tracked; or clutter.

An additional user selectable feature is the option to

add "curving" accelerations to trackable fragments two, three, and four. These curving accelerations cause fragments two and three to have effectively "crossing tracks," while the curving acceleration applied to fragment four causes it to curve downward sharply ($-Z_W$ and $-Z_I$ directions).

The multitude of fragment trajectory combinations possible for this program to simulate allows thorough testing of the EKF/JPDA/RTS package from "easy" single-fragment straight trajectories, to relatively "difficult" multi-fragment, multi-clutter, crossing tracks, and curving trajectory scenarios.

IV.2 EKF/JPDA/RTS Tracker/Smother Program

The EKF/JPDA/RTS program reads the datafile generated by the fragment generator program exactly as though the file were pre-processed image measurements. These measurements include for each camera: the number of fragments present in the image, the centroid (x,y) coordinates for each observed fragment, and the diameter of each observed fragment. These truth model measurements are provided in both the four-dimensional measurement space as $\underline{z}(t_k)$ and as three-dimensional position, velocity, and acceleration state values, $\underline{x}(t_k)$, in the I-frame for subsequent error and statistical analysis of the algorithm's performance.

IV.2.a Measurement Corruption Noise Technique. The truth model measurements (fragment position and diameter) are

corrupted by the summation of a pseudo-random number noise sequence to the uncorrupted measurement vector (see Figure 4.1). The unity strength noise generator shown in Figure 4.1 includes a pseudo-random number generator from which twelve independent outputs are summed, followed by subtracting six from this accumulated sum for each element of a four-dimensional noise vector. This results in a 6D (4D position + 2 diameter measurements) approximate zero-mean Gaussian distributed noise vector of covariance equal to a six-by-six identity matrix. The overall output variance of this noise vector is adjusted by multiplying each covariance term by the desired standard deviation, σ_{mm} , then summed to the 4D truth

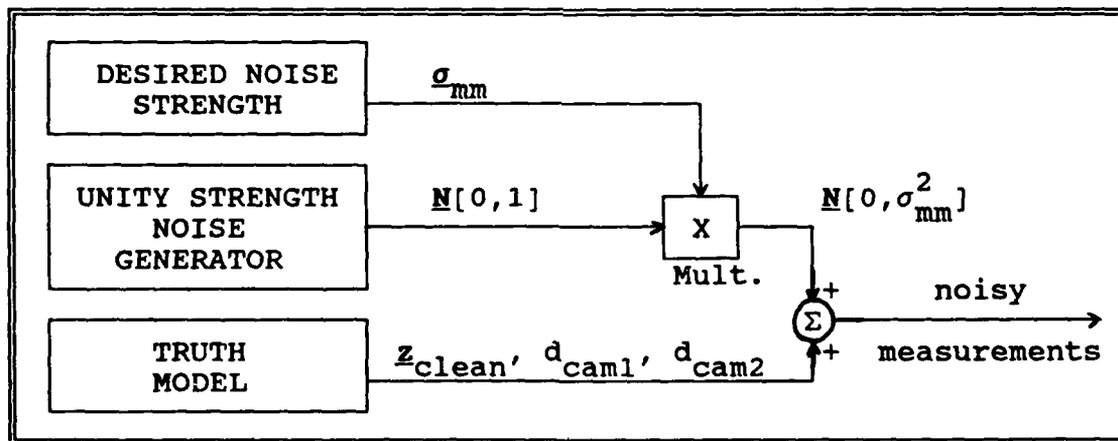


Figure 4.1. Measurement Noise Corruption Process

measurement vector, z_{clean} , and fragment diameters, d_{cam1} and d_{cam2} to give a noise-corrupted measurement vector and diameter measurements for processing by the EKF/JPDA/RTS tracker/smoothing package.

IV.2.b Tracker/Smoother Specifics. This program allows the user to adjust the following parameters at the beginning of each simulation run:

The truth model fragment diameter measurement noise variance.

The truth model fragment position measurement noise variance.

The JPDA algorithm validation gate size: g , of Eq.(2.18).

The JPDA algorithm fragment probability of detection: P_D , of Eq.(2.20) (all fragments are assumed to have equal P_D).

The JPDA algorithm Poisson clutter density: C , of Eq.(2.20).

For all runs presented in this study, both of the above truth model noise levels were set equivalent to the filter's expected measurement noise levels of one pixel or $4.444 \times 10^{-4} \text{ ft}^2$. This noise level was selected simply as an apparently logical place to start any analysis. The selection of the validation gate size was rather arbitrary, using trial and error to arrive at reasonable values. Values of $g = 3.0$ gave good tracks for straight fragment trajectories while $g = 4.0$ became necessary to maintain reliable tracking for curving trajectories. Since the validation gating process takes place in the 4D measurement space of units ft^4 , the volume represented by g is a 4D "statistical" volume that cannot be easily interpreted into 3D real-world units of ft^3 .

The probability of fragment detection, P_D , was arbitrarily set at 0.95, and the Poisson clutter density, C , was set to unity for all presented simulation runs. These

values were chosen as "nice" values work with and due to lack of fragment data to select otherwise.

Similar to the fragment generation program, this program contains the arena dimension and camera location constants necessary to define the measurement geometry and perform the required transformations between any of the reference frames. These dimensions are especially important in the EKF initialization process, since this process relies heavily on arena dimensional constants in its calculations.

Once all necessary optional inputs are supplied to the tracker/smoothing program, it processes the data as outlined in Chapters II and III. The resulting fragment state estimates for both the EKF and the RTS smoother outputs are compared to the uncorrupted truth model states to generate the time-varying estimation errors, $\underline{x}(\cdot) - \hat{\underline{x}}(\cdot)$; filter computed standard deviation values, σ_f ; and true error standard deviation values, σ_{tr} ; for position, velocity, and acceleration in the X_I , Y_I , and Z_I directions. These statistical values are then formatted and written out into two separate MATRIX_x loadable datafiles for plotting. The first datafile contains the EKF performance data and the second contains the RTS smoother performance data. A representative example plot is shown in Figure 4.2. The solid center line is the estimate error, e ; the top solid and bottom dotted lines that are symmetric about zero are the filter's standard deviation, $\pm \sigma_f$; and the top dotted line

and bottom dashed line symmetric about the estimate error line describe the true one-sigma standard deviation bound of the estimate error, $e \pm \sigma_{tr}$; where each of these values are calculated in the vector sense by:

$$\underline{e}(\cdot) = \frac{1}{n} \sum_{i=1}^n \left[\underline{x}_i(\cdot) - \hat{\underline{x}}_i(\cdot) \right] \quad (4.1)$$

$$\underline{\sigma}_f(\cdot) = \frac{1}{n} \sum_{i=1}^n \left[\underline{P}_i(\cdot) \text{diag} \right]^{1/2} \quad (4.2)$$

$$\underline{\sigma}_{tr}(\cdot) = \left[\frac{1}{(n-1)} \sum_{i=1}^n \left[\underline{x}_i(\cdot) - \hat{\underline{x}}_i(\cdot) \right]^2 \right]^{1/2} \quad (4.3)$$

where

$\underline{x}(\cdot)$ = true states

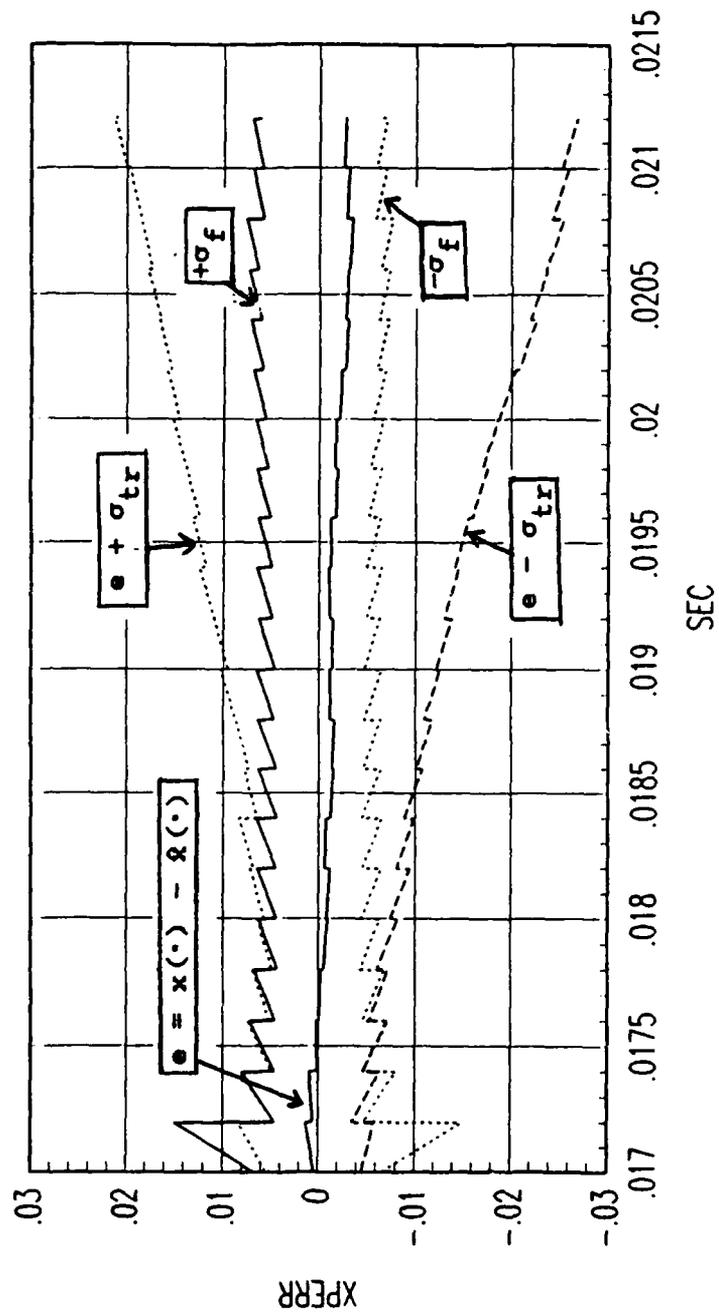
$\hat{\underline{x}}(\cdot)$ = estimate states

$\underline{P}(\cdot) \text{diag}$ = diagonal elements of the filter estimate covariance matrix

n = number of Monte-Carlo runs

IV.3 Summary

This chapter has presented the characteristic descriptions of the FORTRAN fragment generation and tracker/smoothing simulation programs developed in this study. This was followed by a detailed description of the system's applied noise corruption technique. The chapter was completed with a description of the simulation generated information displayed in its output plots; including an example plot.



EXAMPLE OUTPUT PLOT OF EKf (RTS SMOOTHER IS SIMILAR)

Figure 4.2. Example Performance Plot

V. ANALYSIS

VI.1 Simulation Runs

Three major trajectory categories were run against the presented EKF/JPDA/RTS package by Monte-Carlo simulation:

- Category 1.) Single fragment: straight trajectory with no clutter.
- Category 2.) Multiple simultaneous fragments (three): each with straight trajectory and no clutter.
- Category 3.) Multiple simultaneous fragments (five): one with straight trajectory, two with curving trajectories that cross, and two extraneous clutter fragments.

Each of the above trajectory categories were run at three initial fragment speeds, $|\underline{v}(t_0)|$, of 3000, 6000, and 10000 ft/sec (total of nine simulation runs). These runs produced performance plots for each modelled fragment's EKF and RTS smoother tracking errors in position, velocity, and acceleration for each of the X_I , Y_I , and Z_I directions (18 plots per fragment). A total of 324 simulation performance plots were generated from these runs and are displayed in Appendix C.

The desired number of Monte-Carlo runs for each simulation was initially set to 100; however, computational run-time constraints required this number be reduced to 20 runs per simulation. This degraded the overall "smoothness" of the plots but still allowed the important performance trends to stand out. Example plots of 20 versus 100 Monte-

Carlo run simulation comparisons are shown in Figures 5.1 for the EKF output and Figure 5.2 for the RTS smoother output.

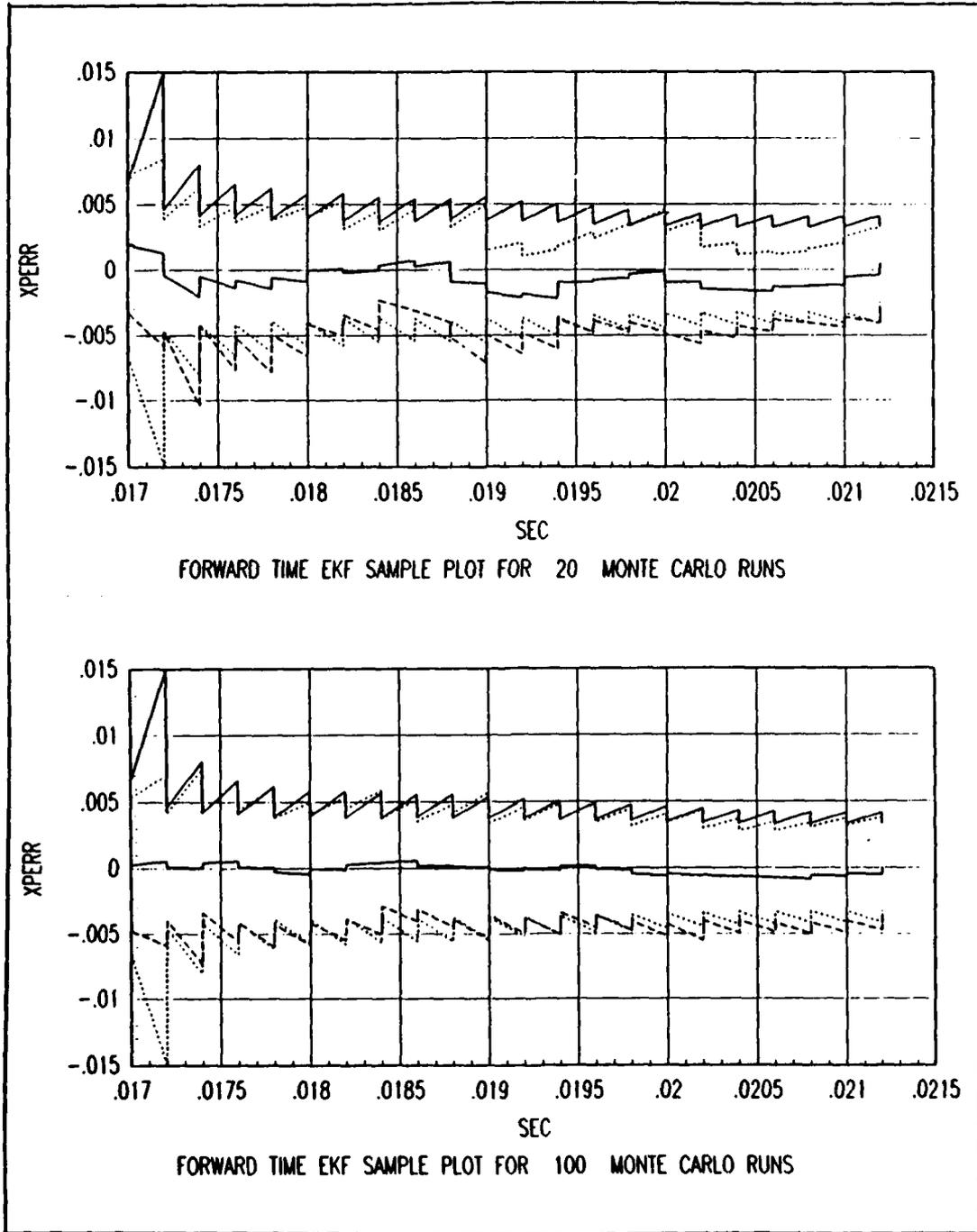


Figure 5.1. EKF Plots for 20 and 100 Monte-Carlo Runs.

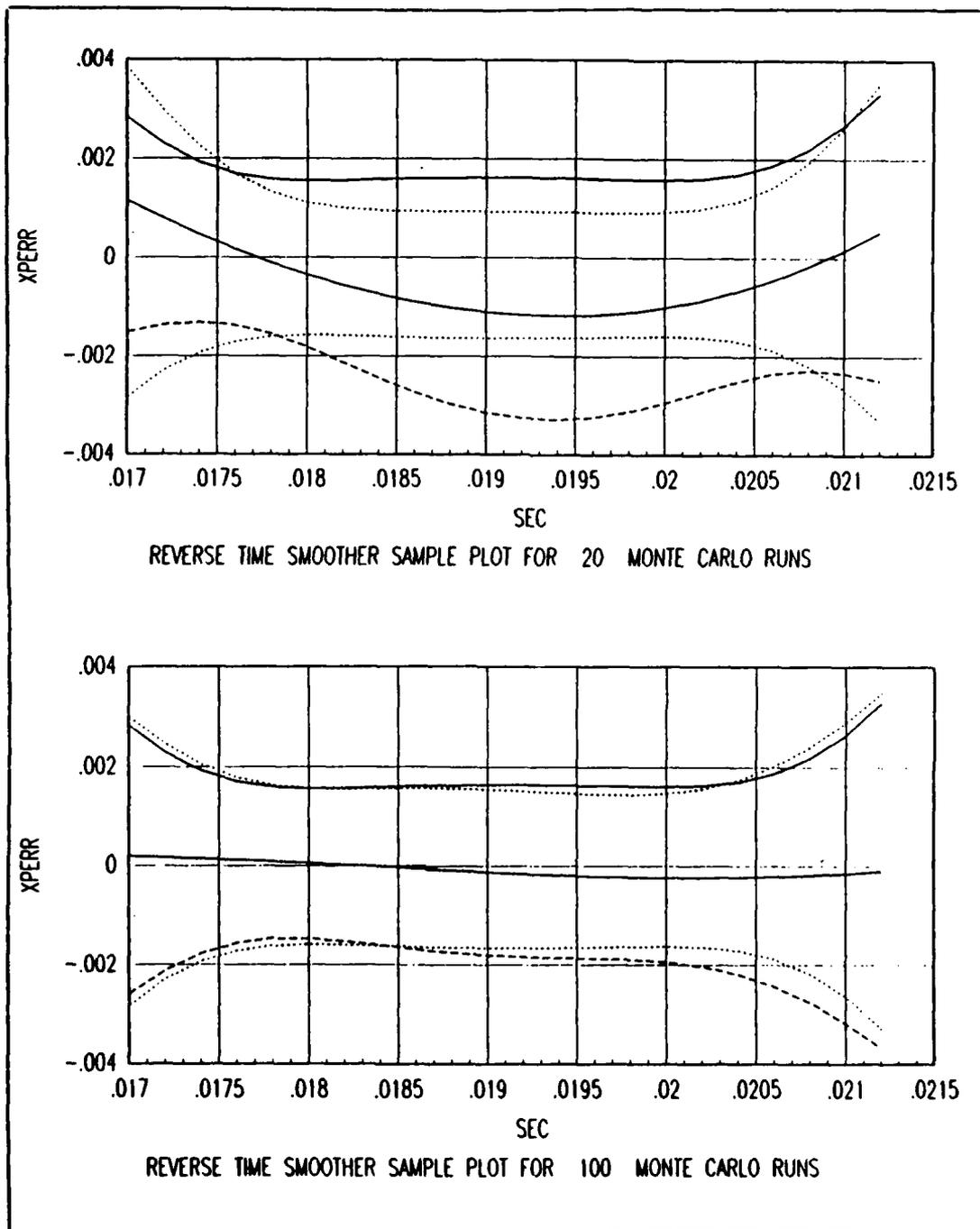


Figure 5.2. RTS Plots for 20 and 100 Monte-Carlo Runs.

The previous example plots serve to emphasize that the presented simulation results (in plots and tables) would, in general, show better performance if the 100 runs could have been applied in all cases.

V.2 Performance Summaries

V.2.a Category One. Position and velocity tracking performances were excellent. This was attributed largely to the "expected straight path" assumption imposed by the filter initialization routine and the single fragment, with its sole measurement, allowing the JPDA algorithm to remain dormant. The obtained position estimation accuracies, even with a three-sigma measurement noise level of \pm one pixel, often exceeded the one pixel resolution level (≈ 0.02 ft) of the CCD cameras by an order of magnitude. This "better than expected" performance raised suspicion that a coding error may exist in the pseudo-random noise generator, causing lower measurement noise corruption levels than desired. Therefore, additional coding was added to calculate and plot the true standard deviations for each of the following:

- 1.) Each noise corrupted element of the 4D position measurement vector, $\underline{z}(\cdot)$.
- 2.) Each camera's noise corrupted diameter measurement, d_{cam1} and d_{cam2} .
- 3.) The equivalent 3D noisy position components of the state vector, $\underline{x}(\cdot)$, resulting from transforming the 4D $\underline{z}(\cdot)$ vector to the $\underline{x}(\cdot)$ space.

An additional simulation run was made for a 3000

ft/sec, category one fragment, with the position and measurement noise levels set for a three-sigma standard deviation level of one-pixel (0.02 ft) as it was in all previous simulation runs. This defined the expected one-sigma value to be $0.02 / 3 \approx 0.0067$ ft. In addition, 1000 Monte-Carlo runs were chosen to ensure statistically steady-state results due to the 1000 "sample" size. The top plot of Figure 5.3 shows the resulting standard deviations for each element of the noise corrupted $\underline{z}(\cdot)$ vector, while the bottom plot shows that for each camera's noise corrupted diameter measurement. The plotted levels of $\sigma \approx 0.0067$ ft validate the measurement noise corruption process and further validate the category 1 and 2 results. However, an interesting system characteristic was discovered when plotting the standard deviations of the equivalent noises into the $\underline{x}(\cdot)$ space, as shown in Figure 5.4. This plot indicates that even though each direction of the 4D $\underline{z}(\cdot)$ vector have equal noise levels (as in Figure 5.3), the nonlinear transformation into the 3D position components of $\underline{x}(\cdot)$ results in unequal noise levels for the X_I , Y_I , and Z_I directions. This plot indicates the X_I direction having the least noise sensitivity while the Y_I direction having the most (worse) noise sensitivity.

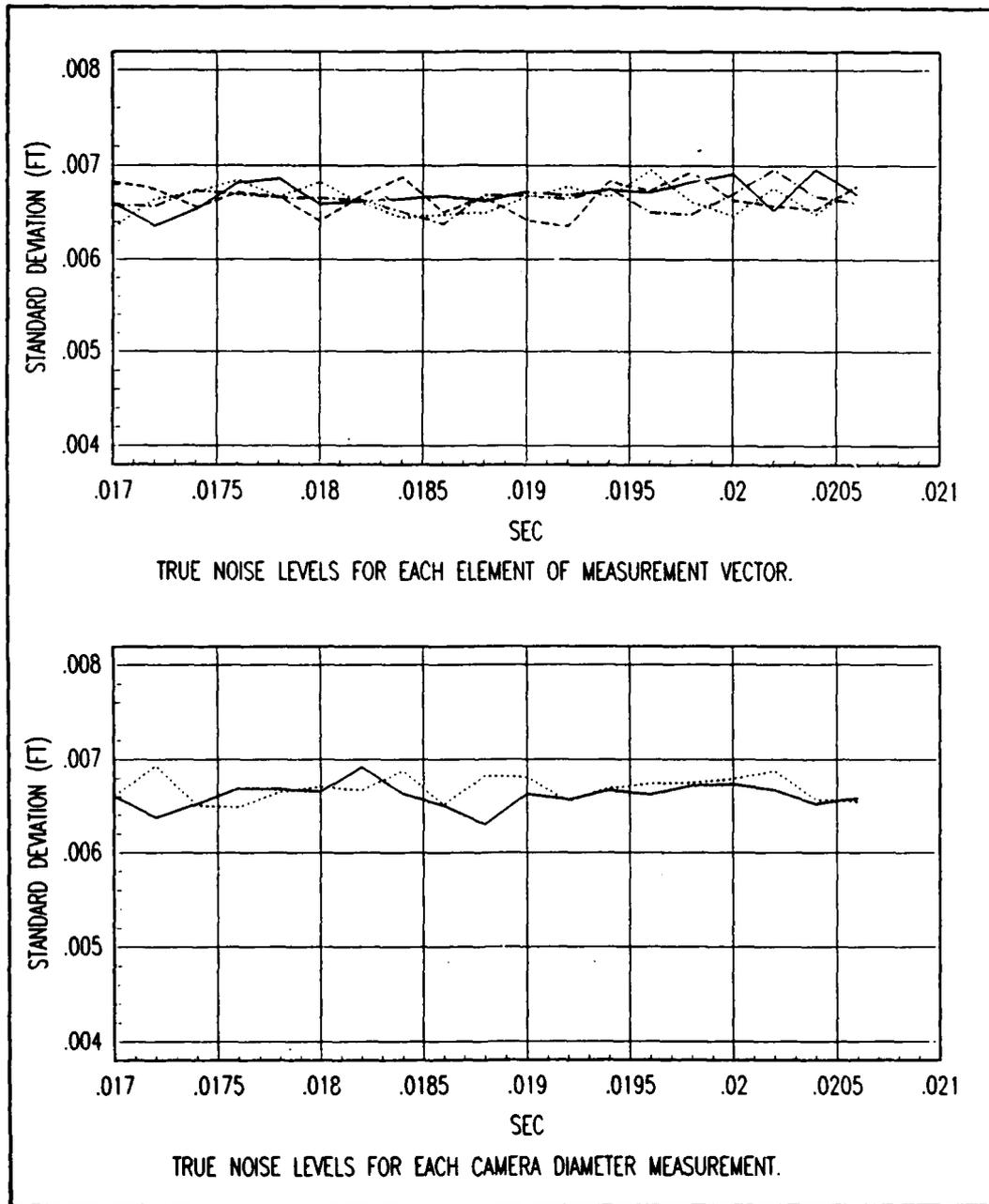


Figure 5.3. True Standard Deviation Noise Level Plots from Desired Level of ≈ 0.0067 ft and 1000 Monte-Carlo Runs.

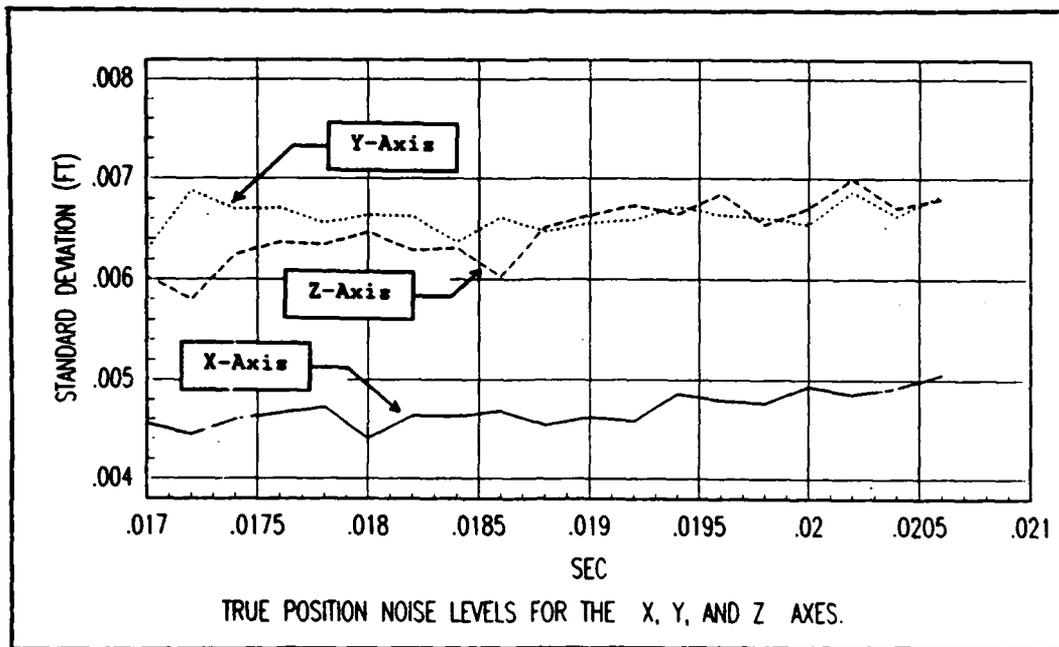


Figure 5.4. Resulting Noise Level Standard Deviations for the X_I , Y_I , and Z_I Axis Directions from Noise Levels Indicated in Figure 5.3.

V.2.a.1 Convergence. The category one filter estimate covariances, $\underline{P}(\cdot)$, for position and velocity, converged quickly and correlated well with the true error covariances to indicate reasonably good "tuning" of the EKF's dynamics noise strength, $\underline{Q}_d(\cdot)$.

In contrast, the filter acceleration state covariances (plotted as standard deviations) in all simulation runs demonstrated poor correlation to the true error standard deviations. In all runs, the filter acceleration covariance was substantially larger than the true covariance. This was mainly attributed to the long transient response of the EKF's

acceleration model. Typical acceleration time constants, τ , were observed to range from 0.01 to 0.5 second while a 10,000 ft/sec track interval would only last 80 μ sec. Therefore, the comparatively "slow" acceleration model could not be converged quickly enough by the "brief" interval of measurements.

Table 5.1 summarizes the performance results taken from the category one RTS smoother outputs. Error data presented in this table (and following tables) includes the worst-case error value for the entire track duration, $[t_o, t_f]$ and the percentage of that error to the true state value at its time of occurrence. For example, the X-axis velocity estimate error value in Table 5.1 for a 3000 ft/sec fragment was determined by observing Figure C.4 in Appendix C and noting that the largest error (worst-point) was approximately 2.0 ft/sec and occurred at time 0.0212 sec. The error percentage is then calculated by dividing the above error value, 2.0, by the true state value at the worst-point time of 0.0212 sec (obtained from truth model computer listings) as 2407.354 ft/sec:

$$\left[\frac{2.0}{2407.354} \right] \cdot 100.0 \approx 0.083\% \quad (6.1)$$

For cases where the truth model value is zero or close to zero, Eq.(6.1) becomes undefined or grows misleadingly

large. An infinity, ∞ , symbol is placed in the table's "percent" square to indicate such occurrences.

Regarding position estimation errors, since the true position values vary across an interval of $\approx[-5.0,+5.0]$ ft, then expressing such errors as percentages would also be misleading. Therefore, percentage values are considered not applicable for position and an N/A is placed in the tables' appropriate "percent" squares. For all figures and tables, position error values are in feet, velocity errors are in ft/sec, and acceleration errors are in ft/sec².

Pointing out some highlights from Table 5.1 and Figures C.1 through C.54 in Appendix C include:

- 1.) No smoothed position estimate errors greater than 0.002 ft (0.024 inches).
- 2.) No smoothed velocity estimate errors greater than 0.253% of the true value.
- 3.) Smoothed acceleration estimate errors vary across a rather large range from 0.503% to 24.14% (a good indicator of potentially unreliable performance).
- 4.) The EKF initialization method devised in this research demonstrated low sensitivity to the fragment diameter measurement noise (critical to derive τ parameter) and performed well at all three fragment speeds.

Overall, the EKF/JPDA/RTS package performed very well (excluding acceleration performance) for category one simulation runs.

Table V.I. Category 1 (single fragment) Worse-Case Error Performances

For initial speed: $ \underline{v}(t_0) = 3000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0012	N/A	-0.0018	N/A	-0.0011	N/A
Velocity	+2.0	0.083	-0.085	∞	+0.50	0.059
Accel.	+1125	6.45	+30.0	∞	-500	8.41
For initial speed: $ \underline{v}(t_0) = 6000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0008	N/A	-0.0019	N/A	+0.0010	N/A
Velocity	-1.6	0.032	-0.9	∞	+2.2	0.133
Accel.	+2000	2.89	+800	∞	+3400	13.54
For initial speed: $ \underline{v}(t_0) = 10000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.002	N/A	-0.0007	N/A	-0.002	N/A
Velocity	+7.0	0.085	+0.8	∞	+7.2	0.253
Accel.	-10150	0.503	-125	∞	-16000	24.14

Regarding overall steady-state convergence, the small number of propagate/update cycles that occur for tracking 6000 ft/sec (twelve updates) and 10000 ft/sec (six updates) fragments do not allow the EKF states to fully converge down to a steady-state condition. This becomes obvious when

comparing a 3000 ft/sec run to a 10000 ft/sec run. The 3000 ft/sec run receives enough updates to converge and "level off" to a steady-state condition while the 10000 ft/sec run does not receive enough updates to do the same.

It was noted that the EKF's first-order Gauss-Markov acceleration model as defined in Eq.(3.4), fit very close to the acceleration truth model in Eq.(3.1) as long as the acceleration time constant, τ , was correctly matched as a function of velocity. This close model matching becomes beneficial for cases of "no valid measurements" in the JPDA algorithm update cycle. Such a situation was observed frequently during simulation runs when one or more noise-corrupted position measurements fell outside the validation gate. This required the EKF acceleration model to continue propagating without a measurement update for one or more recursion cycles. Therefore, the closer the model match, the smaller the tracking error growth rate during an extended "no valid measurements" propagation period. Although the Singer dynamic acceleration model demonstrated good matching to the truth model characteristics, the overall acceleration estimate errors were considerably high. This was attributed to the fact that the acceleration states are two differentiations away from the incoming noisy position measurements. The two differentiation steps accentuate the high frequency system noises to levels that dominate the acceleration state estimates. This poor estimation

performance combined with the previously discussed poor covariance performance leaves no choice but to regard the acceleration estimates as unreliable and unusable.

V.2.b Category Two. These multi-fragment (multi-measurement) simulation runs are where the JPDA algorithm was first put to work. For these runs, the three fragments, with their straight trajectories, did not pose any difficulty whatsoever to the JPDA algorithm, as long as the fragment trajectories were not too close together. Close comparison of Figures C.7 and C.73 in Appendix C reveals the filter standard deviation lines in Figure C.73 begin to diverge after 0.0195 seconds. This peculiarity imposed the need for a closer look by running two additional special runs: one run for two relatively "close" fragment trajectories (≈ 1.8 ft apart) and another run for two relatively "distant" trajectories (≈ 2.6 ft apart). These scenarios were run for 100 Monte-Carlo runs to rule out if Figure C.73 was only the result of its relatively small 20 Monte-Carlo run sample size. Figure 5.5 displays two indicative plots that summarize the results. The top plot shows the X-axis position performance for one of two fragments with ≈ 1.8 ft spacing between their trajectories, while the bottom plot is the same except for ≈ 2.6 ft spacing. The relatively degraded performance of the top plot is attributed to both fragments' position measurements falling within both filters' validation gates. This requires the JPDA algorithm to jointly

distribute the "weight" of each measurement among both tracking filters.

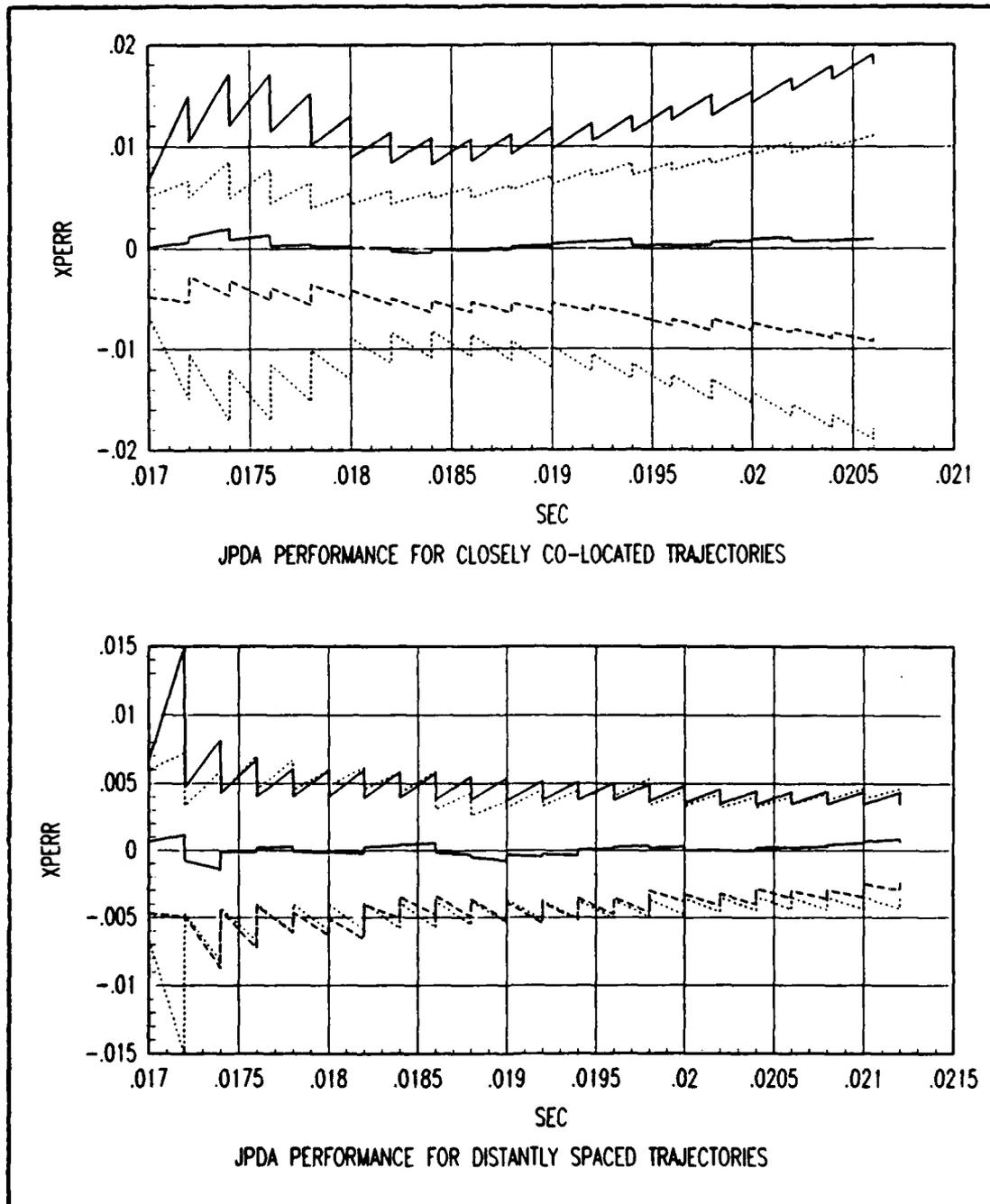


Figure 5.5. Special Test-Run of "Close" Versus "Distanced" Dual-Fragment Trajectories from 100 Monte-Carlo Runs.

This jointly distributed series of "weak" updates creates growing estimation errors along with filter covariance divergence that correctly indicates the filter is less confident of its estimate. In contrast, the lower plot of Figure 5.5, illustrates the case where each fragment's measurement falls solely within its own tracking filter's validation gate and getting "full weight;" resulting in low error and convergence to steady-state.

Although the top plot shows less-than-perfect tracking, it does have the correct track for the correct fragment that it started with. Therefore, the backward-running RTS smoother does a good job at reducing these inaccuracies as shown in Figure 5.6.

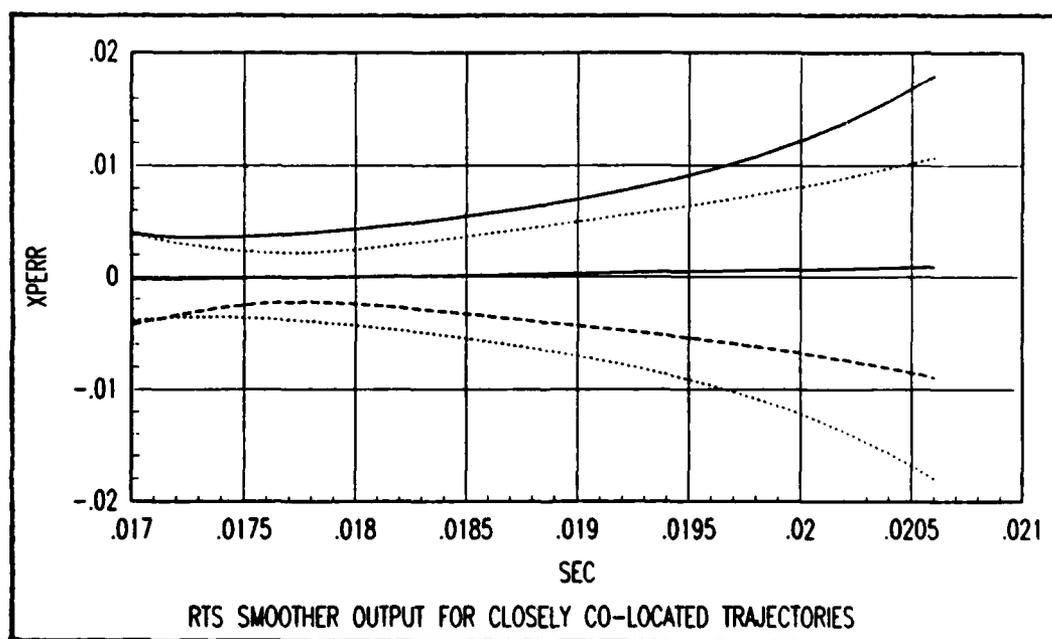


Figure 5.6. RTS Smoother Output from Top Plot of Figure 5.5.

The resulting final output of the RTS smoother at time t_0 may now be fed back into the EKF/JPDA/RTS package as initial conditions and the whole reduction process repeated. Time constraints in this study's completion prevented the necessary software additions to do such a "loop-around" for multiple passes; therefore, it was not attempted.

The overall review of the category two performance plots, Figures C.55 through C.216, Appendix C, indicated no significant performance degradation compared to the category one plots, other than just discussed. Comparing RTS smoother outputs displayed in Table 5.1 with Tables 5.2A, 5.2B, and 5.2C indicates near identically excellent performance except for the acceleration estimates (poor in both categories).

Table V.IIA. Category 2, Fragment 'A', Worse-Case Error Performances

For initial speed: $ \underline{v}(t_0) = 3000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	+0.0008	N/A	-0.0002	N/A	-0.0021	N/A
Velocity	+1.25	0.052	-0.10	∞	+1.4	0.164
Accel.	+1100	6.0	-0.14	∞	-800	13.3
For initial speed: $ \underline{v}(t_0) = 6000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0013	N/A	+0.0003	N/A	+0.0018	N/A
Velocity	+1.0	0.020	+0.25	∞	+3.2	0.228
Accel.	+20.0	0.29	0	∞	+3850	15.3
For initial speed: $ \underline{v}(t_0) = 10000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0019	N/A	+0.0005	N/A	-0.0011	N/A
Velocity	+2.0	0.024	+0.7	∞	+1.0	0.036
Accel.	-6500	3.22	0	∞	+4000	5.75

Table V.IIB. Category 2, Fragment 'B', Worse-Case Error Performances

For initial speed: $ \underline{v}(t_0) = 3000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	+0.0005	N/A	+0.0014	N/A	+0.0006	N/A
Velocity	-1.0	0.40	+0.28	0.636	-0.70	0.097
Accel.	+600	3.31	+10.0	3.25	+500	9.34
For initial speed: $ \underline{v}(t_0) = 6000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0015	N/A	+0.0008	N/A	+0.0008	N/A
Velocity	-3.2	0.065	+0.25	0.284	-3.0	0.208
Accel.	-4000	5.40	-100	7.74	+4500	21.2
For initial speed: $ \underline{v}(t_0) = 10000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.003	N/A	+0.002	N/A	+0.0006	N/A
Velocity	+10.0	0.119	+1.8	1.26	-1.2	0.050
Accel.	-17000	8.28	+500	14.6	+6000	10.2

Table V.IIC. Category 2, Fragment 'C', Worse-Case Error Performances

For initial speed: $ \underline{y}(t_0) = 3000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	+0.0006	N/A	-0.0009	N/A	-0.0008	N/A
Velocity	-0.8	0.032	+0.42	0.984	-0.50	0.065
Accel.	+500	2.71	-20.0	6.54	-250	4.64
For initial speed: $ \underline{y}(t_0) = 6000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.001	N/A	-0.0017	N/A	-0.0016	N/A
Velocity	-1.5	0.030	-2.5	2.86	+1.7	0.111
Accel.	+1000	1.36	+550	42.8	-2900	13.5
For initial speed: $ \underline{y}(t_0) = 10000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0018	N/A	-0.0008	N/A	-0.0014	N/A
Velocity	-5.0	0.061	+0.5	0.347	+2.1	0.083
Accel.	-6000	2.94	-100	2.92	-5000	8.40

One other impact of tracking multiple fragments versus a single fragment was the increase in computer run time. For a single fragment run, the EKF/JPDA/RTS package needs only to run one EKF and the measurement validation process simplifies to one g -sigma test. Such simulation runs were completed in a matter of minutes on a VAX 8650 host. In contrast, for

tracking three fragments, the package must run nine simultaneous EKFs and test each filter's expected measurement, $\underline{h}[\cdot, \cdot]$, against nine candidate measurements, \underline{z}_k . Following the event algebra discussion in Appendix B, for a no-clutter scenario, Eq.(B.1b) defines the JPDA algorithm must "step through" over $9! = 362,880$ event permutations. However, for this study, which includes clutter consideration, Eq.(B.7), Appendix B, increases the total events to 17,572,114 permutation/combination events for each update cycle. The resulting computational time commonly extended to over 24 hours. For a case where four valid fragments are to be tracked, Eq.(B.7) defines a total of 6,199,668,952,530,000 possible events of measurement-to-target permutations and all clutter combinations. Applying this to a hypothetical computer that could do ten-million permutations per second, the above run would require over 19 years to do one update. Needless to say, the event generation routine developed in this study that must "step through" every possible event to find all $\hat{\Omega}(X)$ event matrices for a given Ω validation matrix must be optimized by some manner for practical data reduction use.

V.2.c Category Three. This category, with its curving and crossing trajectories, combined with the presence of non-trackable clutter fragments, posed the "worse case" challenge to the EKF/JPDA/RTS package. It is emphasized here, that these category three runs were modelled with unrealistically

large magnitude lateral accelerations in order to get the trajectories to cross. This was intended to thoroughly test the EKF/JPDA/RTS package's tracking capabilities under adverse conditions. Such curving acceleration magnitudes are highly unlikely in a true test blast fragment dispersion. The resulting runs (as expected) gave the least overall performance of the three categories. For these runs, two fragments were given a substantially large lateral acceleration (mostly in the Y-axis), such that their trajectories curved and crossed tracks approximately one foot from each other, while within both cameras' fields-of-view. Simultaneously, two clutter fragments, each only visible to one camera, were generated to create "extra" measurements that the JPDA algorithm had to process out. Specifically, the clutter fragment trajectories were designed such that one passed through the far-field view of camera #1 and the other passed through the near-field view of camera #2, as shown in Figure 5.7. Such a clutter situation is quite realistic for the arena test application.

From Figures C.217 through C.324 in Appendix C, one can quickly determine the EKFs experiencing severe transient conditions since they were initialized assuming straight-line trajectories from the arena center.

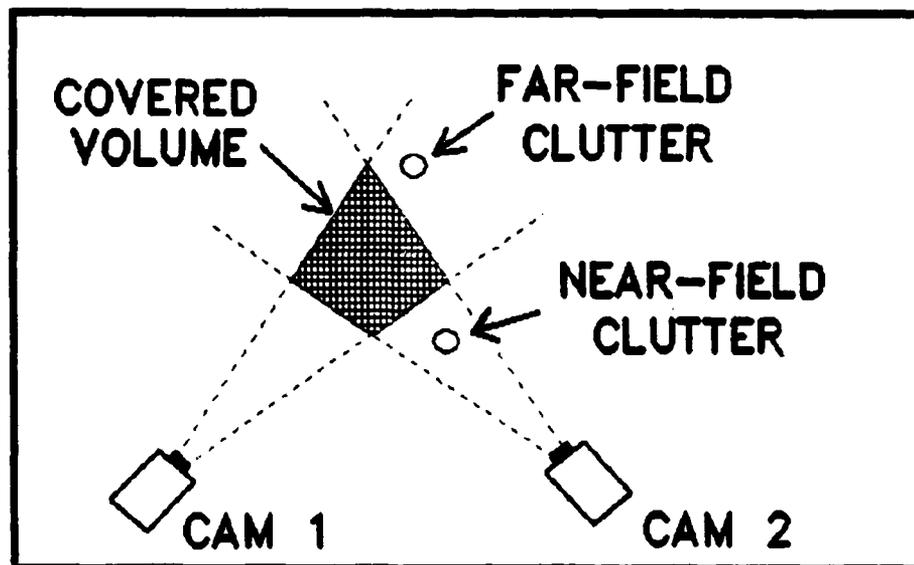


Figure 5.7. Clutter Fragment Trajectory Locations.

For the slow, 3000 ft/sec fragment run, the EKFs receive enough measurements to recover good tracking in the X_I and Z_I axis directions since their lateral acceleration and velocity components are minimal. However, the defined arena geometry combined with the "straight path" assumption leads the EKF initialization procedure discussed in Chapter 3 to expect the greatest fragment acceleration in the X_I axis (radial) direction and near-zero acceleration in the Y_I axis (tangential) direction (review Figures A.3, A.4 and A.5 in Appendix A). Therefore, the unexpected large Y_I acceleration "fools" the EKFs as they propagate a straight trajectory while the fragment is actually curving. Fortunately, the a_{Dmin} threshold feature of the initialization process prevents the Y_I acceleration

component from being set lower than one-third the X_I acceleration value. This keeps the Kalman gains for the Y_I axis direction high enough to help "pull" the EKF states toward the correct values from the succeeding measurement updates. Although the presented tracks in the Y_I axis direction contain diverging errors that surpass the EKFs' standard deviation values, the results obtained from the RTS smoother outputs greatly reduced these inaccuracies. These RTS outputs for time t_0 may now be substituted as initial conditions back into the EKFs for reprocessing, where this time, the EKFs are better initialized to reduce the previous transient behavior. Such a multi-cycle process seems especially applicable to the 6000 and 10000 ft/sec fragment runs since they need the "extra" recursion cycles to approach steady-state.

Table 5.3A and 5.3B summarize the category three results where the Y_I axis shows the only significant degradation in performance. Similar to categories one and two, the acceleration estimates for this category are considered unreliable, and are extremely poor in the Y_I axis direction.

Special notice is brought to the fact that no degrading effects could be found from the presence of the two clutter fragments (except computational run-time). This makes perfect sense since candidate measurements created by the clutter should consistently "fail" all validation tests.

Table V.IIIA. Category 3, Fragment 'A', Worse-Case Error Performances

For initial speed: $ \underline{v}(t_0) = 3000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.002	N/A	+0.009	N/A	+0.0003	N/A
Velocity	+4.5	0.18	+11.0	17.1	-1.4	0.91
Accel.	-2150	10.3	+5500	99.5	-700	12.5
For initial speed: $ \underline{v}(t_0) = 6000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0035	N/A	+0.0102	N/A	+0.0007	N/A
Velocity	+8.2	0.16	+27.0	21.3	-2.8	0.18
Accel.	+9300	14.4	+22000	99.5	+3300	14.1
For initial speed: $ \underline{v}(t_0) = 10000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0058	N/A	-0.025	N/A	-0.0024	N/A
Velocity	-33.0	0.04	-70.0	34.6	+1.9	0.08
Accel.	+56000	24.2	-61000	99.7	-5000	9.20

Table V.IIIB. Category 3, Fragment 'B', Worse-Case Error Performances

For initial speed: $ \underline{v}(t_0) = 3000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	-0.0006	N/A	-0.007	N/A	-0.0005	N/A
Velocity	-1.0	0.042	-10.5	16.4	-0.90	0.12
Accel.	-700	3.50	-5000	90.8	+720	13.9
For initial speed: $ \underline{v}(t_0) = 6000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	+0.0045	N/A	-0.0085	N/A	-0.0011	N/A
Velocity	-9.0	0.18	-26.0	20.7	-1.8	0.13
Accel.	+8800	10.5	-22000	99.8	-2000	10.2
For initial speed: $ \underline{v}(t_0) = 10000.0$ ft/sec						
State	X-Axis		Y-Axis		Z-Axis	
	Error	Percent	Error	Percent	Error	Percent
Position	+0.008	N/A	+0.027	N/A	+0.003	N/A
Velocity	+41.0	0.48	+72.0	35.3	+4.7	0.19
Accel.	-61000	34.3	+62000	100	+8000	12.4

Therefore, when a JPDA event matrix matches a clutter measurement, the algorithm will simply step past it and go on to the next event. This precludes a "weight" being assigned to the clutter measurement and subsequent "weakening" of the resulting update.

V.3 Summary

This chapter has presented the three categories of fragment trajectories simulated in this study along with a special run to indicate the effects of 20 versus 100 Monte-Carlo runs on simulation results. For each category, specific performance highlights and shortcomings were discussed along with any "attempted fixes." Further discussion described the computational workload shortfall encountered with this study. One other special run included in this chapter compared the tracking performance of "close" versus "distant" fragment trajectories, relative to each other.

VI. Conclusions and Recommendations

This study has developed, and simulated a multiple fragment tracking system for use in munitions testing. Simulation analysis has shown overall positive results and the usefulness of the JPDA algorithm. Some important observations learned from the research, design, simulation, and analysis are given below:

VI.1 Conclusions

1.) The designated design specification for a CCD camera of 512 x 512 pixel resolution, operating at 5000 frames per second, is a very demanding specification to meet because of the multi-gigahertz (GHz) signal processing speeds required to read, convert (analog to digital), and store the image data from the camera chip to a suitable storage medium. Present technology, non-imaging CCD devices have been operated at the GHz range, but no report literature was found to support the 5000 frame/sec rate as truly achievable with present technologies.

2.) The orthogonal sensor (camera) geometry simplified the applied measurement model by eliminating cross-covariance noise terms.

3.) The linear, first-order, Gauss-Markov acceleration model employed within the EKFs proved to be a very close match to the nonlinear fragment acceleration truth model, Eq. (3.1).

4.) Tracking error performance of straight-trajectory fragments in position often out-performed the camera pixel resolution by an order of magnitude, even though the camera measurements were corrupted by a three-sigma noise level of one pixel. Similarly, velocity tracking errors rarely surpassed 0.2% of the true state value.

5.) The acceleration state estimates for all simulation runs proved inconsistent and unreliable to the point that they were determined unusable.

6.) The EKF initialization process developed in this study performed extremely well for straight-path and mildly curving fragments. This process also demonstrated good matching of the acceleration time constant parameters, τ_x , τ_y , and τ_z , to the true acceleration decay rates.

7.) The slow adaptation rate of the EKFs resulted in their inability to hold track on strong-curving fragments.

8.) The inclusion of the RTS smoother greatly improved the quality of the estimated outputs without significantly increasing the computational run-time.

9.) Although this study developed an operational software algorithm to generate all possible events for a given validation matrix, its computational load made it impractical for applications involving more than three fragments visible in each camera (nine measurement combinations).

10.) The JPDA validation gate process employing the "g-sigma" test proved very effective at eliminating undesirable candidate measurements, such as the simulated clutter fragments.

11.) The degraded performance of the JPDA algorithm for "close" trajectories could not be improved by simply adjusting the validation gate size, g . The gate size could be reduced to exclude the "other track" measurements, but at the cost of the gate being so small, that the measurement noise would often "jig" the correct track measurements outside the proper gate. This resulted in a significantly reduced number of updates and complete loss of track.

VI.2 Recommendations

Many of the assumptions made while completing this study were made either to simplify the problem or because no data could be found to support alternative choices. The following recommendations consider variations to the general assumptions outline in Chapter I:

1.) The simplified, spherical shaped, fragment drag model needs to be expanded to non-spherical shapes, such as platelets, wedges, etc.

2.) The simplified, white, Gaussian measurement noise needs to be expanded to include biases for camera misalignment and time-correlated noise components for camera vibration, etc.

3.) The image pre-processing, that was assumed already present, must be designed that will extract the pertinent fragment data from the raw digitized image data. This process must extract as a minimum, the quantity of fragments visible, fragment diameters, and fragment image centroid coordinates.

4.) This study set the JPDA clutter density value, C , to unity for all simulation runs since no data was available to do otherwise. Likewise, the camera probabilities of detecting fragments within their field-of-view, P_D , were arbitrarily set at 0.95 simply because it was a "nice" number. Variations to these parameters need investigation and their resulting effects on the JPDA algorithm performance.

5.) This study, through ad hoc tuning, determined that validation gate values, g , were best set at 3.0 for straight-trajectory fragments and 4.0 for curving trajectory fragments. A more analytic means of determining this parameter as a function measurement noise strength, $R(\cdot)$, and other input sources should be investigated.

6.) This study held the acceleration time constants, τ_x , τ_y , and τ_z , constant once they were initialized at t_0 . The consideration to augment these to the EKF as estimated parameters may be considered.

7.) The simulation software developed in this study should be modified to allow the RTS smoother output at time

t_0 be "looped back" into the EKF input as initial conditions. This would facilitate multiple, bi-directional processing runs for a given set of measurements.

8.) The algorithm developed in this study to generate each possible event matrix, $\hat{\Omega}(X)$, for a given validation matrix, Ω , must be rewritten to be less computationally burdensome.

APPENDIX A: Measurement Geometry

This appendix details the development of the applied observation and measurement geometry, including specifics defining reference frames and transform functions between reference frames.

The general arena setup involves the following set of three reference frames:

Frame 1) A two dimensional (2D) x-y coordinate plane for each camera where the origin is located at the center of the image plane (camera bore-sight):

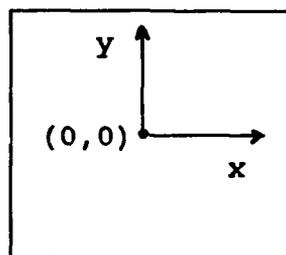


Figure A1. Camera Object Plane Coordinate Frame

For the above image coordinates, the x and y designators are subscripted by the camera number. For example camera one would have a coordinate pair listed as (x_1, y_1) and camera two would have its coordinate pair listed as (x_2, y_2) .

The width and height of the image plane (modelled to be square, so width = height) is determined by the focal distance, D , and the camera's angular FOV, α , determined by camera lens selection as depicted in Figure A2:

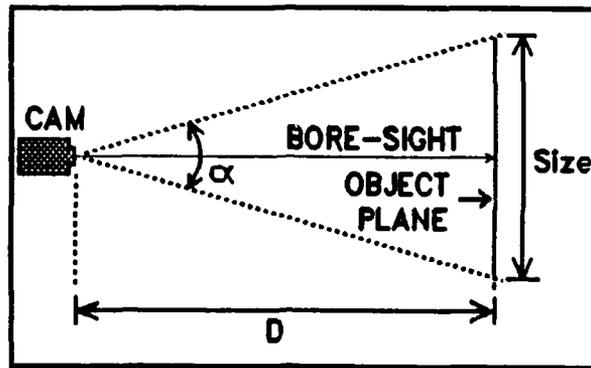


Figure A2. Camera FOV Geometry

The function relating object plane Size to D and α in Figure A2:

$$\text{Size} = 2 D \text{TAN}(\alpha/2) \quad (\text{A-1})$$

From this equation, various image plane sizes for various resolutions or arena set-up requirements may be met by changing camera lenses or adjusting D in the arena set-up.

Frame 2) Each camera pair (x_1, y_1, x_2, y_2) defines a three dimensional (3D) right-hand intermediate reference frame (X_I, Y_I, Z_I) such that the object planes of the two cameras orthogonally intersect and both camera object frame origins and the intermediate frame origin are at the same point. The Z_I direction is pointed in the local vertical (up). The X_I direction is pointed radially outward from the arena center. The remaining Y_I direction is set by the right-hand rule, Z_I cross X_I equals Y_I . See Figures A3, A4 and A5.

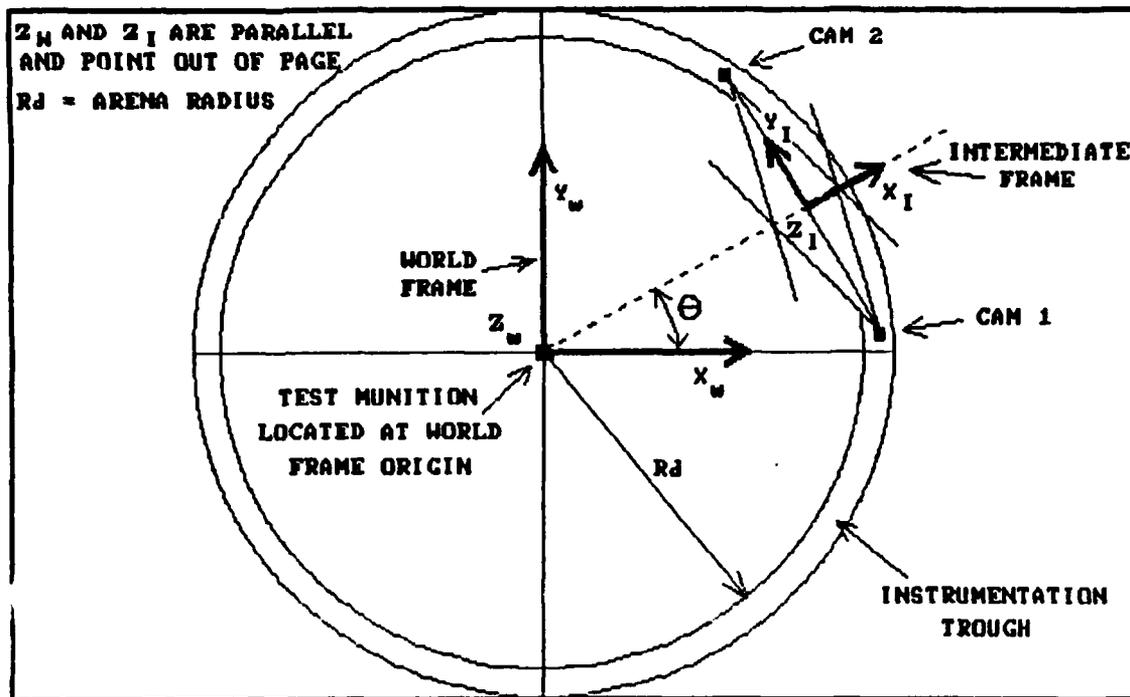


Figure A3. Overhead View of Arena Geometry

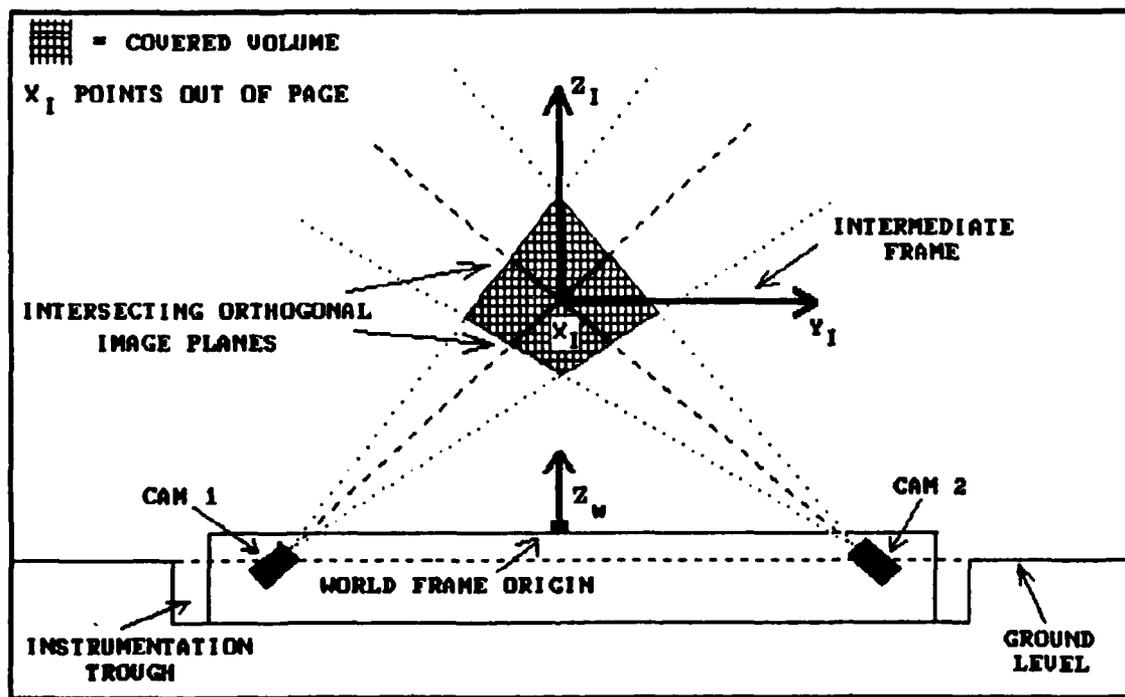


Figure A4. Radial View (looking inward) of Arena Geometry

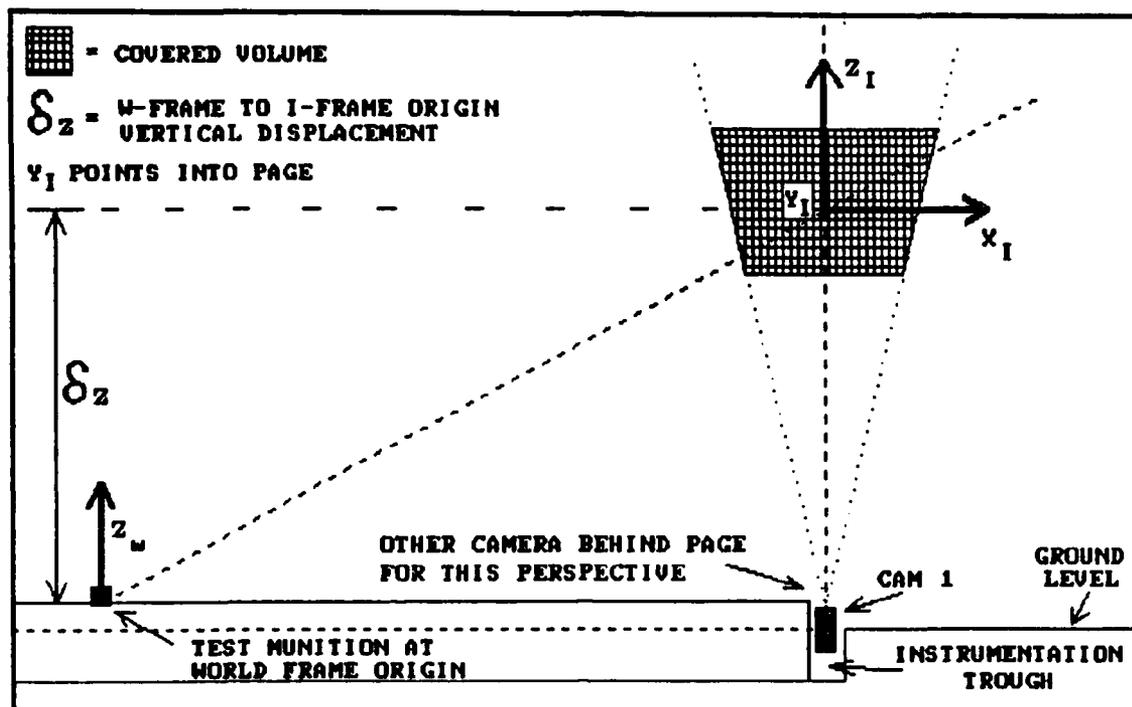


Figure A5. Side View of Arena Geometry

The transformation from the 2D camera image coordinates (C-frame) to the 3D intermediate frame (I-frame) requires a set of three nonlinear functions. The derivation of these functions is based on combining the Pythagorean theorem, properties of similar triangles, intersection point of two lines, and numerous algebraic substitutions to yield:

$$x_I = \frac{2 D x_1 x_2}{x_1 y_2 - x_2 y_1 + D x_2 - D x_1} \quad (A-2)$$

$$y_I = \frac{D^2 (y_1 - y_2)}{R (y_1 y_2 - D^2)} \quad (A-3)$$

$$Z_I = \frac{D [Y_2 (2 Y_1 + D) + D Y_1]}{R (D^2 - Y_1 Y_2)} \quad (A-4)$$

where

$$R = (2.0)^{1/2}$$

Note that Eq.(A-2) is undefined when x_1 , y_1 , x_2 , and y_2 are all zero. This condition must be checked for in any software implementations.

To transform from the I-frame back to the C-frame requires the above system of equations be algebraically manipulated and solved for x_1 , y_1 , x_2 and y_2 (Note: There is a second solution for Y_I other than that shown as Eq.(A-3); therefore, four equations do exist to solve for the four unknowns). The resulting inverse transform equations are:

$$x_1 = \frac{2 D X_I (R Y_I + D)}{2 (Y_I Z_I + Y_I^2 + D^2) + R D (3 Y_I + Z_I)} \quad (A-5)$$

$$y_1 = \frac{R D (Z_I - Y_I)}{R (Z_I + Y_I) + 2 D} \quad (A-6)$$

$$x_2 = \frac{\left[\frac{R D X_I (Z_I + Y_I)}{R (Z_I - Y_I) + 2 D} \right] - D X_I}{D - R Y_I} \quad (A-7)$$

$$y_2 = \frac{R D (Z_I + Y_I)}{R (Z_I - Y_I) + 2 D} \quad (A-8)$$

Frame 3) A right-hand 3D main reference frame or world reference frame X_W, Y_W, Z_W with its origin at the center of the arena (also location of test munition) (again review Figures A.3, A.4, and A.5). The Z_W axis is pointed to local vertical (up, parallel to Z_I). The X_W - Y_W plane is therefore horizontal. The direction to which X_W points is arbitrary (such as, pointing north) since this W-frame is the master "inertial" frame to which all previously defined reference frames eventually transform.

The transformation function from the I-frame to the W-frame simply involves a single direction cosine matrix (DCM) operation for fragment velocity and acceleration vectors, while position vectors also require an additional translation vector component. As indicated in the previous reference frame definitions, the Z_W and Z_I directions are parallel making them the "pivot" for the rotation process.

Based on the selected location of each camera pair, the correspondingly created I-frame origin is located at angle θ from the X_W axis following standard right-hand rotation (see Figure A3). The desired DCM, M_I^W , becomes:

$$M_I^W = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (A-9)$$

and is applied as:

$$\underline{v}^W = \underline{M}_I^W \underline{v}^I \quad (\text{A-10})$$

where the inverse transform from the W-frame to the I-frame is the transpose of the above DCM to give \underline{M}_W^I :

$$\underline{M}_W^I = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A-11})$$

such that:

$$\underline{v}^I = \underline{M}_W^I \underline{v}^W \quad (\text{A-12})$$

When transforming fragment position vectors, a position translation vector from the W-frame origin to the I-frame origin must also be included. The components of this translation vector, \underline{t} are the x, y and z projections of this translation vector onto the X_W , Y_W , and Z_W axes respectively:

Let: Rd = Arena radius (from arena center to camera lens measured along ground plane) (Figure A.3).

δz = The vertical displacement of the W-frame and I-frame origins (preselected for particular arena set-up) (Figure A.5).

$$t_x = (Rd^2 - \delta z^2)^{1/2} \cos \theta \quad (\text{A-13})$$

$$t_y = (Rd^2 - \delta z^2)^{1/2} \sin \theta \quad (\text{A-14})$$

$$t_z = \delta z \quad (\text{A-15})$$

When transforming an I-frame position vector, \underline{p}^I , to the W-frame, \underline{p}^W , the \underline{t} vector is summed to the DCM·I-frame product as:

$$\underline{p}^W = \underline{M}_I^W \underline{p}^I + \underline{t} \quad (\text{A-16})$$

When transforming from W-frame to I-frame the \underline{t} vector is subtracted from the DCM·W-frame product as:

$$\underline{p}^I = \underline{M}_W^I \underline{p}^W - \underline{t} \quad (\text{A-17})$$

This concludes the reference frames and associated transforms discussion pertinent to this research effort.

APPENDIX B: Event Algebra

The following derivations develop the worst-case situation for a given validation matrix, Ω , that is fully unit populated (all ones). Although such an occurrence is highly unlikely the JPDA algorithm must "assume the worse" since there is no "a priori" information regarding which or how many of the candidate measurement-to-target associations are true.

Begin with the following variable definitions:

T = The number of candidate non-clutter targets (tracks).

M = The number of candidate measurements.

f = The number of false measurements (clutter).

E = The integer number of possible feasible events.

t = The index counter for candidate tracks; $0 \leq t \leq T$.

j = The index counter for candidate measurements; $1 \leq j \leq M$.

where also the number of permutations, P, of n objects taken r at a time may be defined as:

$$n^P_r = \frac{n!}{(n - r)!} \quad (\text{B.1})$$

and the number of combinations, C, of n objects taken r at a time may be defined as:

$$n^C_r = \frac{n!}{r! (n - r)!} \quad (\text{B.2})$$

Recall from Chapter 2, a feasible event is defined where no

more than one measurement may originate from each target

(24:176):

$$j \neq r \text{ and } t_j > 0 \text{ implies } t_j \neq t_r \quad (\text{B.3})$$

Also repeating the rules defining an event, $\hat{\Omega}(X)$, from the given validation matrix, $\underline{\Omega}$:

- 1) Scan $\underline{\Omega}$ by rows and pick one unit per row for $\hat{\Omega}(X)$.
- 2) Only one unit from each column $i \geq 1$ can be taken (i.e. at most one measurement could have originated from a target). The number of units from column $i = 0$ is not restricted.

From these definitions and beginning with a simple "no clutter" case, the number of targets, T , and the number of measurements, M , replace n and r respectively in Eq.(B.1) to give an $\underline{\Omega}$ matrix and the total number of events, E , as:

$$\underline{\Omega} = \begin{matrix} & t_1 & t_2 & t_3 & \dots & t_T \\ \begin{matrix} j_1 \\ j_2 \\ j_3 \\ \vdots \\ \vdots \\ j_M \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \end{matrix} ; \quad T^E_M = \frac{T!}{(T - M)!} \quad (\text{B.1a})$$

for $M \leq T$

where two classifications describe as:

Case 1: Eq.(B.1a) applies directly for no-clutter and $M \leq T$.

Case 2: Same as Case 1 except when $M = T$, Eq.(B.1a) simplifies to:

respectively in Eq.(B.2) to give:

$$M^C_f = \frac{M!}{f! (M - f)!} \quad (B.5)$$

The product of Eqs.(B.4) and (B.5) results in the total number of possible events for any ratio of T to M and a fixed number, f, of clutter measurements:

$$T^E_{M,f} = \frac{M! T!}{f! (M - f)! (T - M + f)!} \quad (B.6)$$

where f must be in the range:

$$(M - T) \leq f \leq M$$

Note that Eq.(B.6) reduces to Eq.(B.1a) when $f = 0$ (no clutter) and Eq.(B.1b) when $f = 0$ and $T = M$.

Since the number of clutter measurements is not fixed, all possible numbers of clutter measurements must be accounted for where the number of possible clutter measurements may fall in the range: $(M - T) \leq f \leq M$. Therefore, the absolute total number of possible events including clutter for any given T to M ratio results as a summation of multiple cases of Eq.(B.6) over the range of clutter possibilities:

$$\boxed{T^E_{M,f}}_{\text{Total}} = M! T! \sum_{q=0}^M \left[\frac{1}{f! (M - f)! (T - M + f)!} \right]$$

where

(B.7)

$$\begin{aligned}q &= 0 && \text{for } (M - T) < 0 \\q &= (M - T) && \text{for } (M - T) \geq 0\end{aligned}$$

The final quantity, E , given in Eq.(B.7) grows very rapidly for increasing numbers of targets and measurements due to the factorial operators. Re-emphasizing that a fully populated validation matrix is extremely unlikely, the JPDA event generator algorithm designed in this study, must still "step through" each possible permutation-combination event, $\hat{\Omega}(X)$, and test it for a match against the present validation matrix, Ω . The resulting computational workload may become overwhelming.

APPENDIX C: Simulation Output Plots

Description

The plots contained in this appendix were generated using the MATRIX_x interactive software package plotting utility (32). The following lists summarize the simulation parameters that all the contained plots share in common:

- 1.) Created from 20 Monte-Carlo runs.
- 2.) Object plane size and resolution level:
10.0 x 10.0 ft @ 512 x 512 pixels
(≈ 0.02 x ≈ 0.02 ft per pixel side)
- 3.) EKF measurement model expected three-sigma noise level:
0.02 ft
- 4.) JPDA sensor probability of detection: $P_D = 0.95$
- 5.) JPDA Poisson clutter density: $C = 1.0 \text{ ft}^{-4}$
- 6.) True position and diameter measurement three-sigma noise level:
0.02 ft

The nomenclature used the label the vertical axis of each plot is derived from the following list:

- XPERR : The X-axis position error performance.
- XVERR : The X-axis velocity error performance.
- XAERR : The X-axis acceleration error performance.
- YPERR : The Y-axis position error performance.
- YVERR : The Y-axis velocity error performance.
- YAERR : The Y-axis acceleration error performance.

ZPERR : The Z-axis position error performance.
ZVERR : The Z-axis velocity error performance.
ZAERR : The Z-axis acceleration error performance.

For all plots, the horizontal axis labeled SEC is the time in seconds following detonation. Other parameters, such as fragment initial speed, trajectory category, and fragment index label A, B, or C are indicated on each plot.

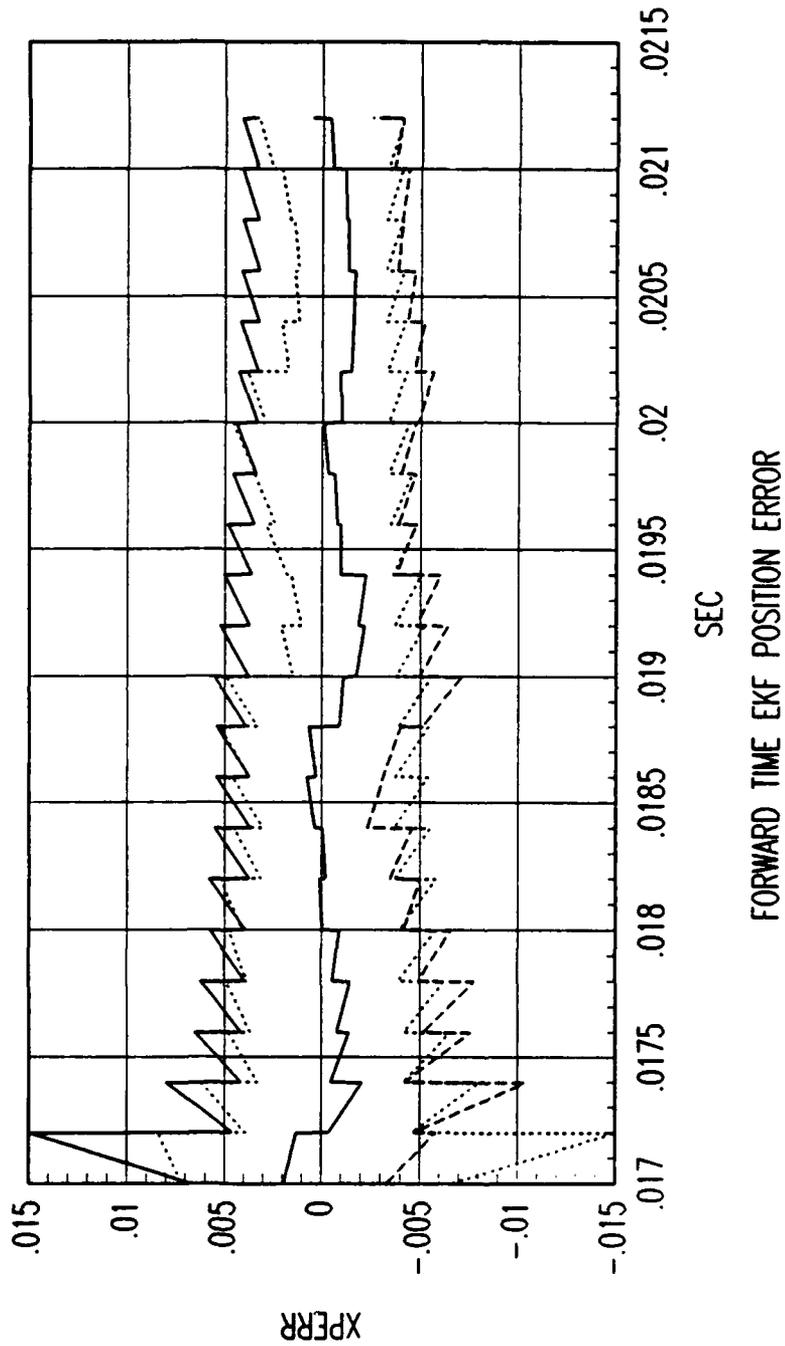


Figure C. 1. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

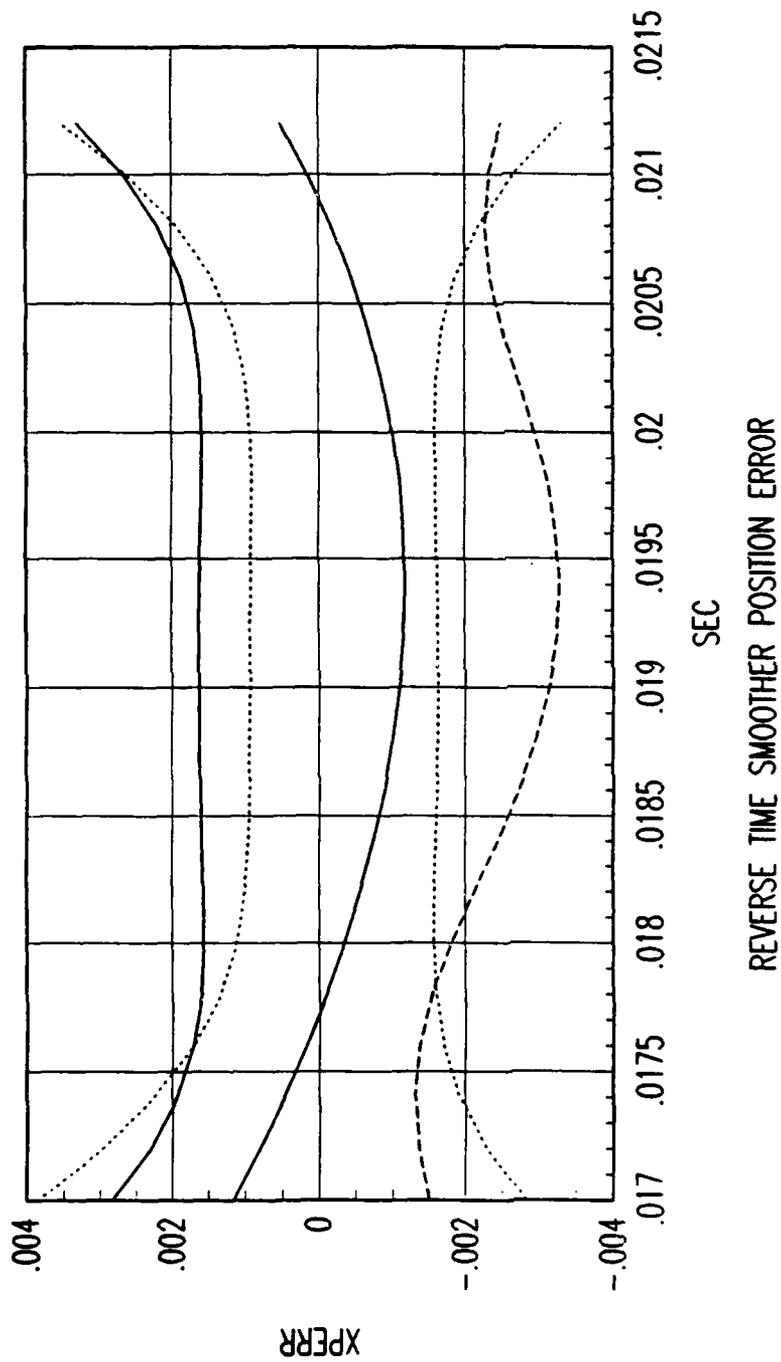


Figure C. 2. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

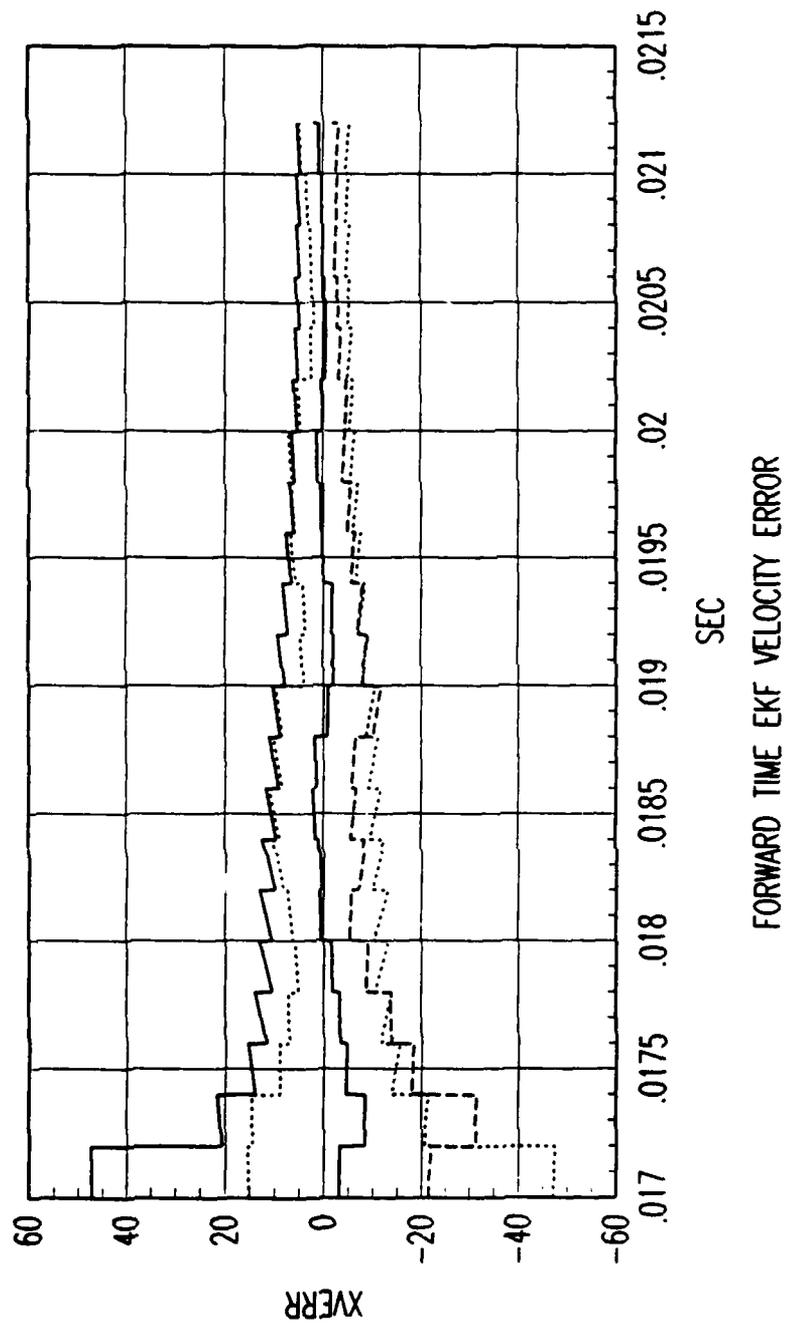


Figure C. 3. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

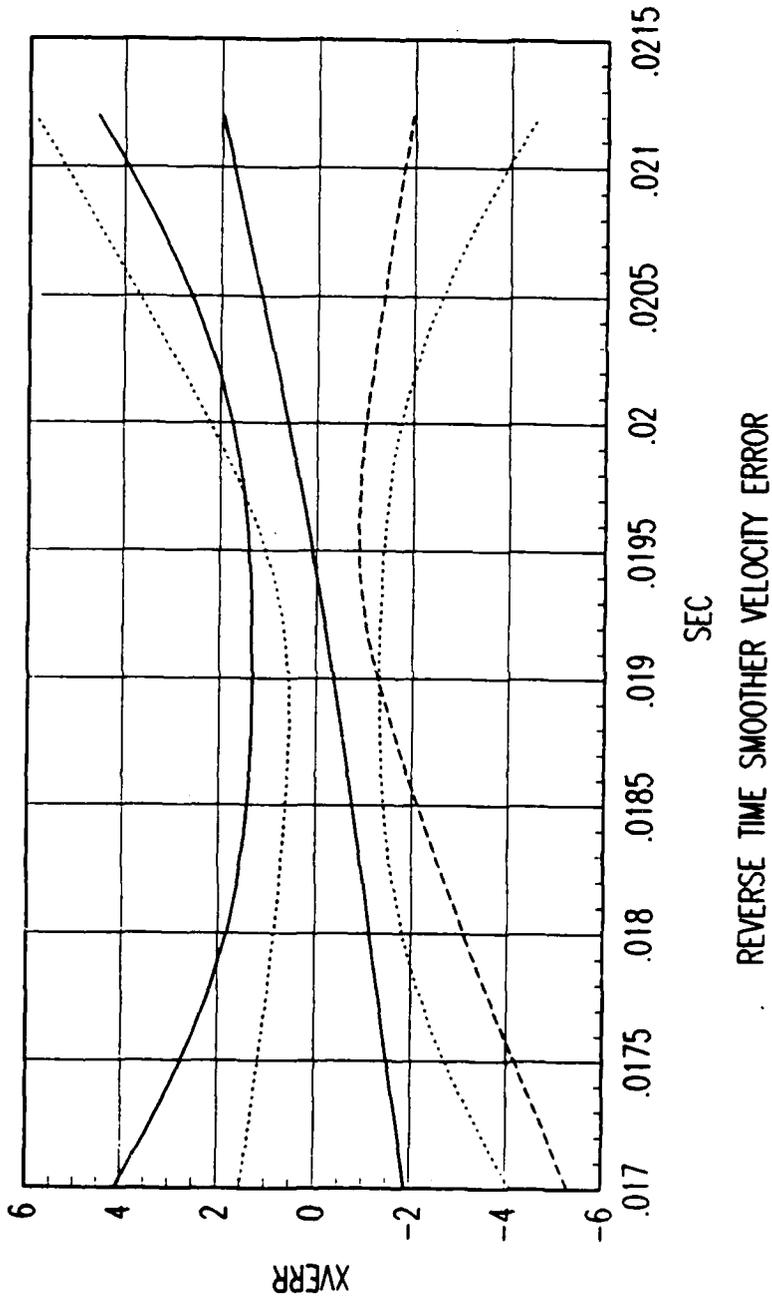


Figure C. 4. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

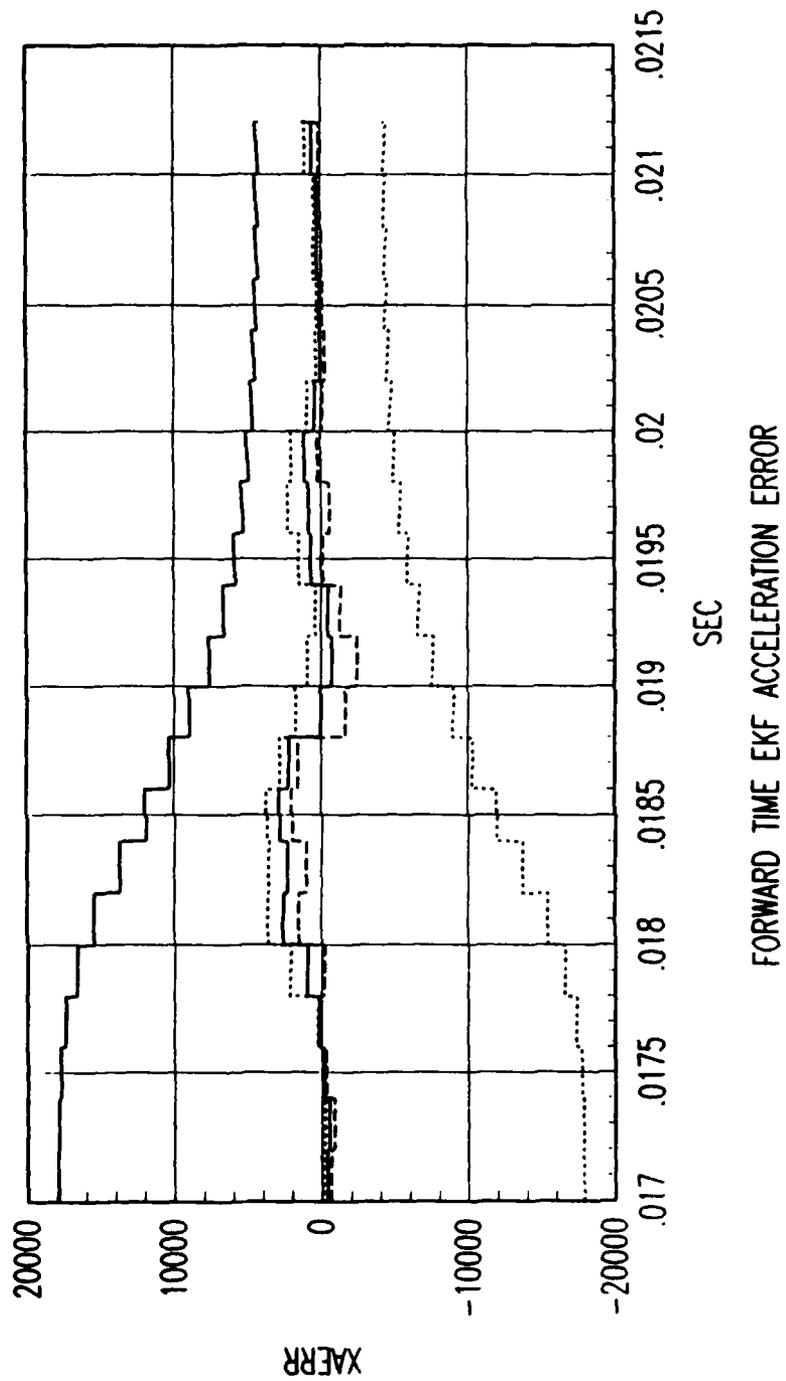


Figure C. 5. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

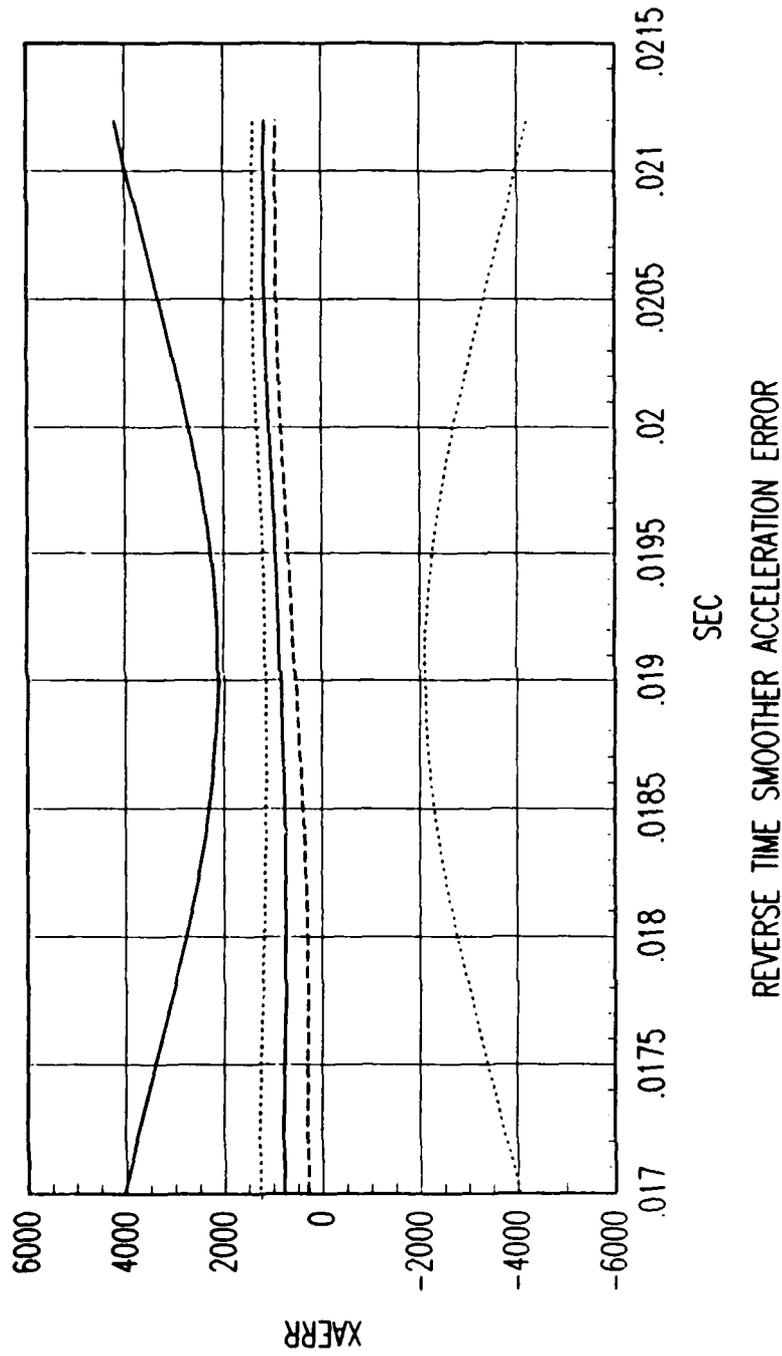


Figure C. 6. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

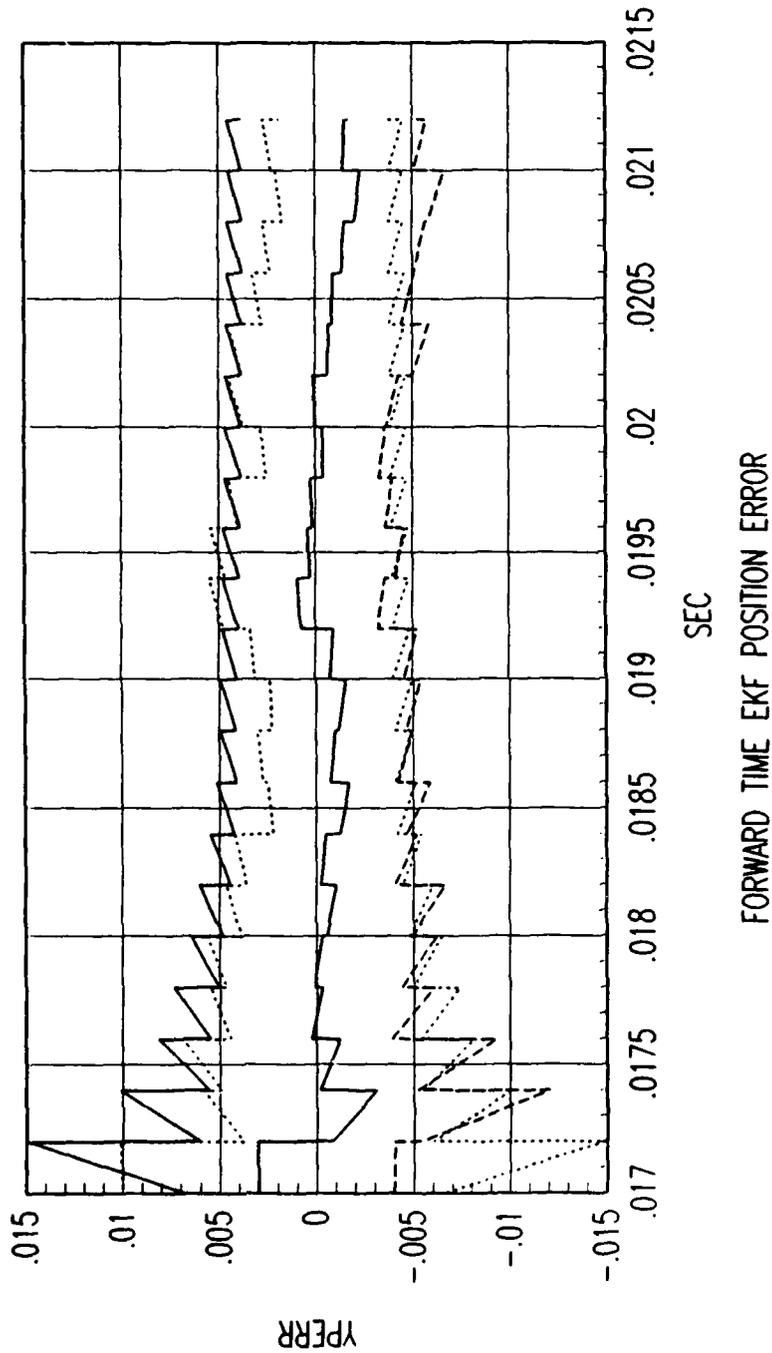


Figure C. 7. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

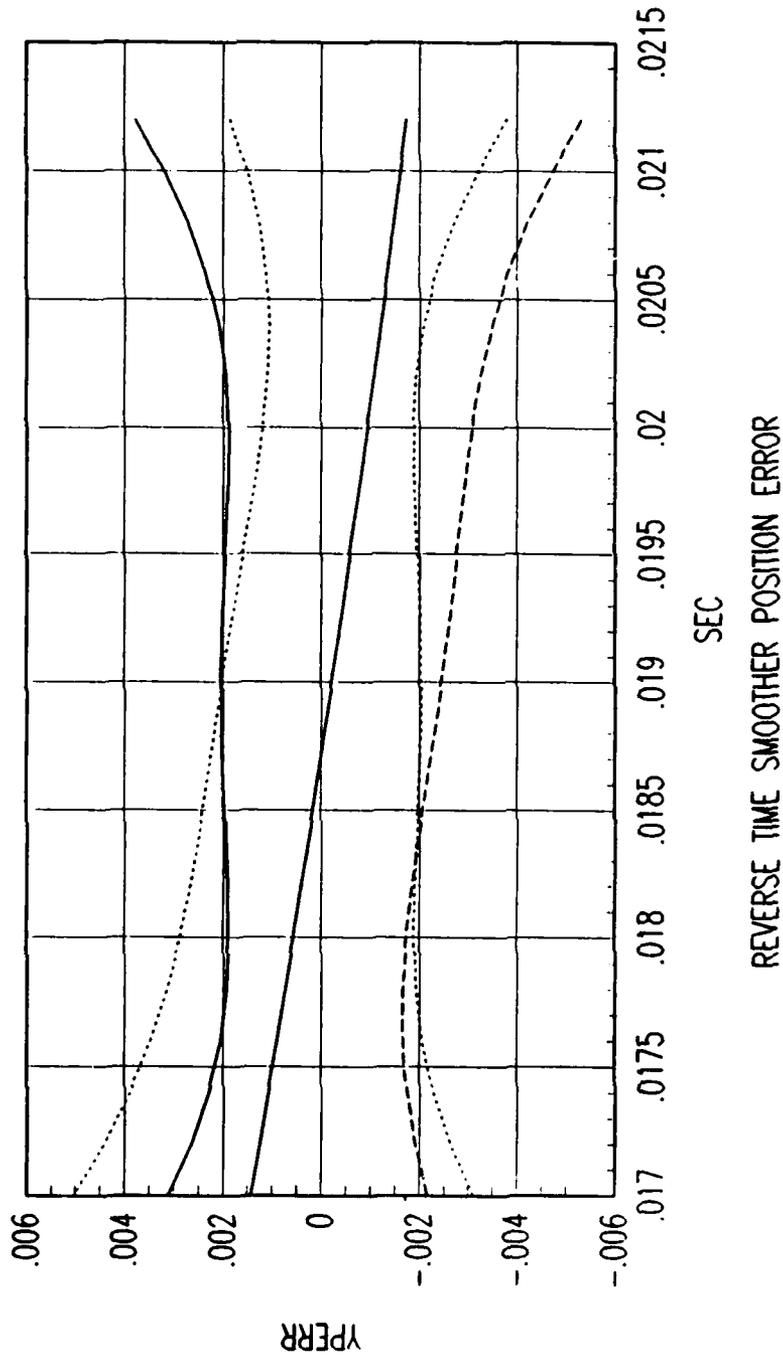


Figure C. 8. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

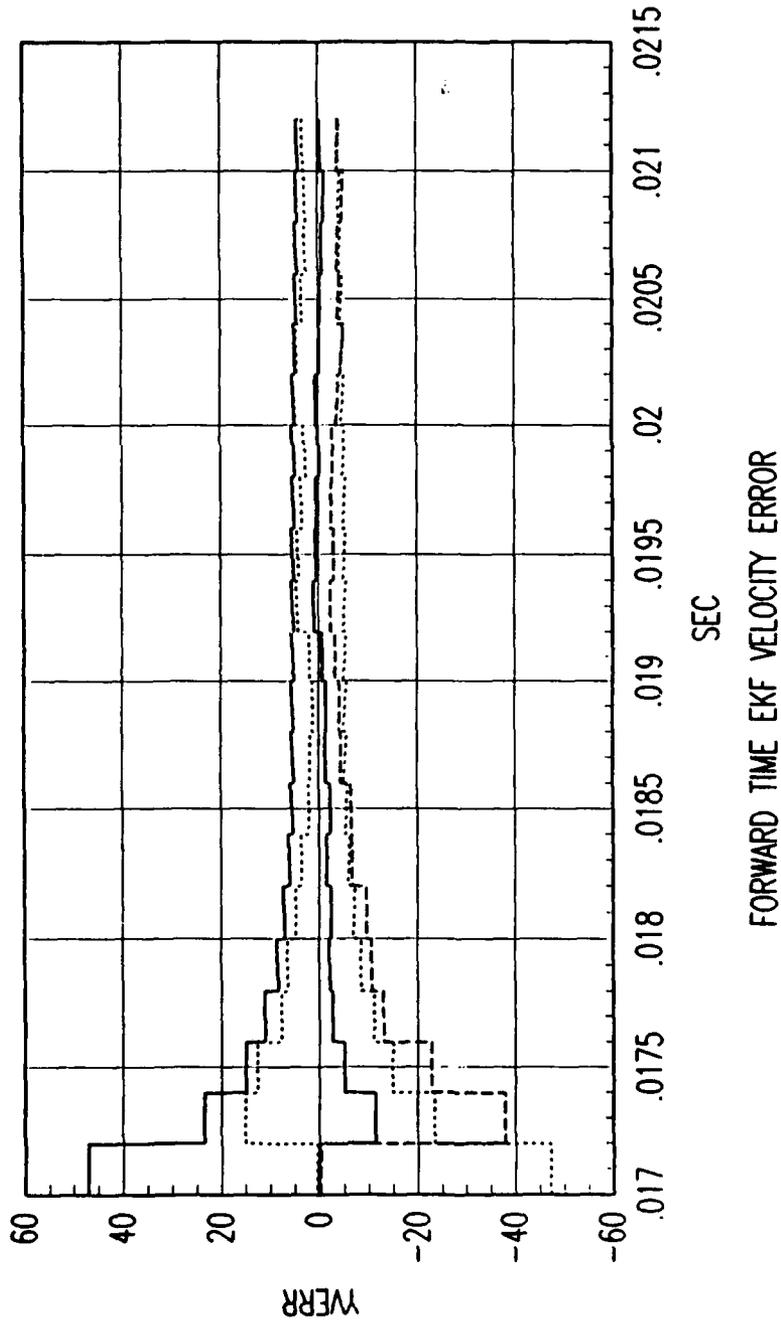


Figure C. 9. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

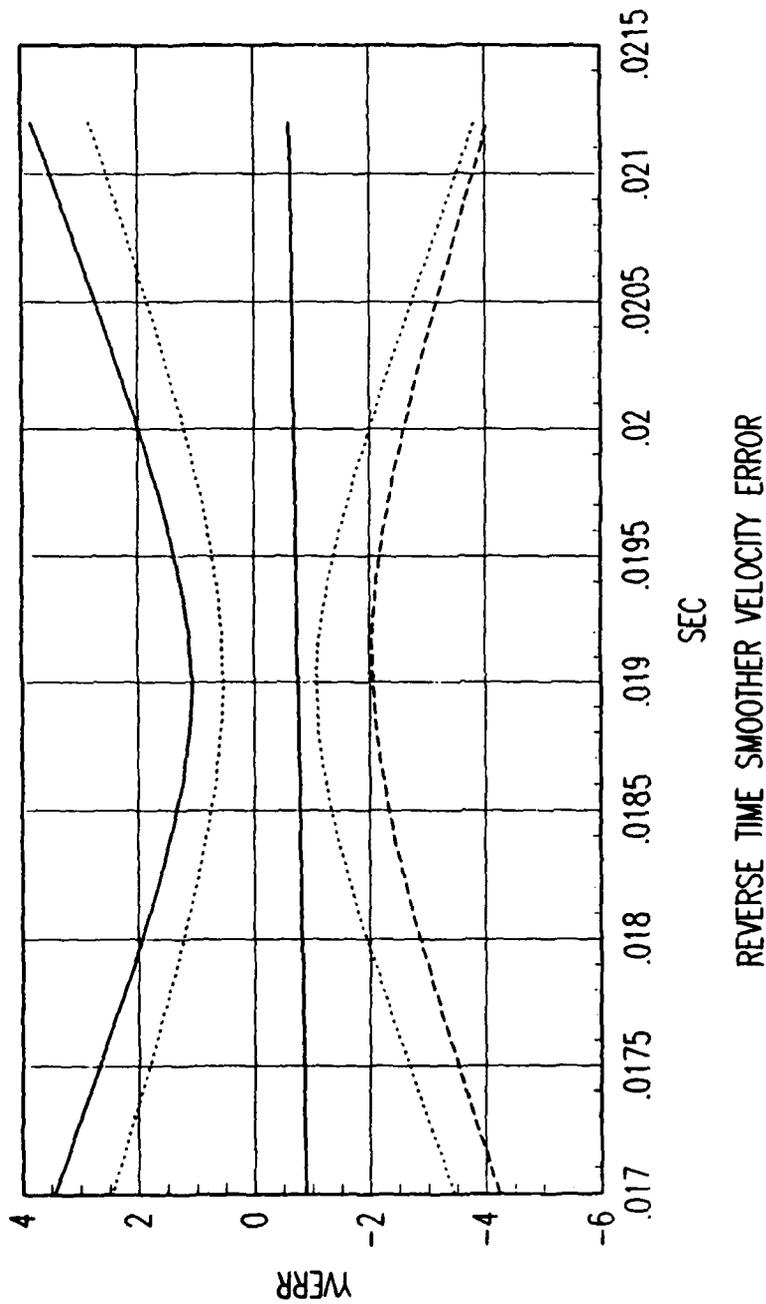


Figure C. 10. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

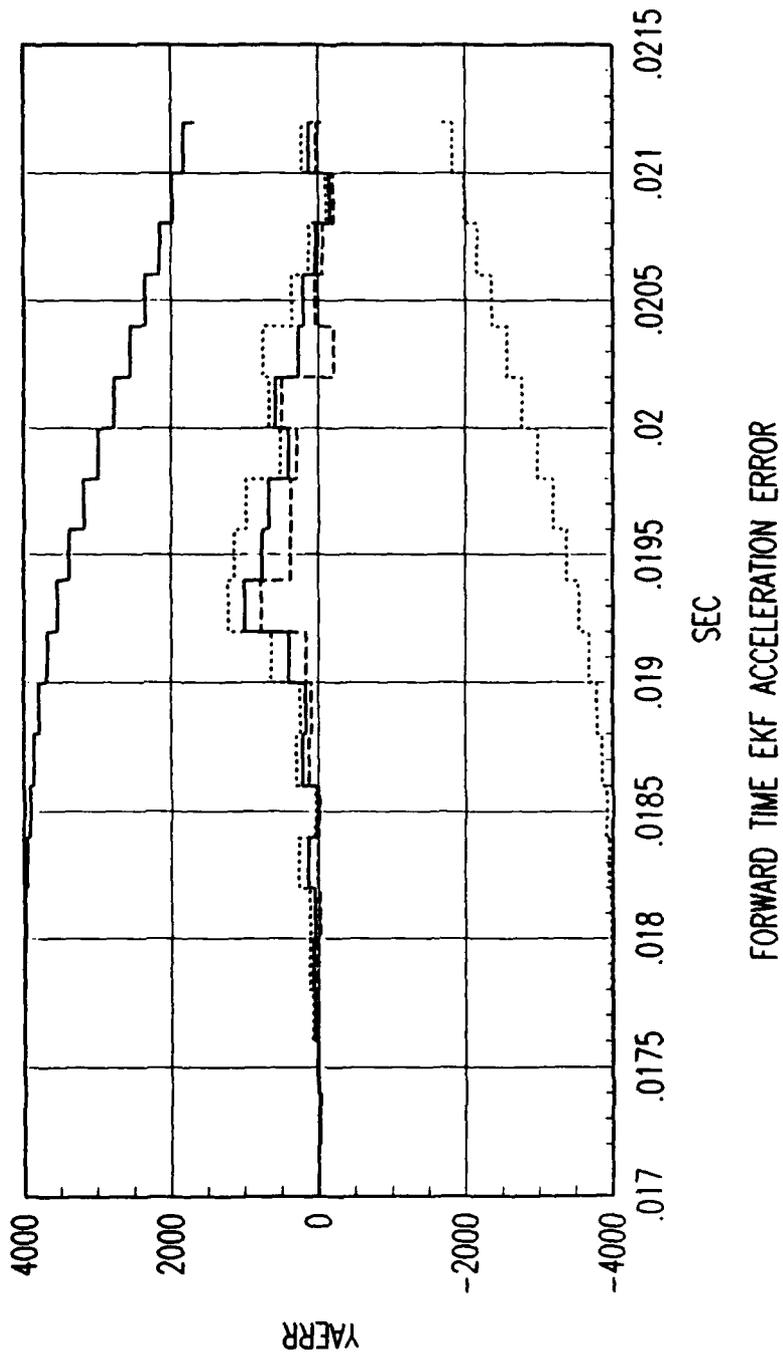


Figure C. 11. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

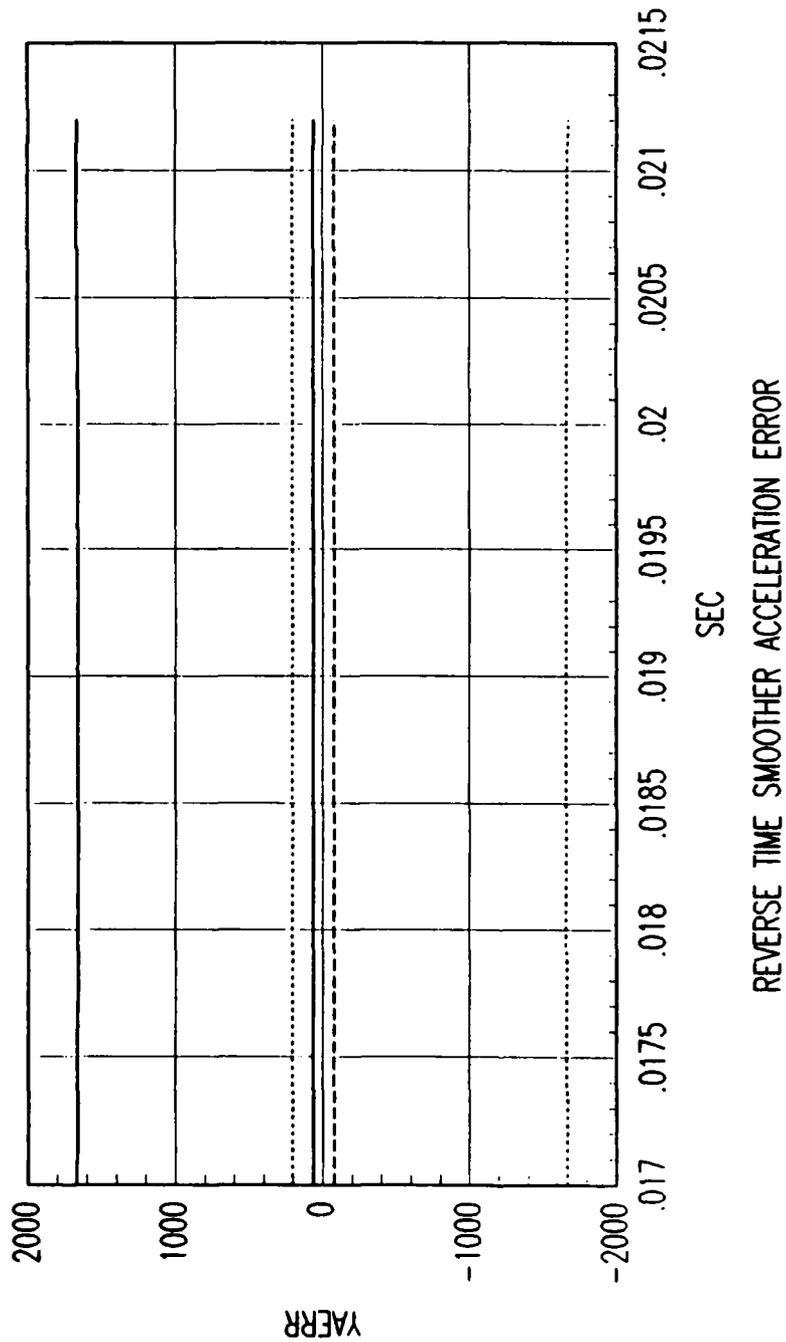


Figure C. 12. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

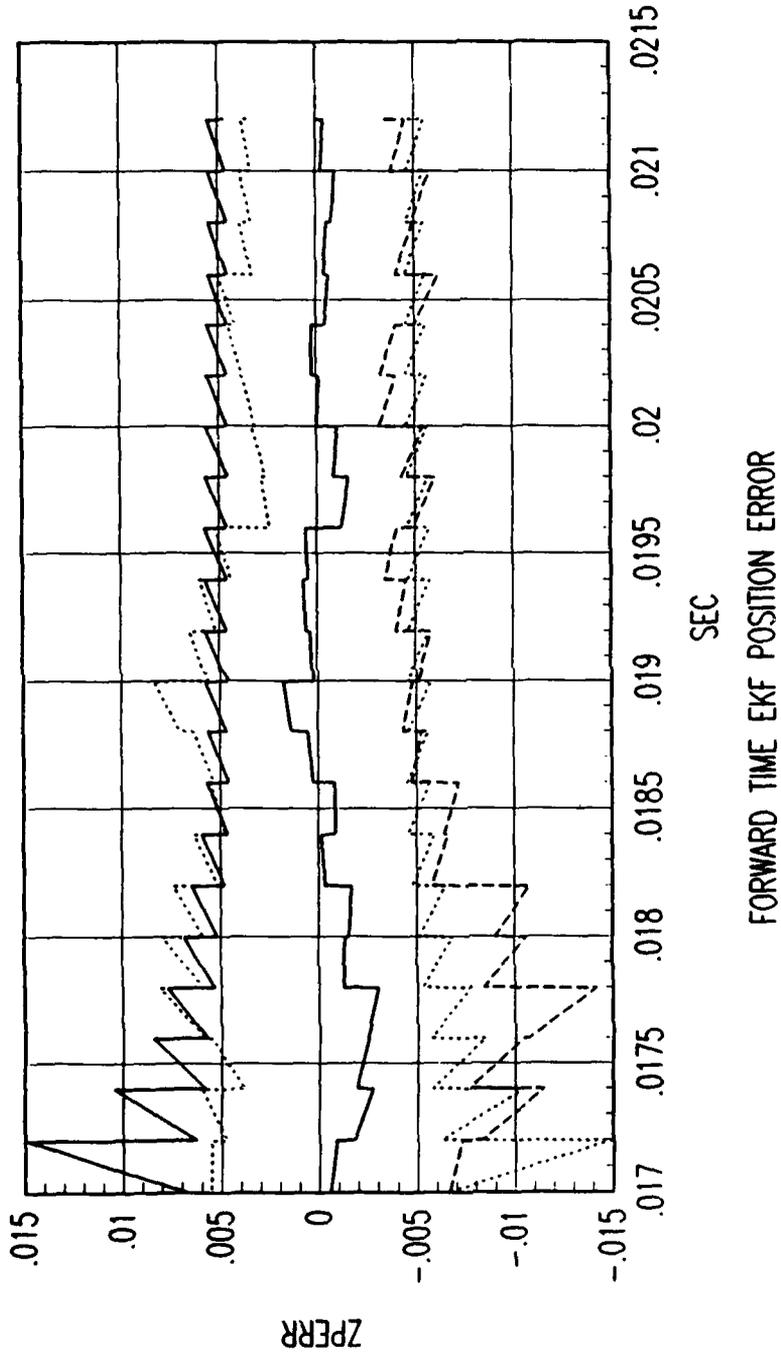


Figure C. 13. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

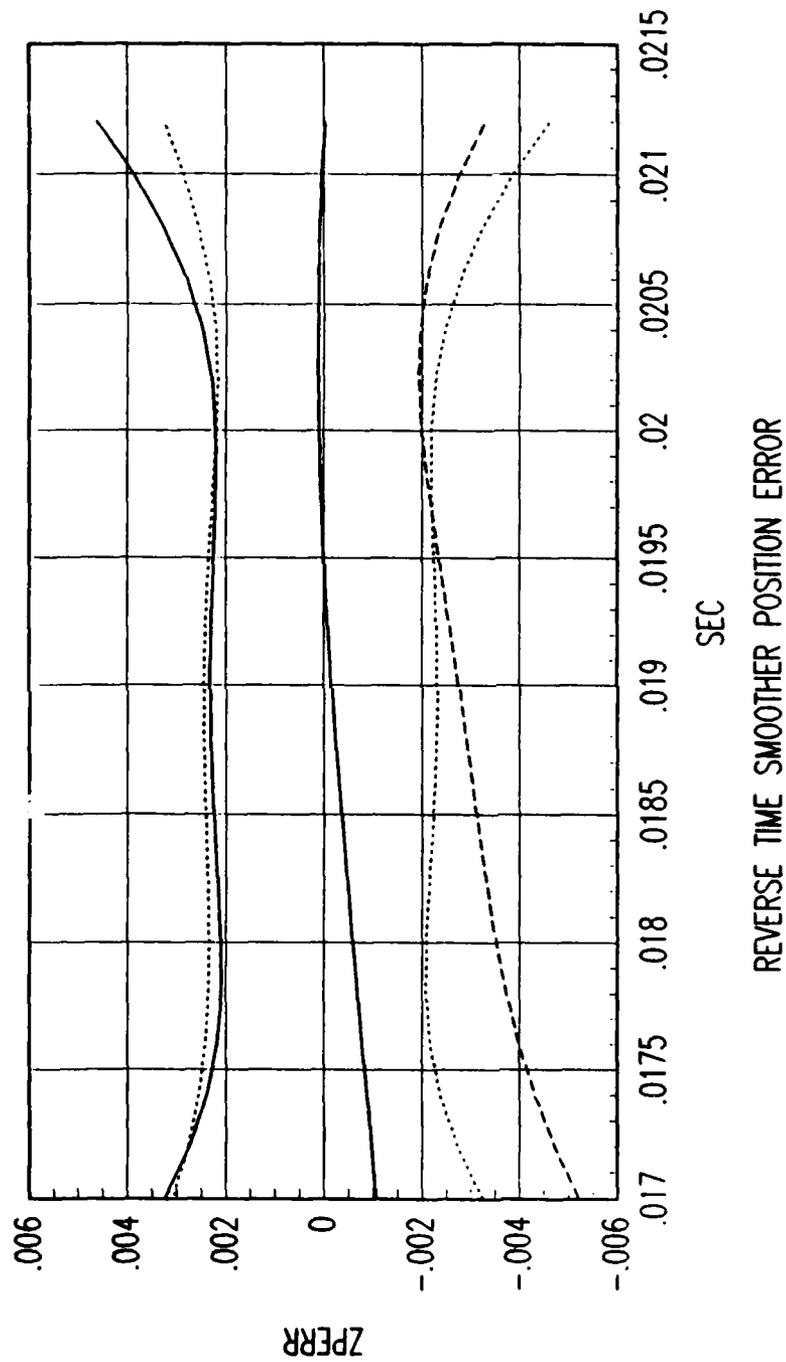


Figure C. 14. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

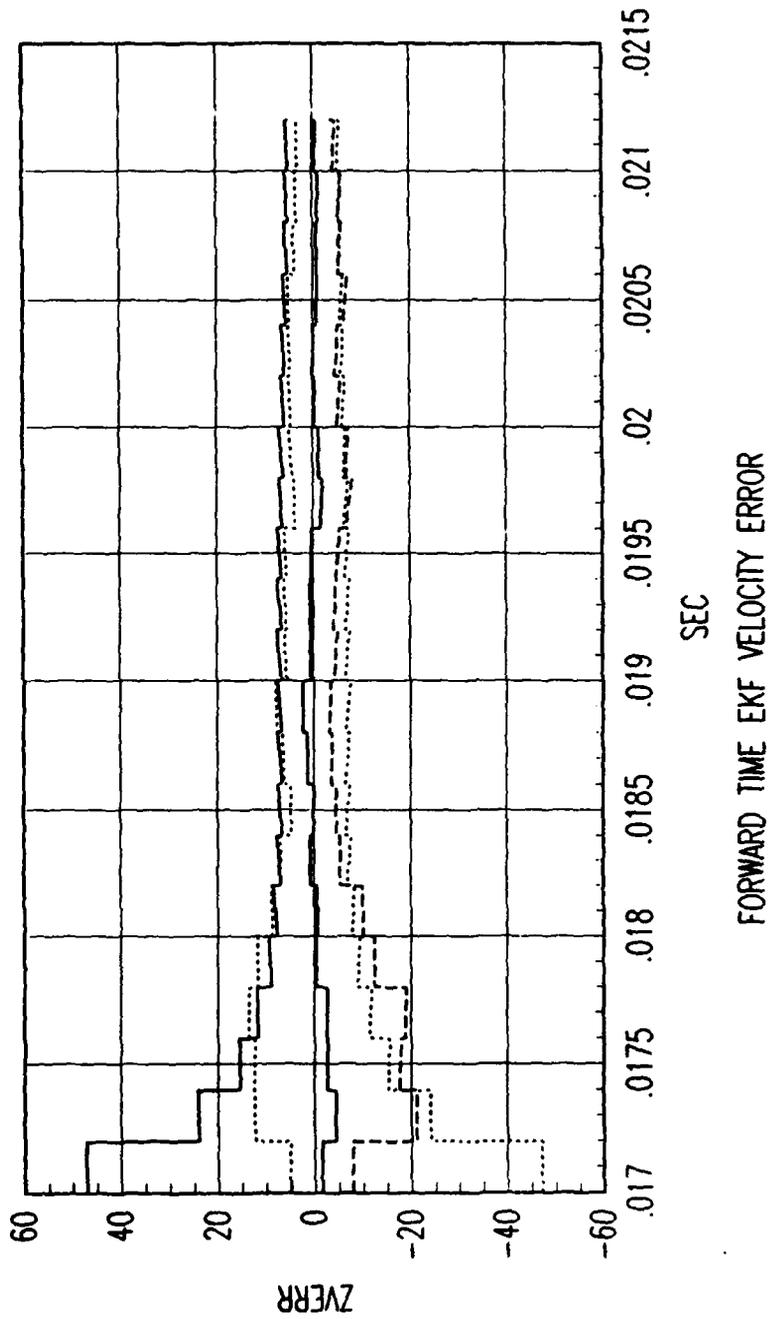


Figure C. 15. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

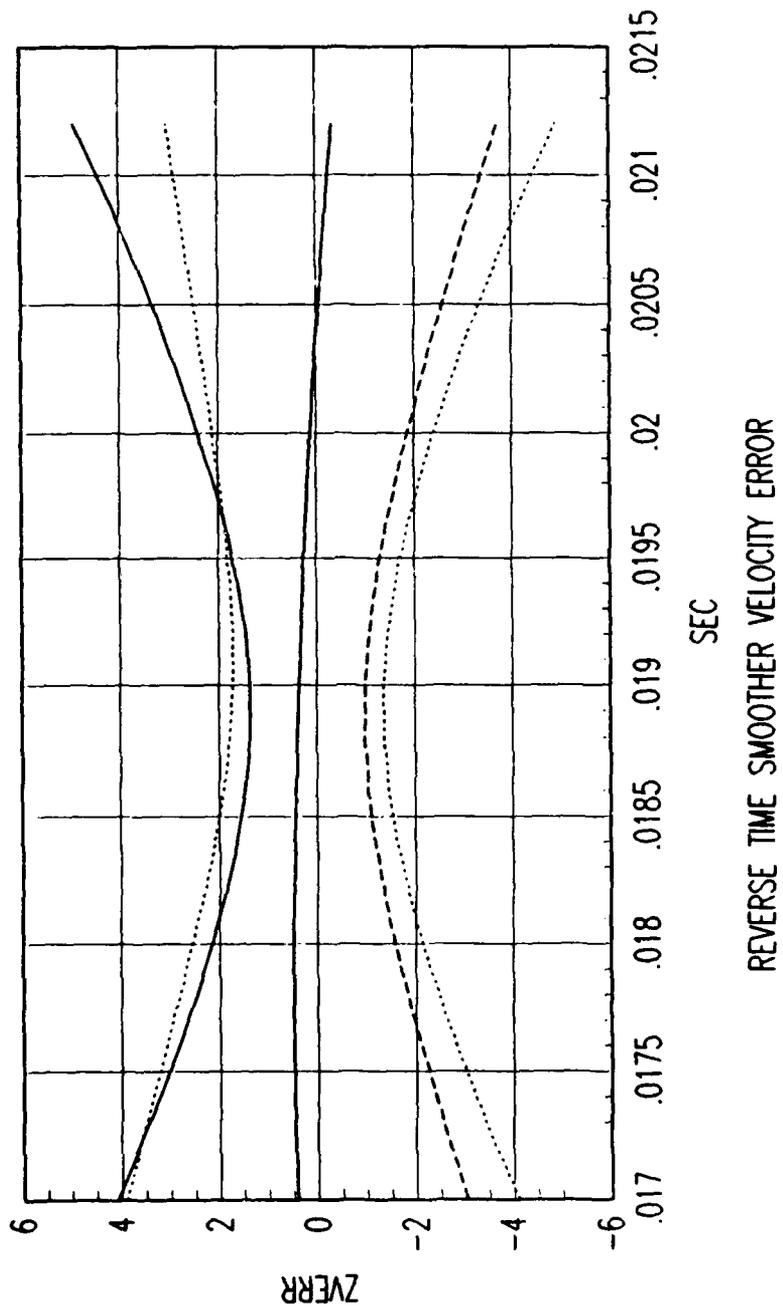


Figure C. 16. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

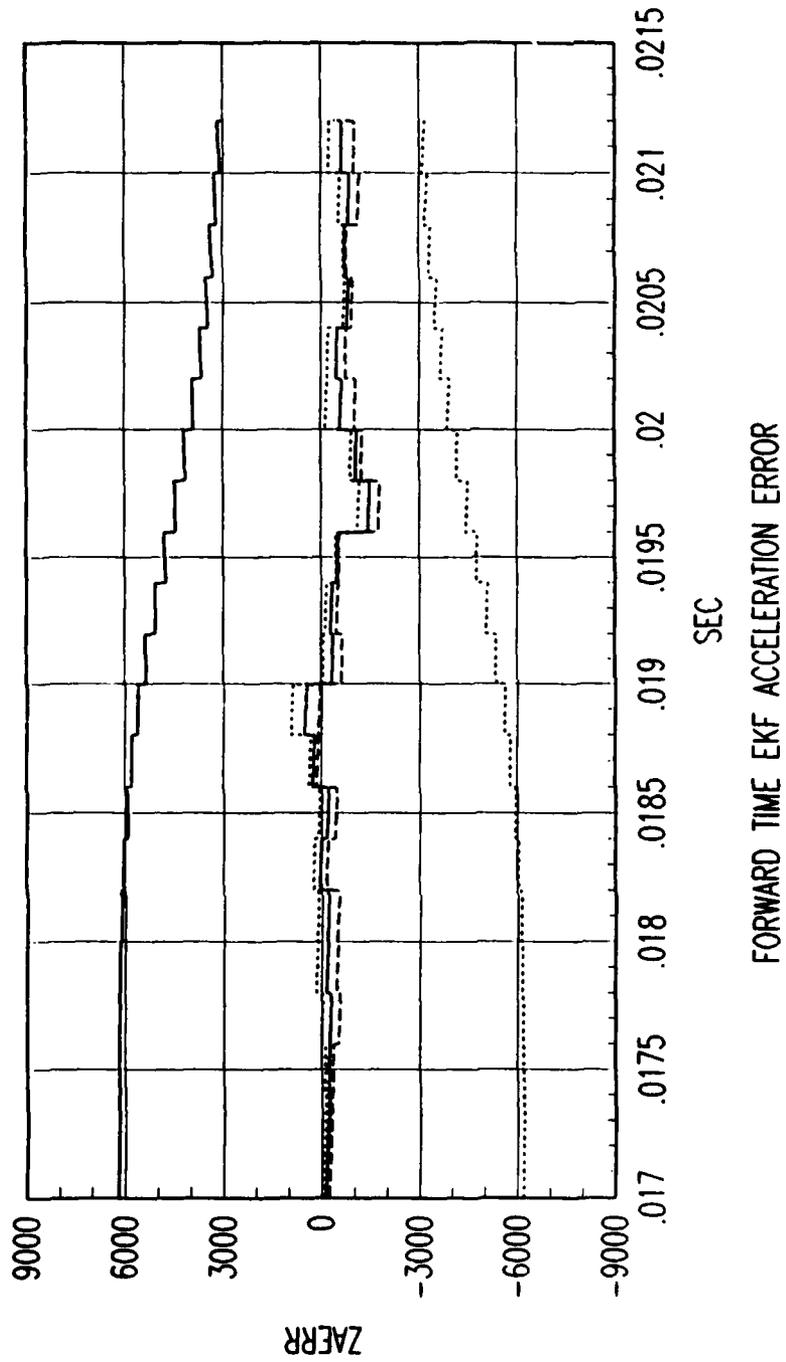


Figure C. 17. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

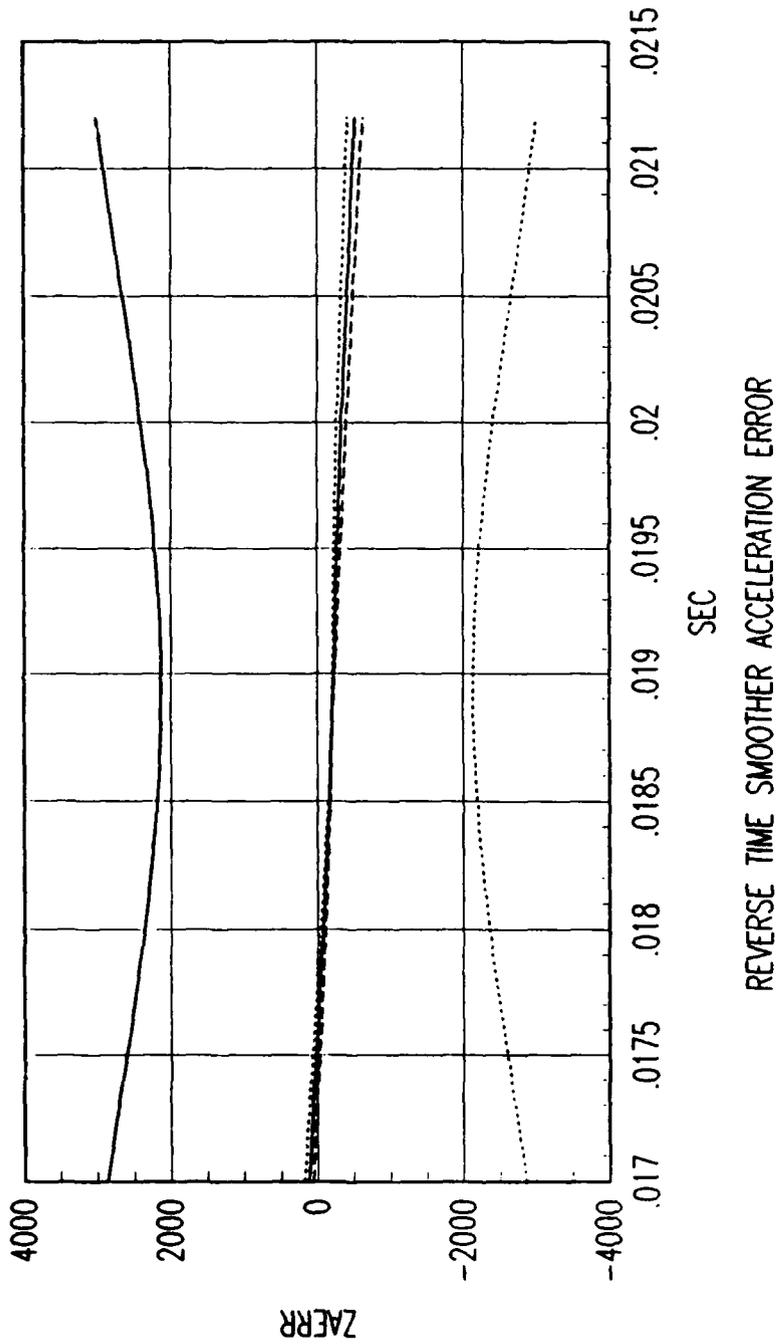


Figure C. 18. Tracking Error Plot, Category 1, $v(t_0) = 3000$ ft/sec.

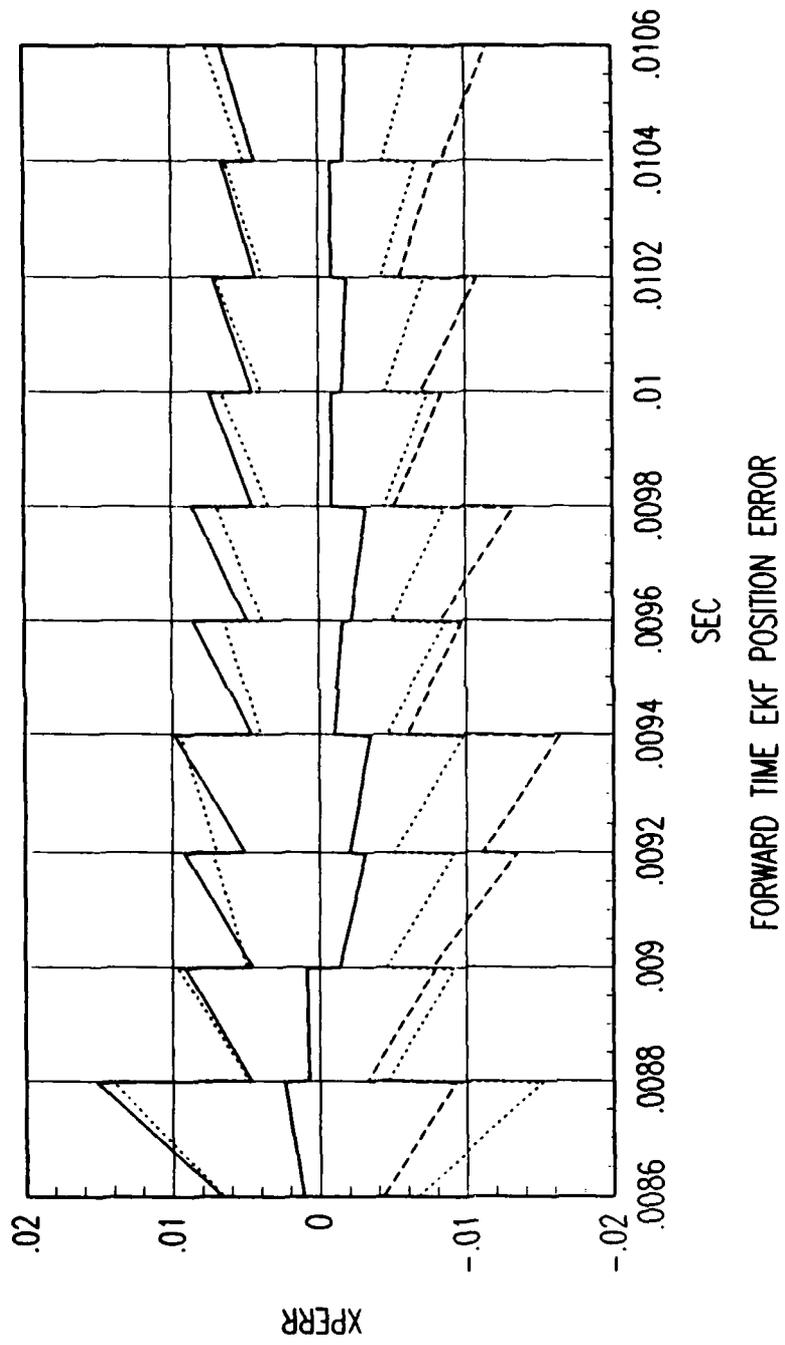


Figure C. 19. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

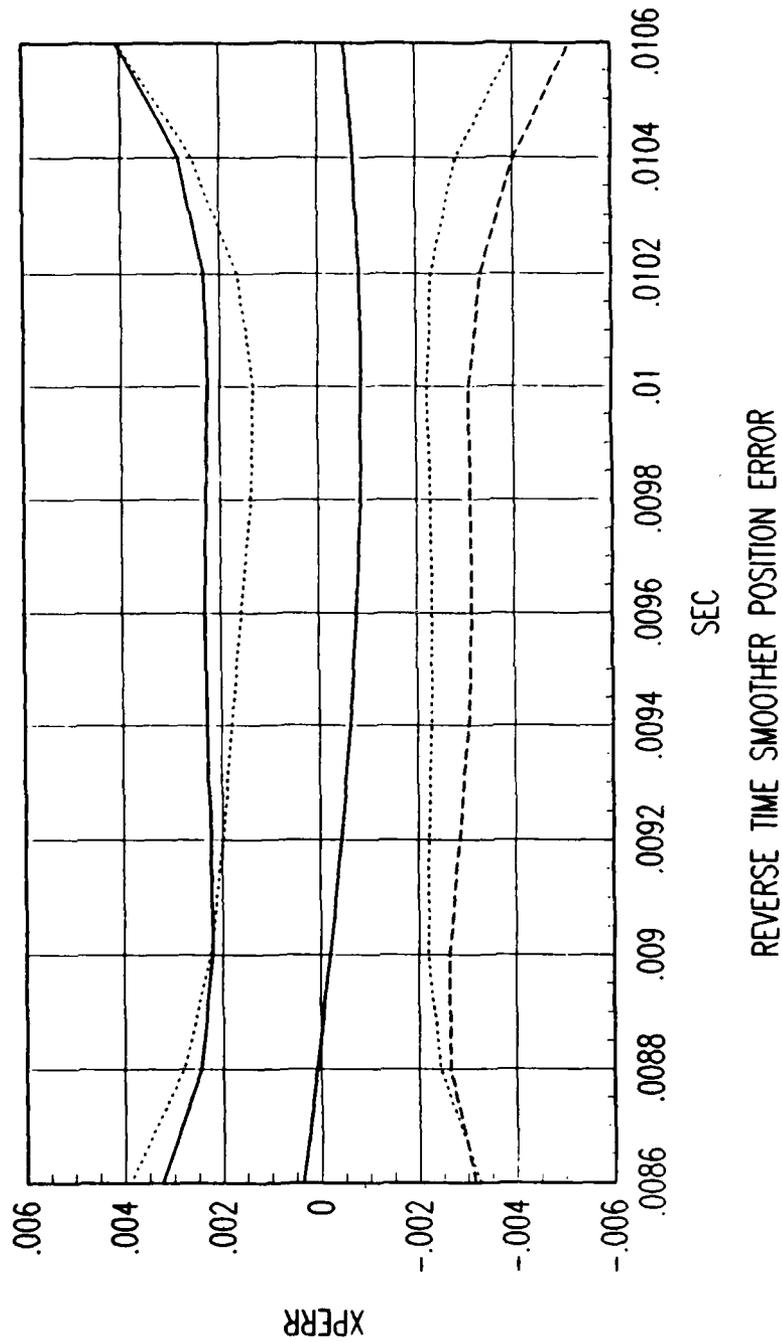


Figure C. 20. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

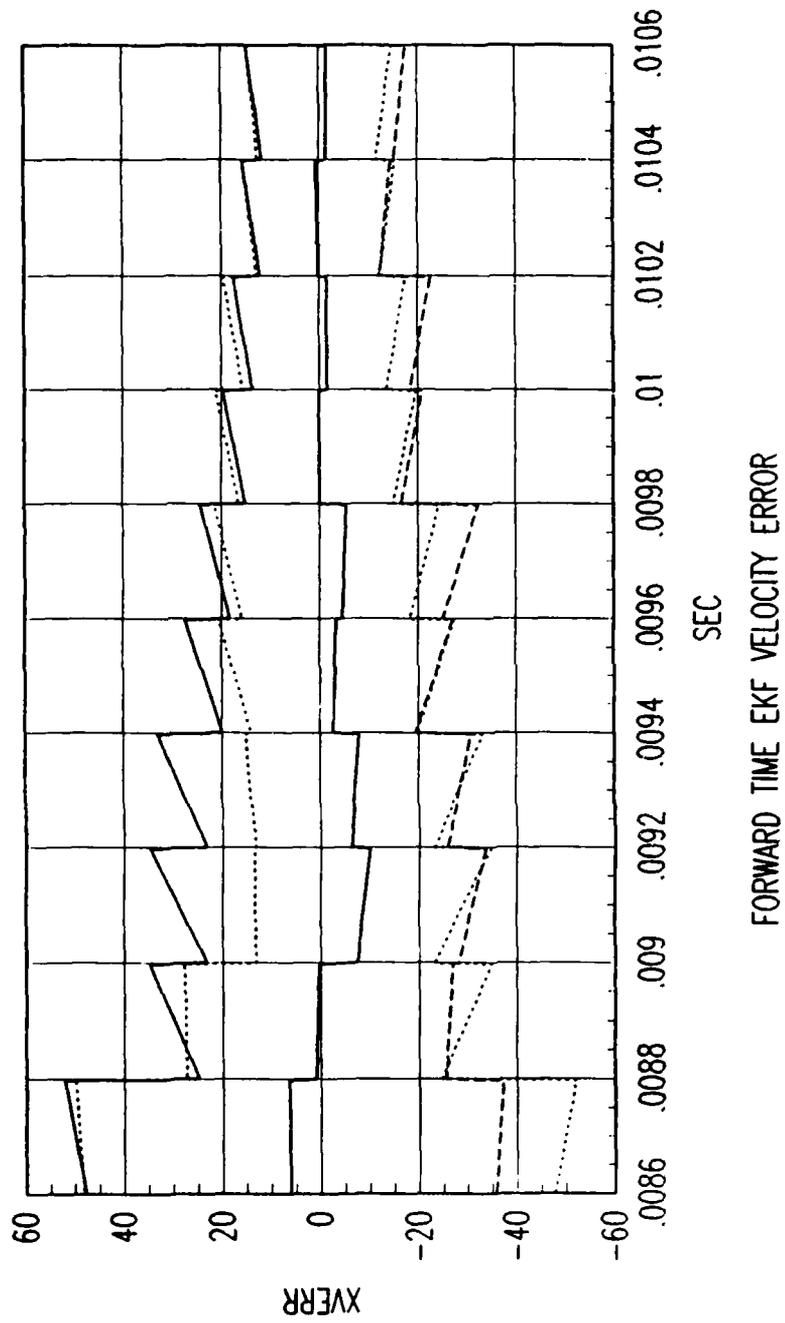


Figure C. 21. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

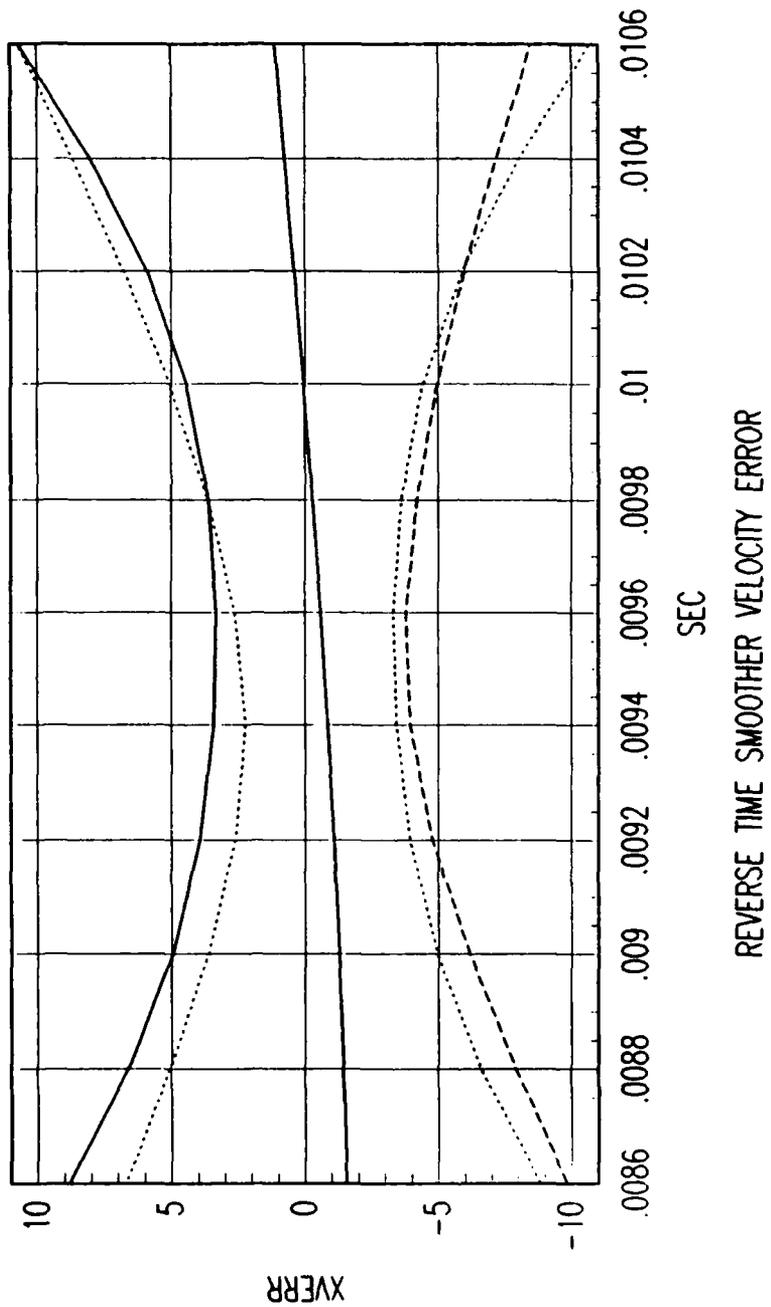


Figure C. 22. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

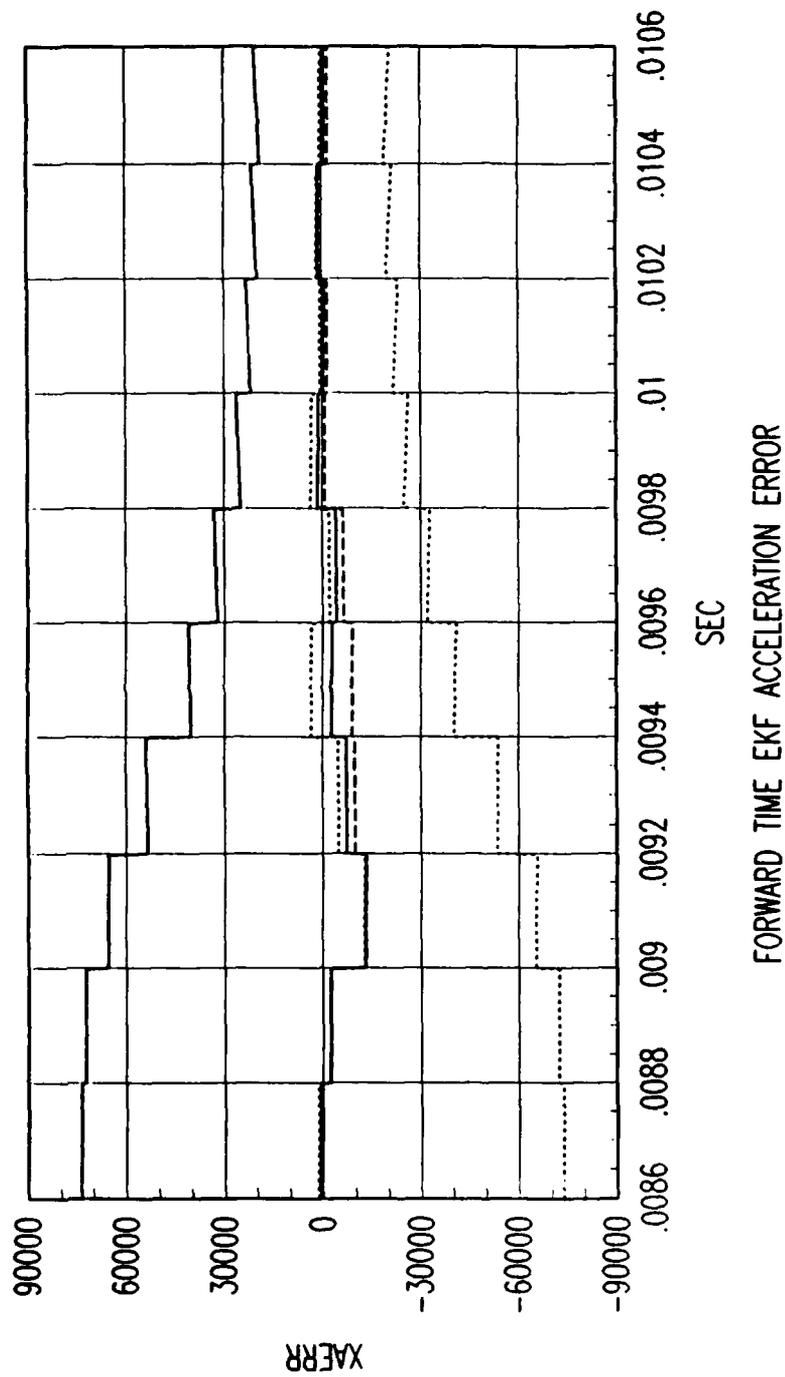


Figure C. 23. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

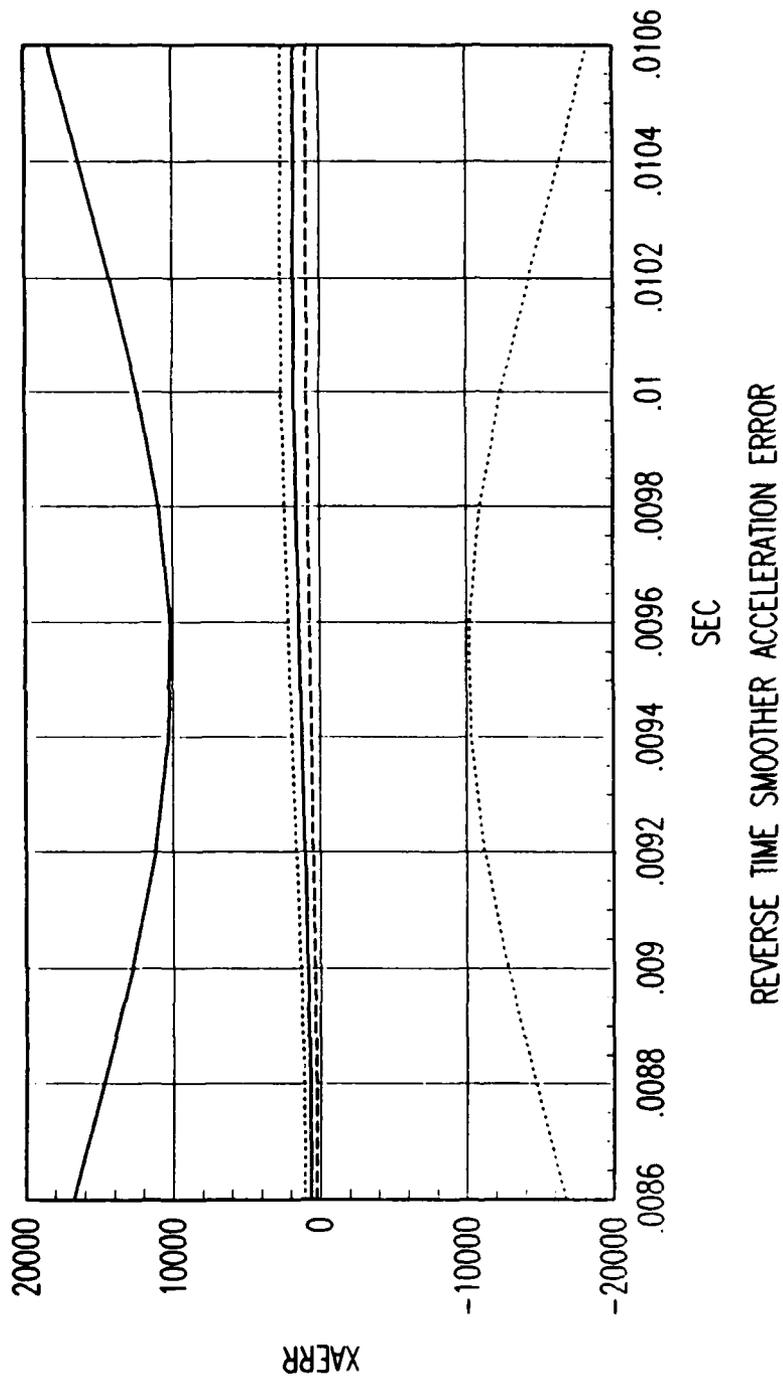


Figure C. 24. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

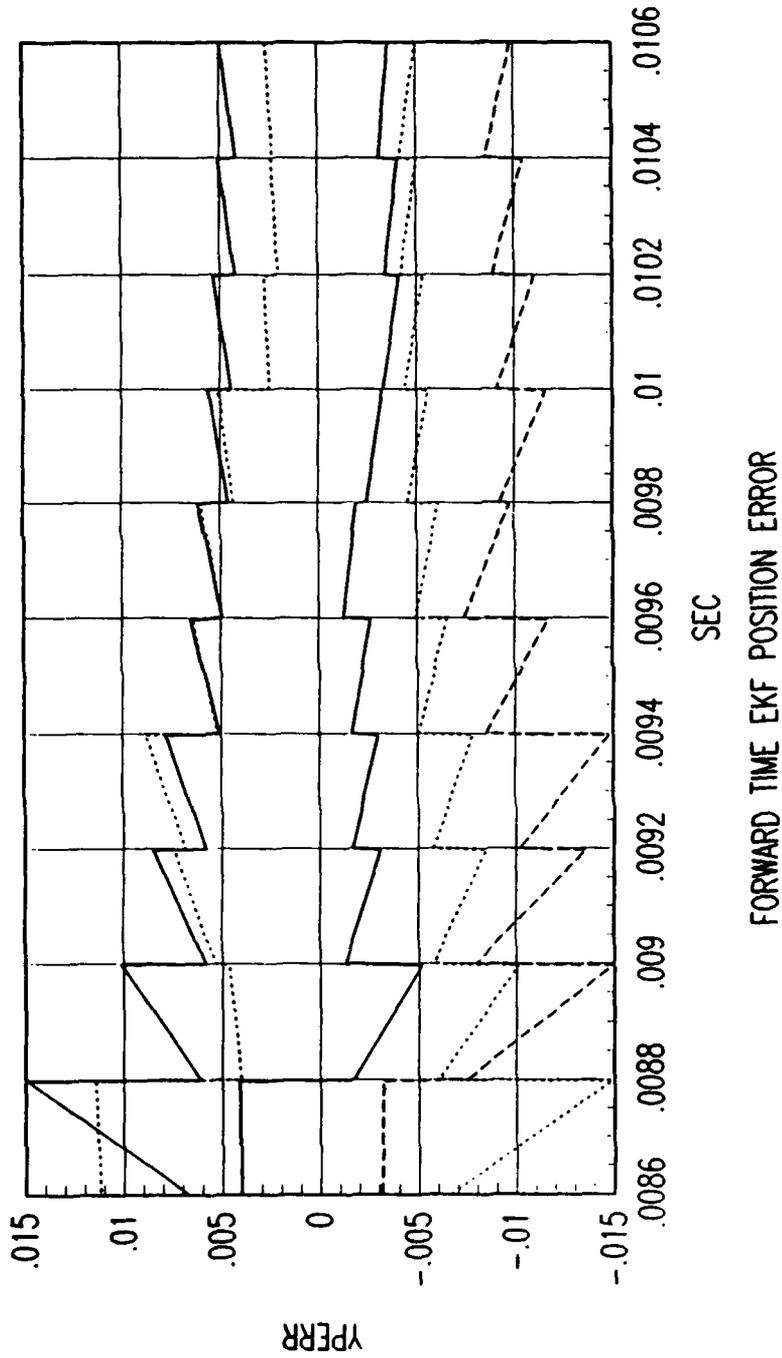


Figure C. 25. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

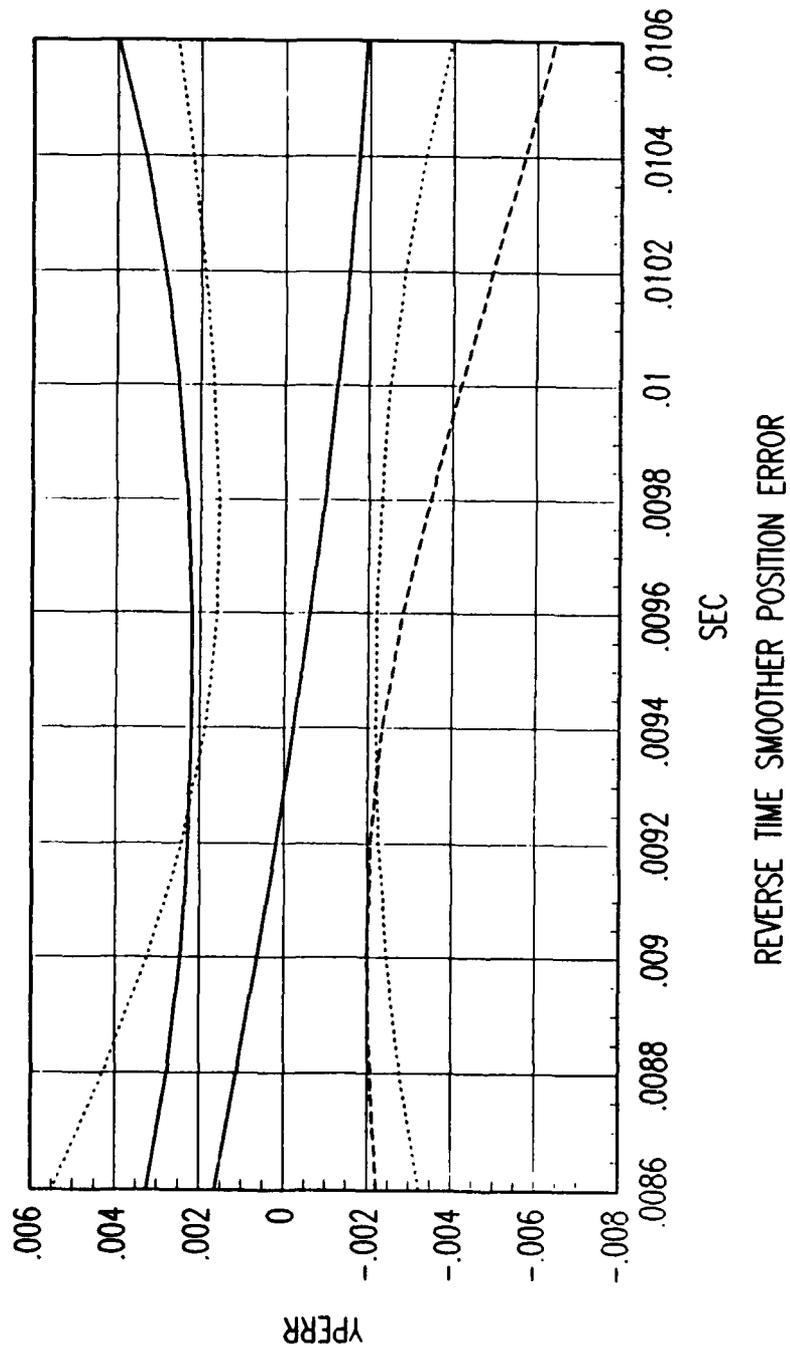


Figure C. 26. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

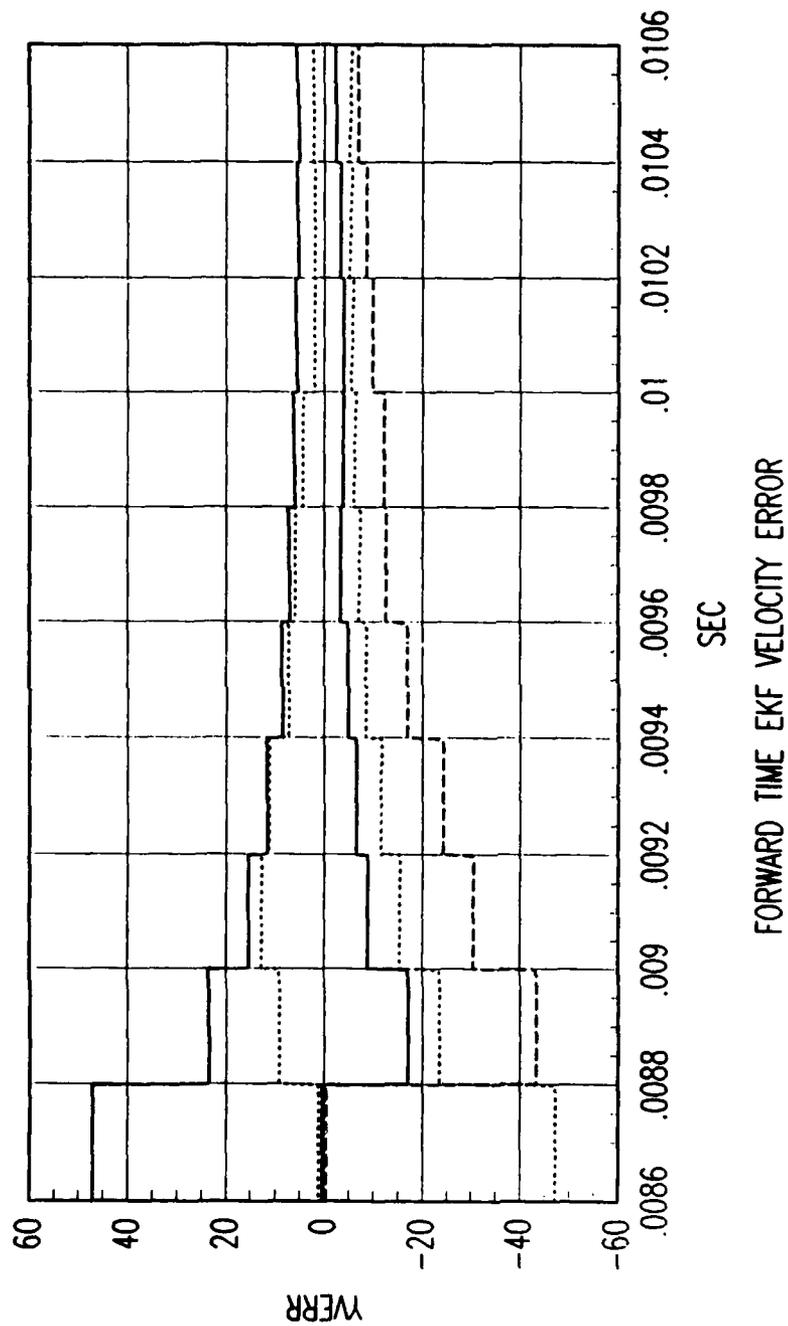


Figure C. 27. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

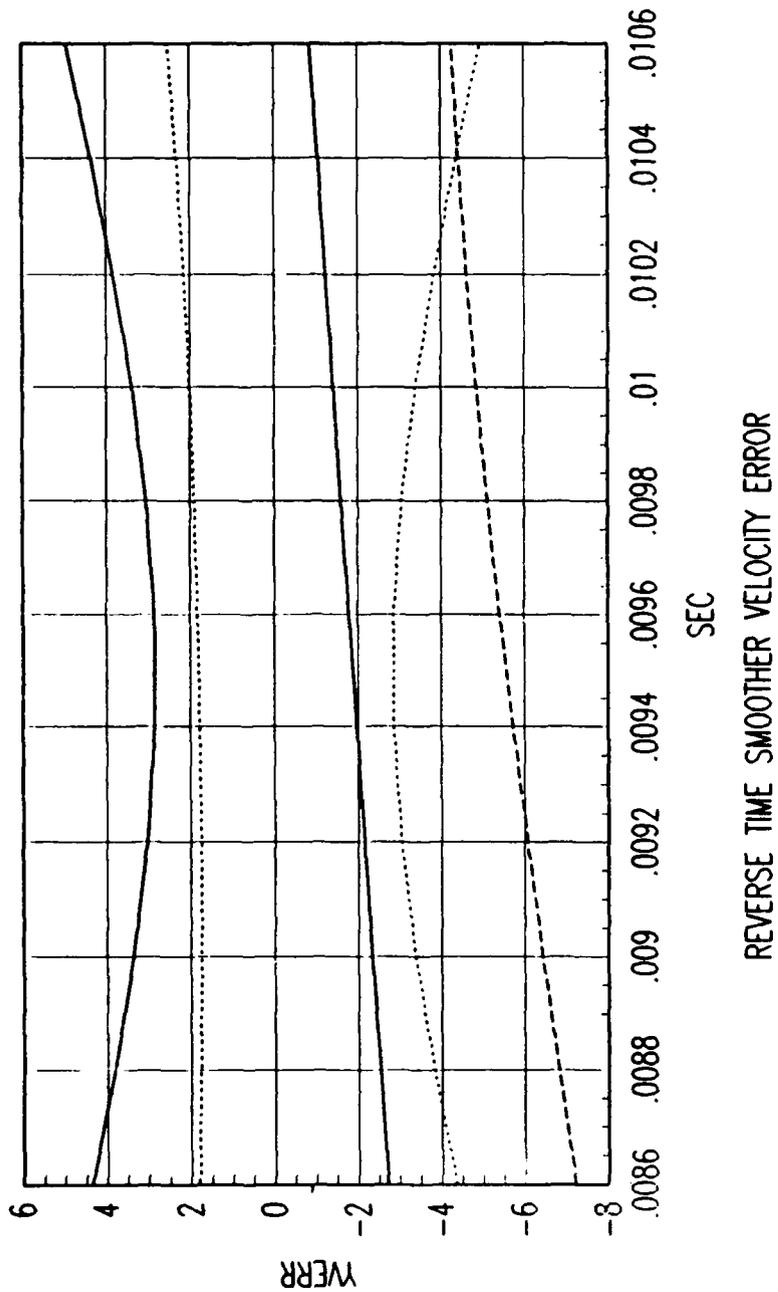


Figure C. 28. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

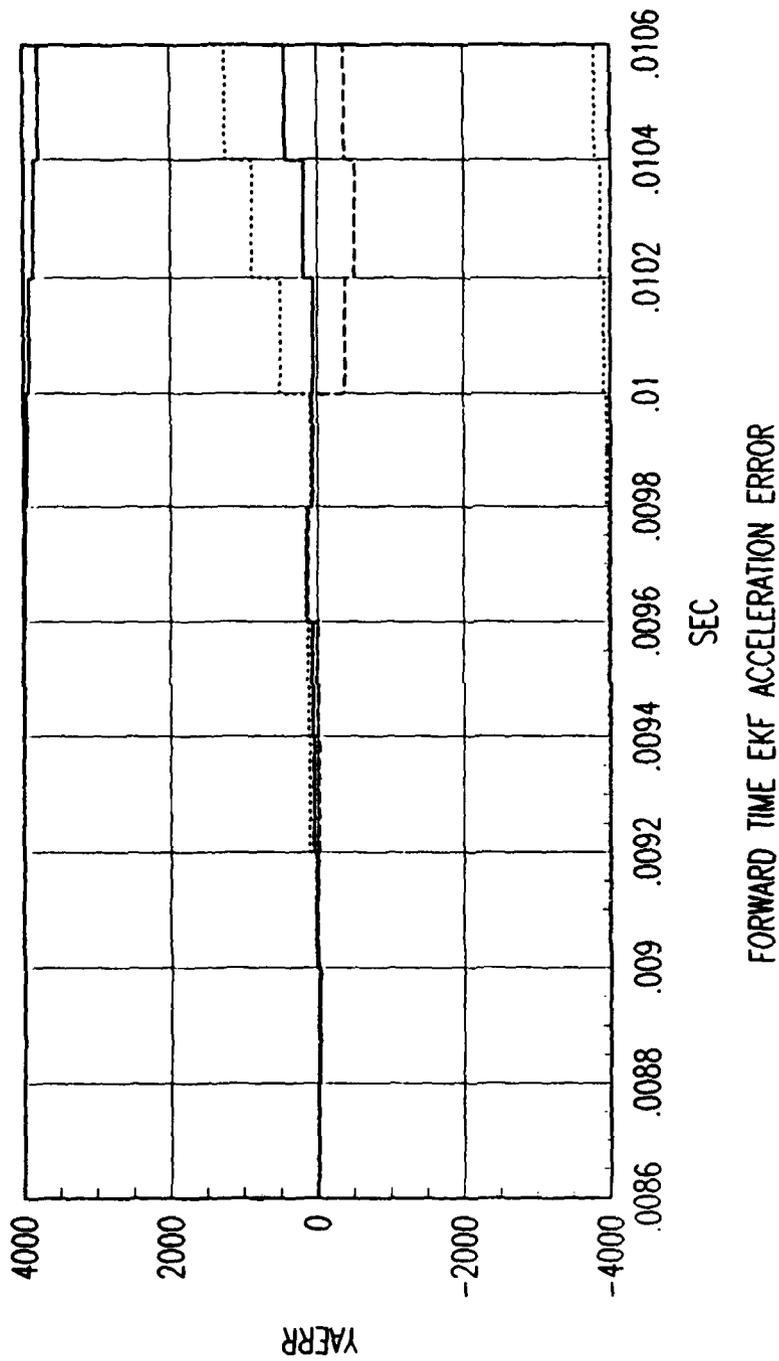


Figure C. 29. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

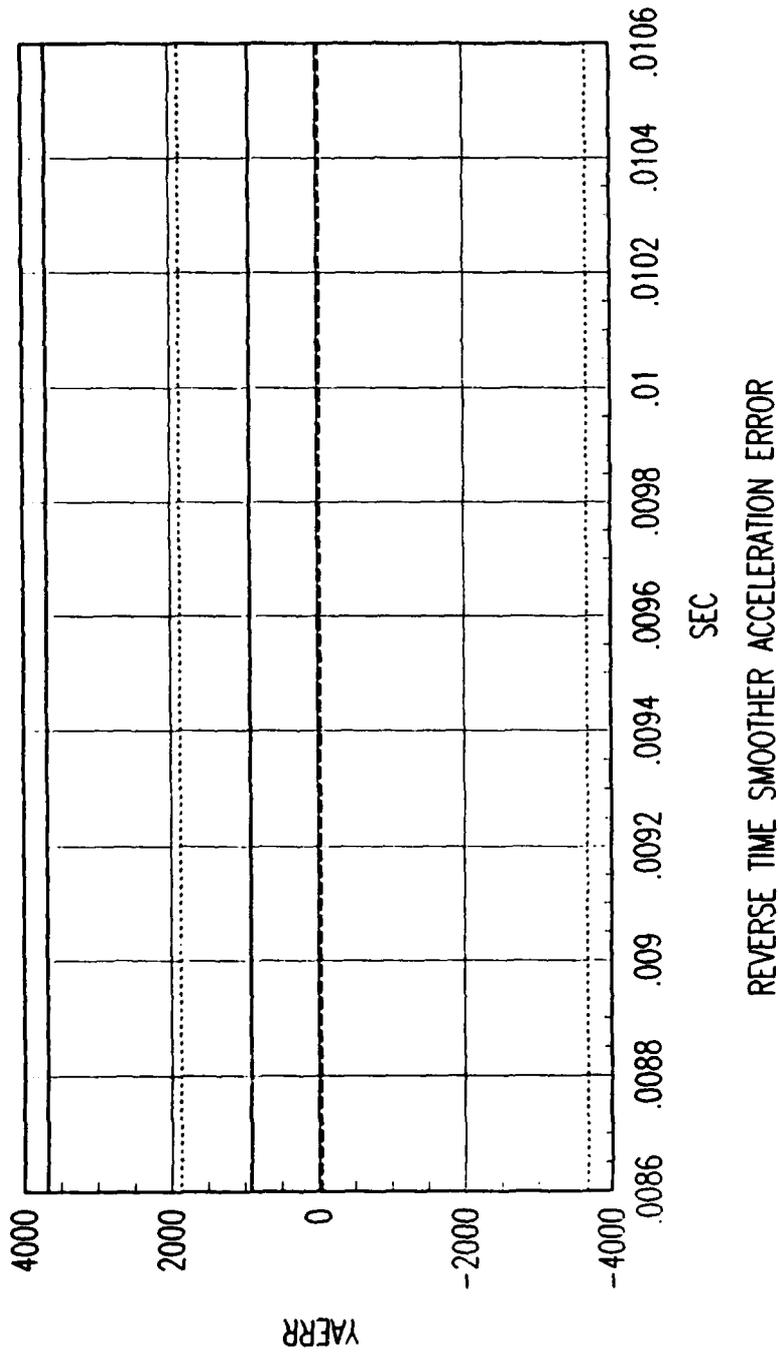


Figure C. 30. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

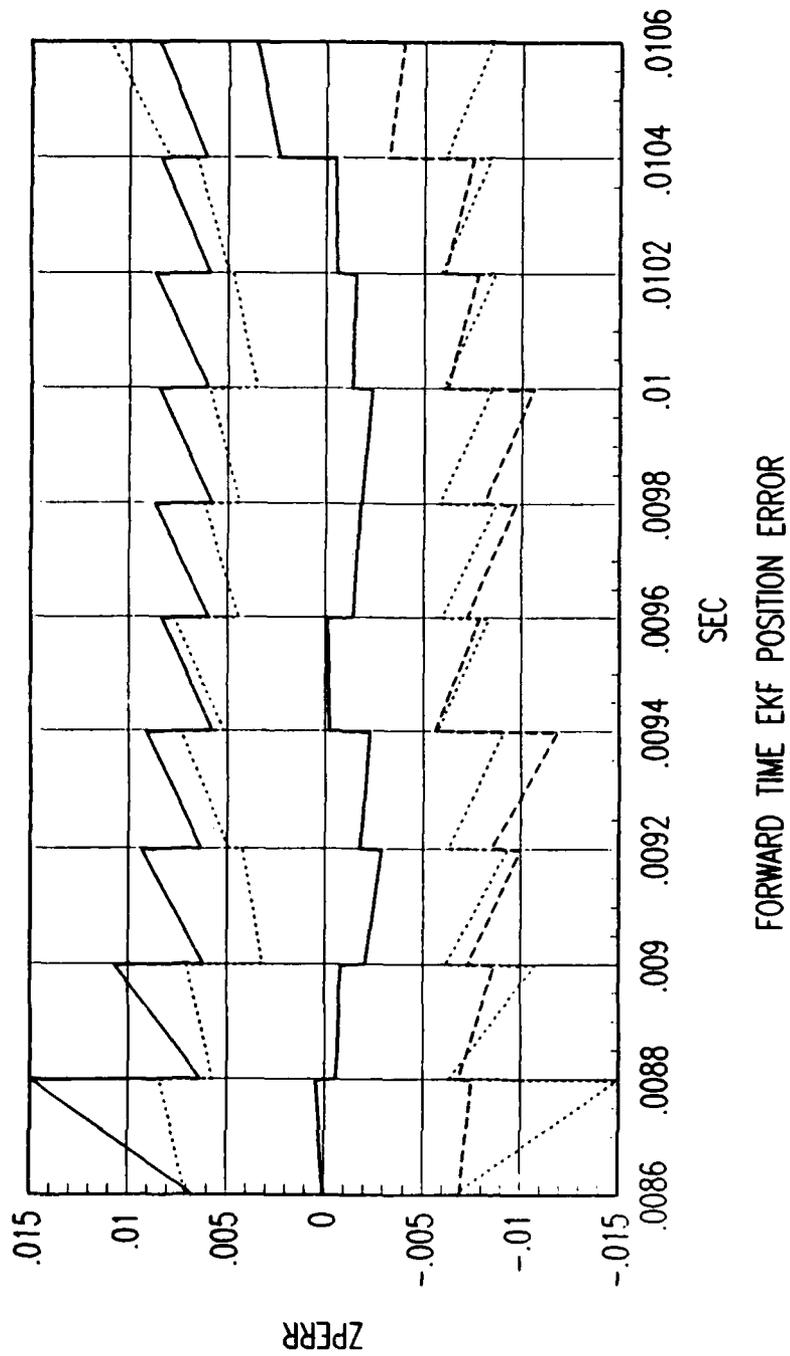


Figure C. 31. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

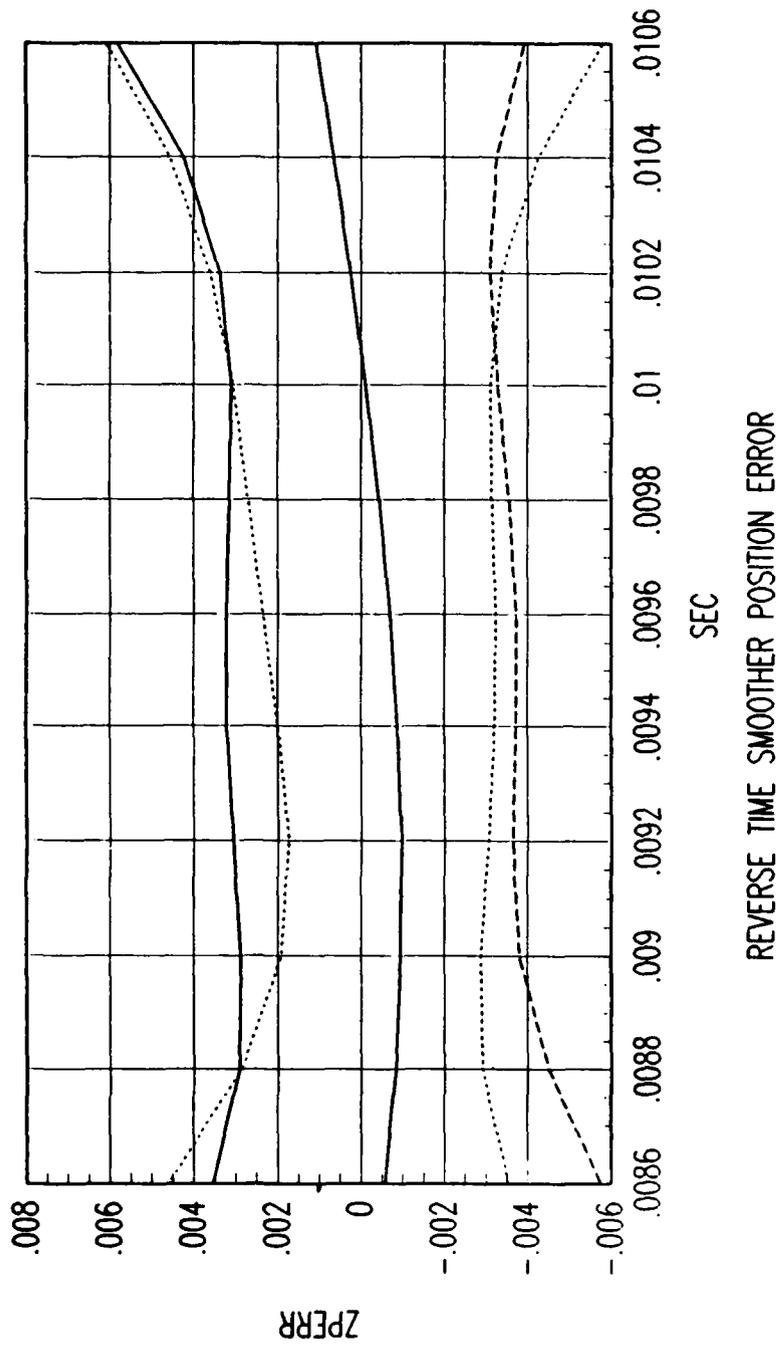


Figure C. 32. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

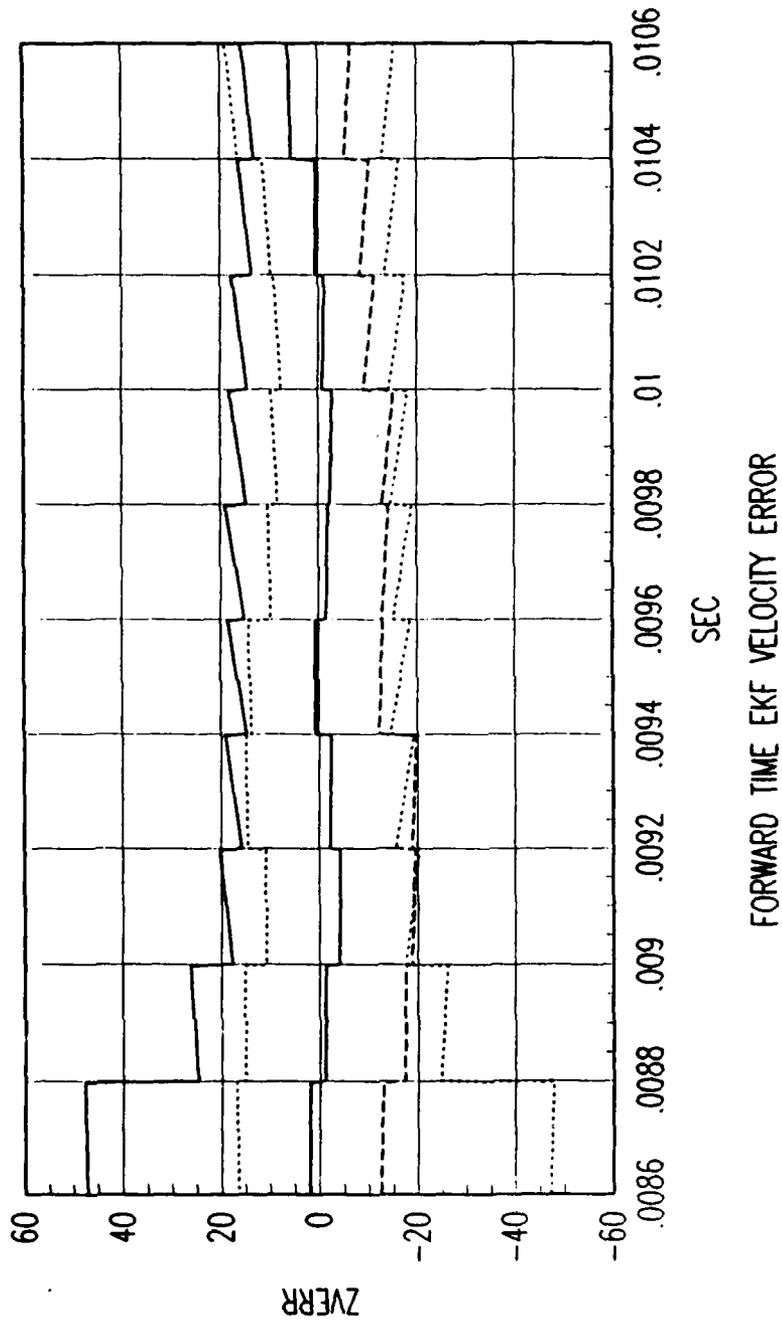


Figure C. 33. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

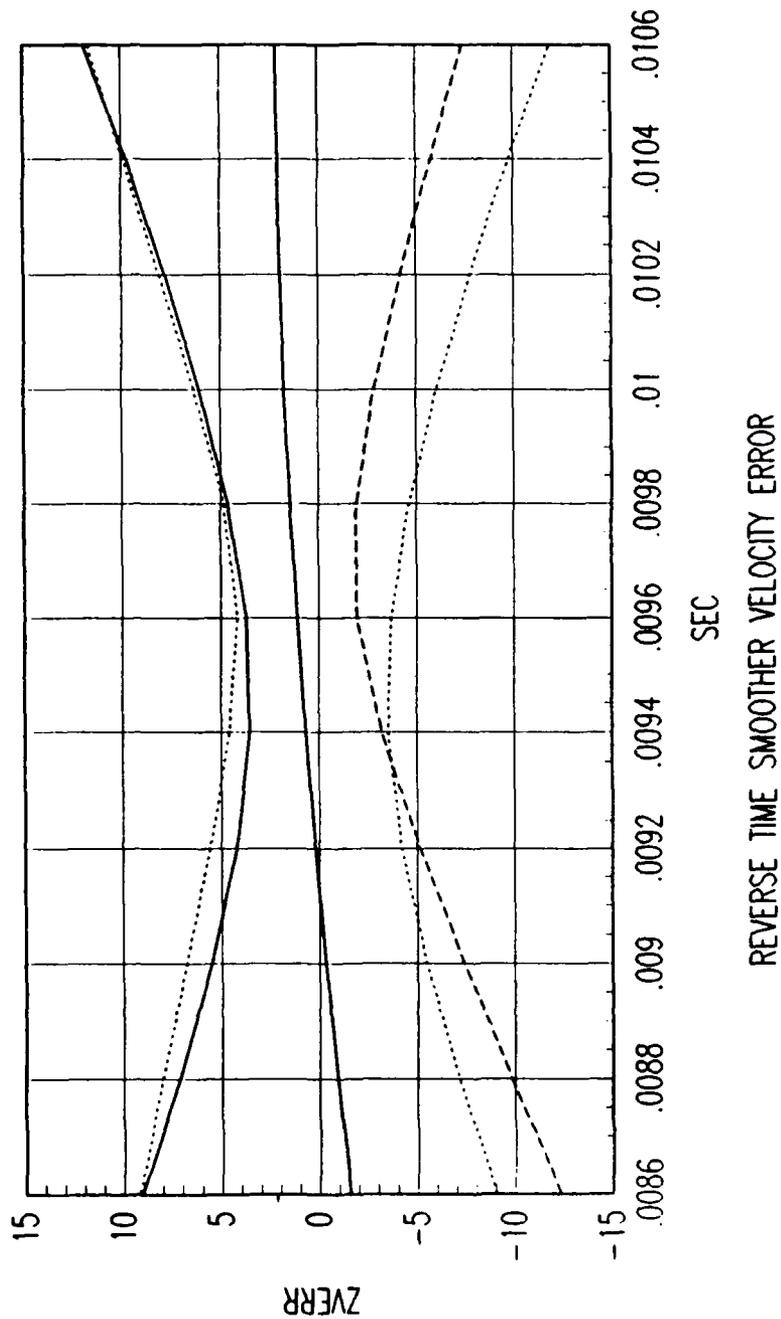


Figure C. 34. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

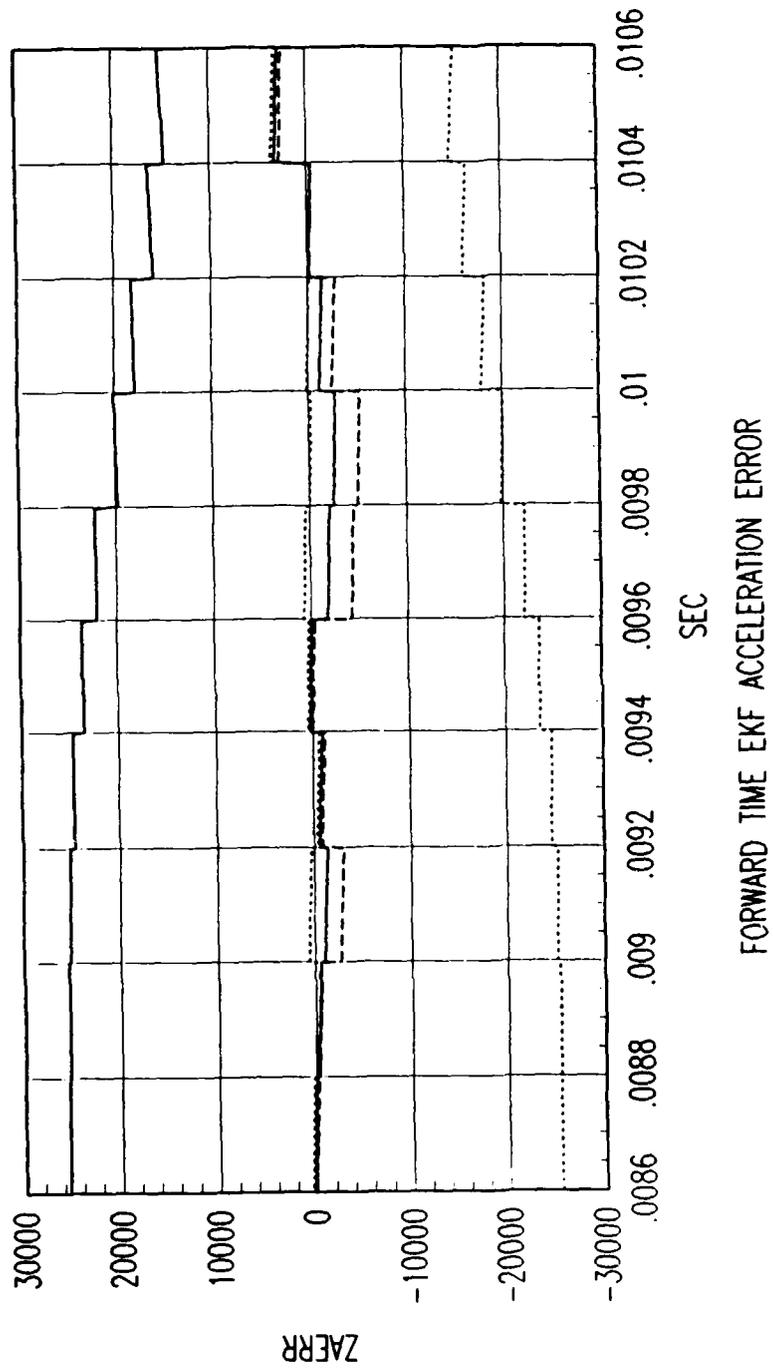


Figure C. 35. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

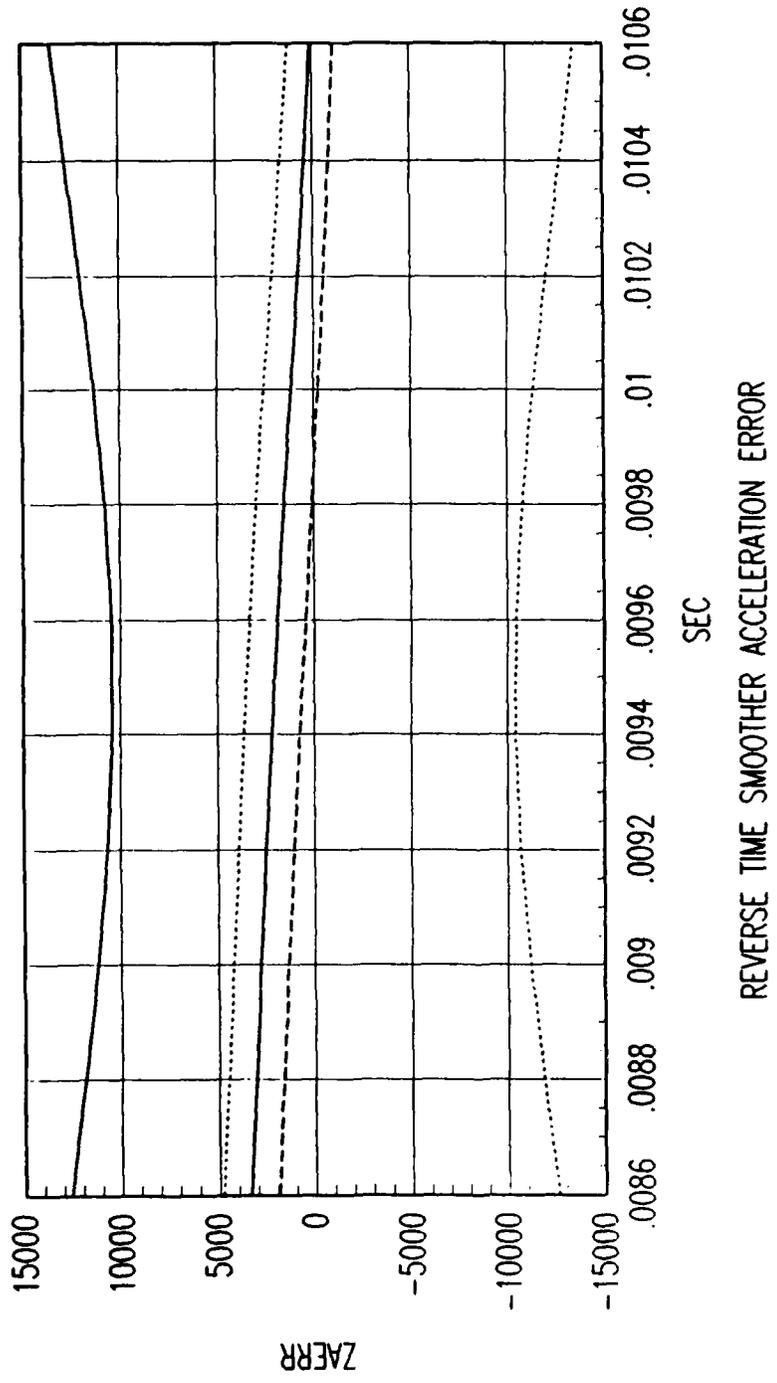


Figure C. 36. Tracking Error Plot, Category 1, $v(t_0) = 6000$ ft/sec.

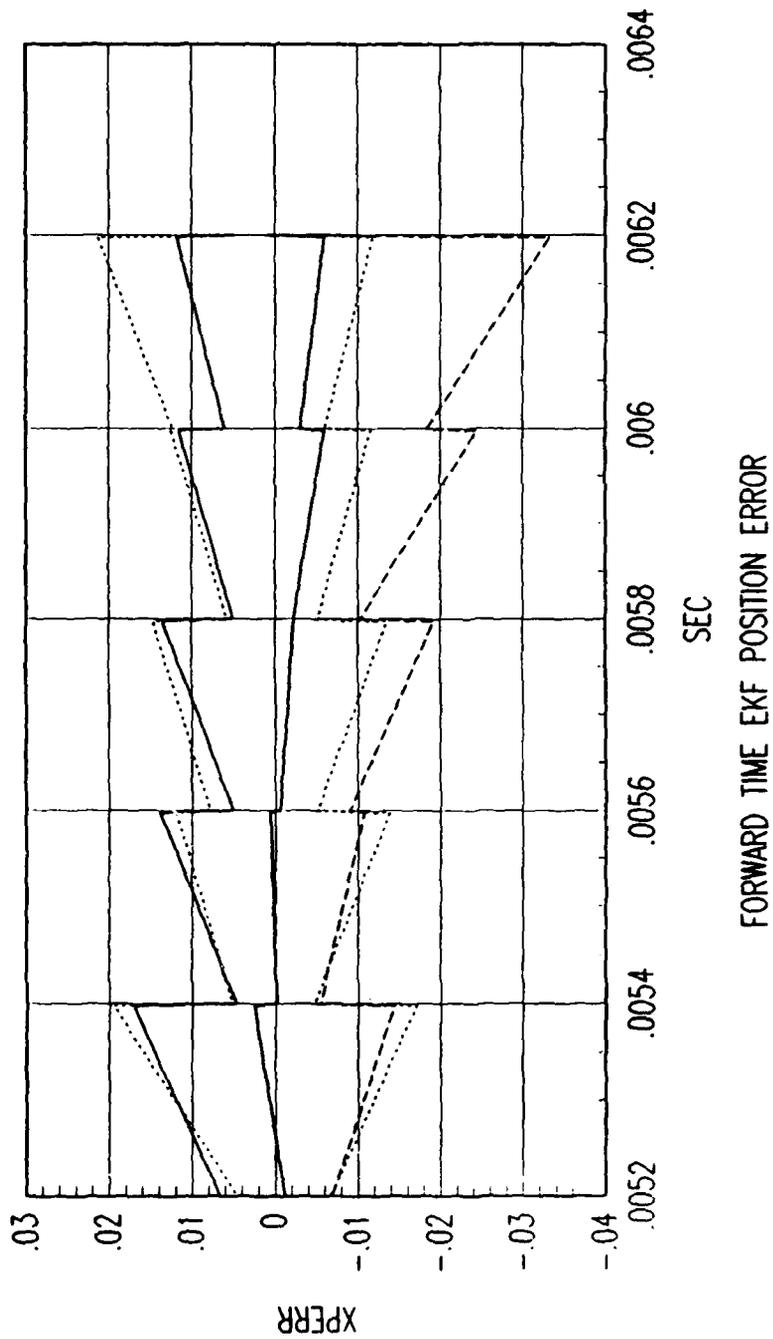


Figure C. 37. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

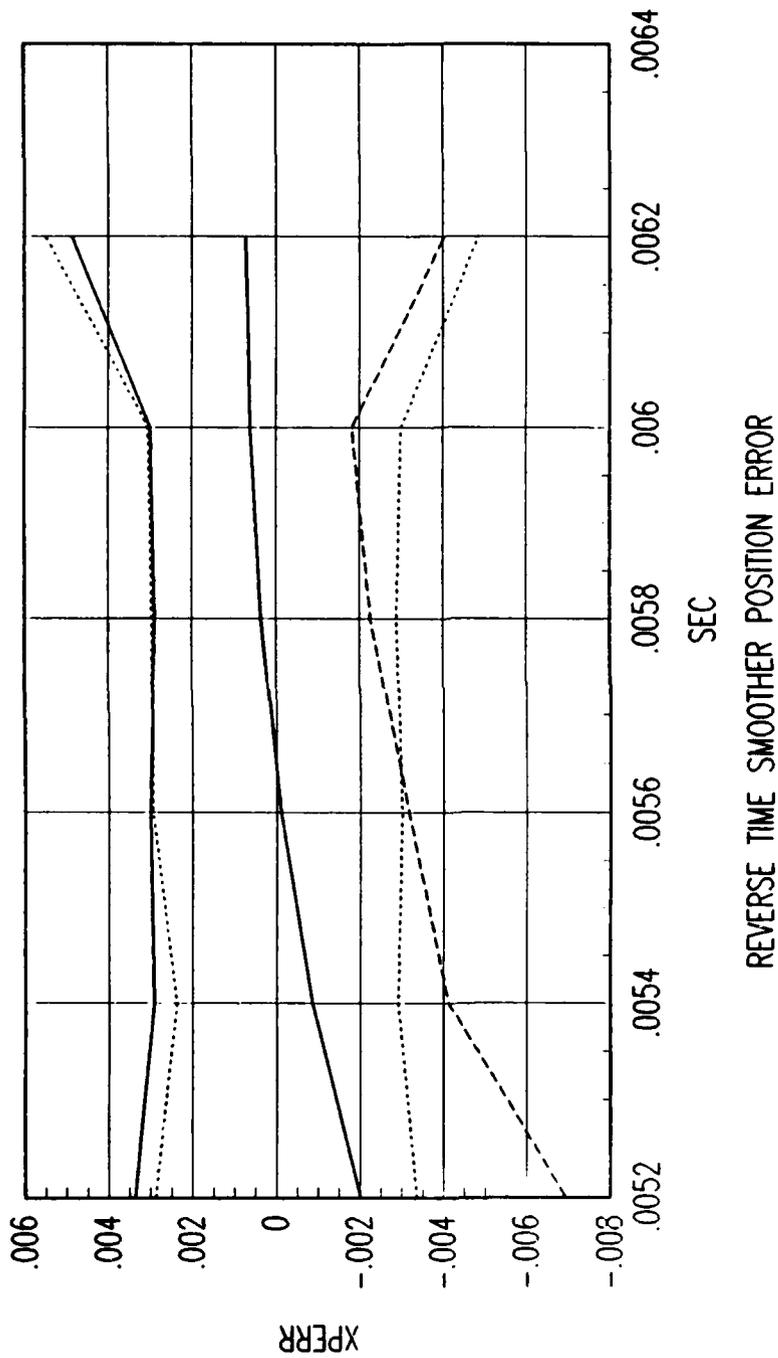


Figure C. 38. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

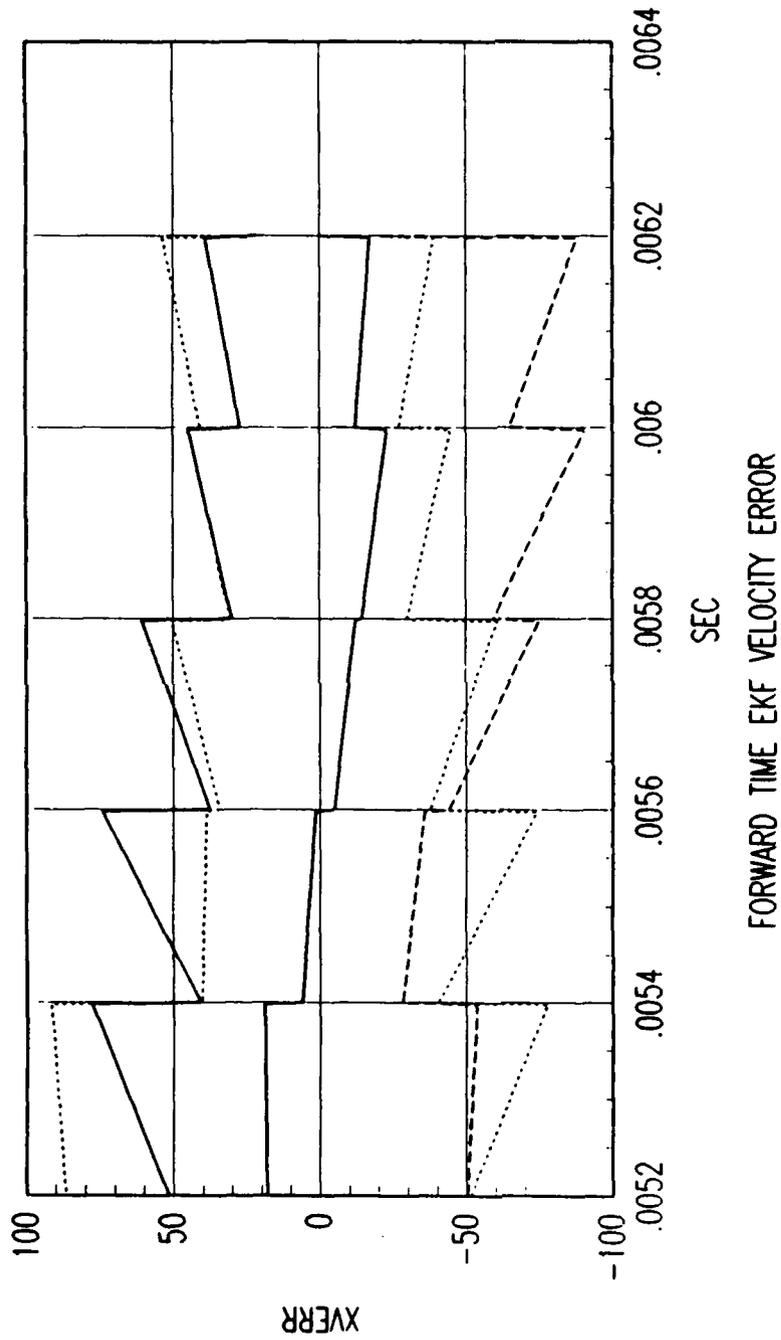


Figure C. 39. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

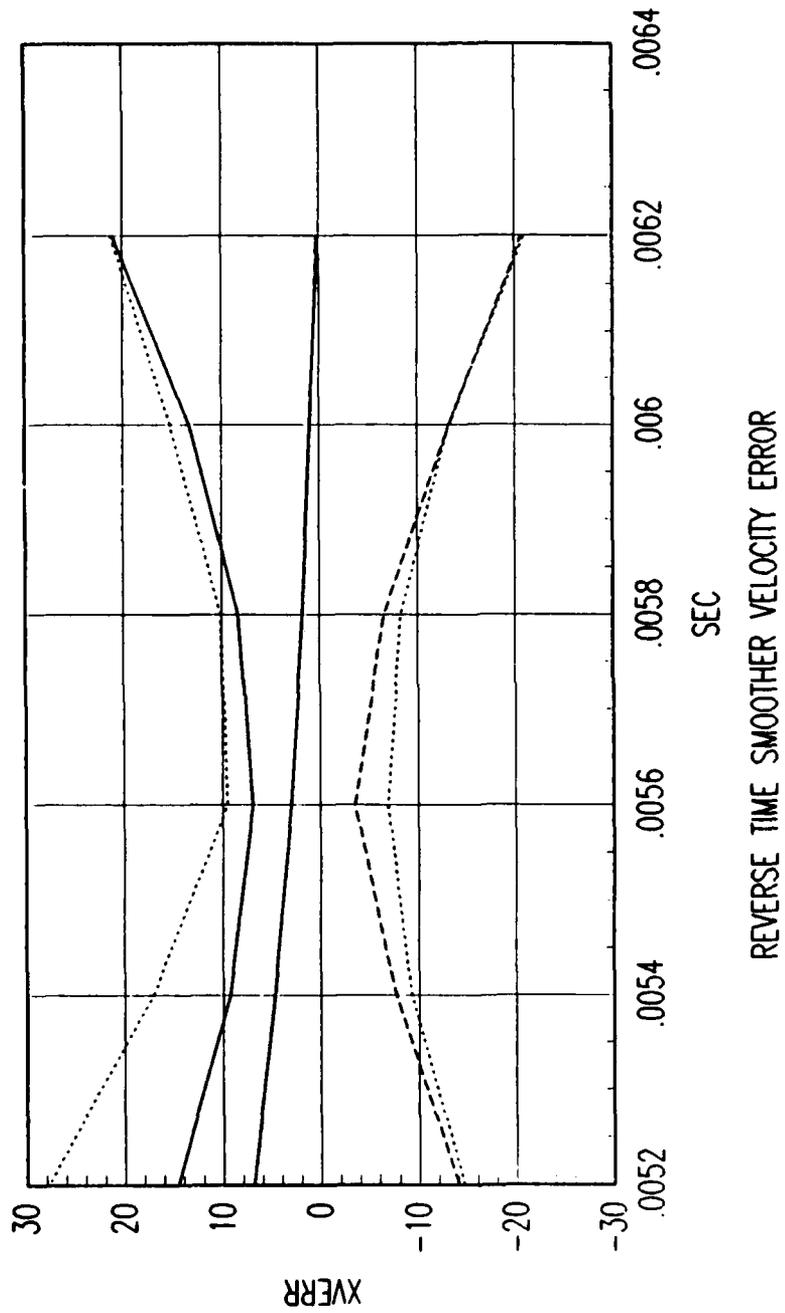


Figure C. 40. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

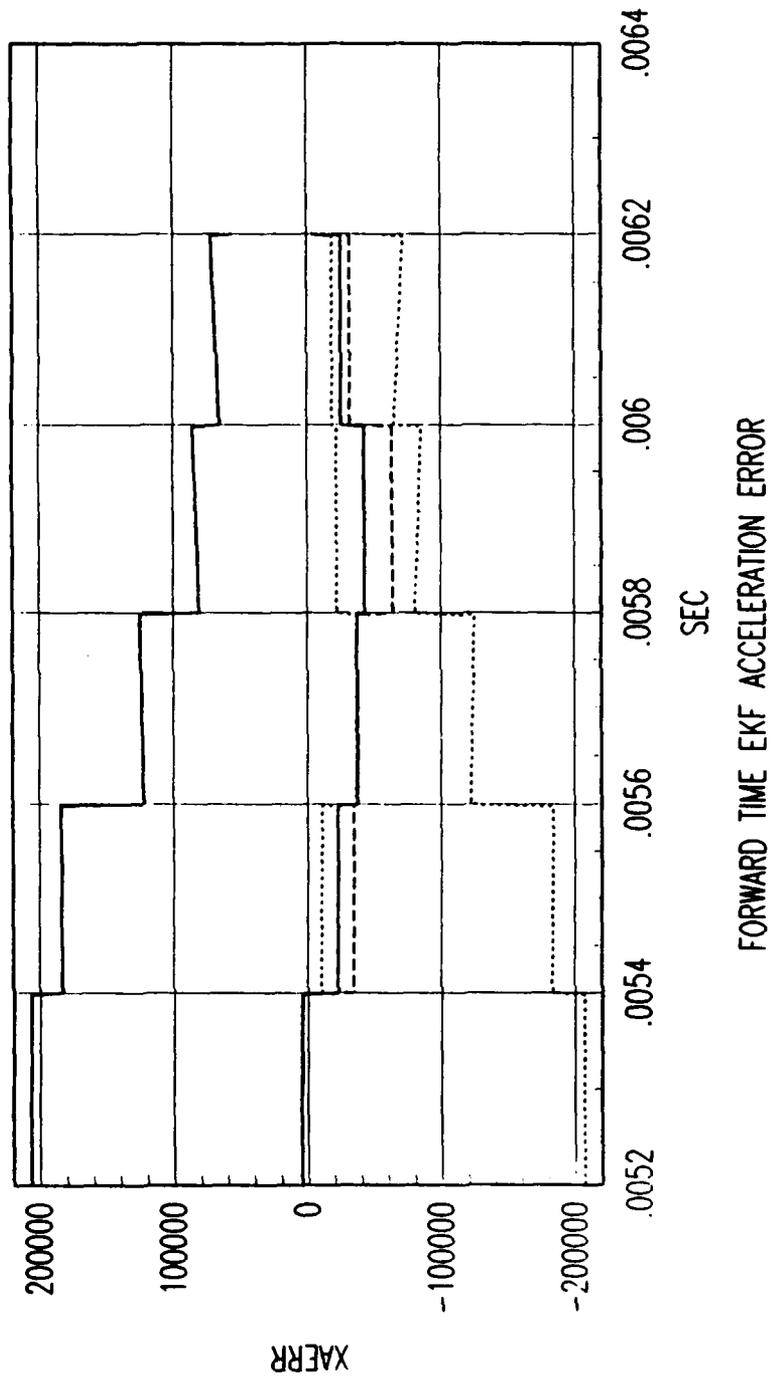


Figure C. 41. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

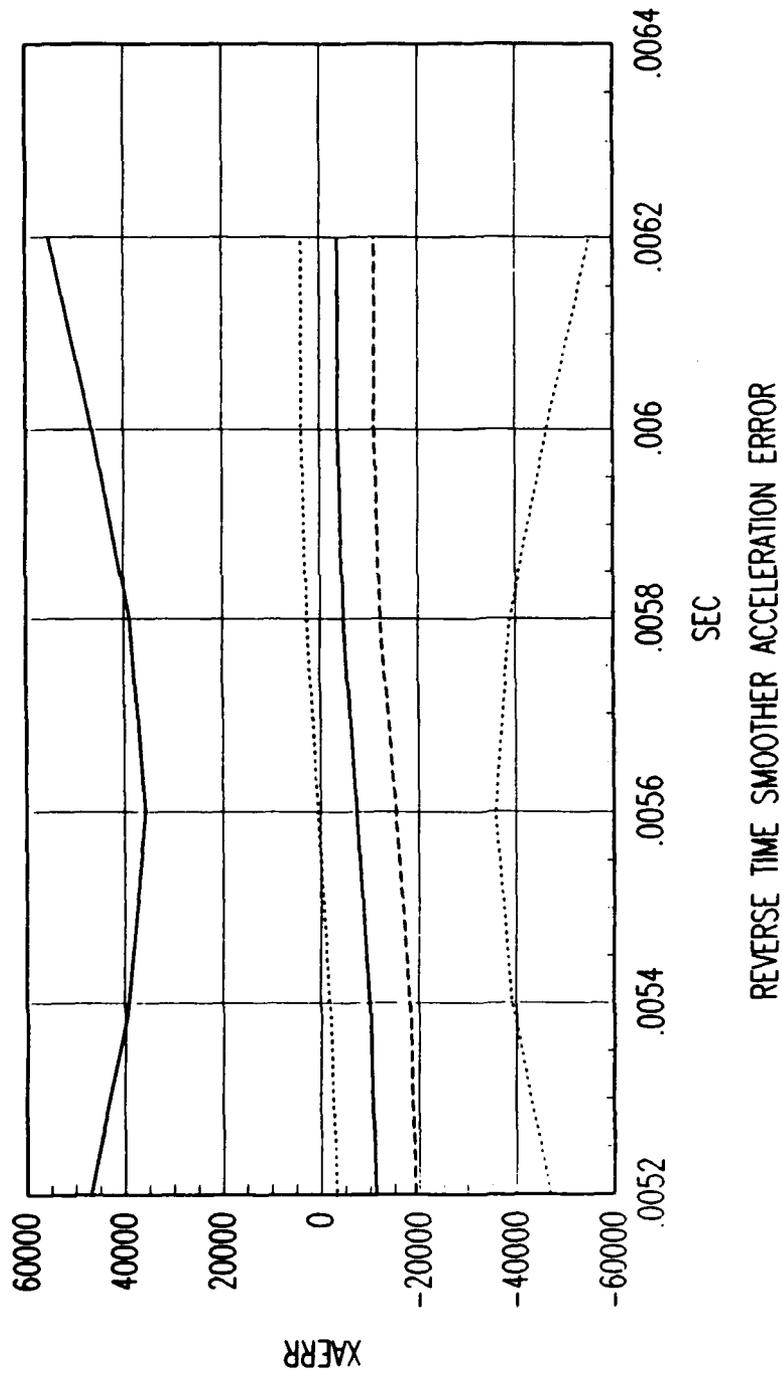


Figure C. 42. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

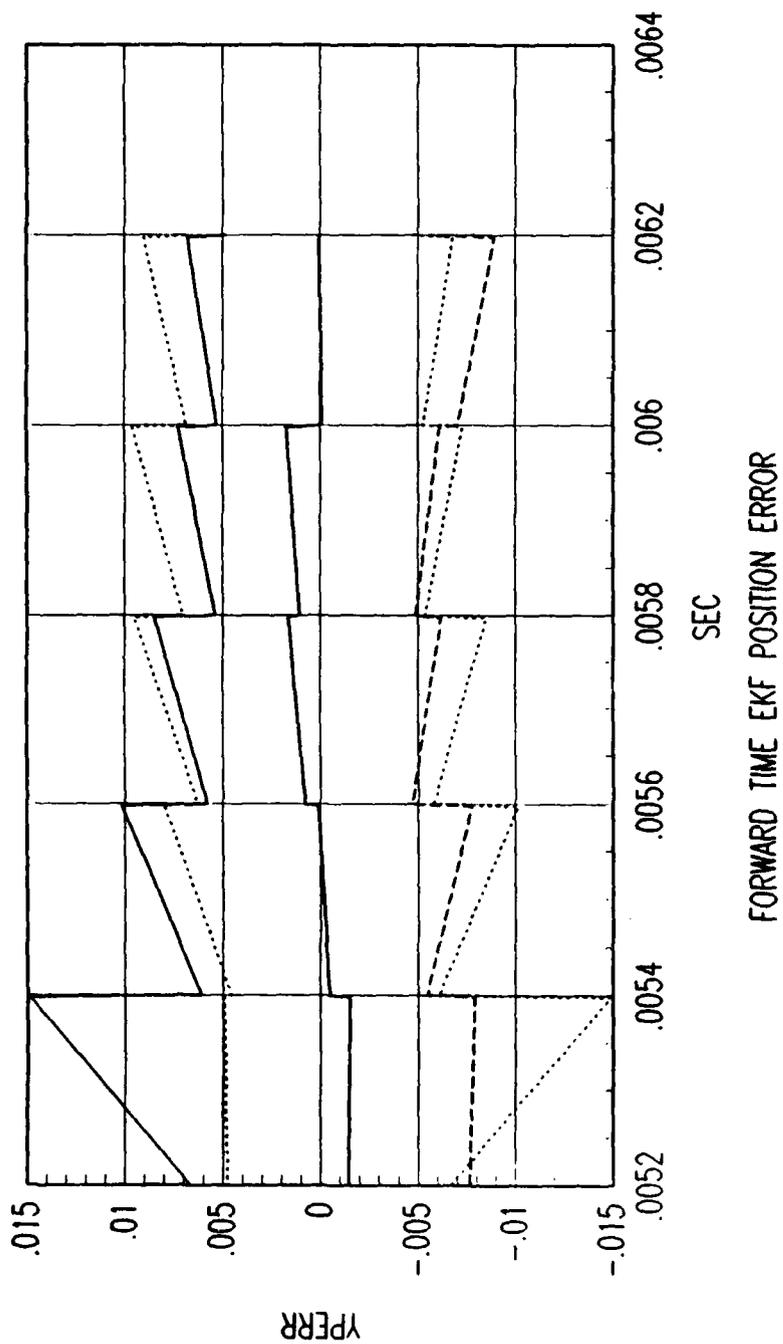


Figure C. 43. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

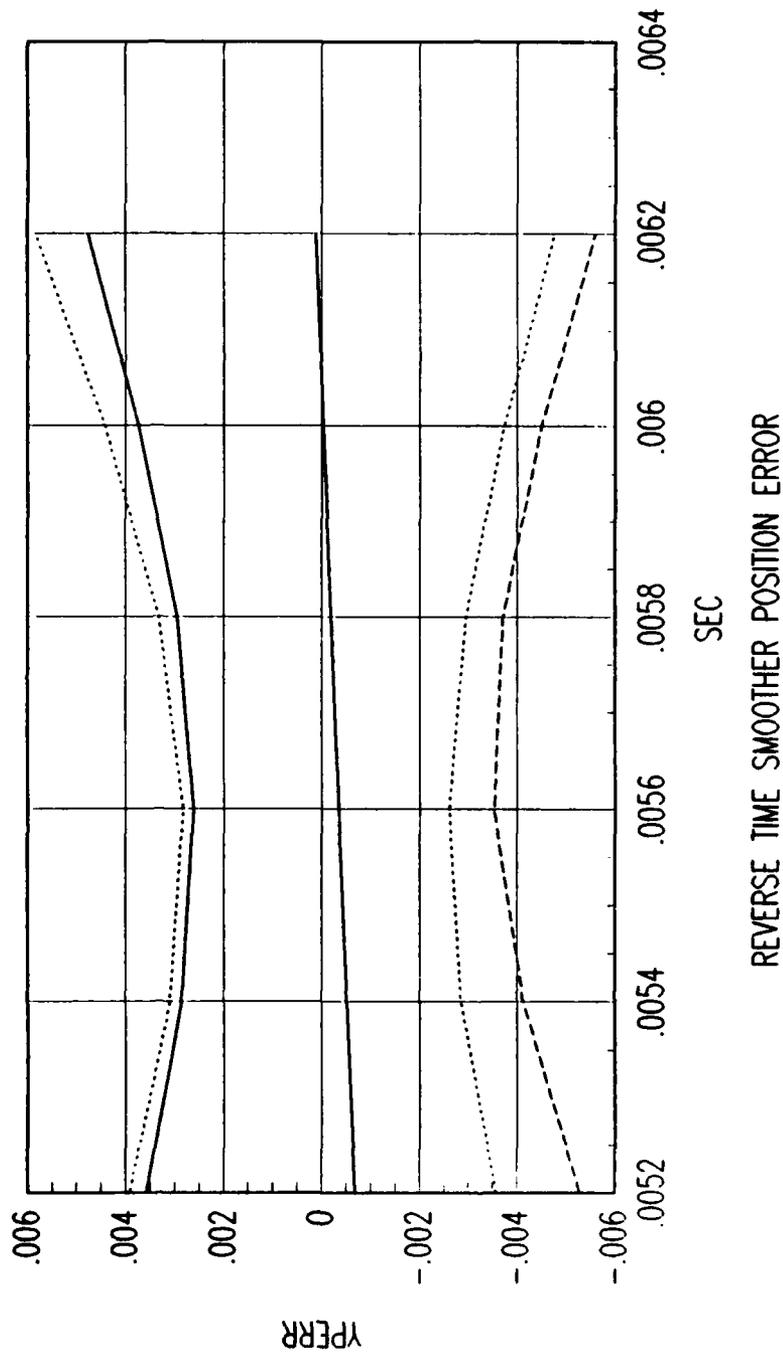


Figure C. 44. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

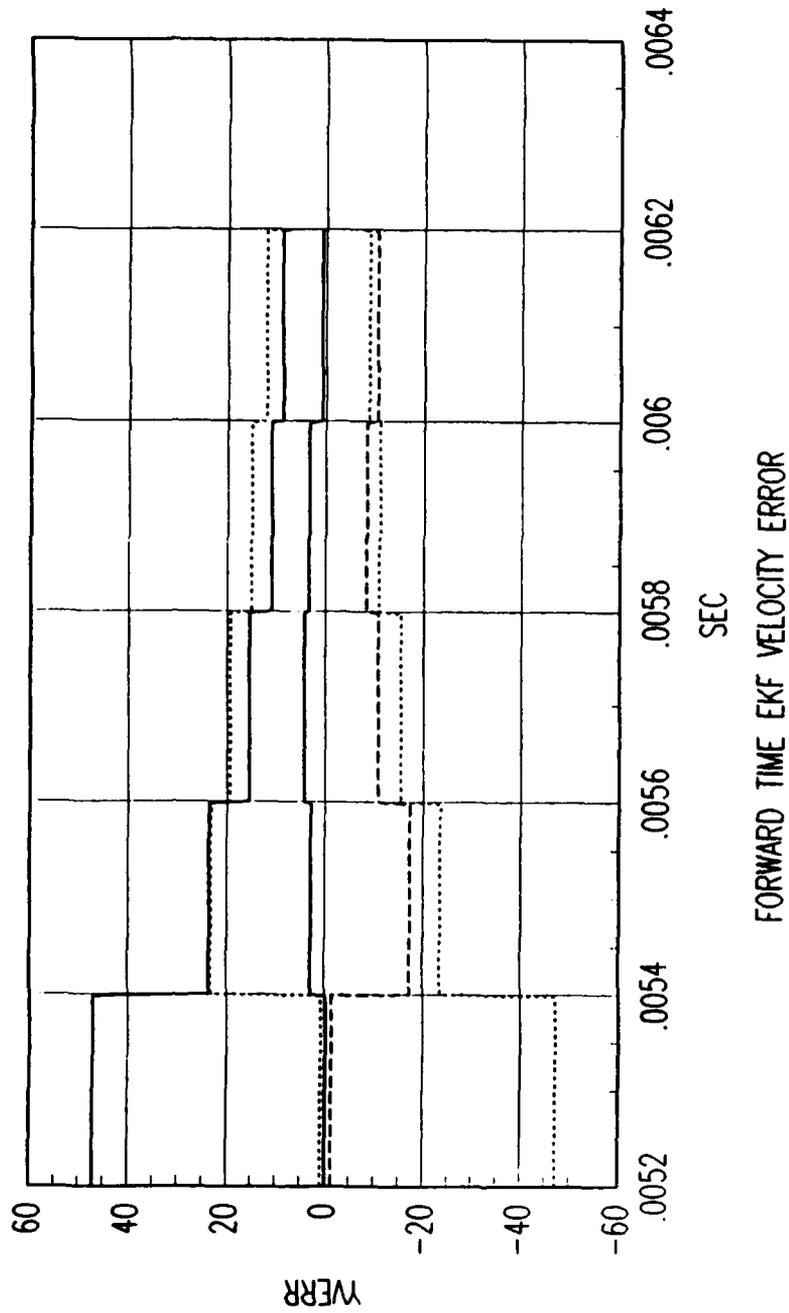


Figure C. 45. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

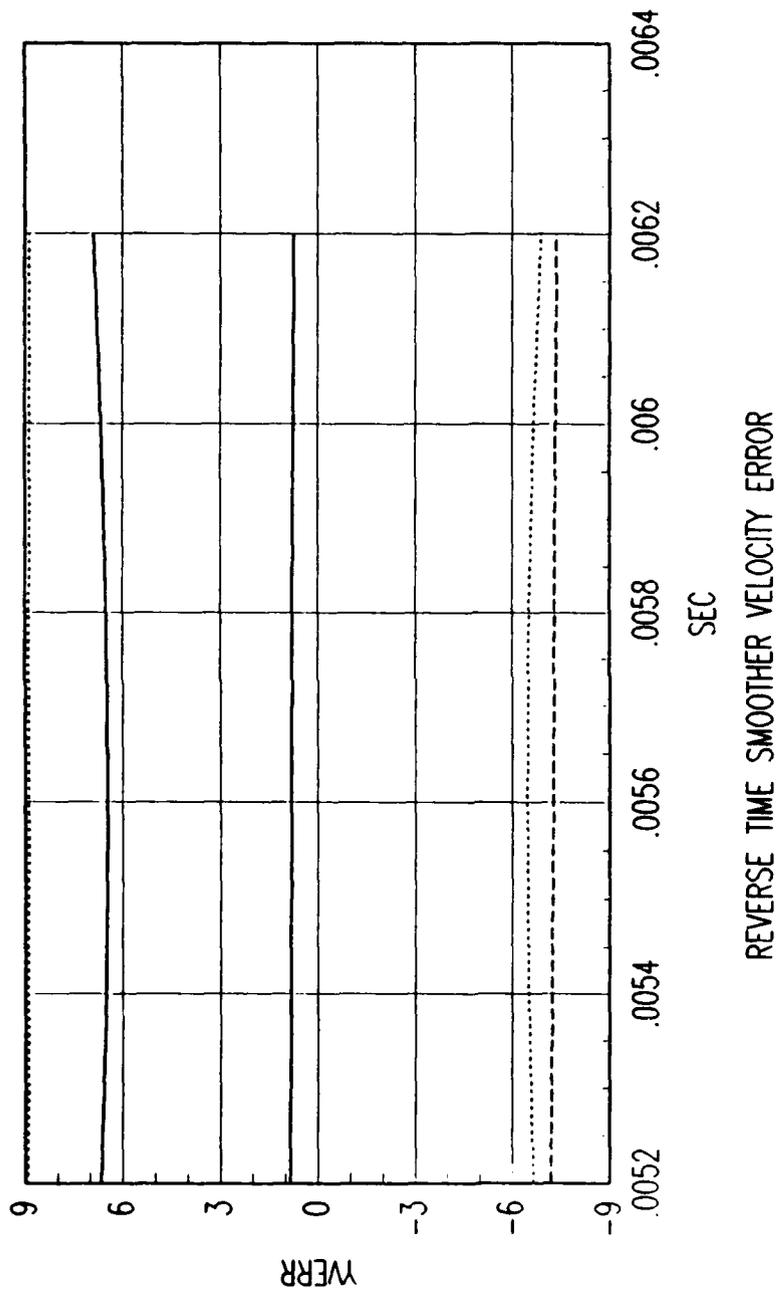


Figure C. 46. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

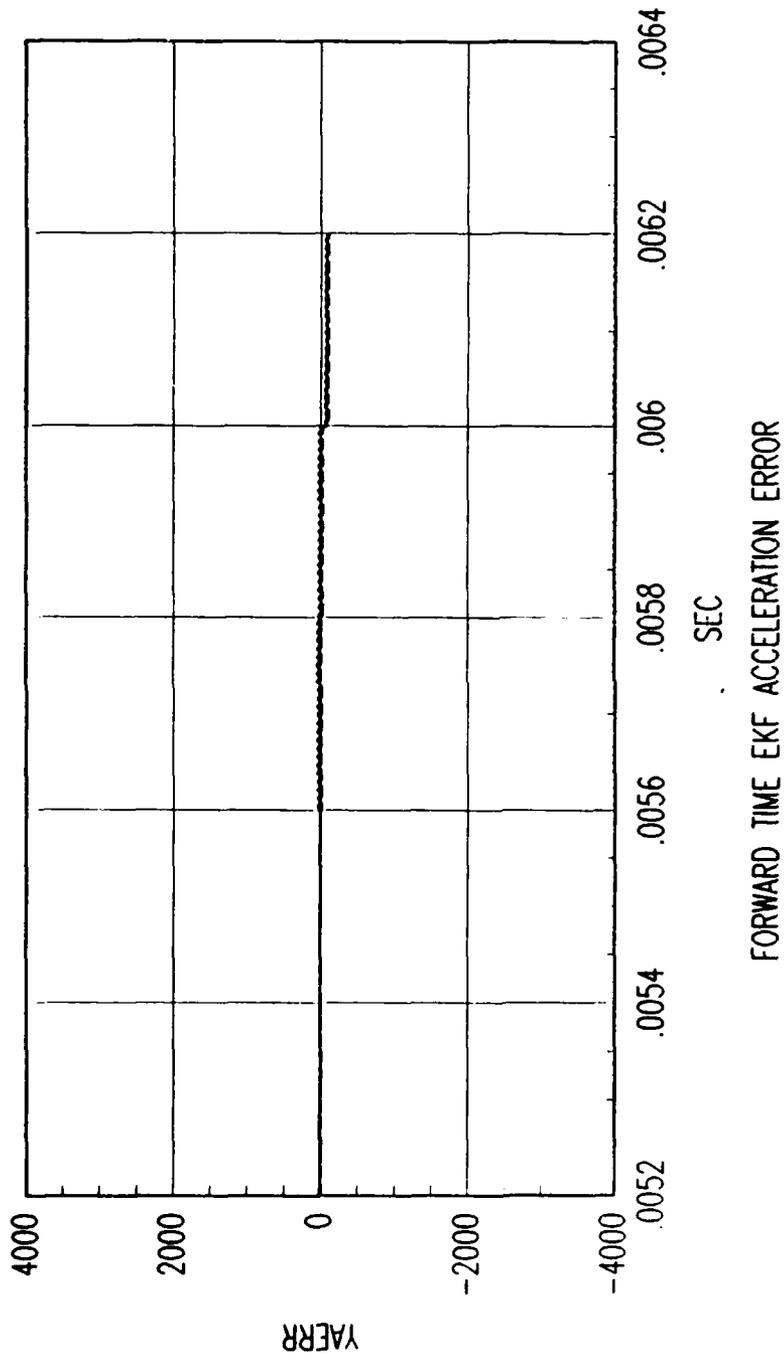


Figure C. 47. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

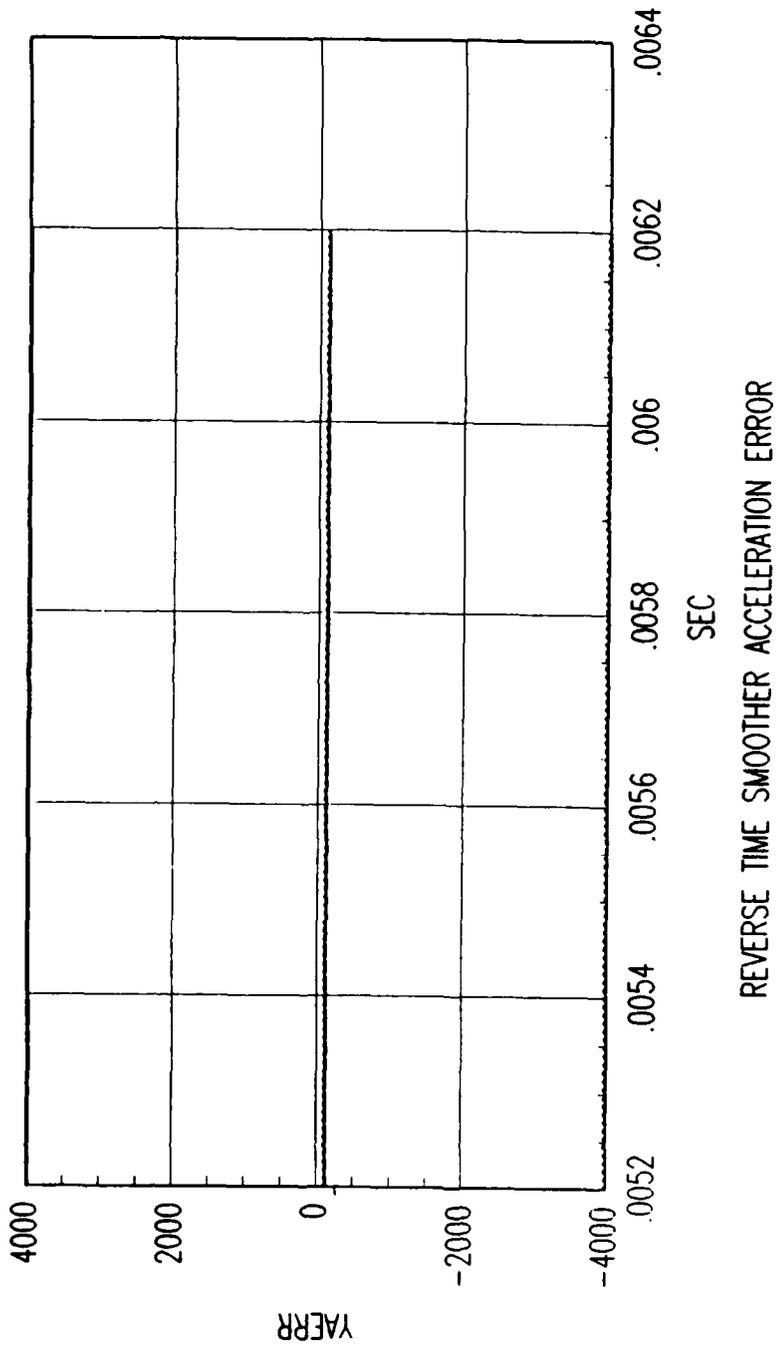


Figure C. 48. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

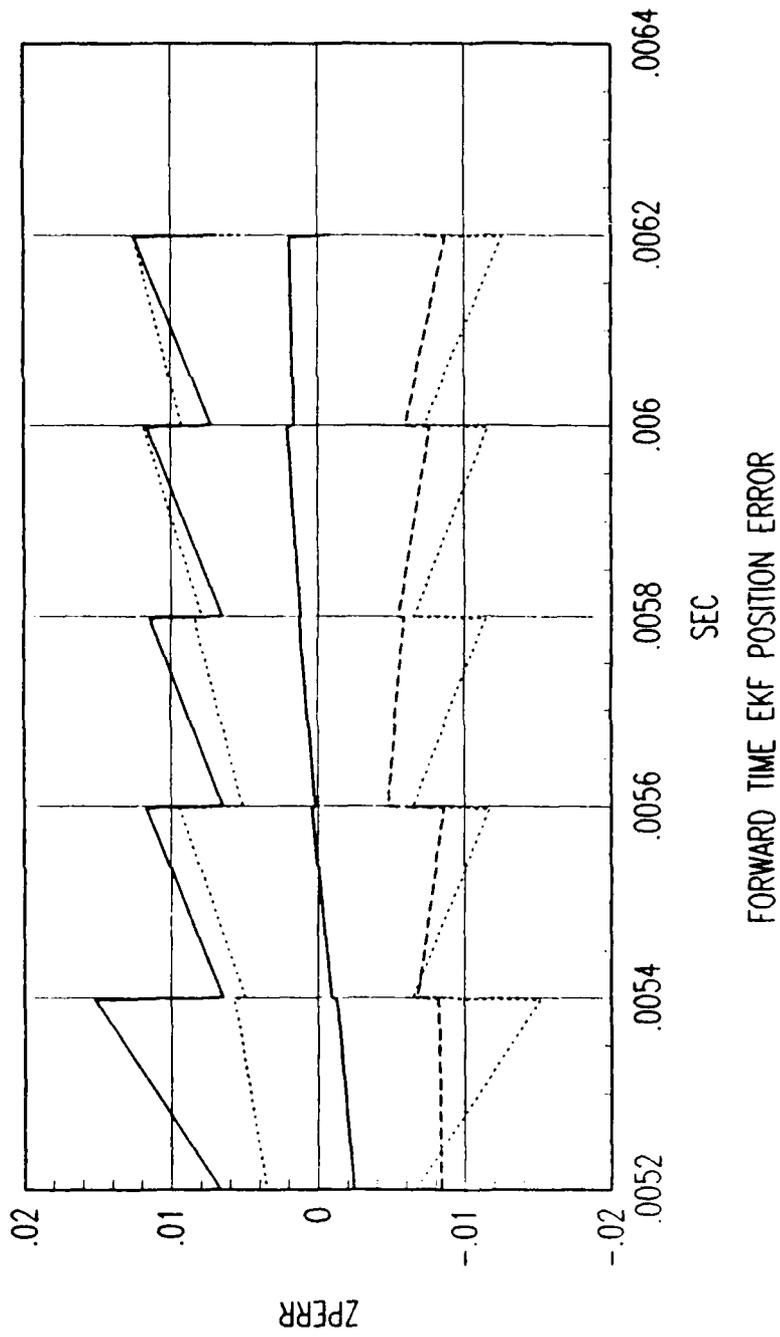


Figure C. 49. Tracking Error Plot, Category 1, $v(t_0) = 1000$ ft/sec.

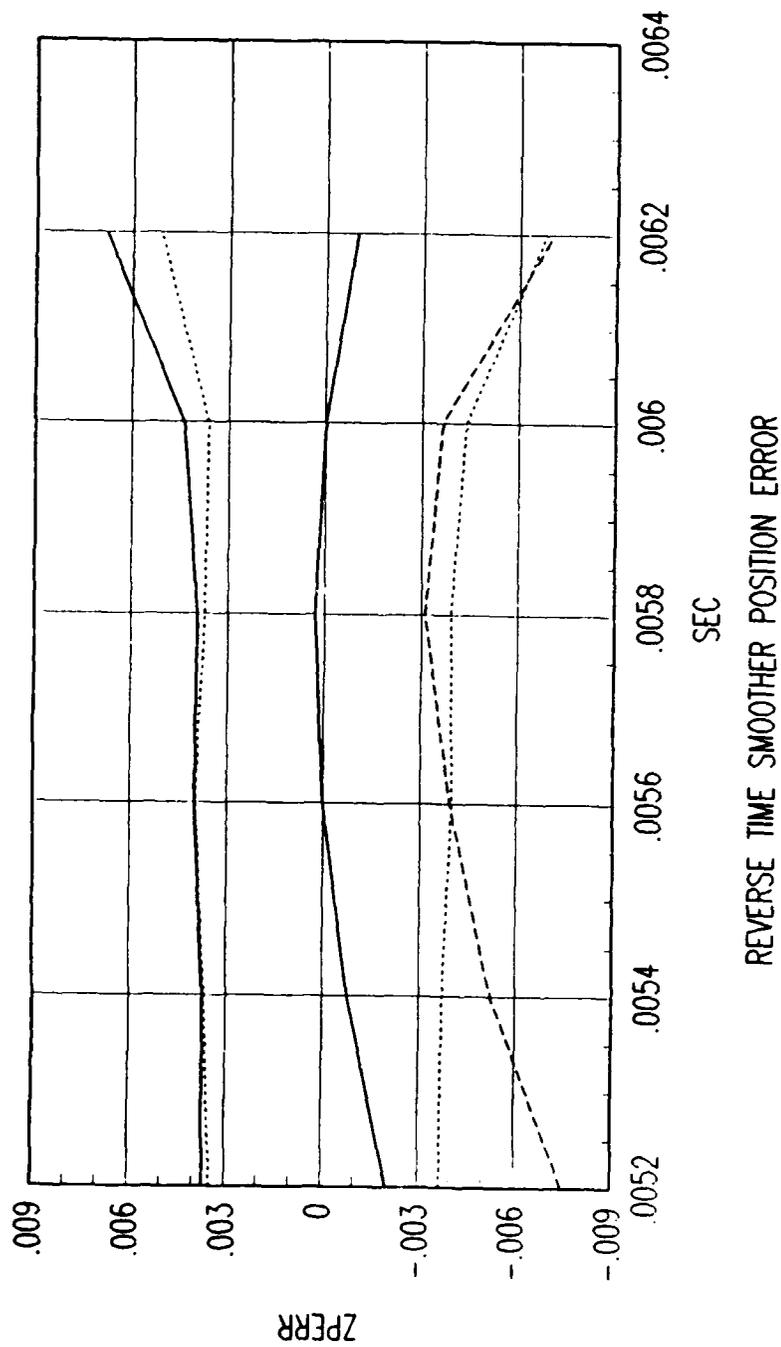


Figure C. 50. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

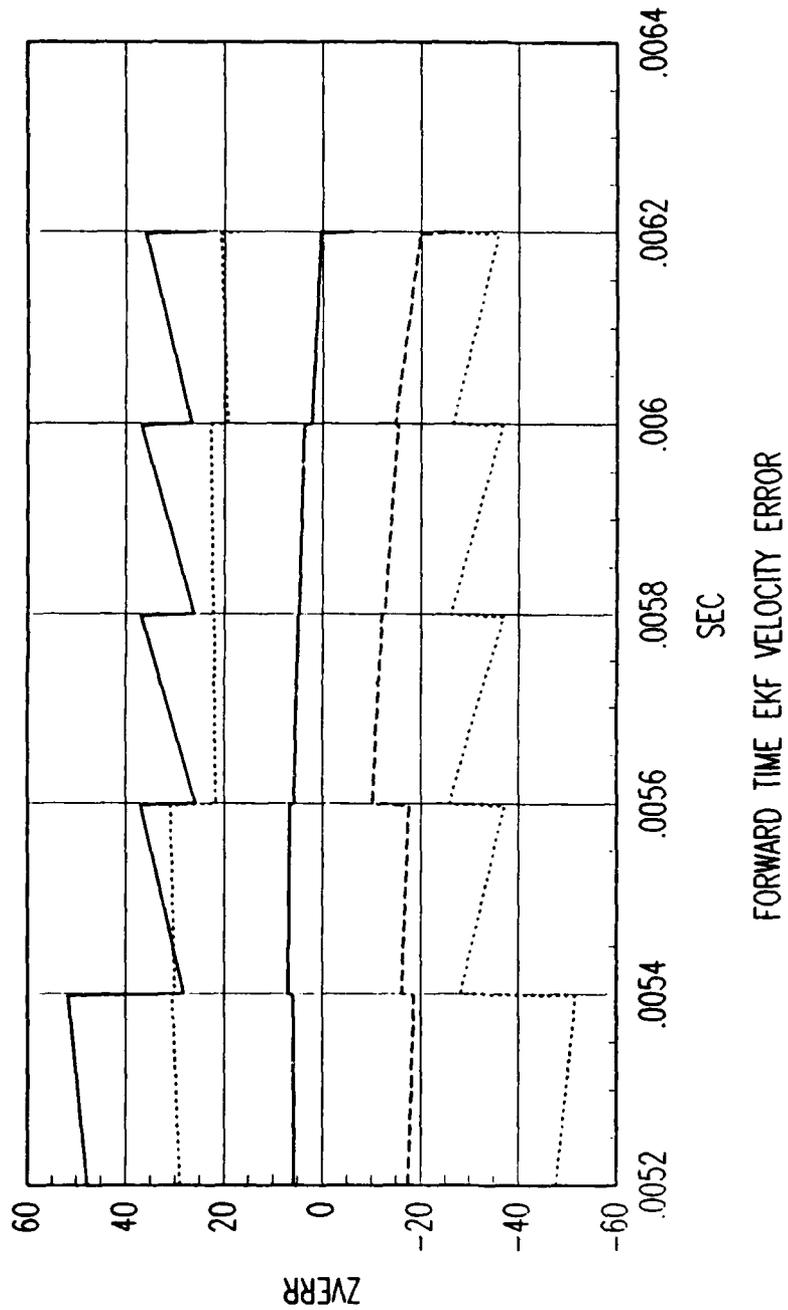


Figure C. 51. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

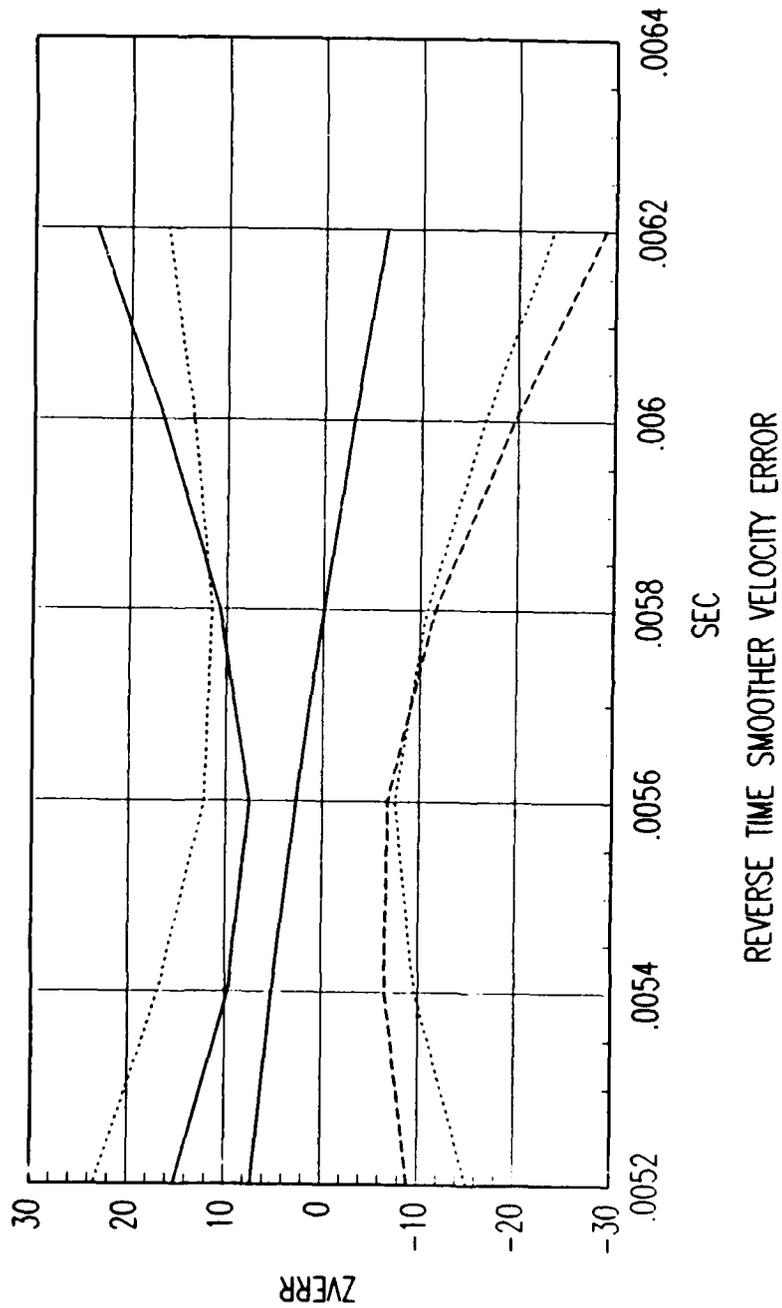


Figure C. 52. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

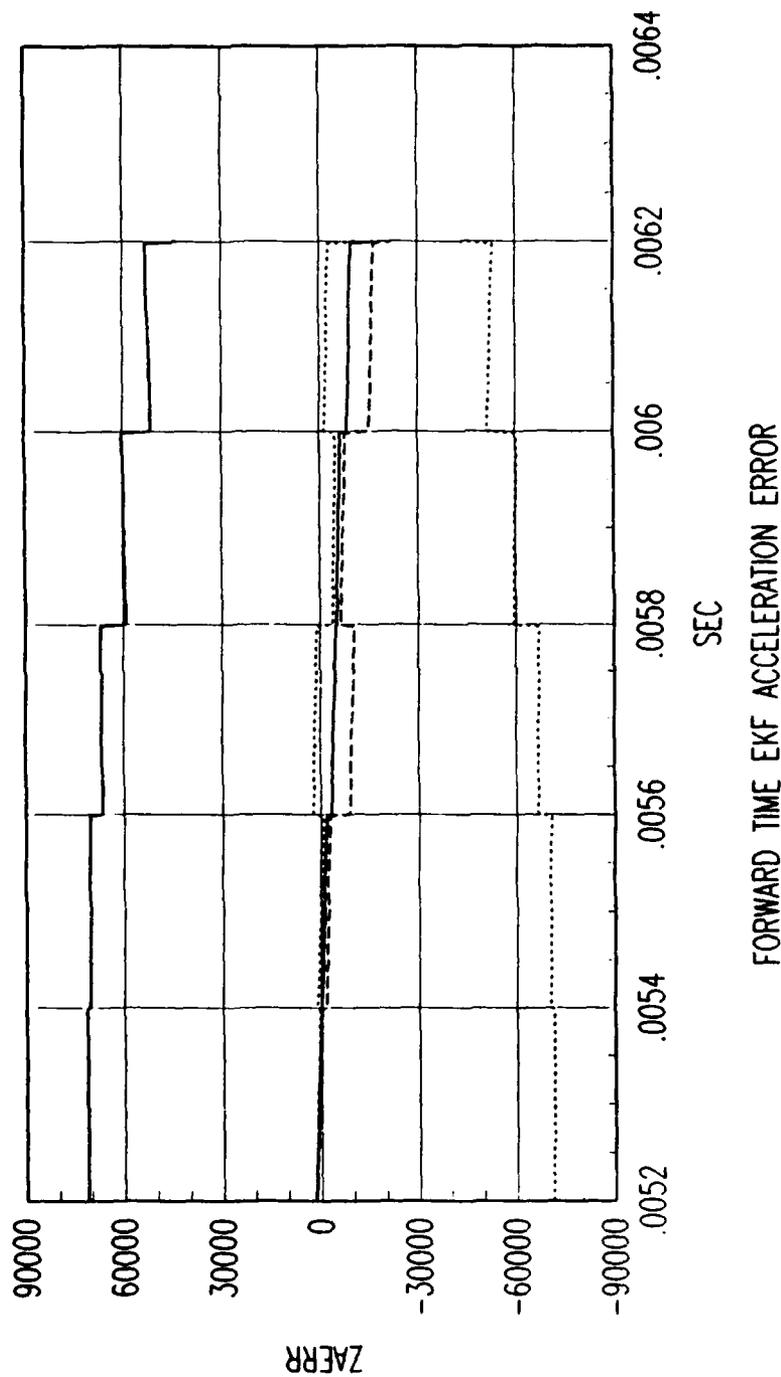


Figure C. 53. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

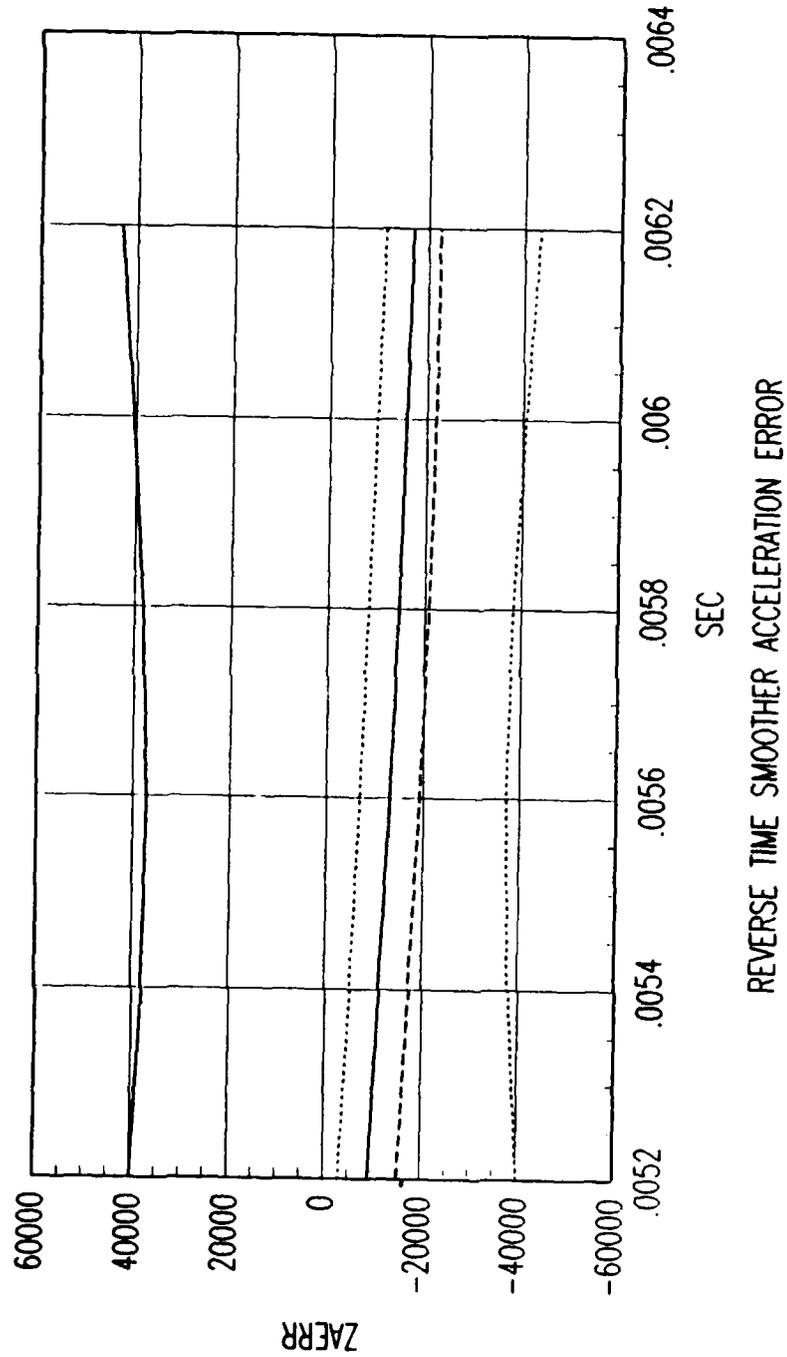


Figure C. 54. Tracking Error Plot, Category 1, $v(t_0) = 10000$ ft/sec.

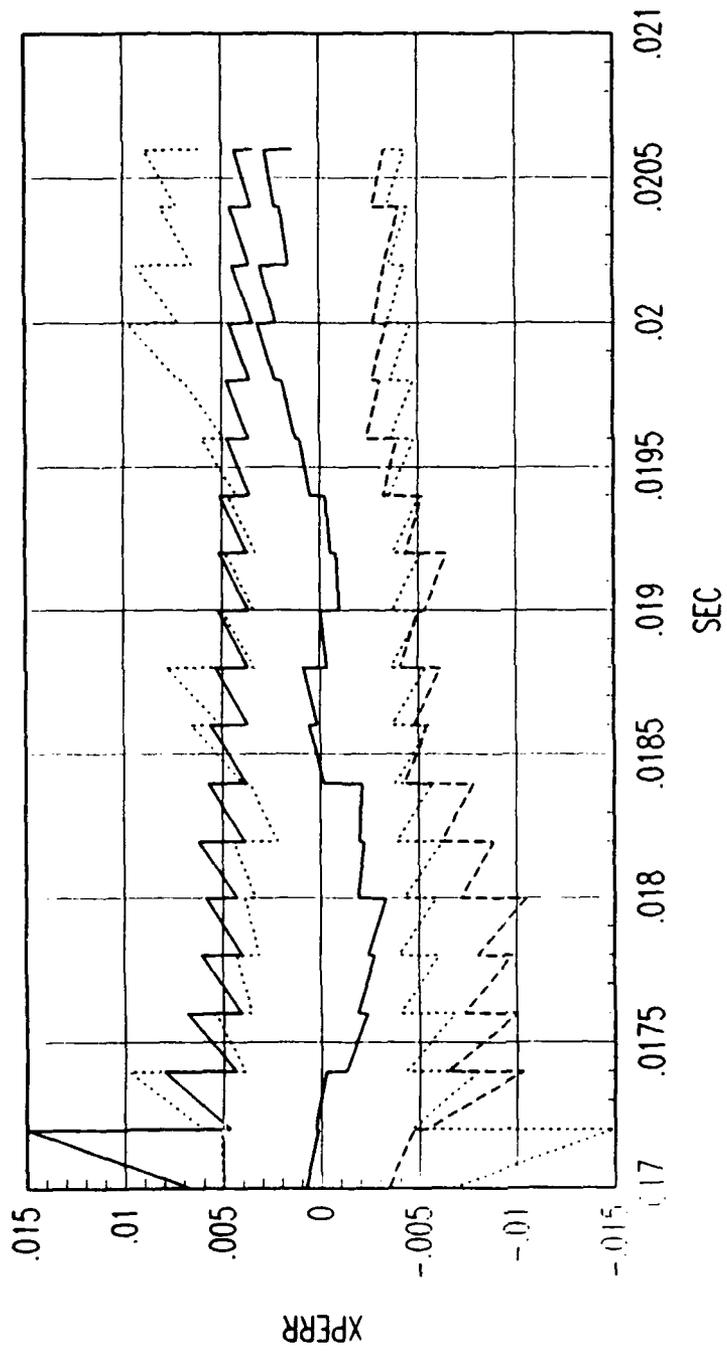


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C. 55. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

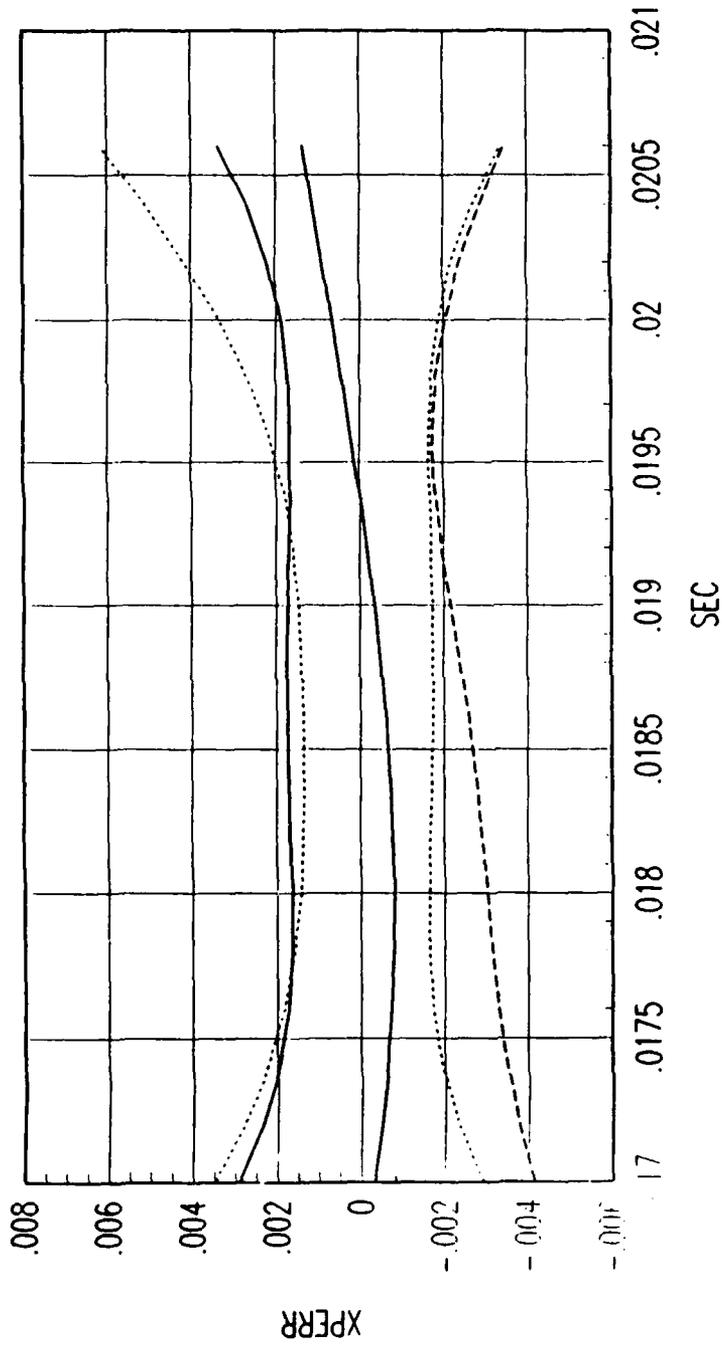


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 56. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

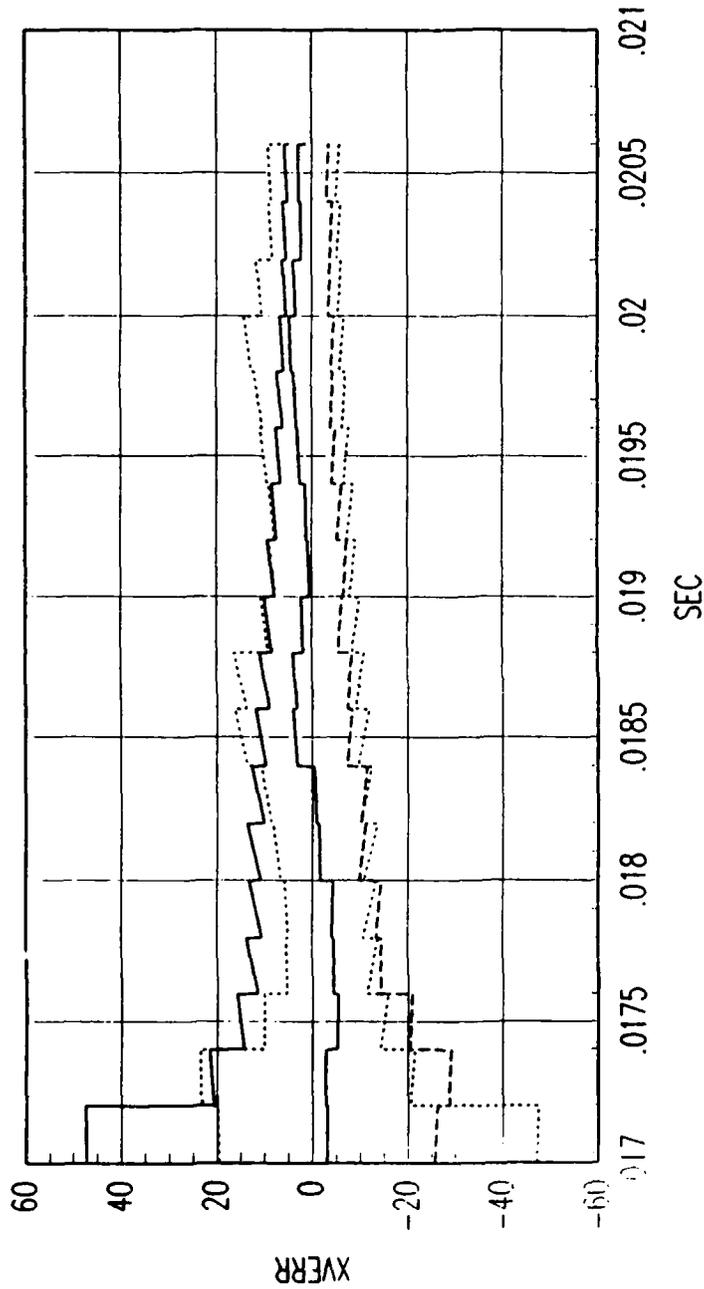


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C. 57. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

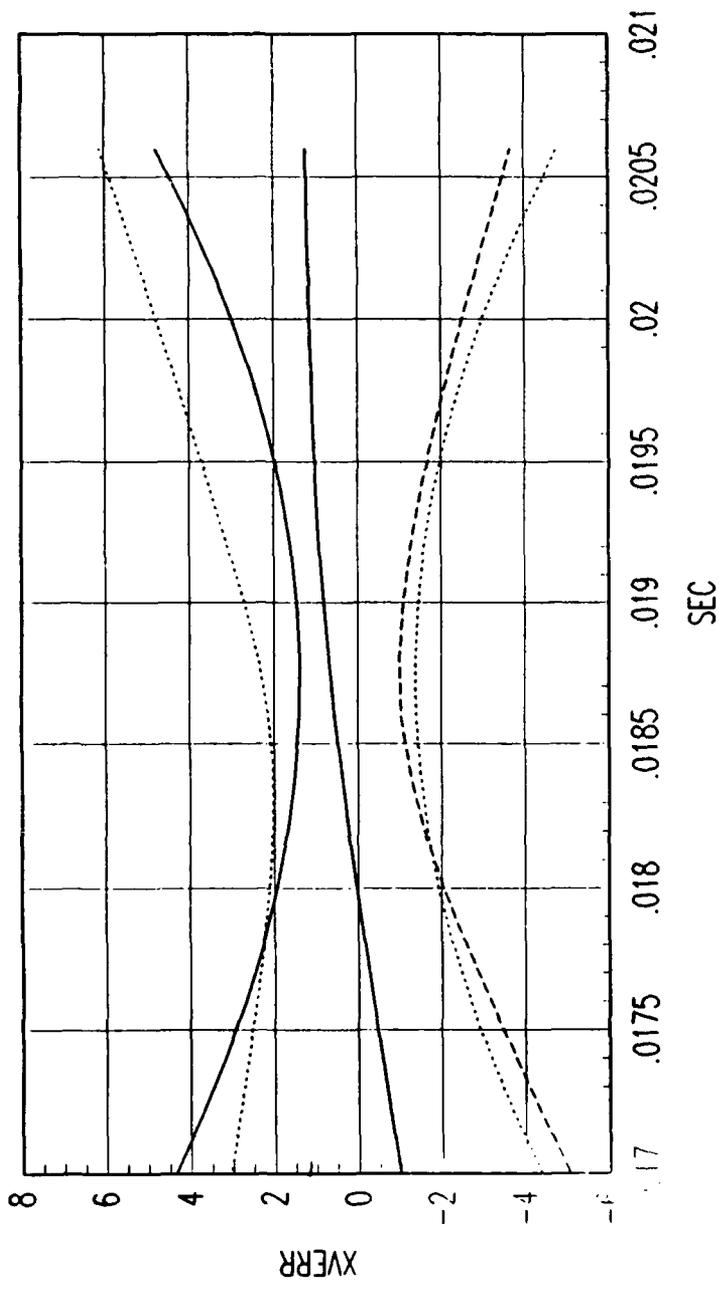


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 58. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

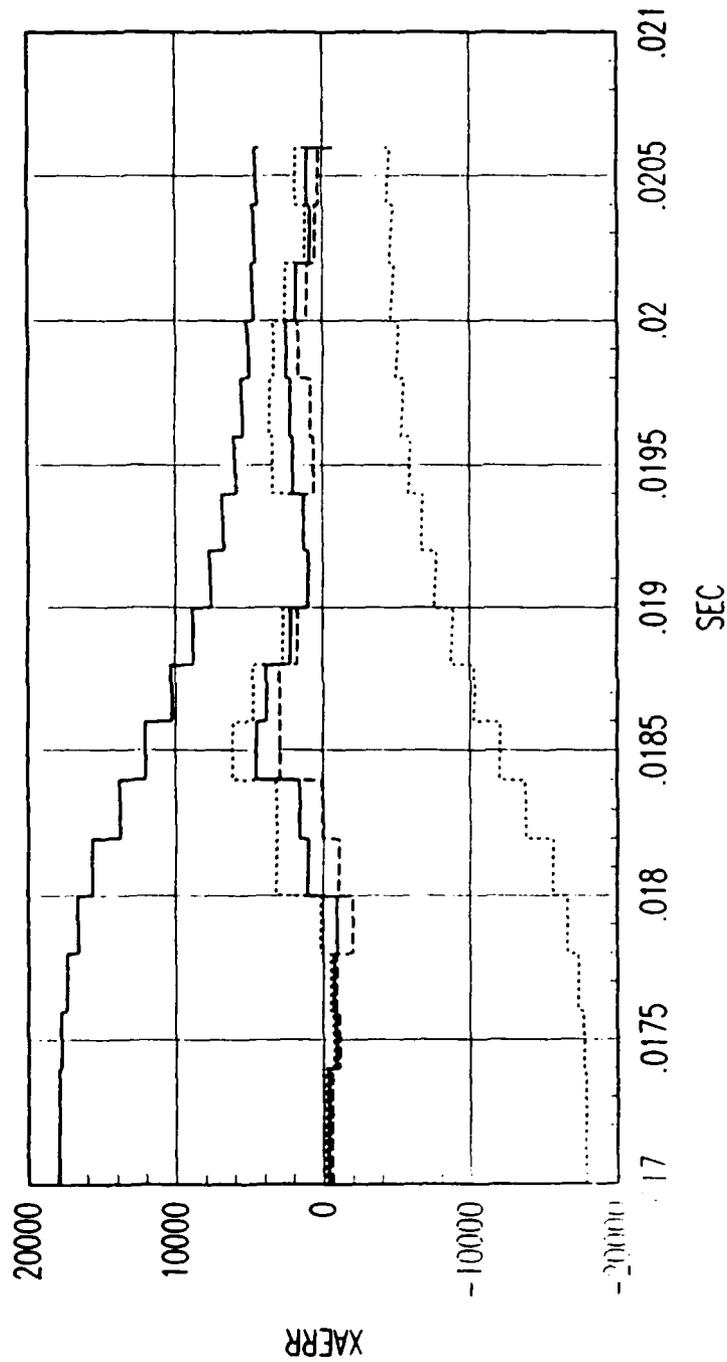


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C. 59. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

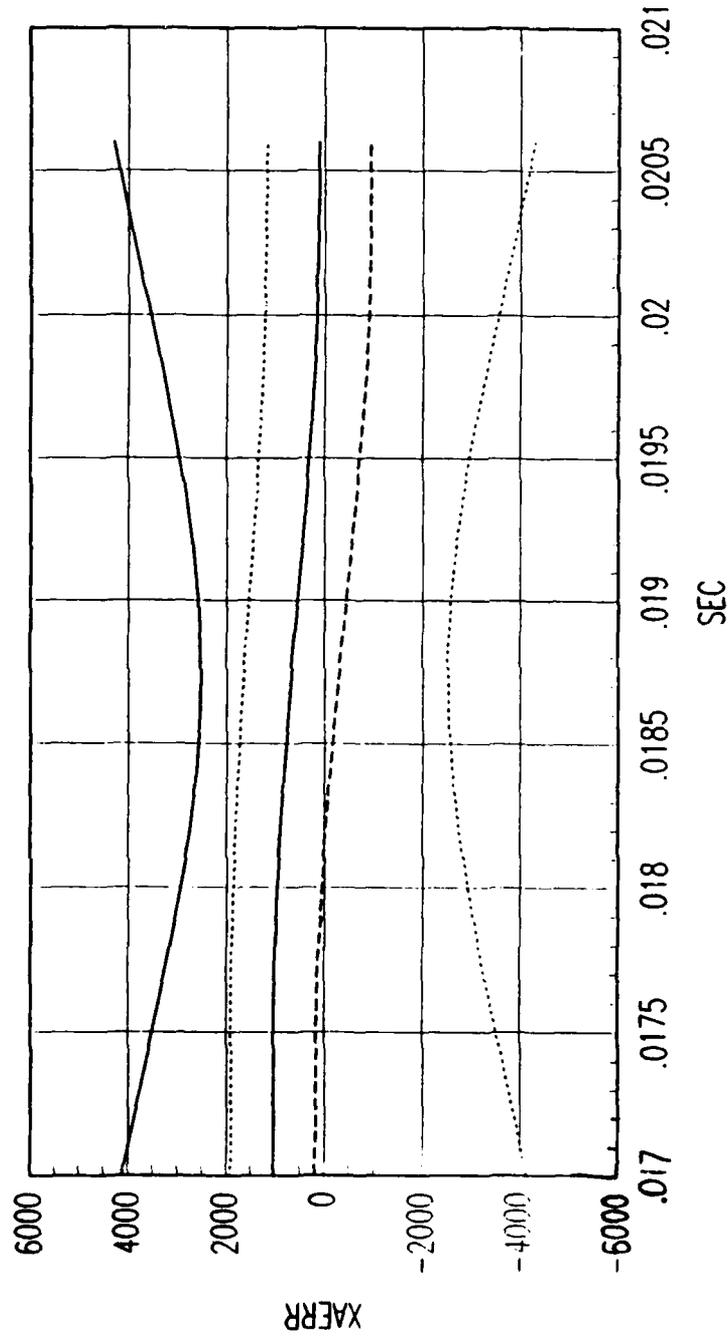


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 60. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

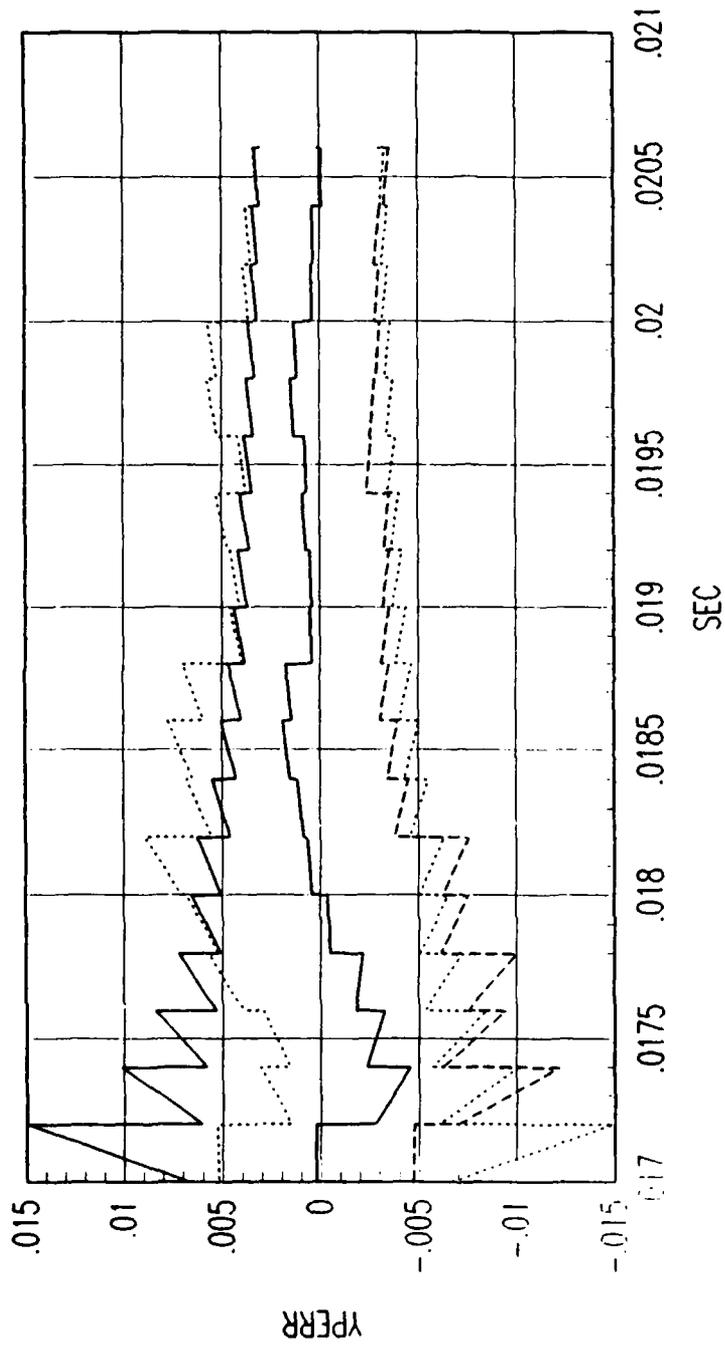


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C. 61. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

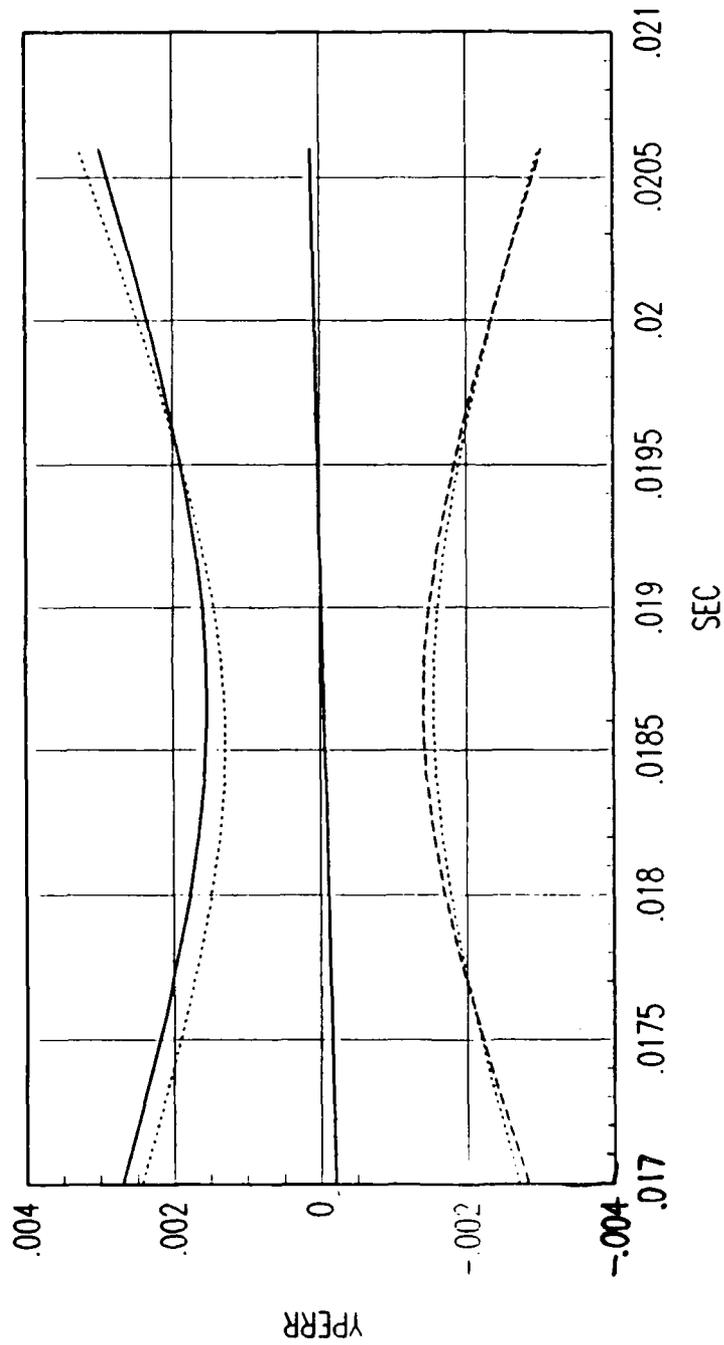
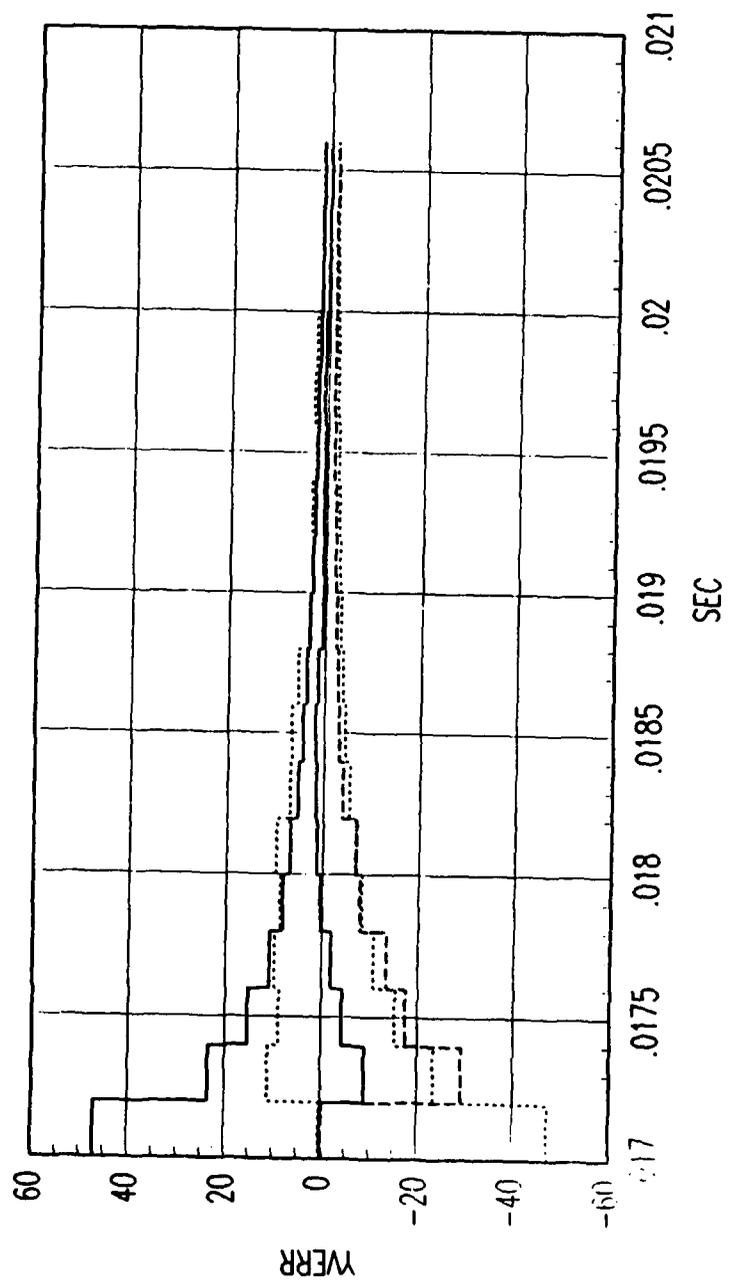


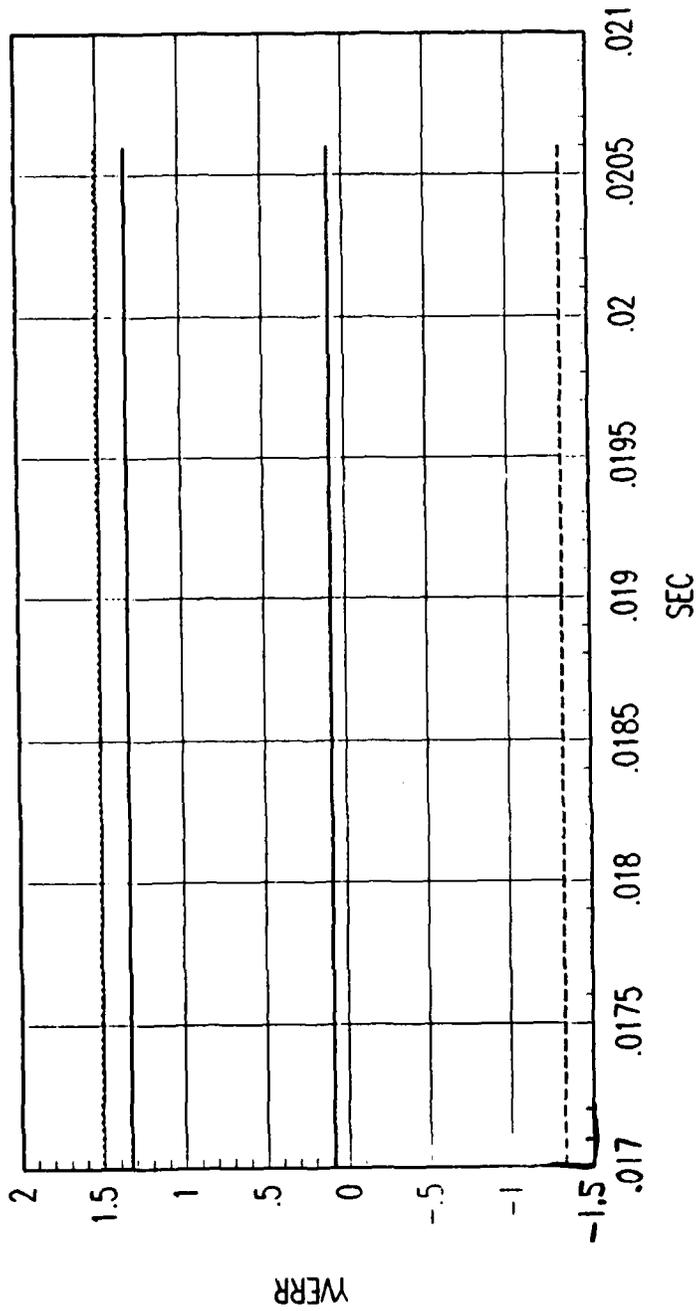
FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 62. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. A FORWARD TIME EKF VELOCITY ERROR

Figure C. 63. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 64. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

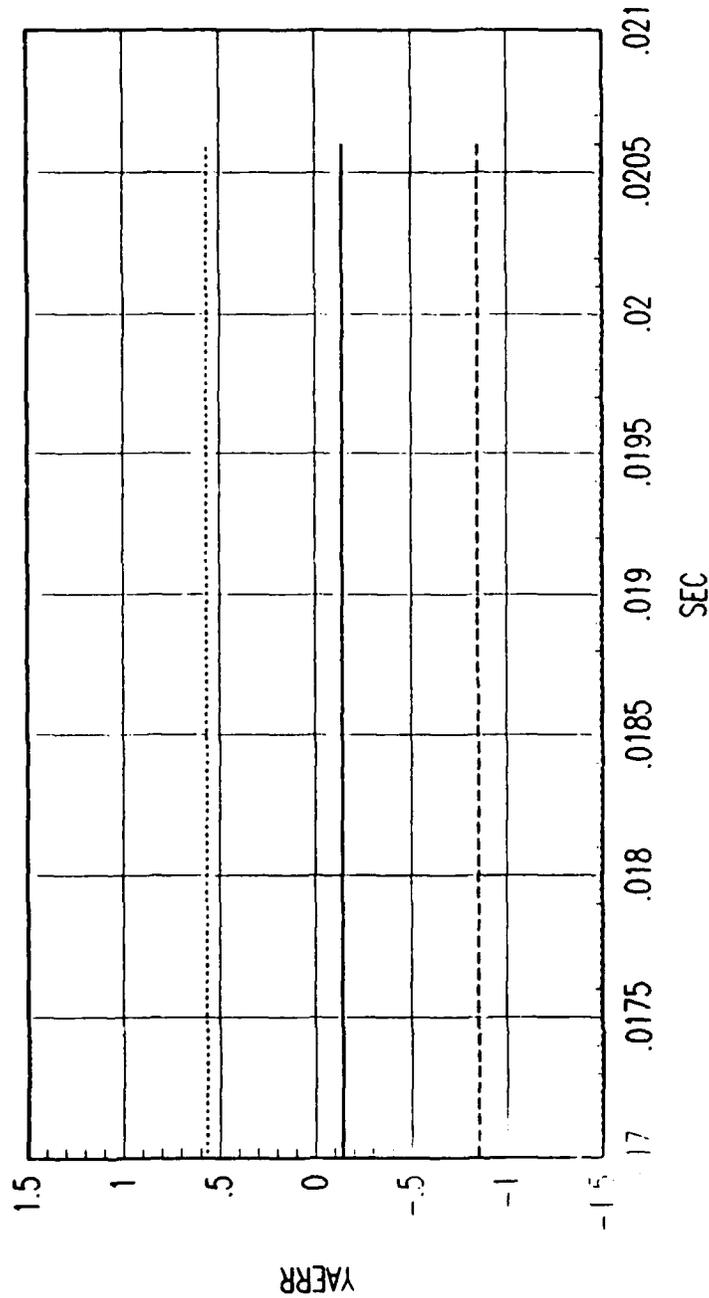


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C. 65. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

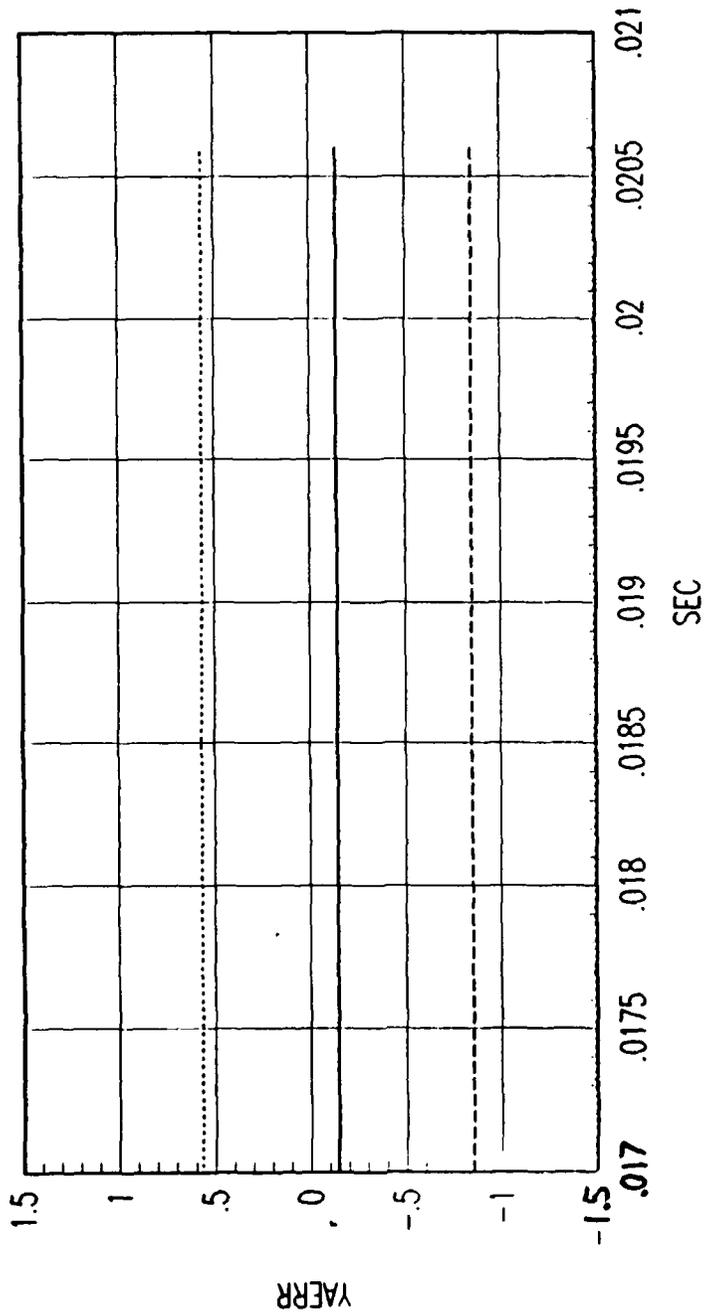
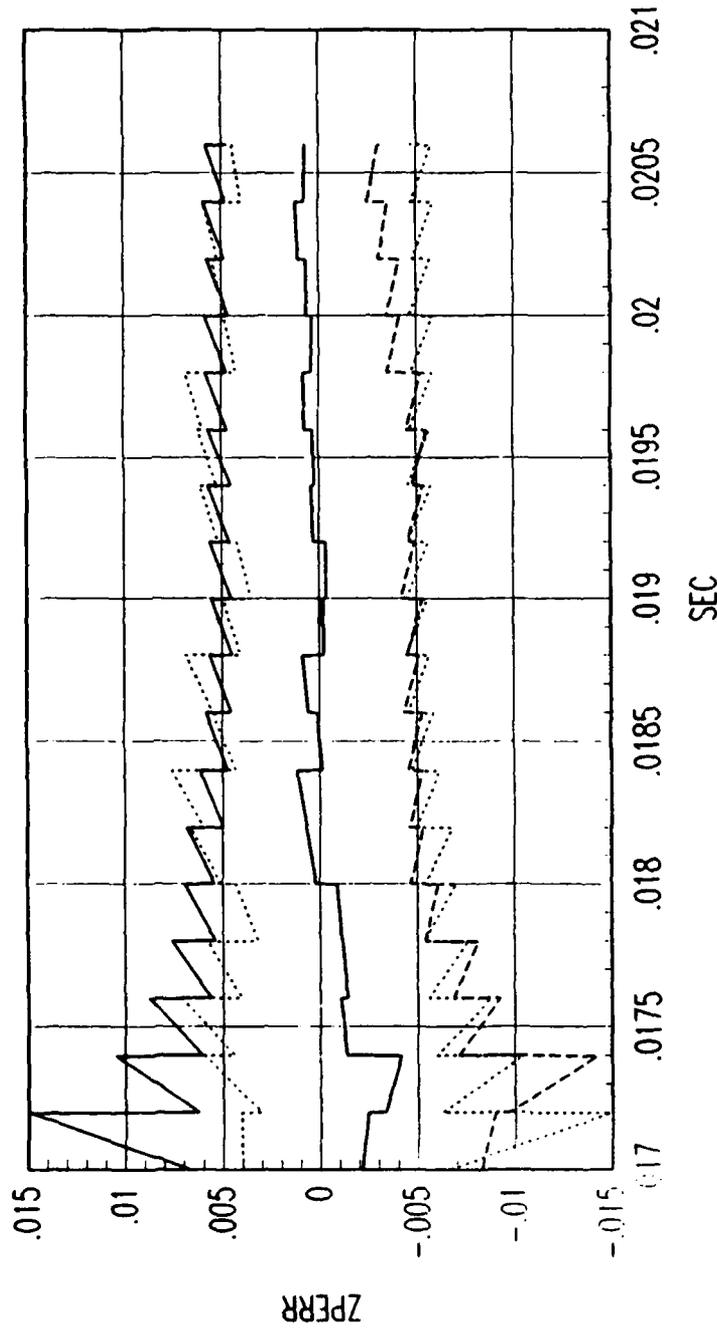


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 66. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. A FORWARD TIME EKF POSITION ERROR

Figure C. 67. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

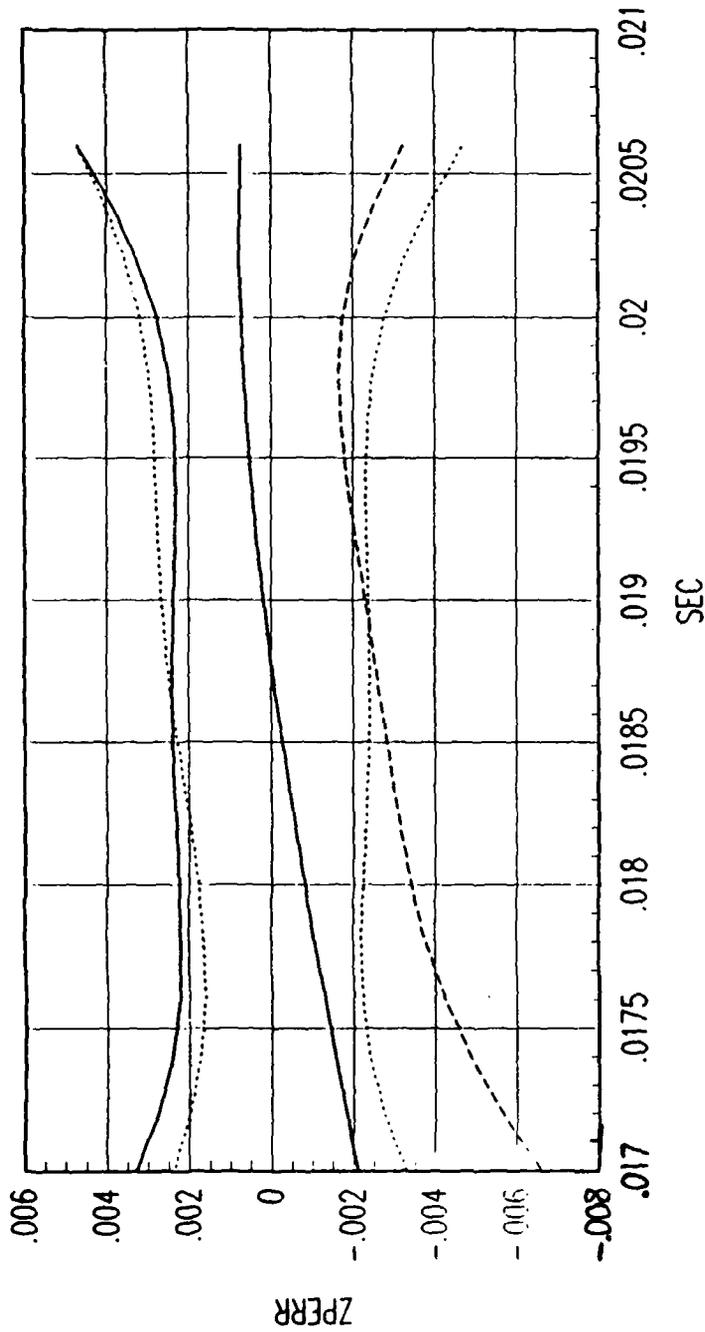
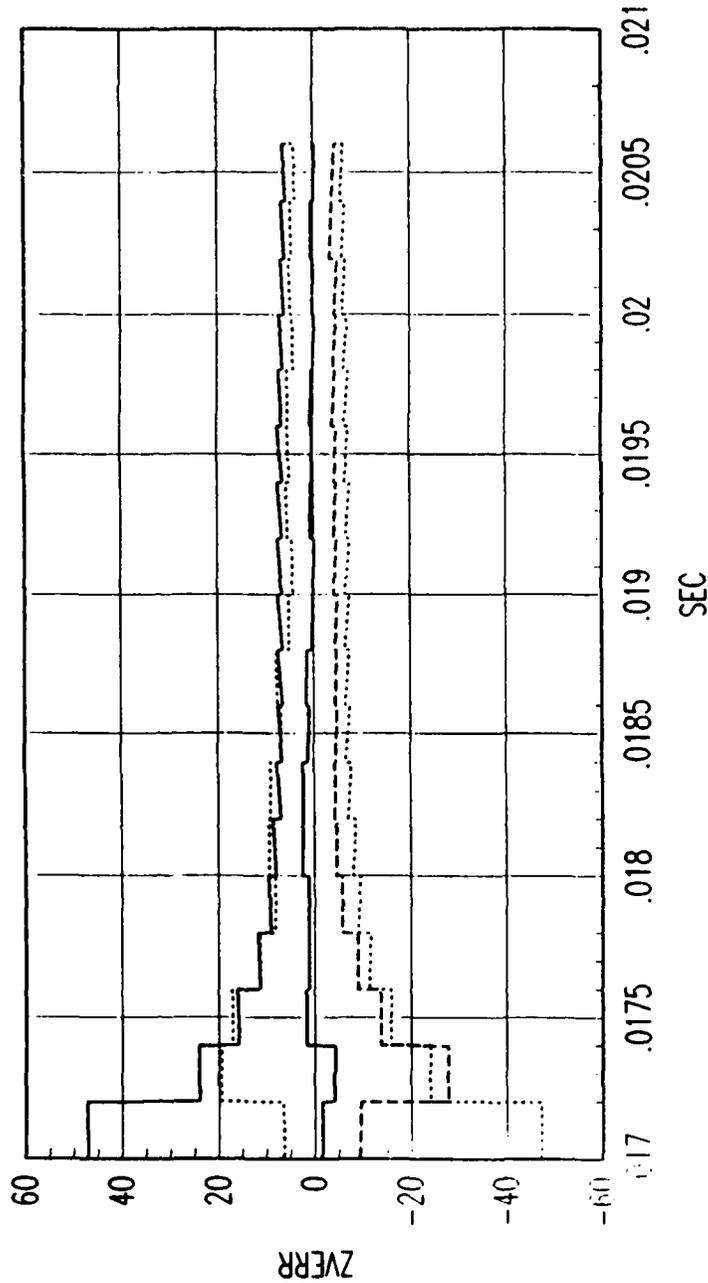


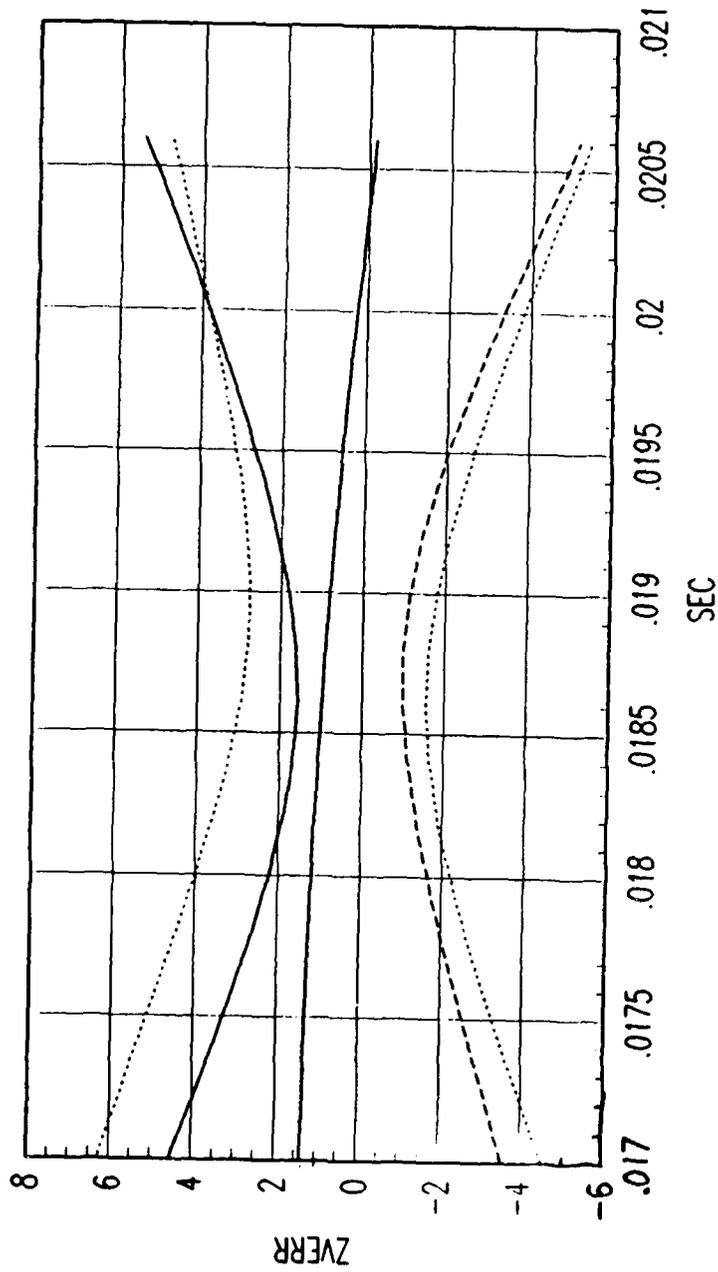
FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 68. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. A FORWARD TIME EKF VELOCITY ERROR

Figure C. 69. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 70. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

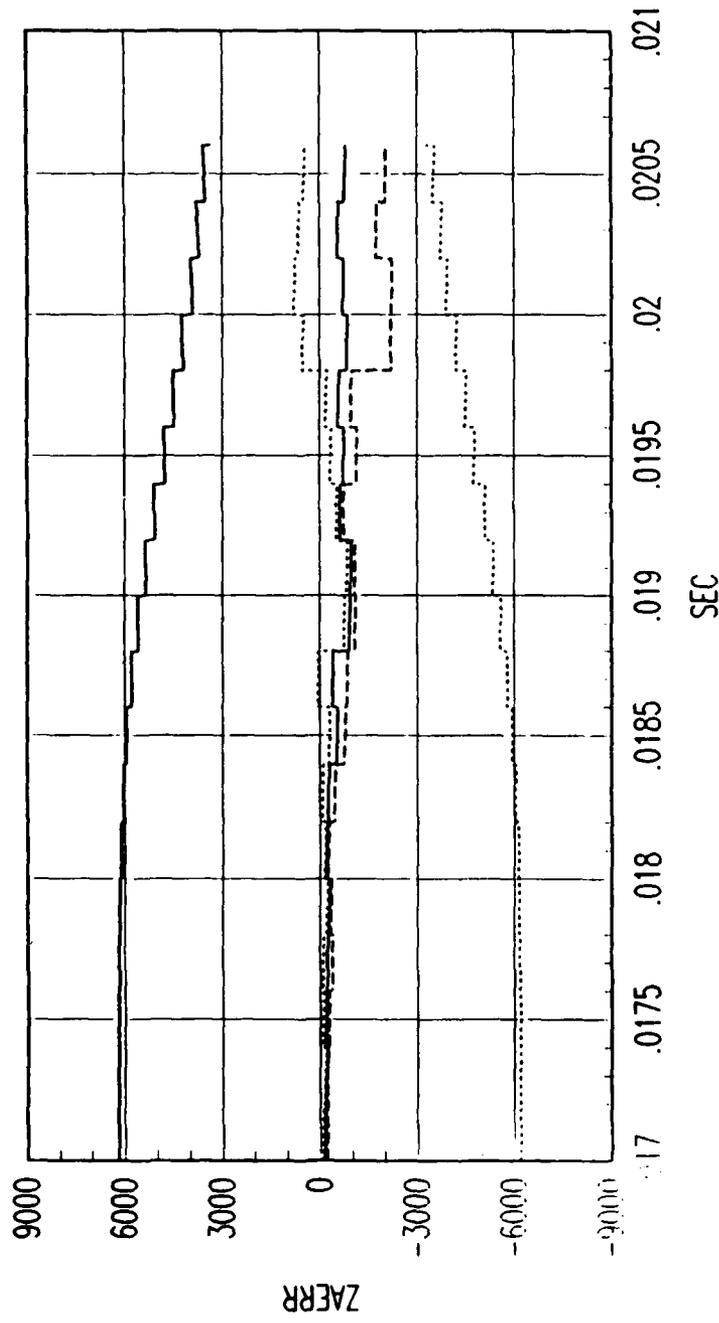
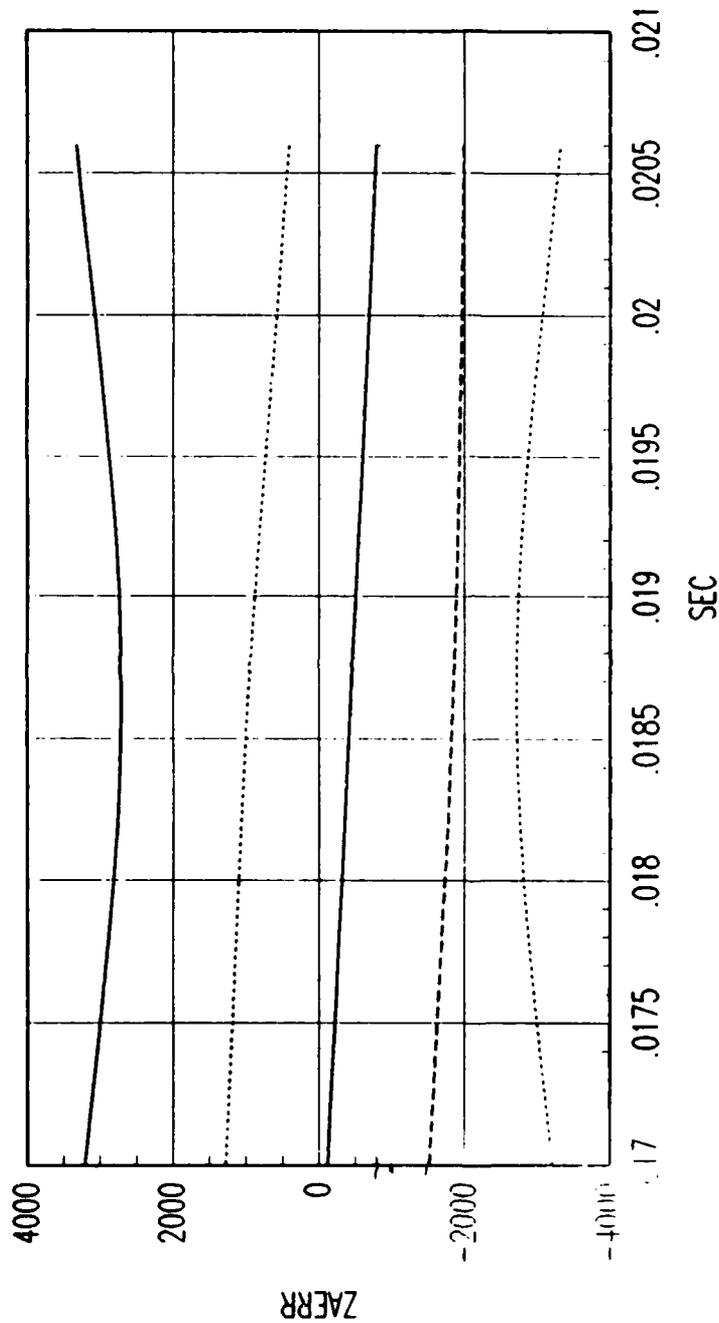


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C. 71. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 72. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

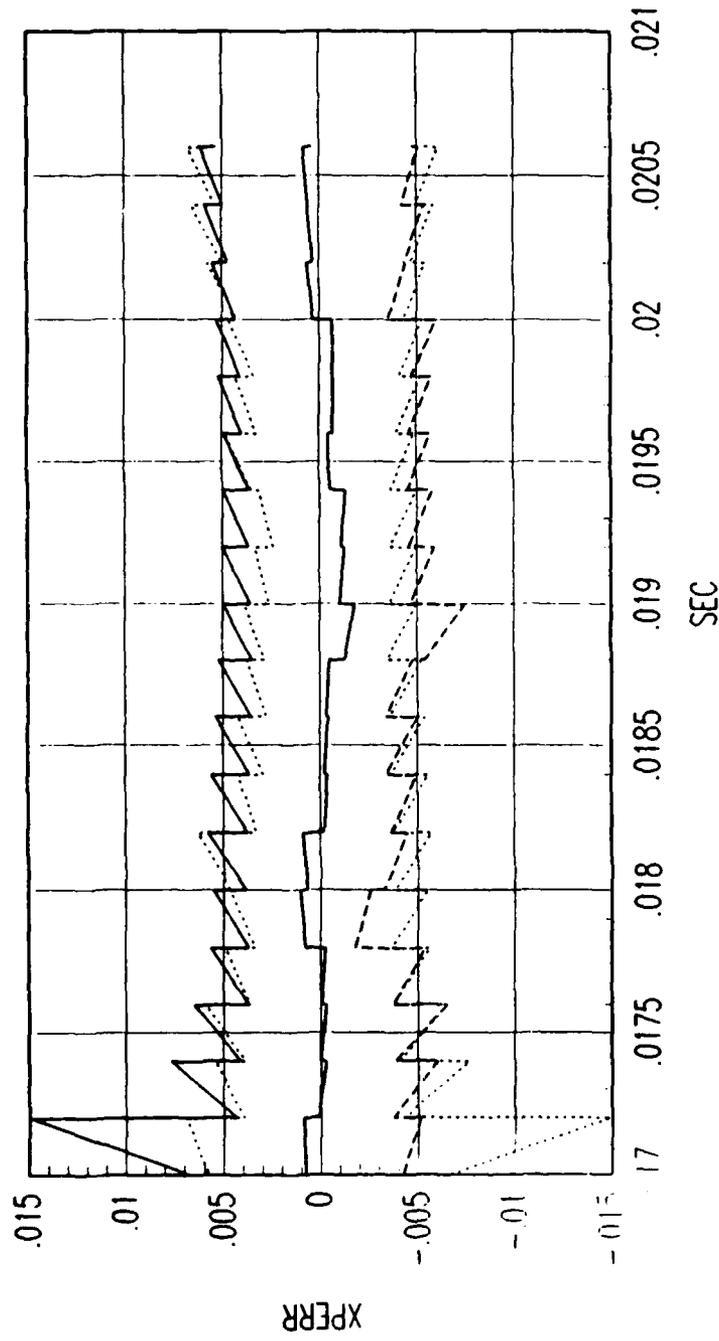
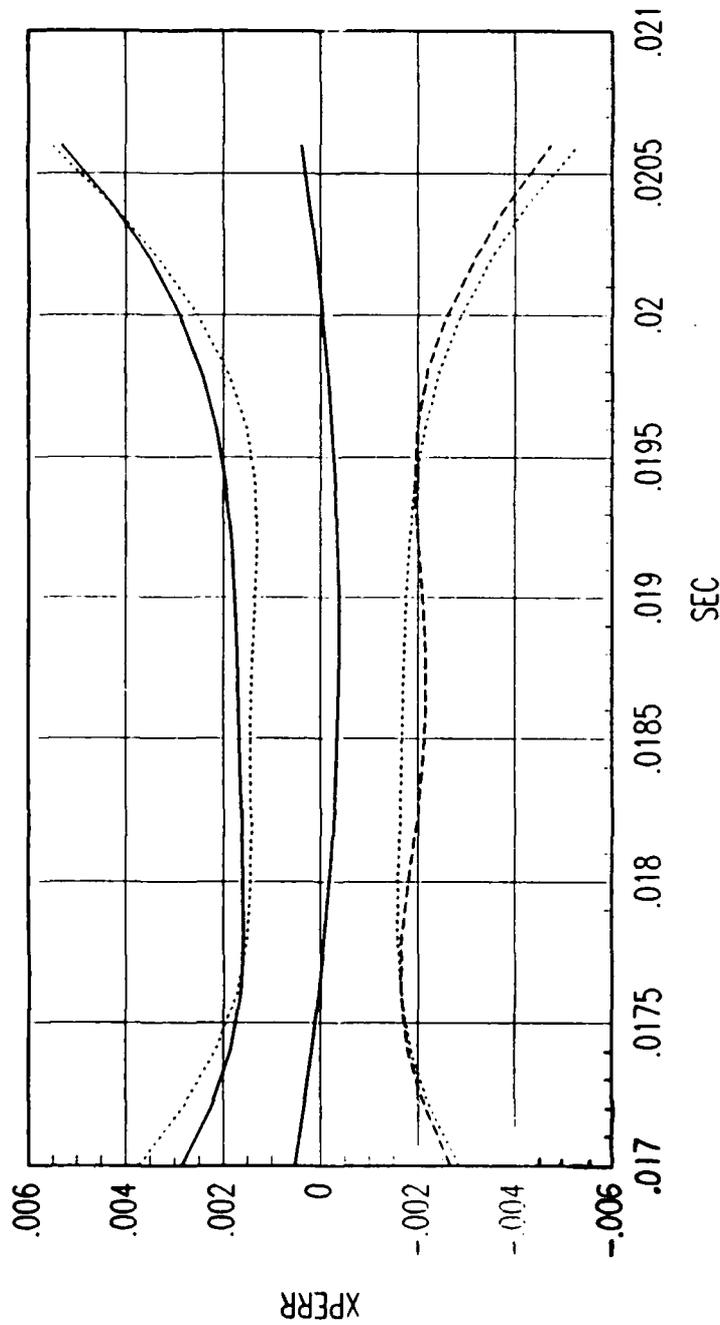


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C. 73. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 74. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

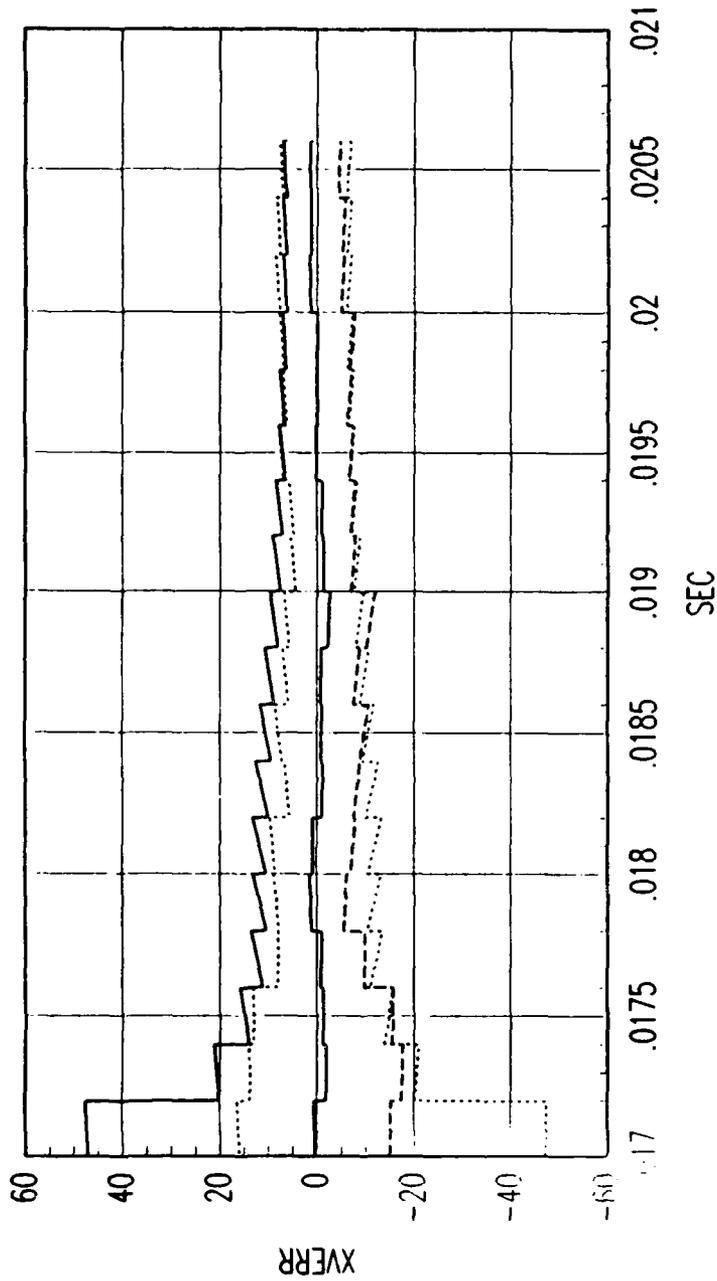
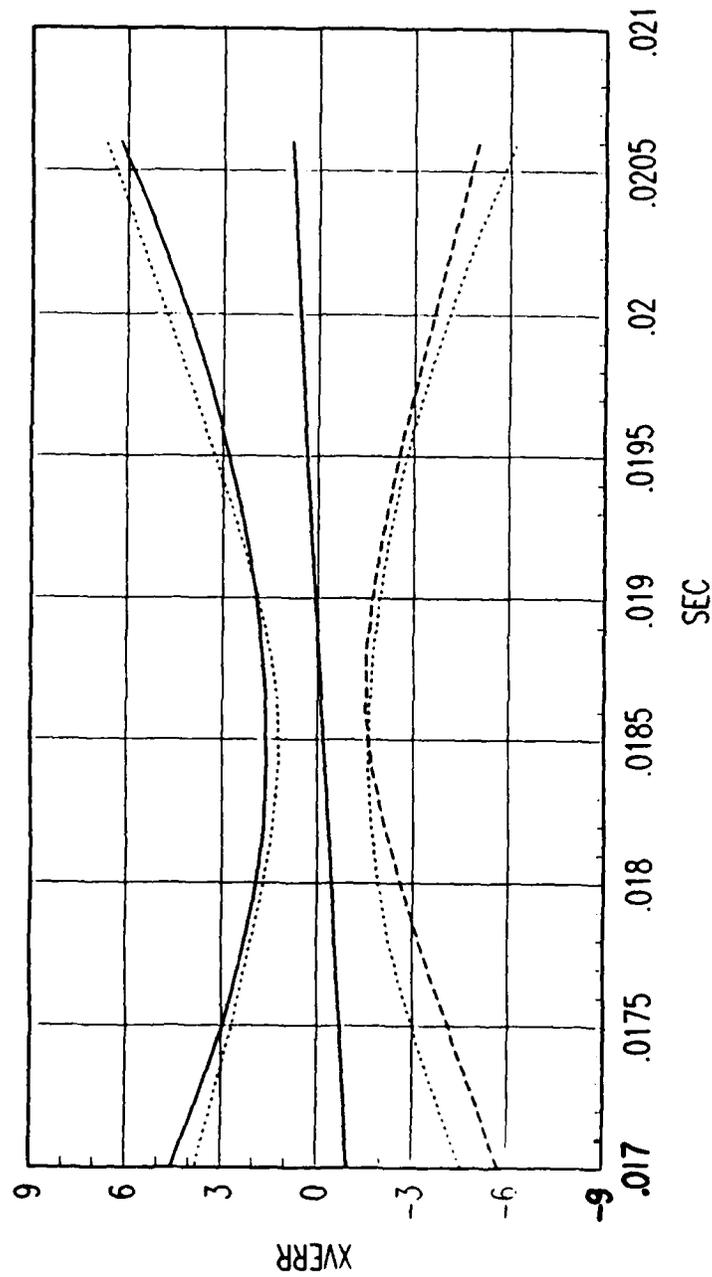


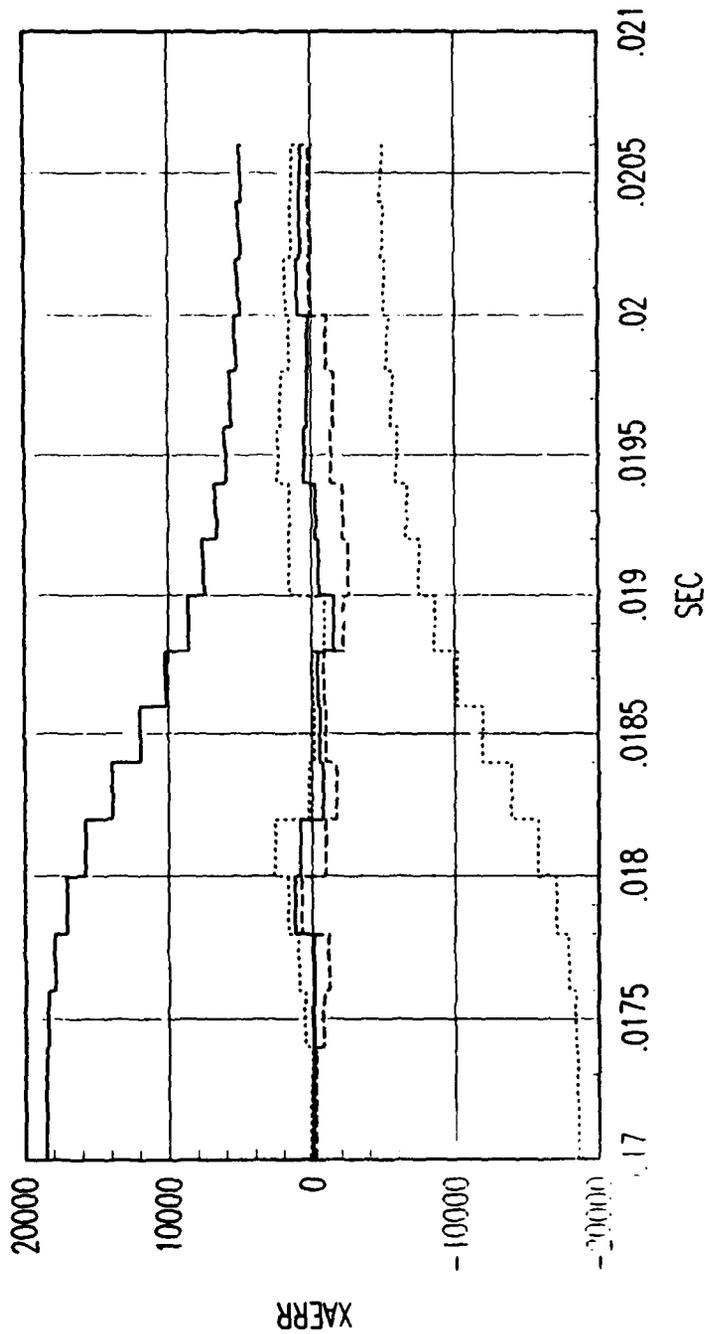
FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C. 75. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 76. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C. 77. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

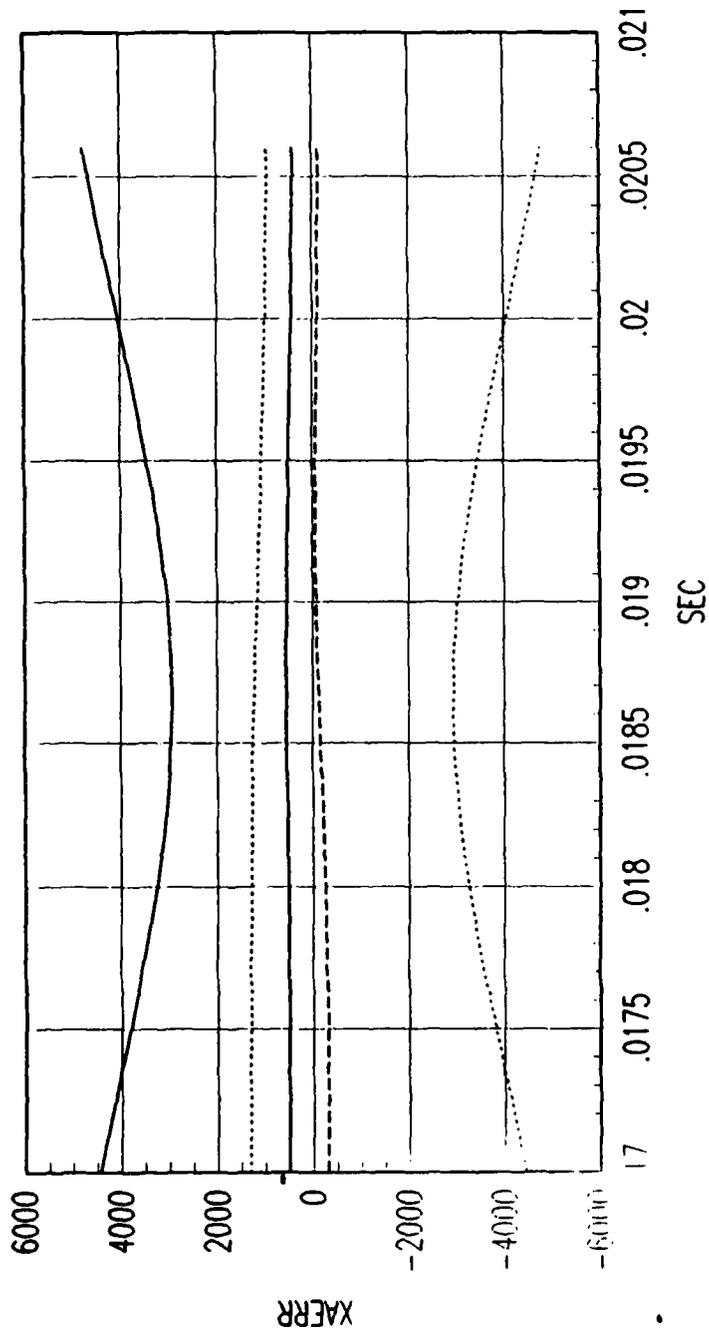
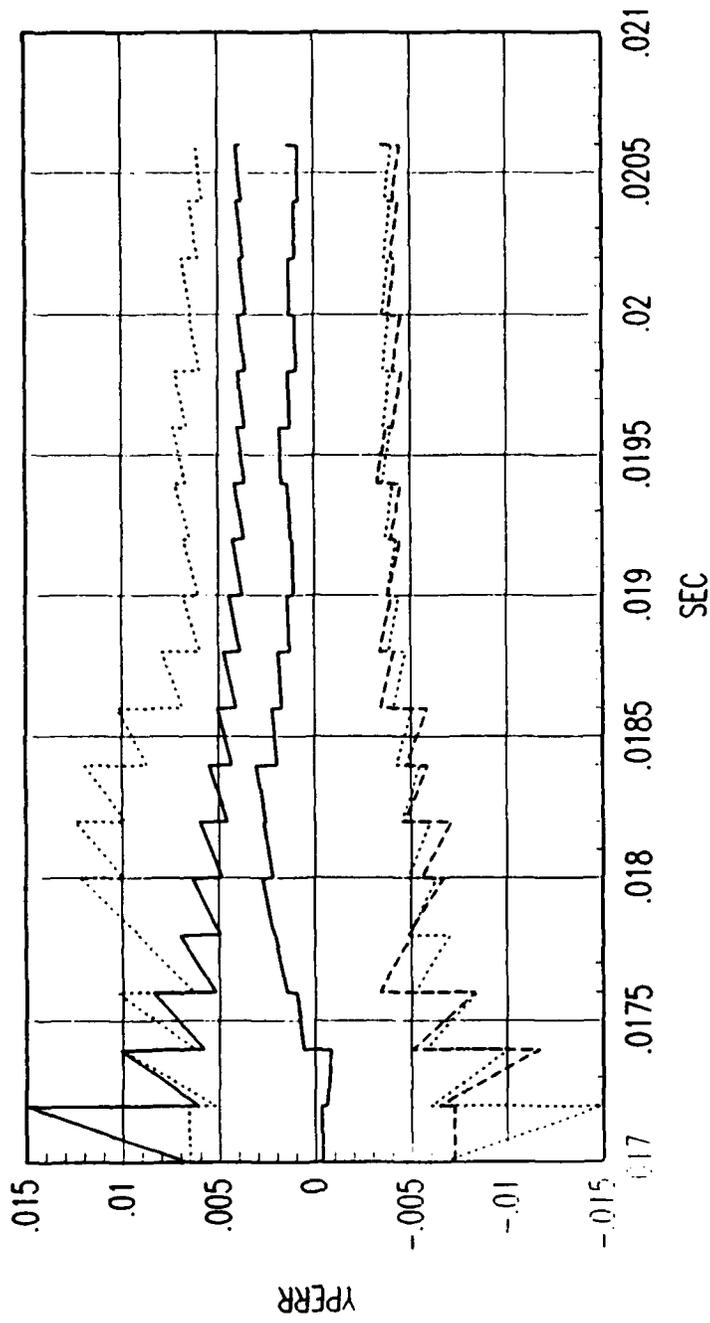


FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 78. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B FORWARD TIME EKF POSITION ERROR

Figure C. 79. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

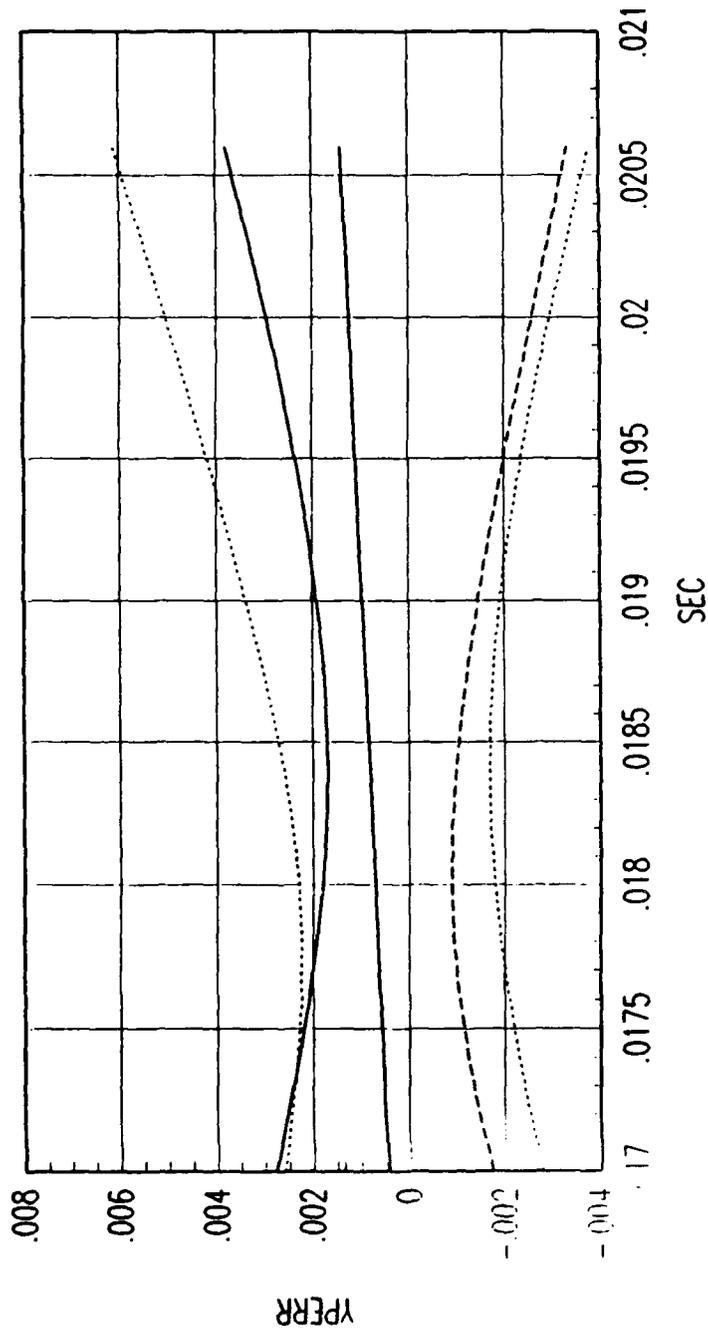
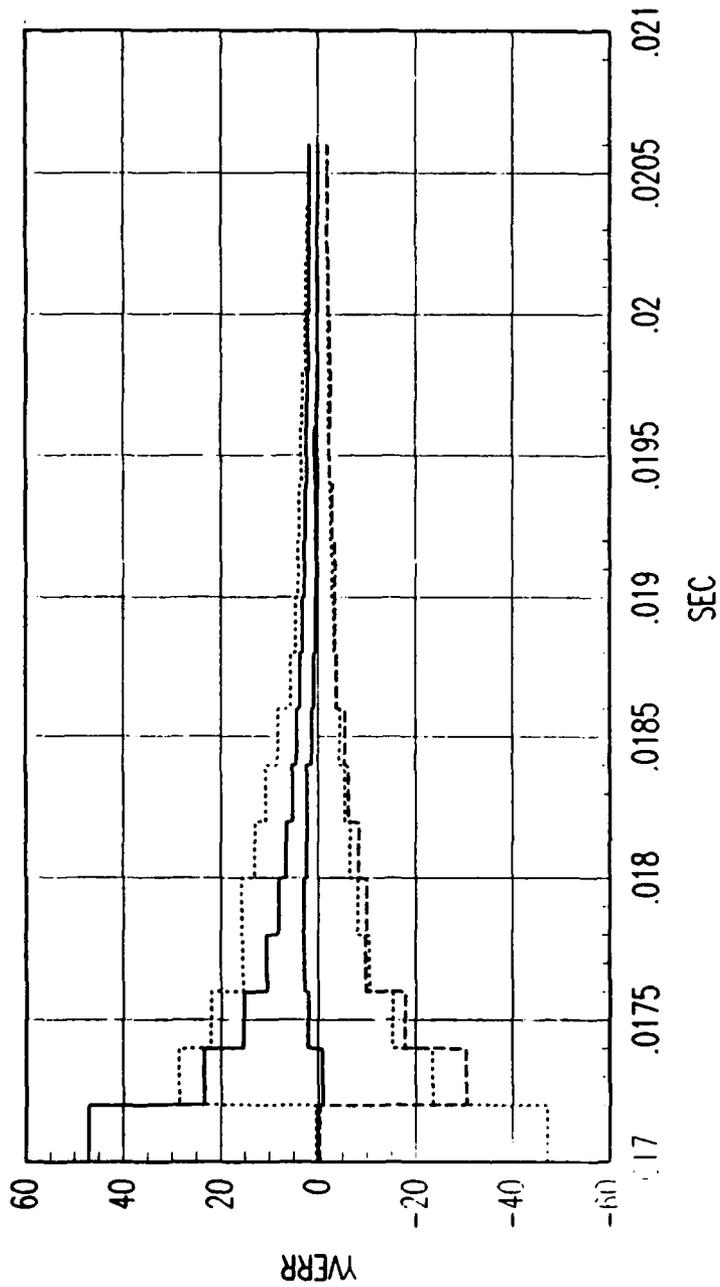


FIG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 80. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B FORWARD TIME EKF VELOCITY ERROR

Figure C. 81. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

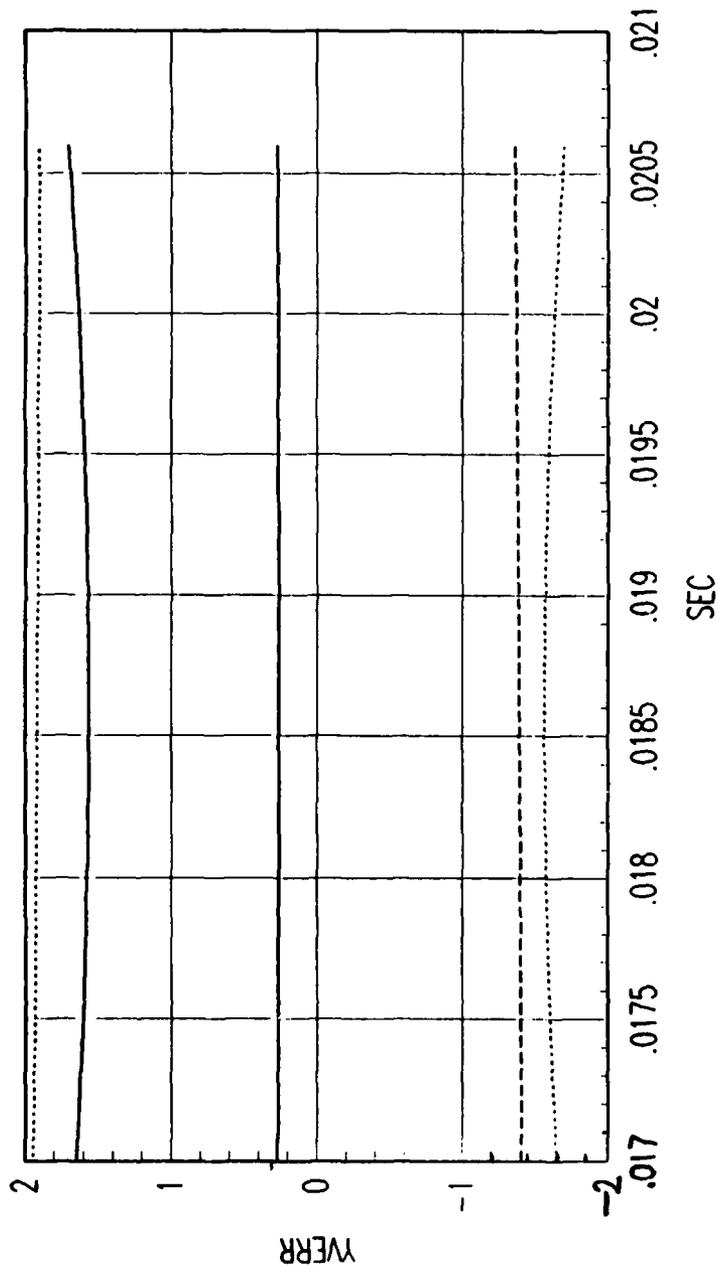
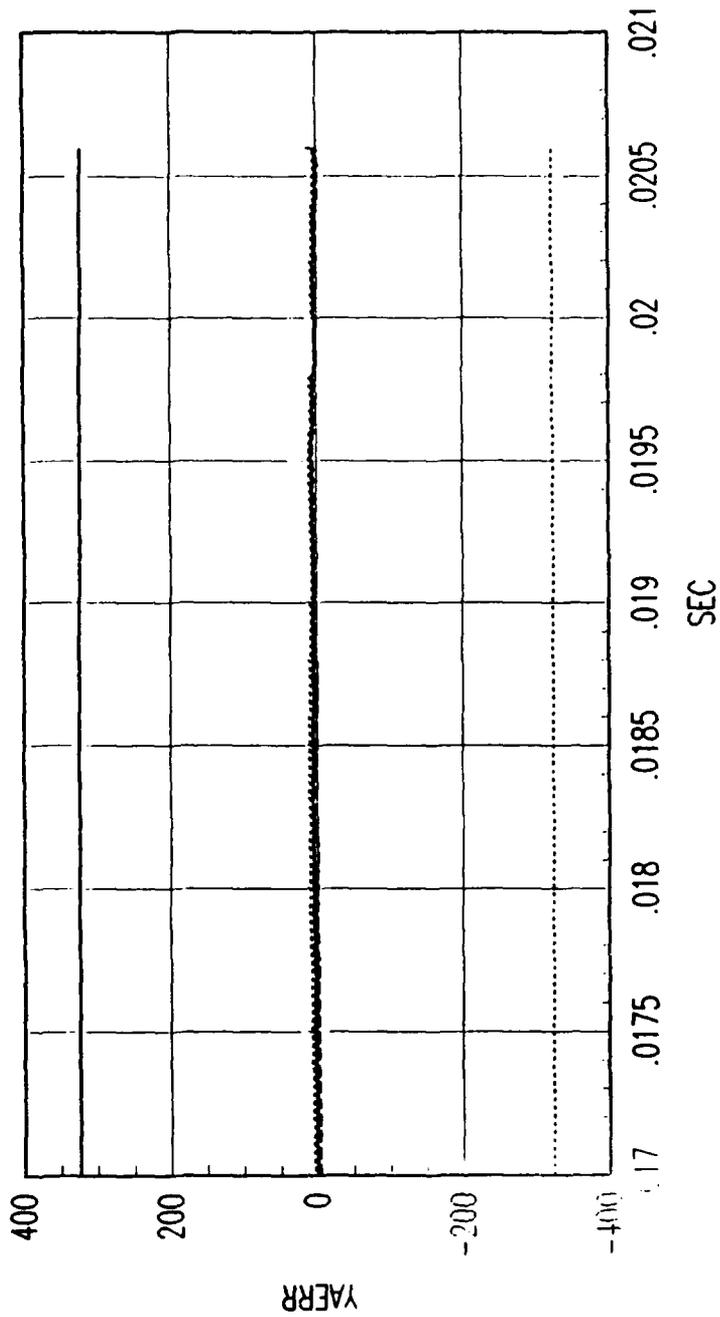


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 82. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C. 83. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

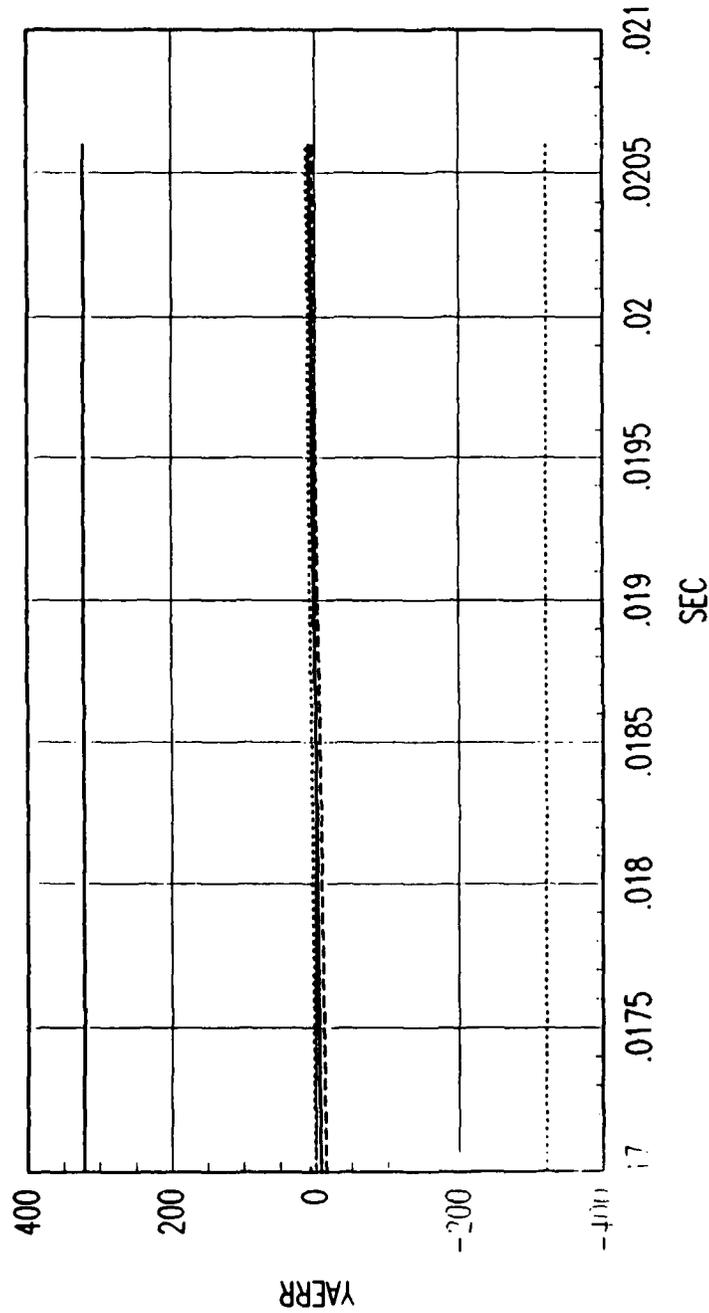


FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 84. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

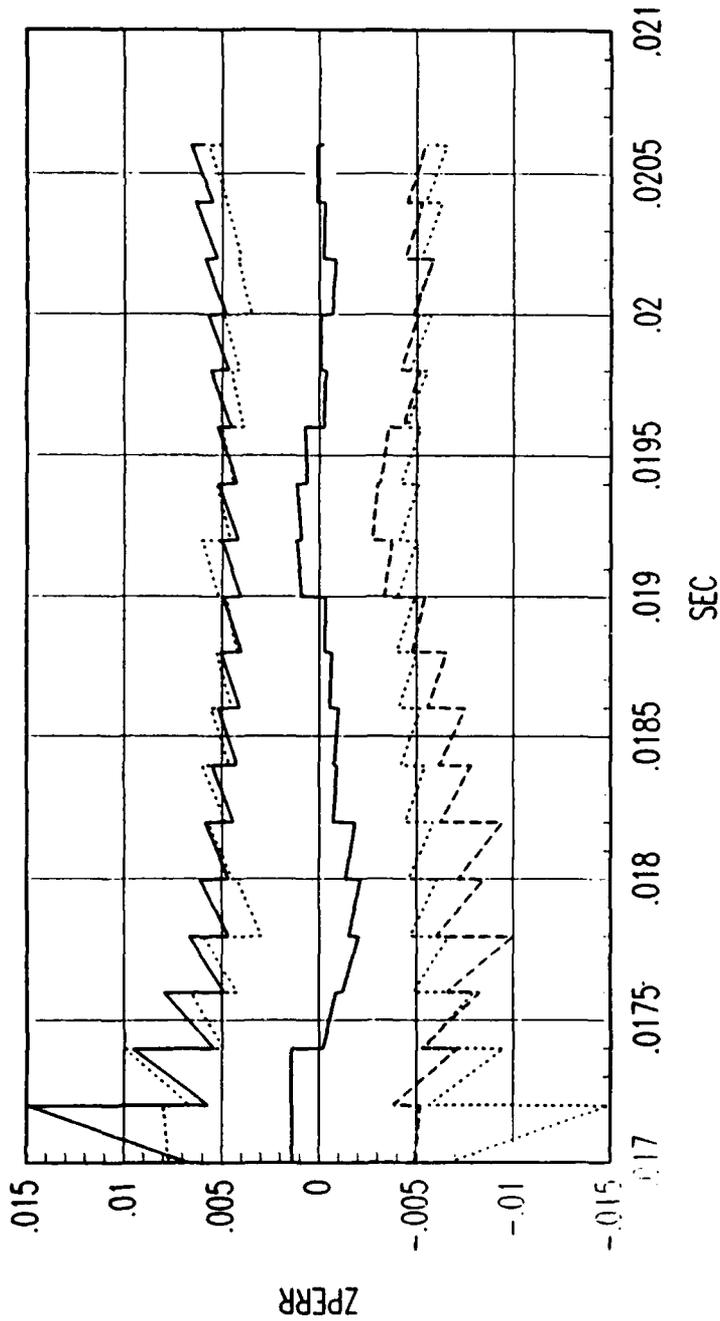
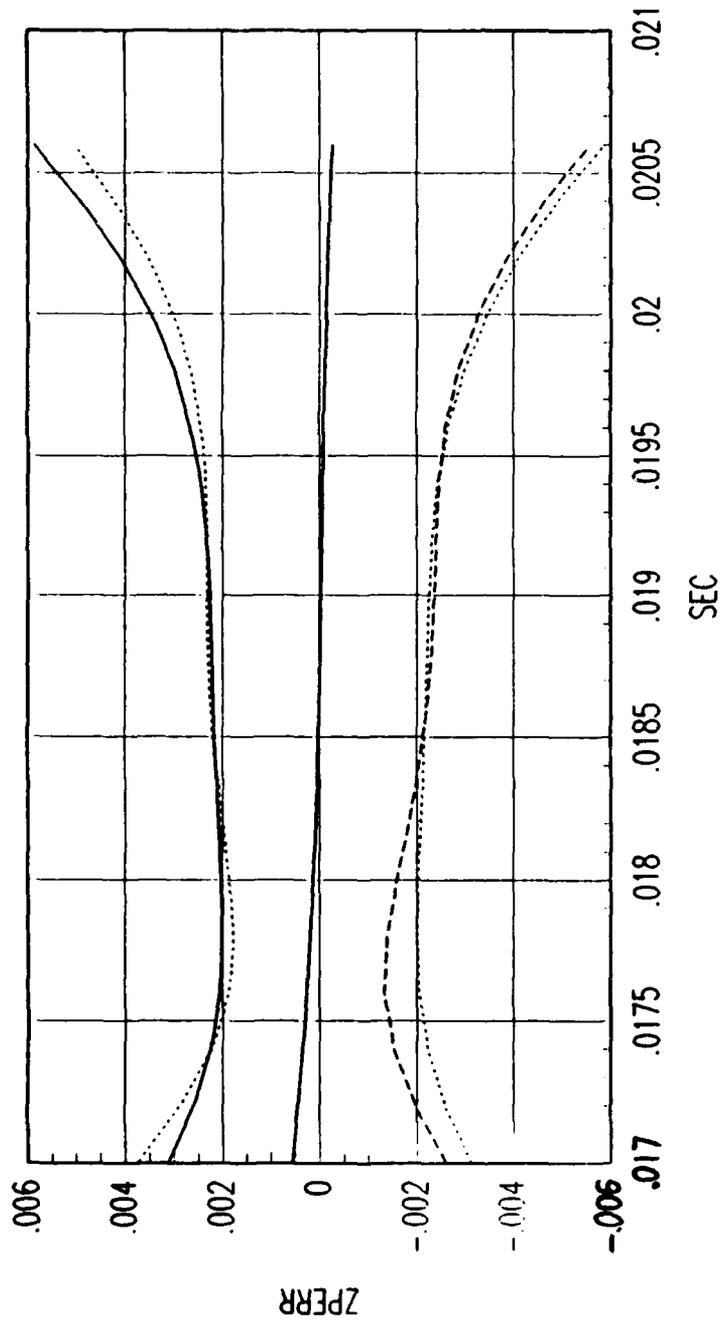


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C. 85. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 86. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

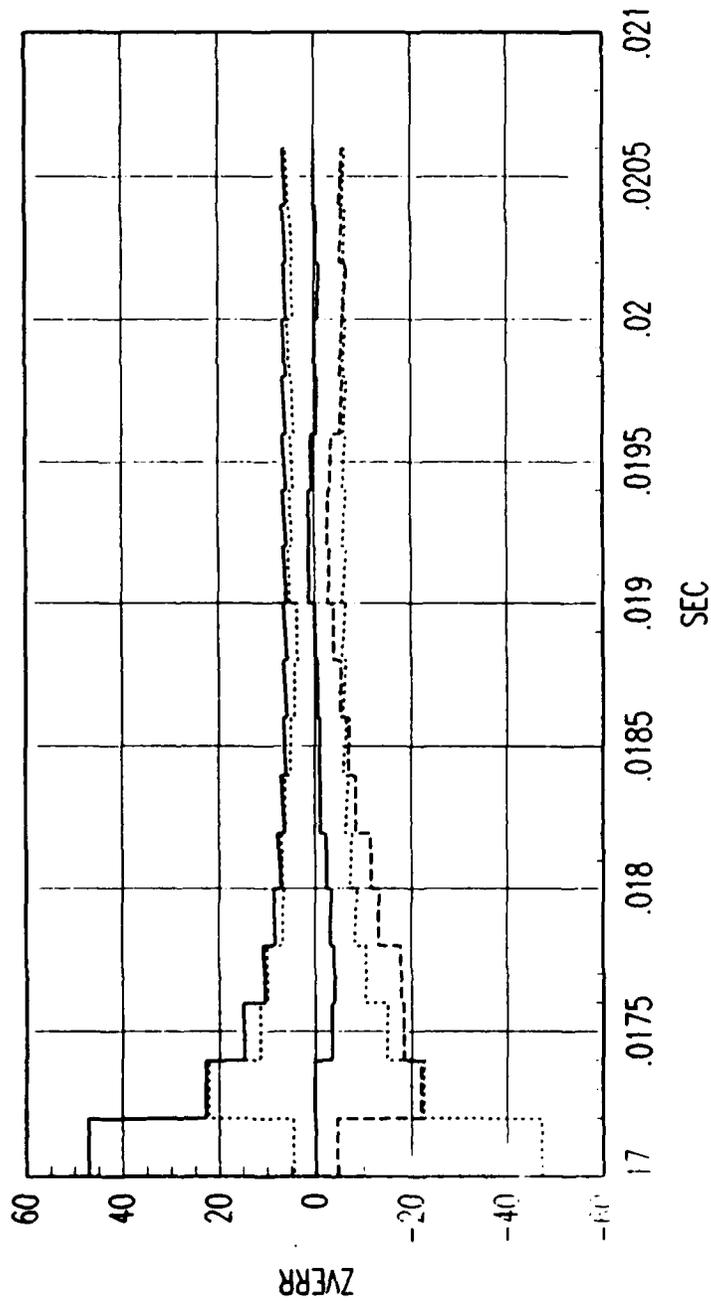
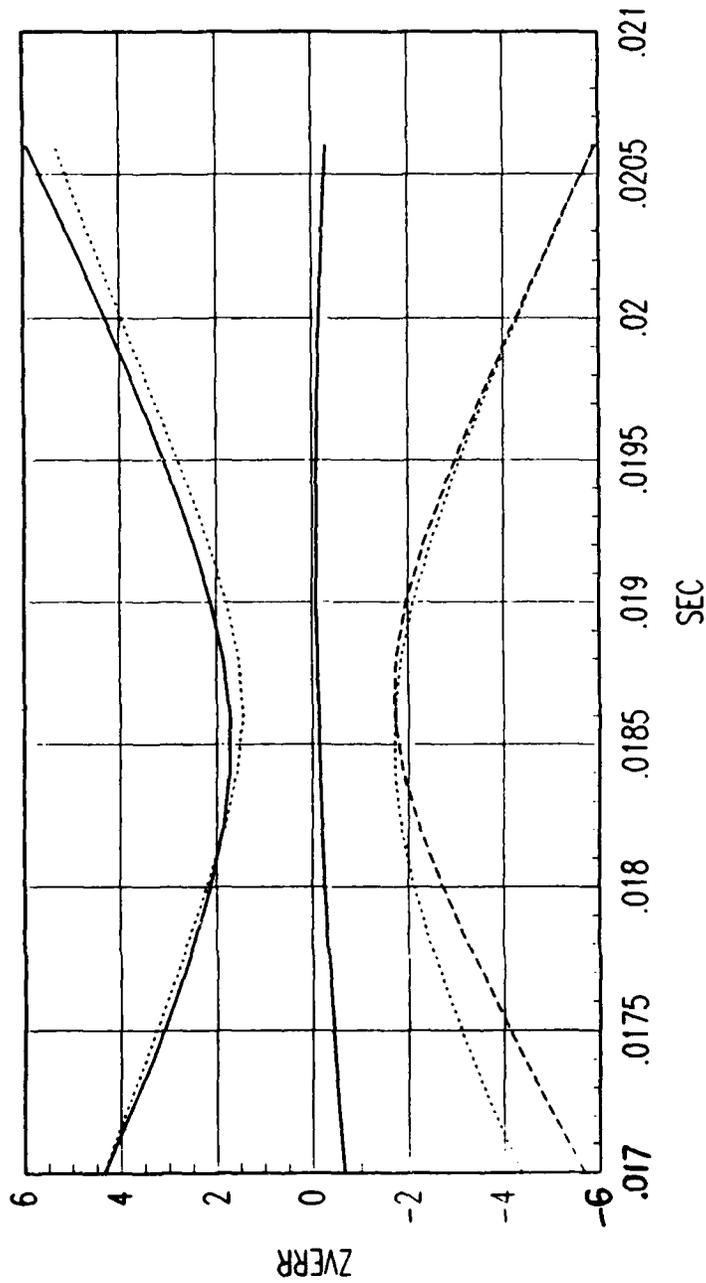


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C. 87. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 88. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

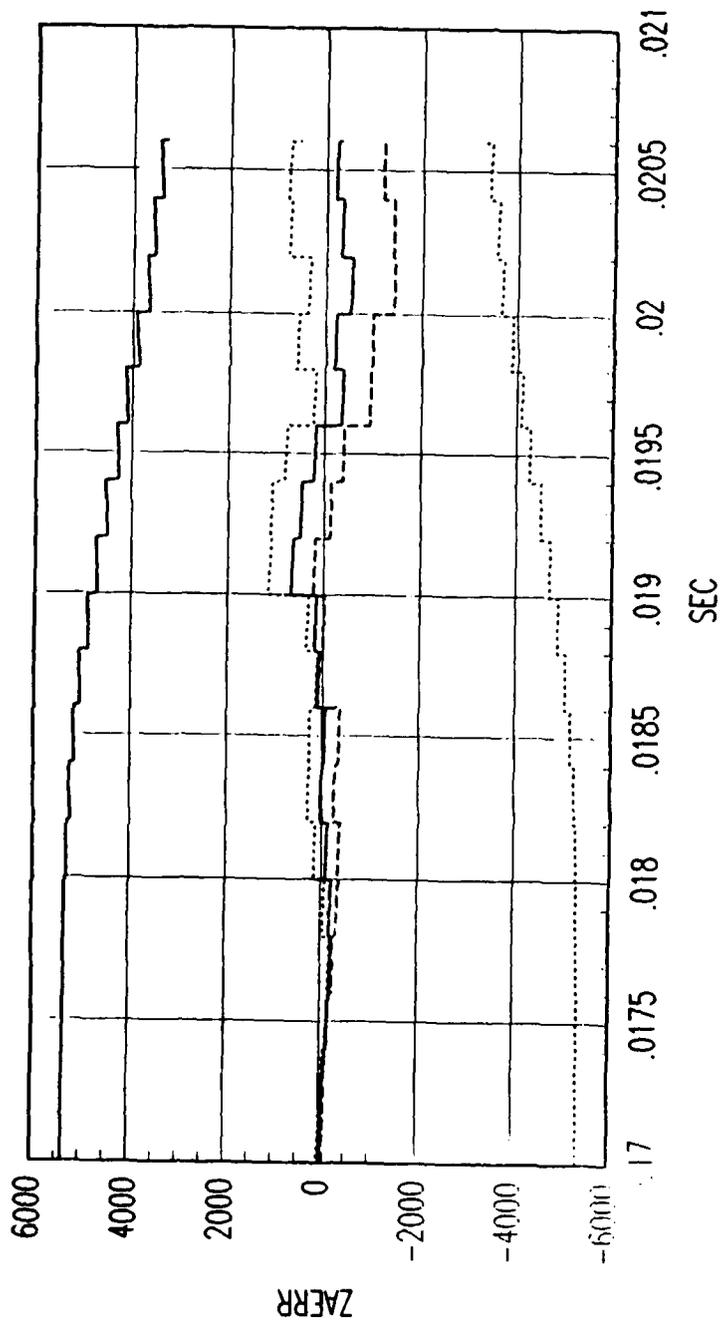
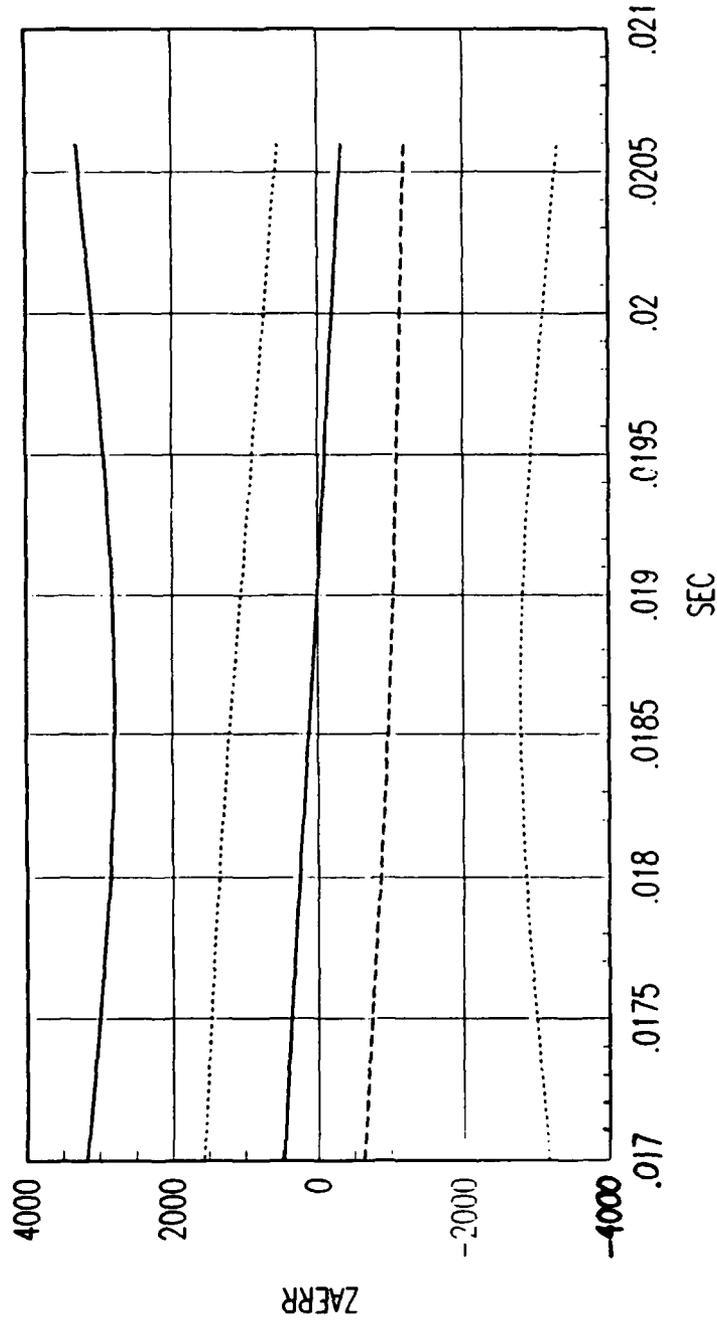


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C. 89. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 90. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

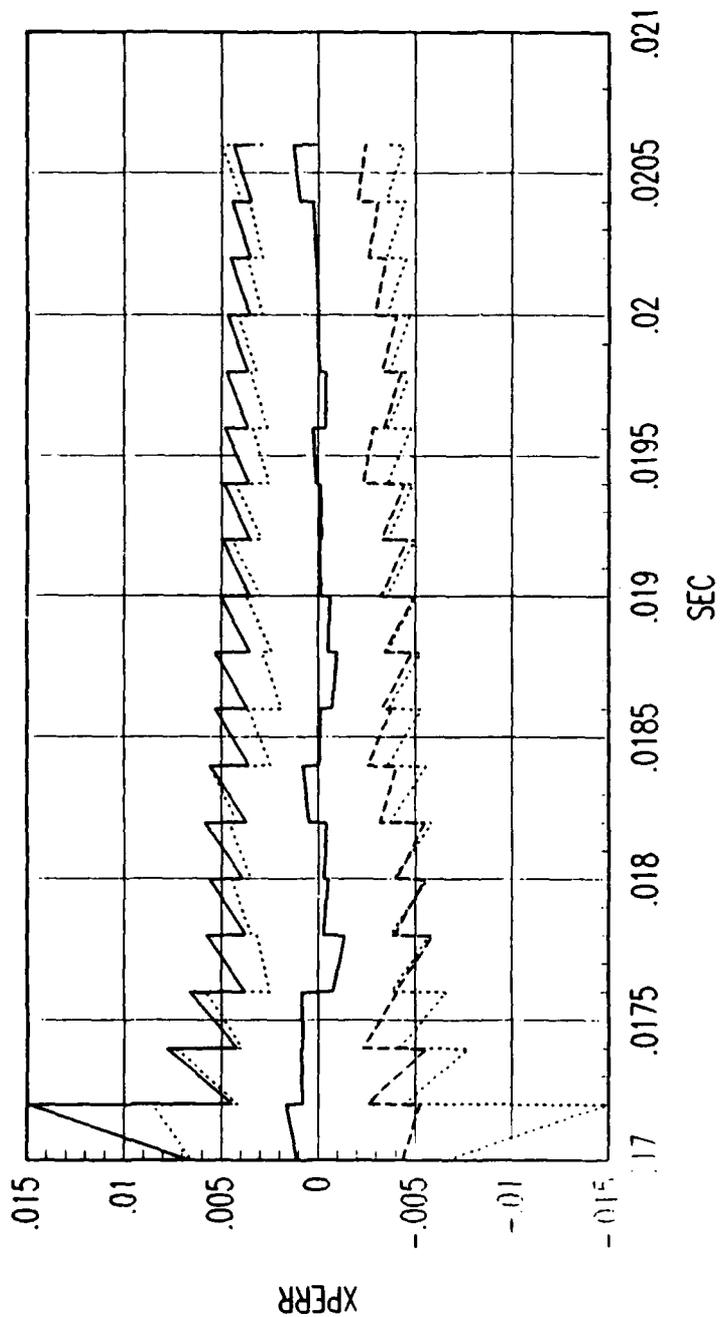
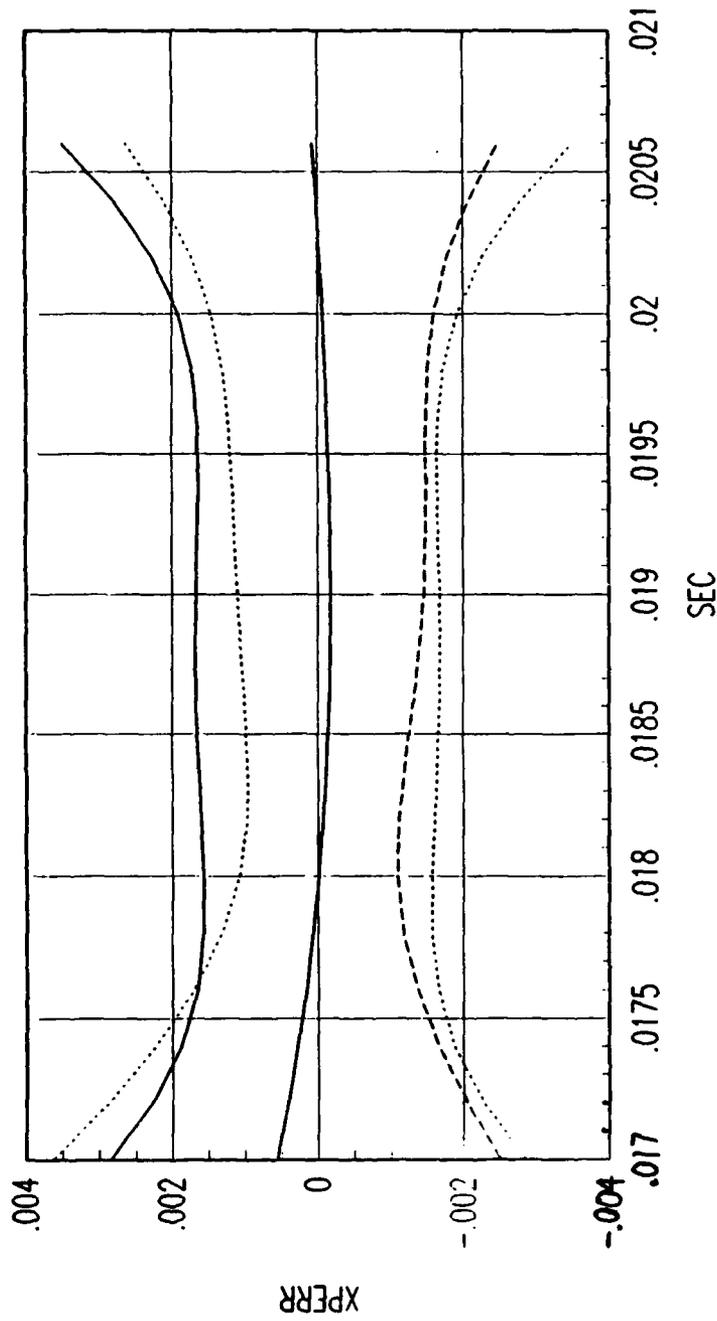


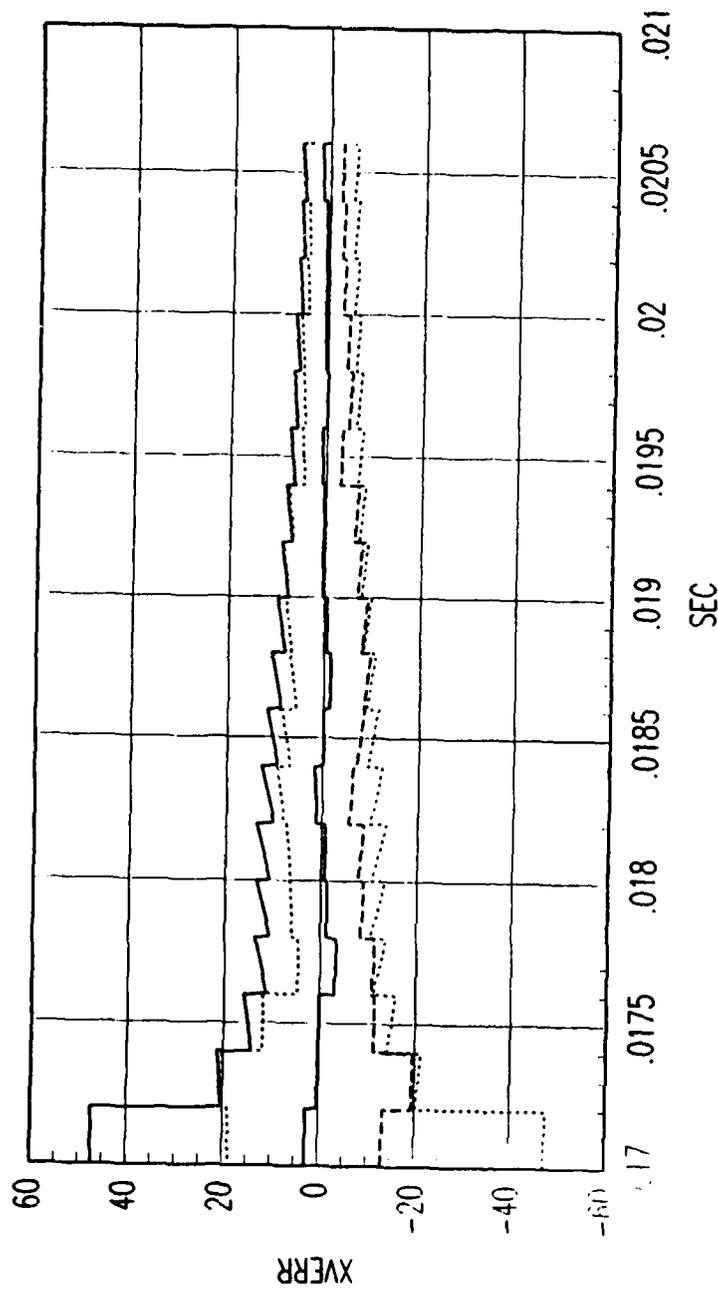
FIG. C FORWARD TIME EKF POSITION ERROR

Figure C. 91. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



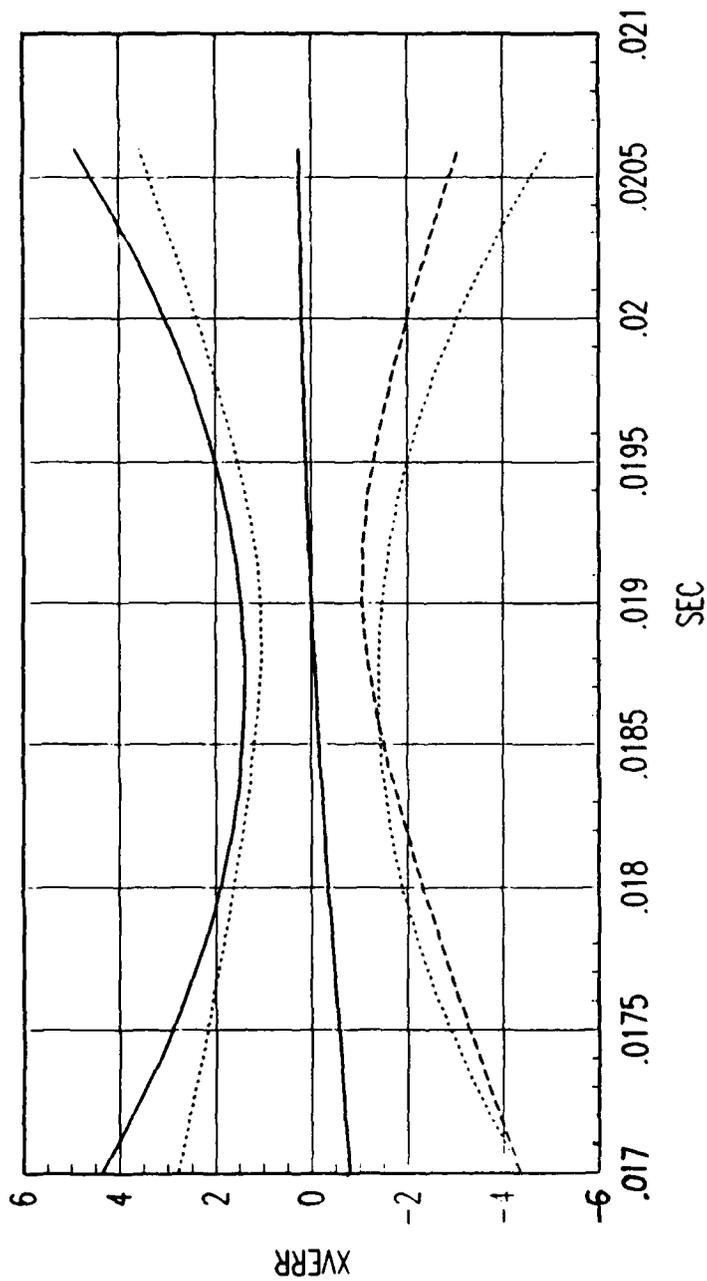
FRG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 92. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



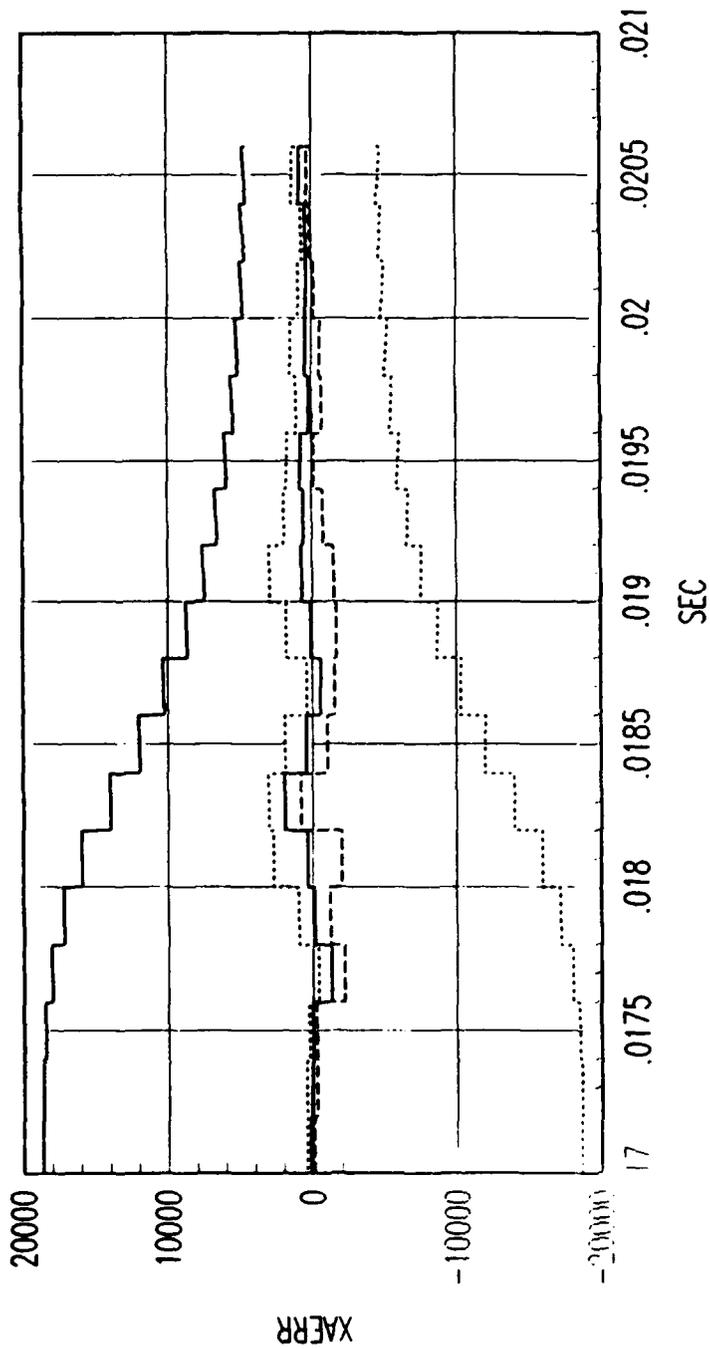
FRG. C FORWARD TIME EKF VELOCITY ERROR

Figure C. 93. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



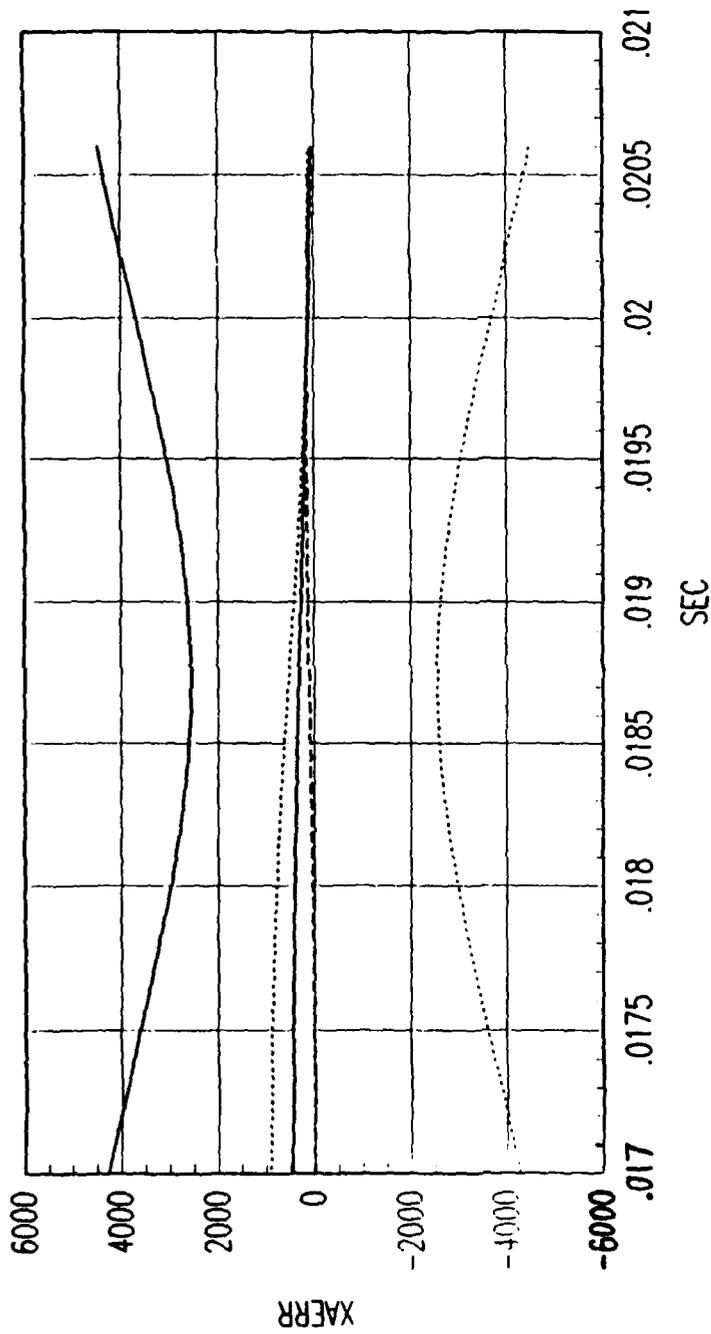
FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C. 94. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



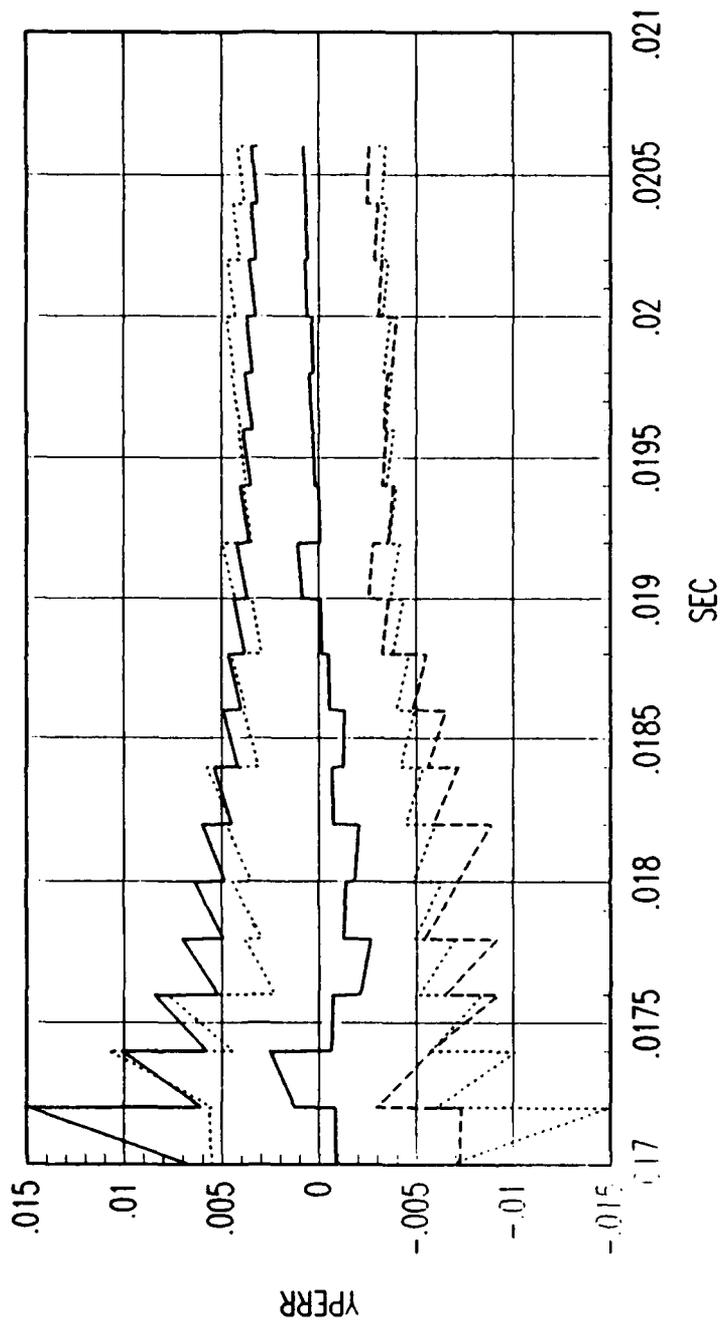
FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C. 95. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



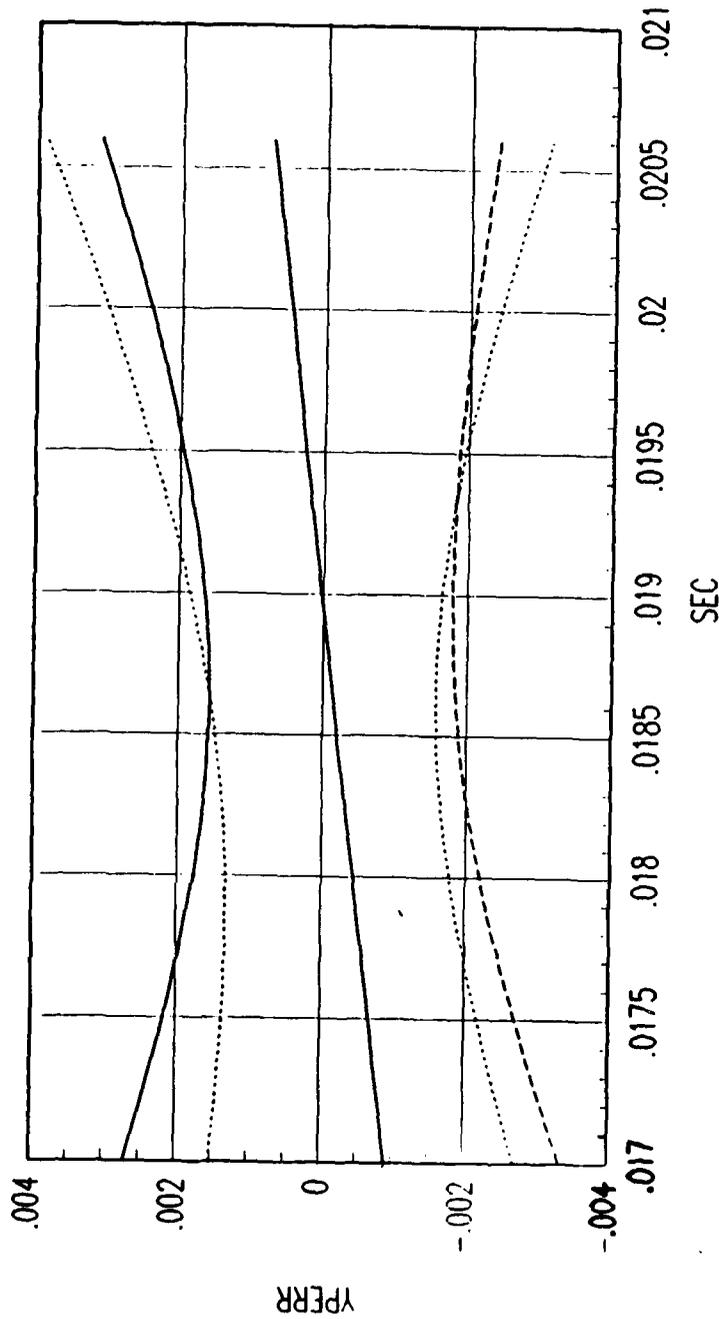
FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C. 96. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. C FORWARD TIME EKF POSITION ERROR

Figure C. 97. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C. 98. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

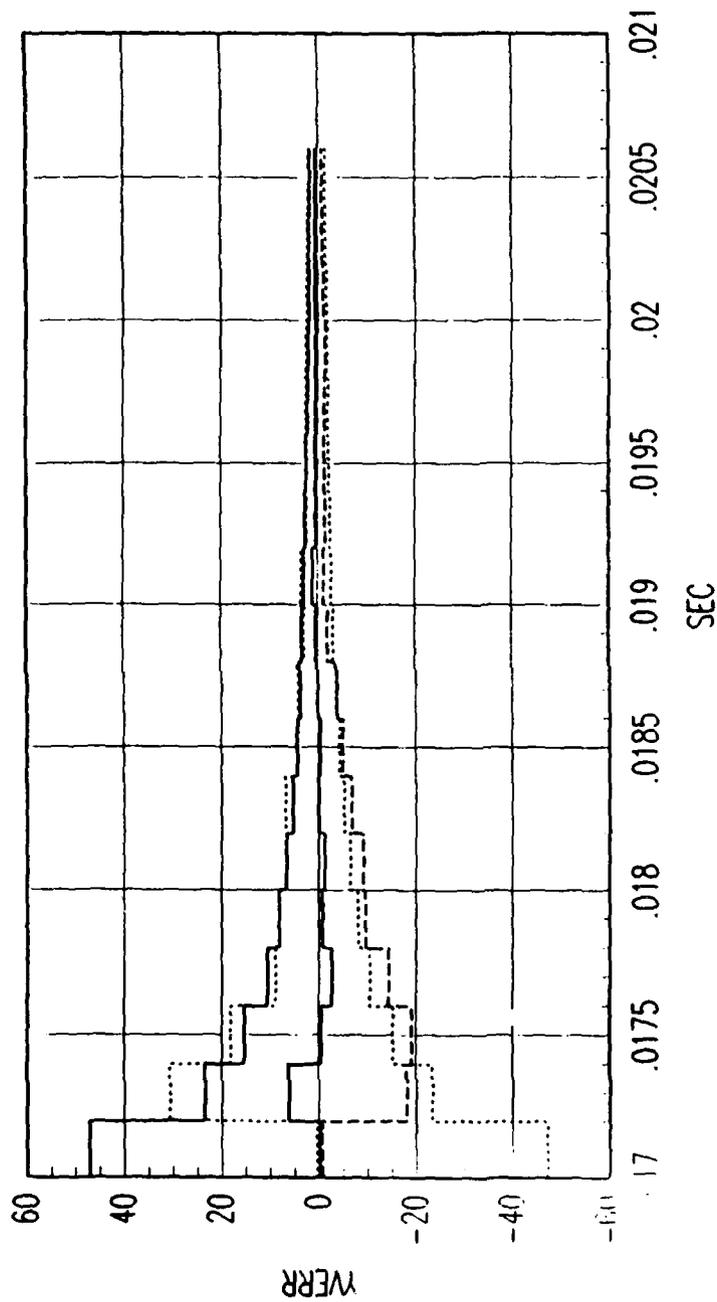
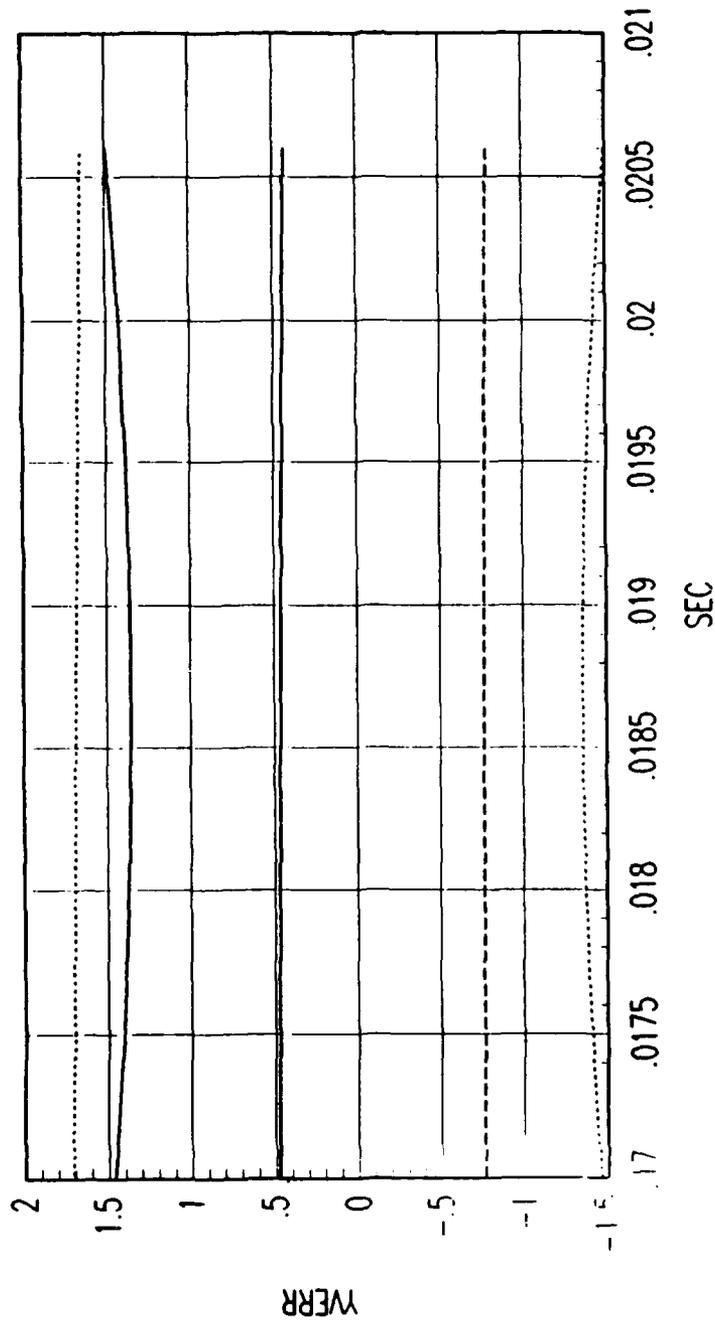


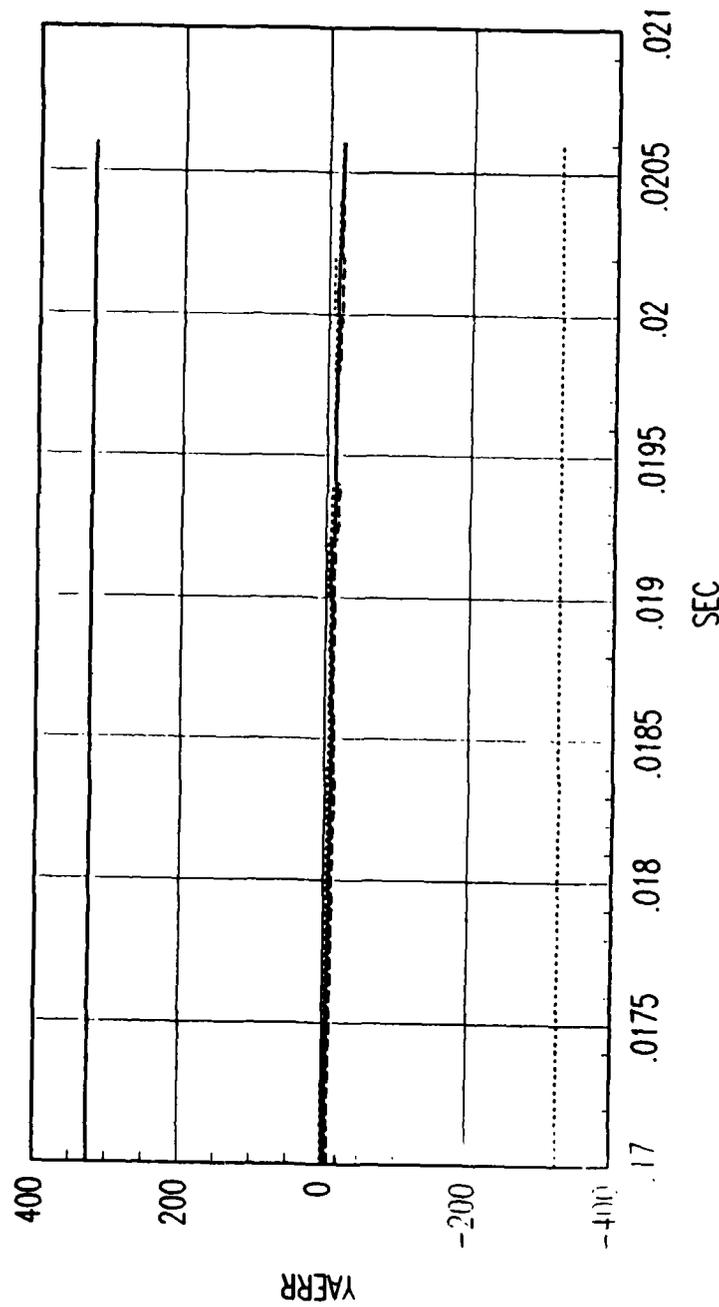
FIG. C FORWARD TIME EKf VELOCITY ERROR

Figure C. 99. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



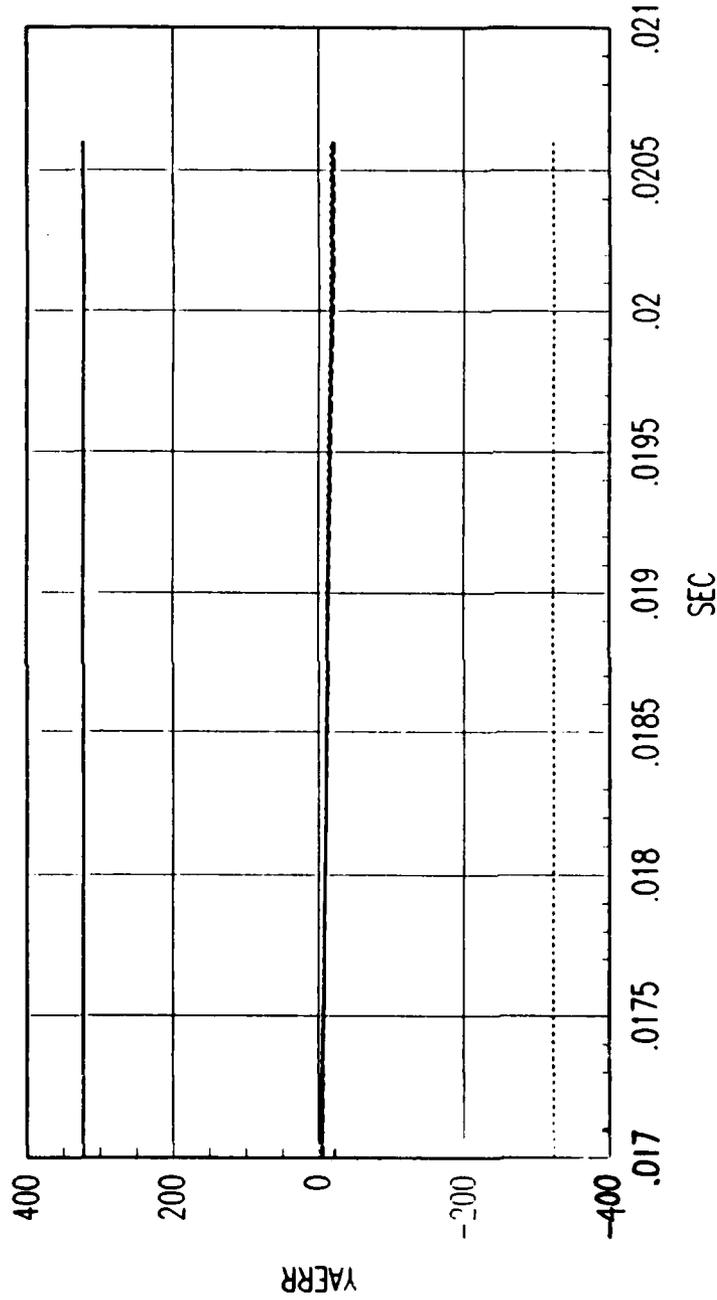
FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.100. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



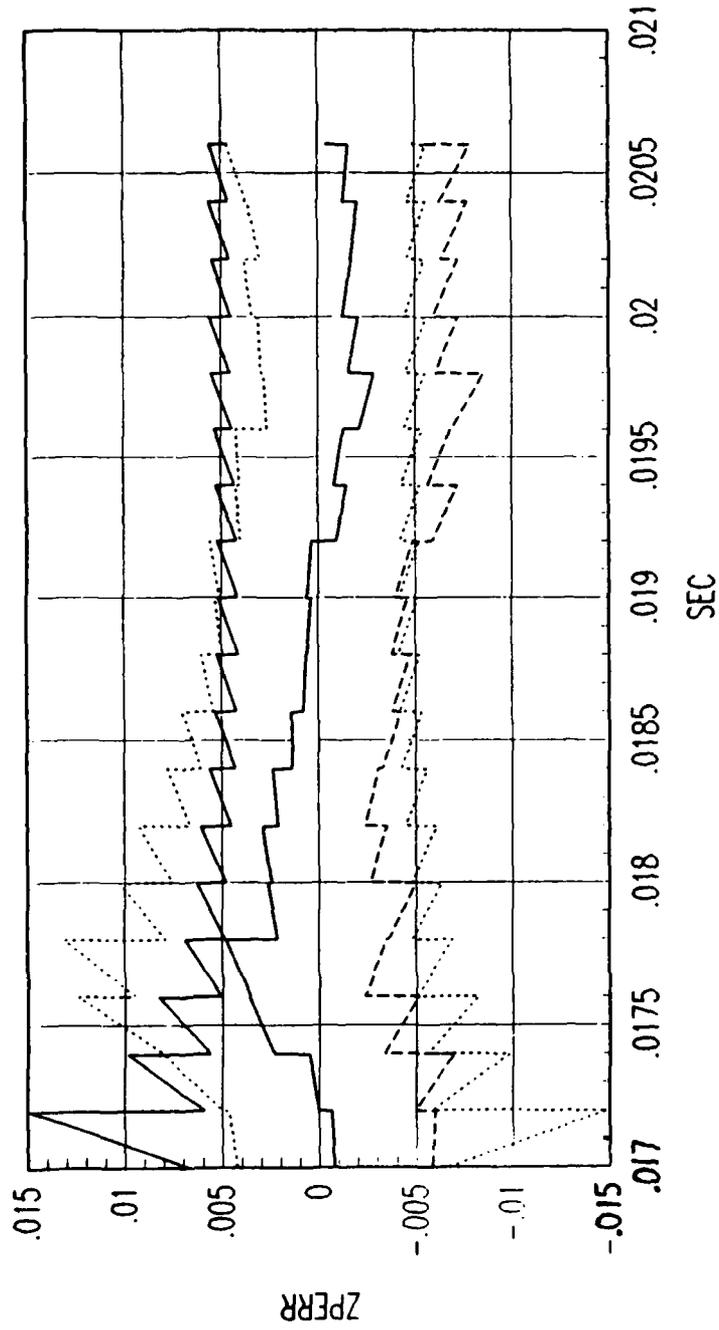
FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.101. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



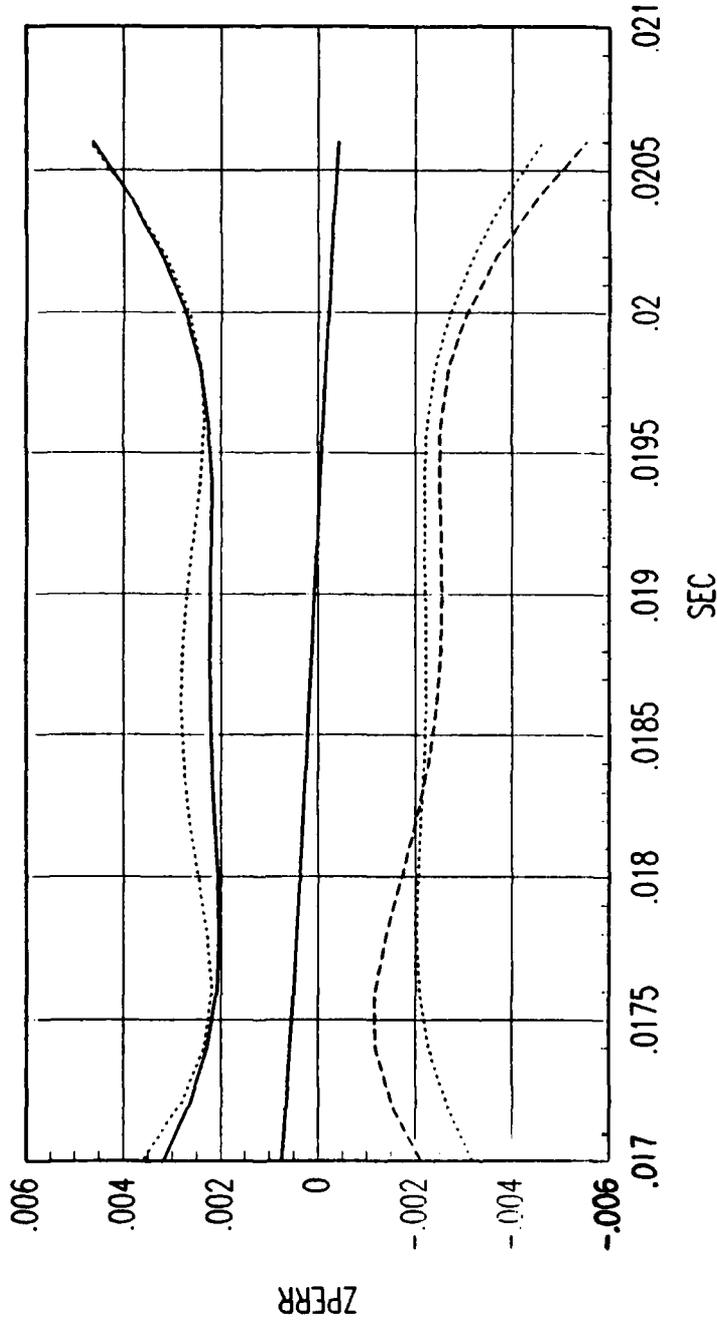
FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.102. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. C FORWARD TIME EKF POSITION ERROR

Figure C.103. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.104. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

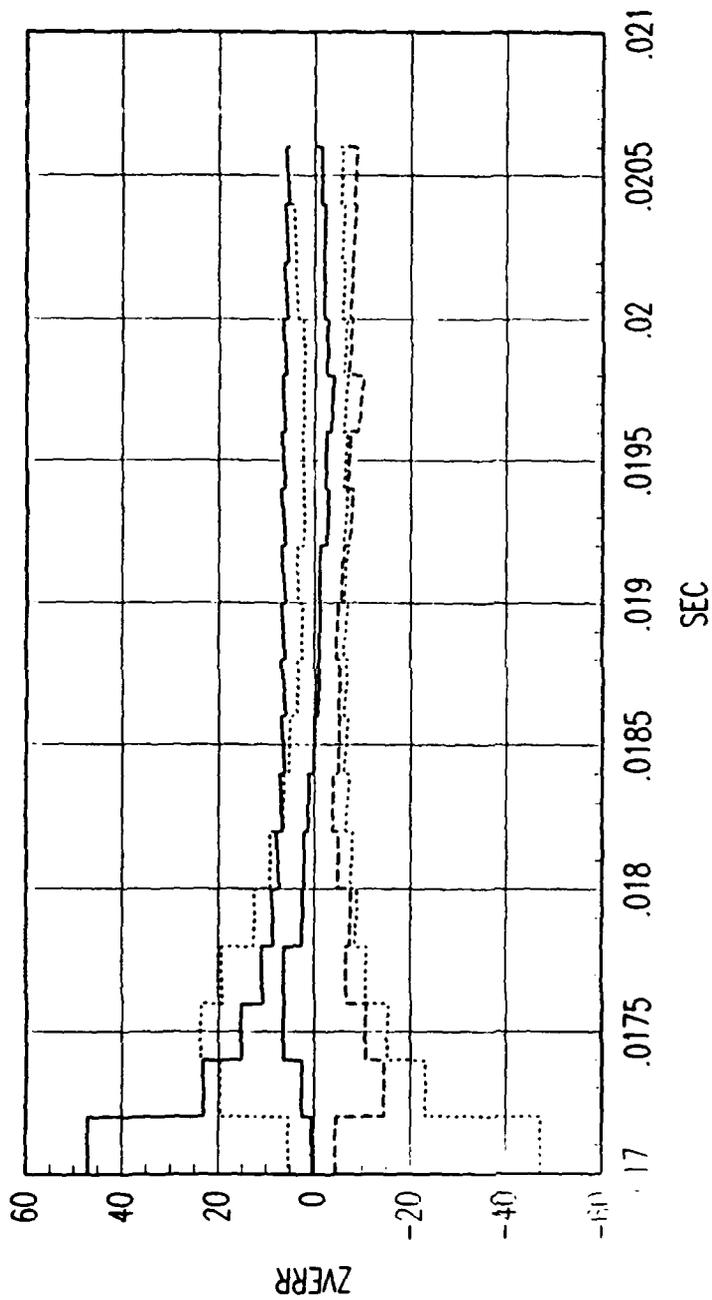
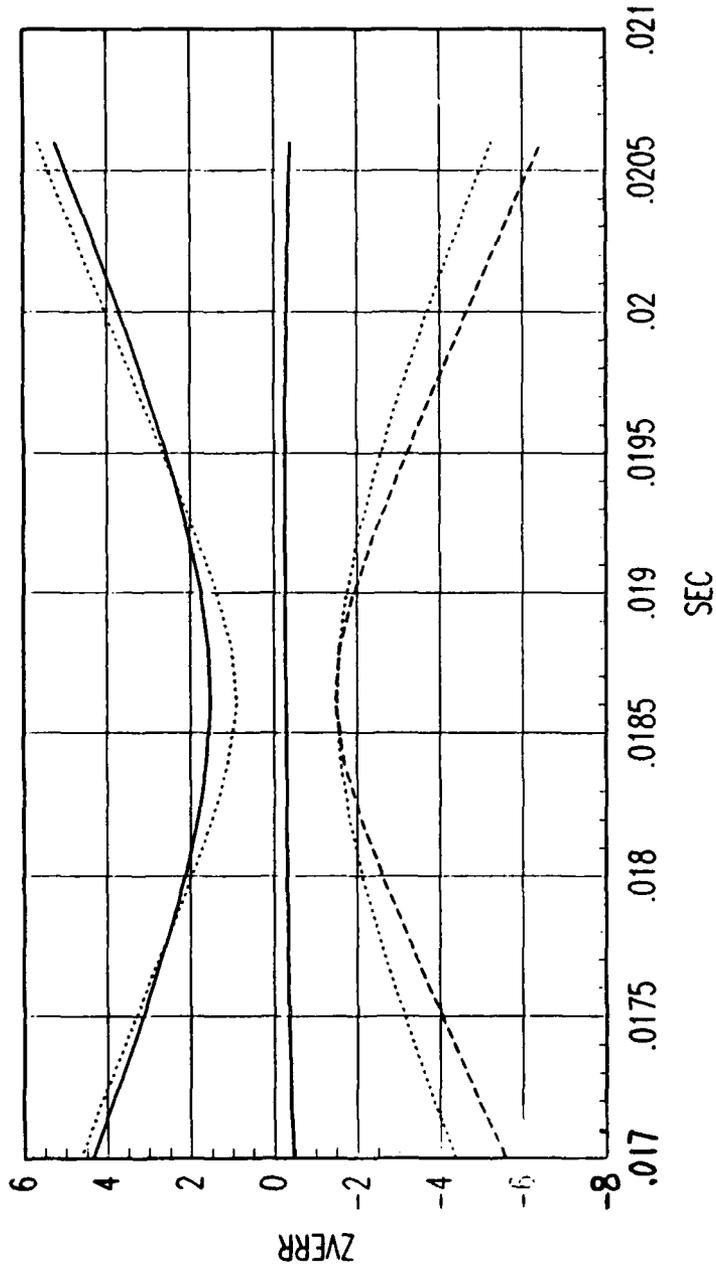


FIG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.105. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.106. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

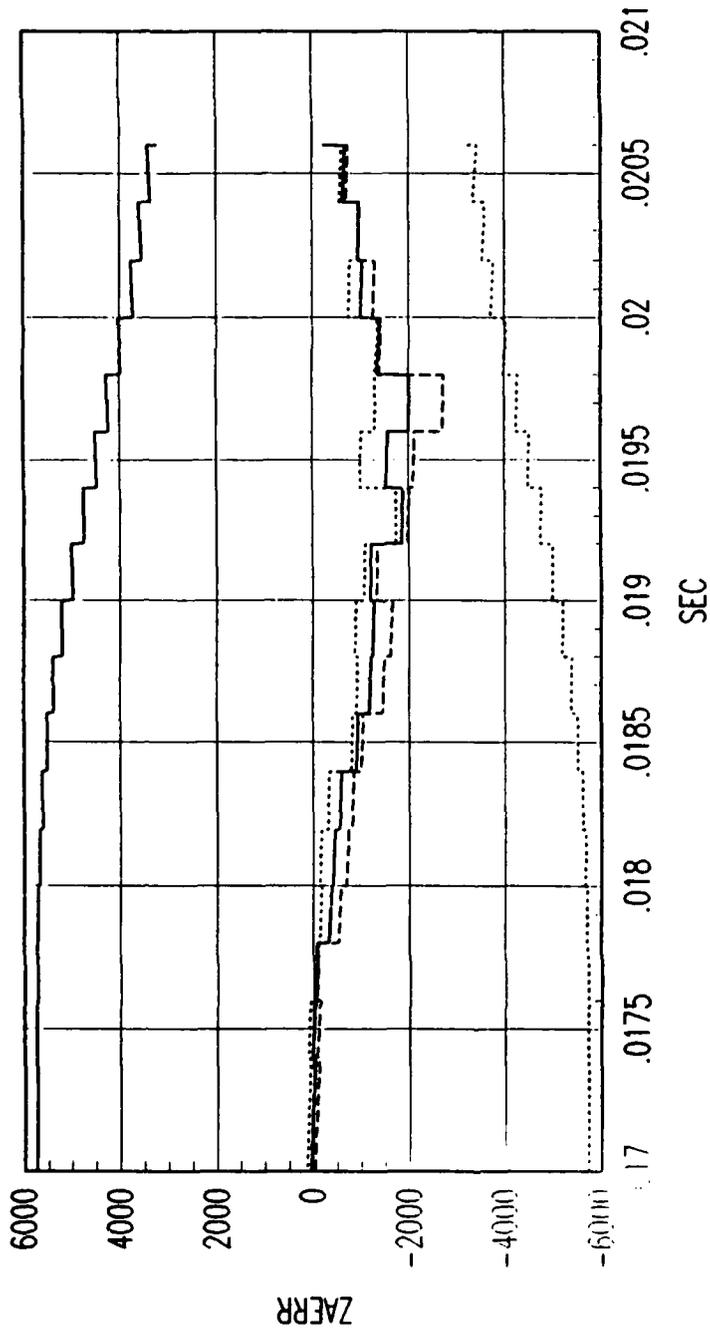
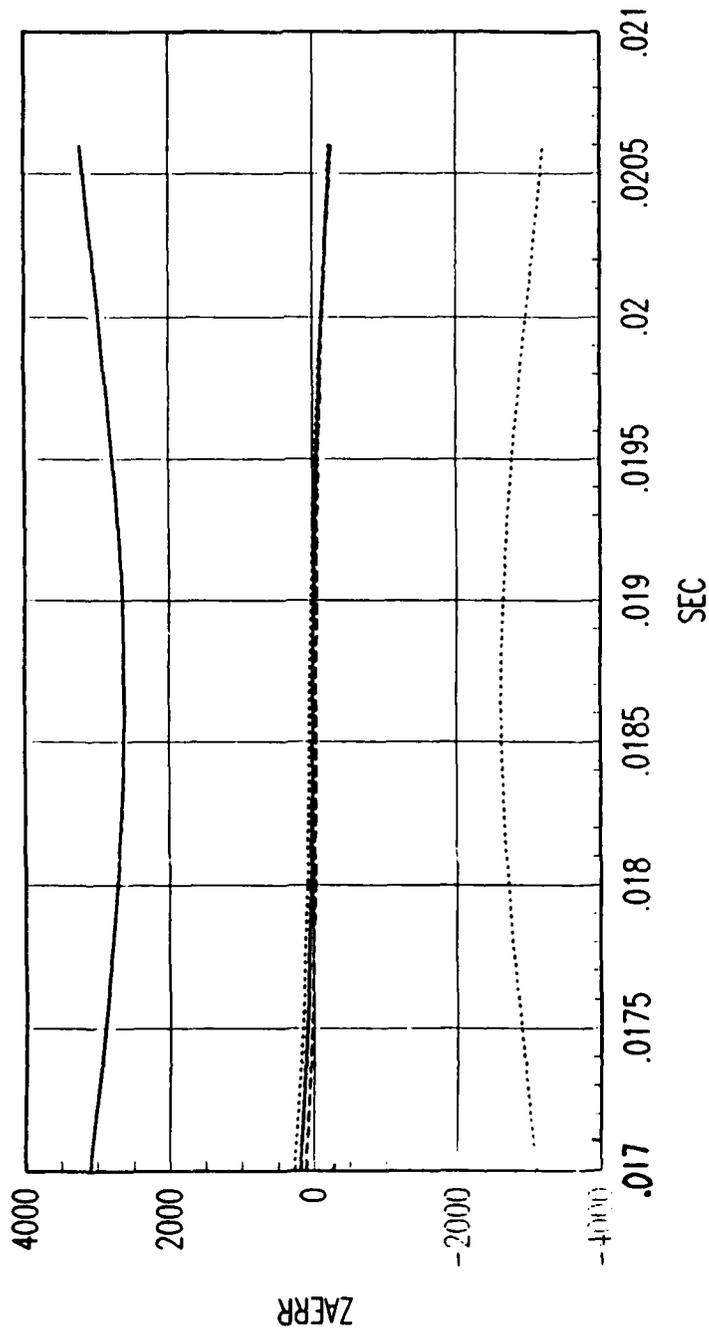


FIG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.107. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.108. Tracking Error Plot, Category 2, $v(t_0) = 3000$ ft/sec.

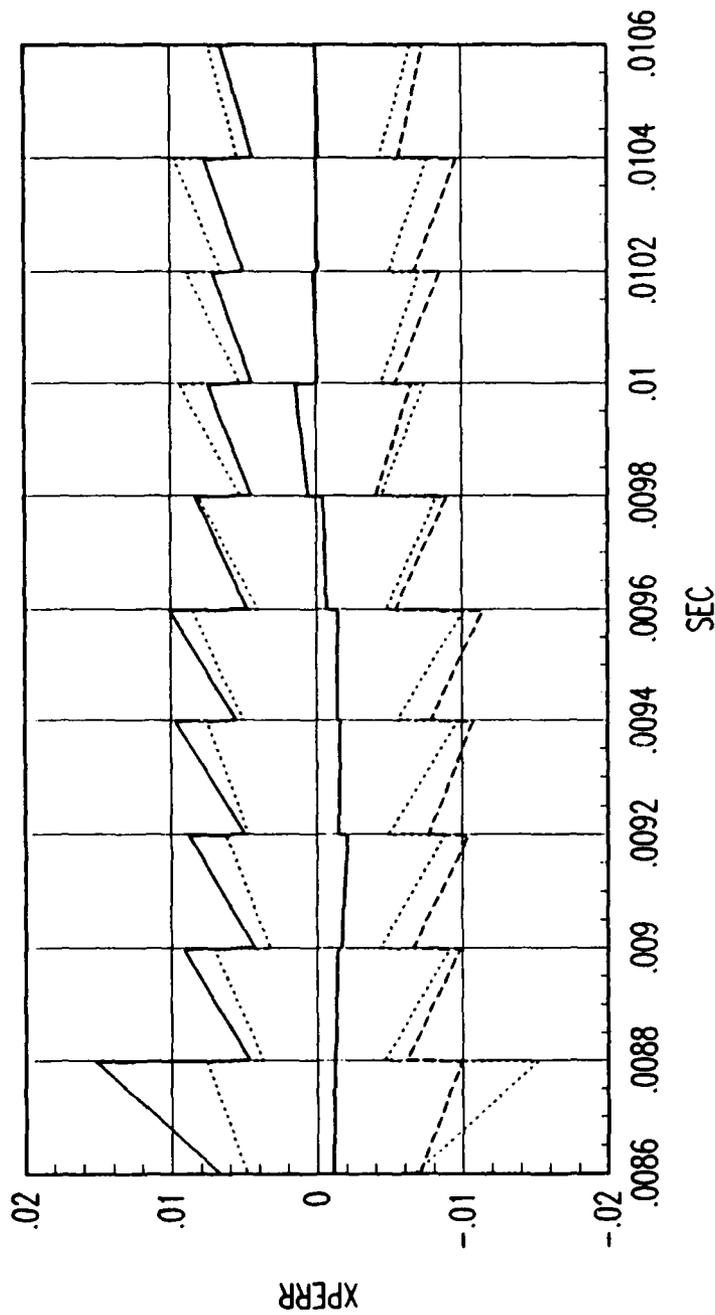


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.109. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

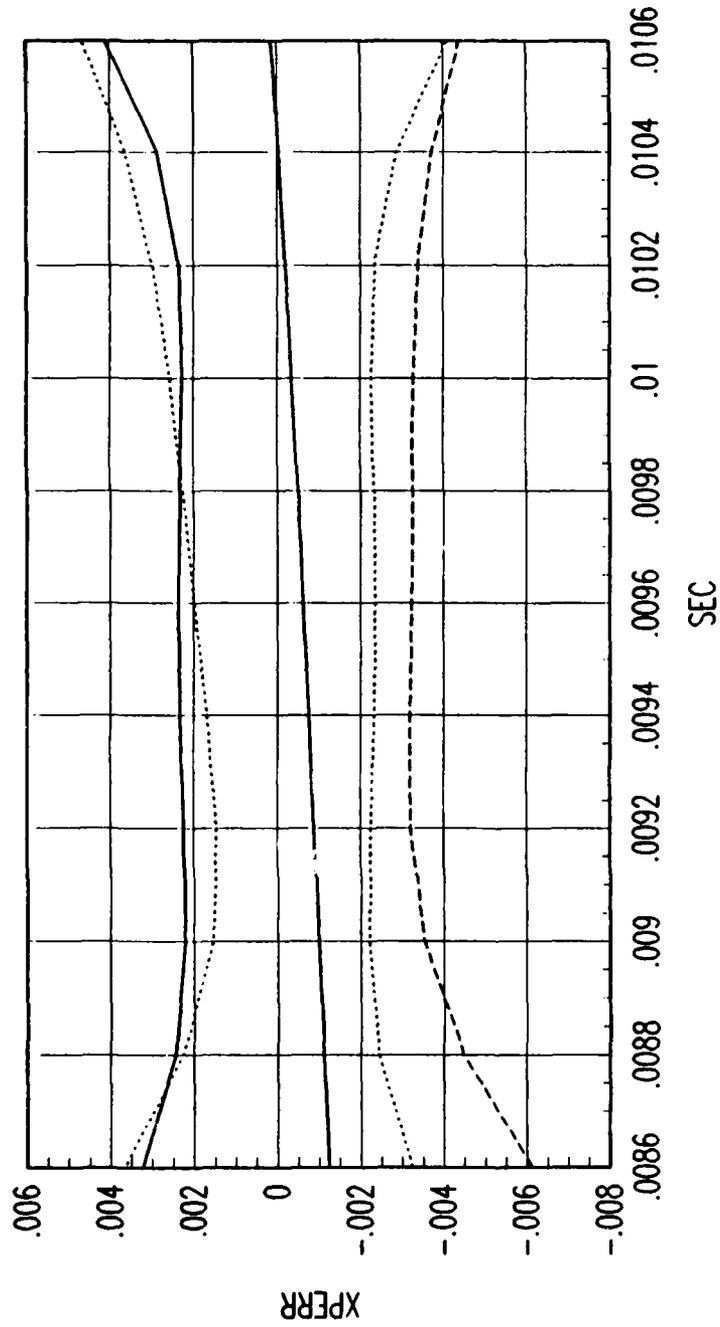


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.110. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

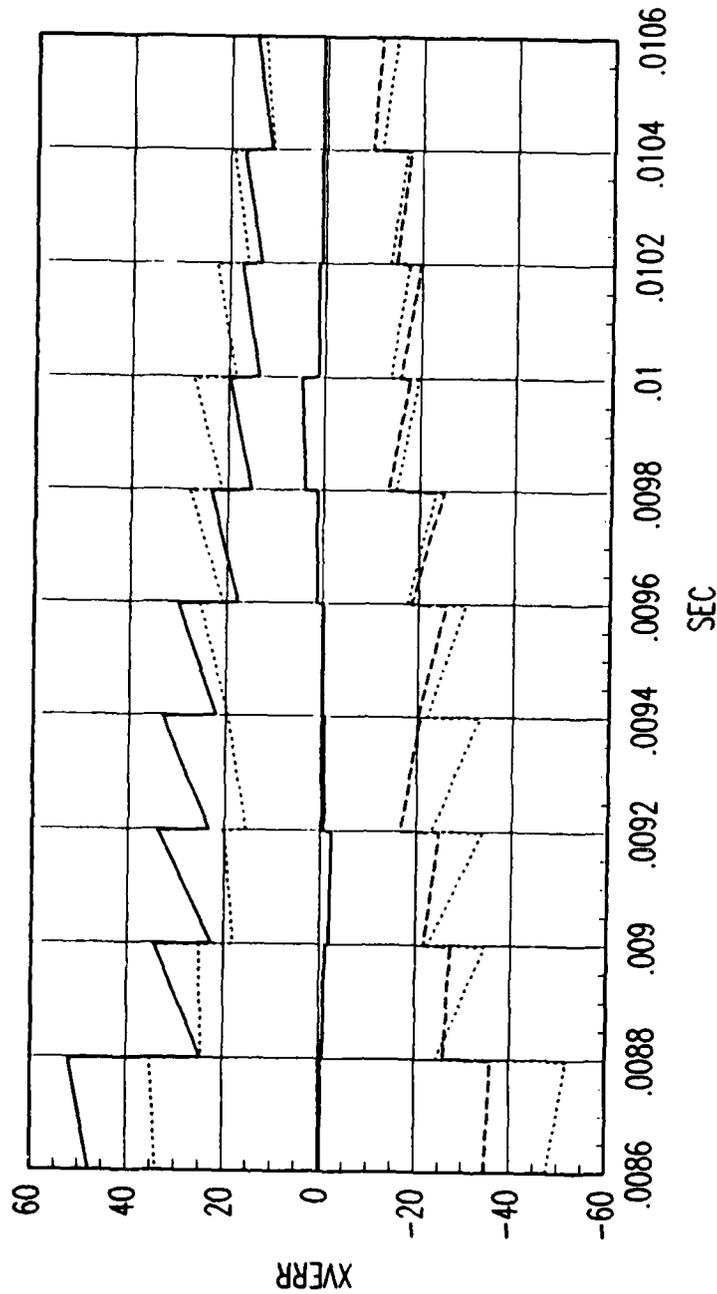


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.111. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

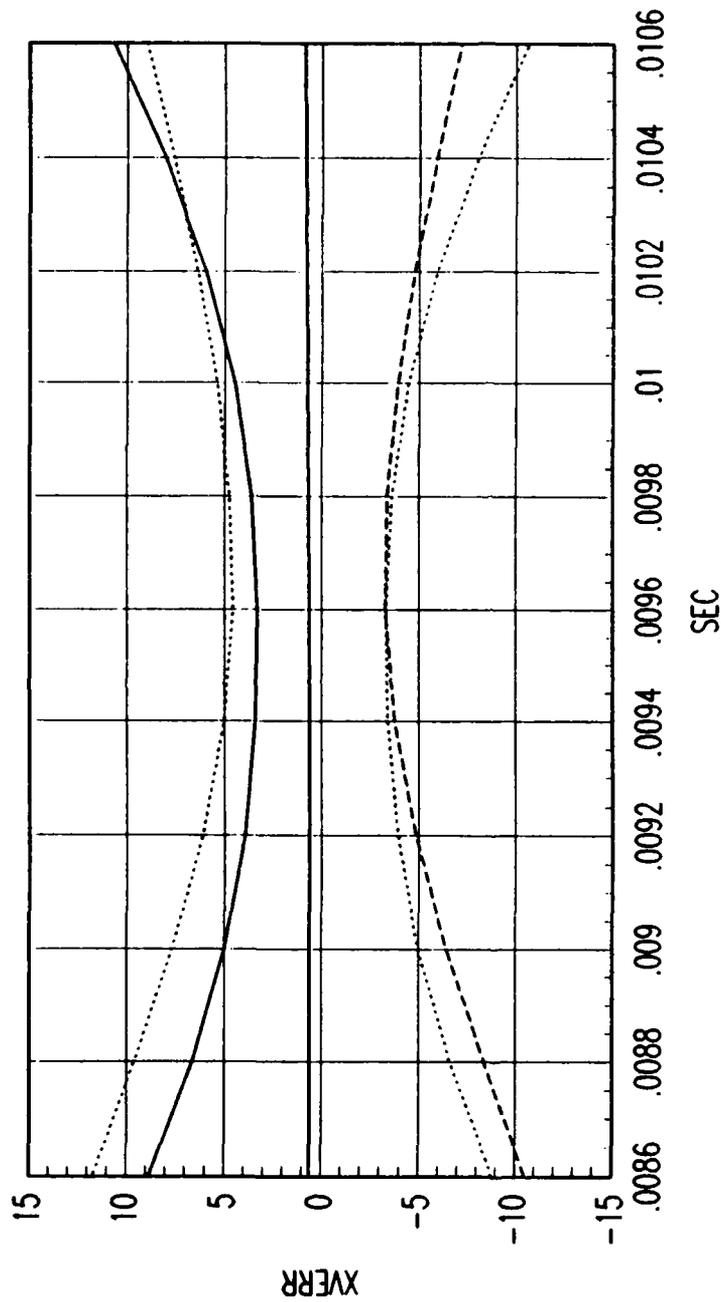
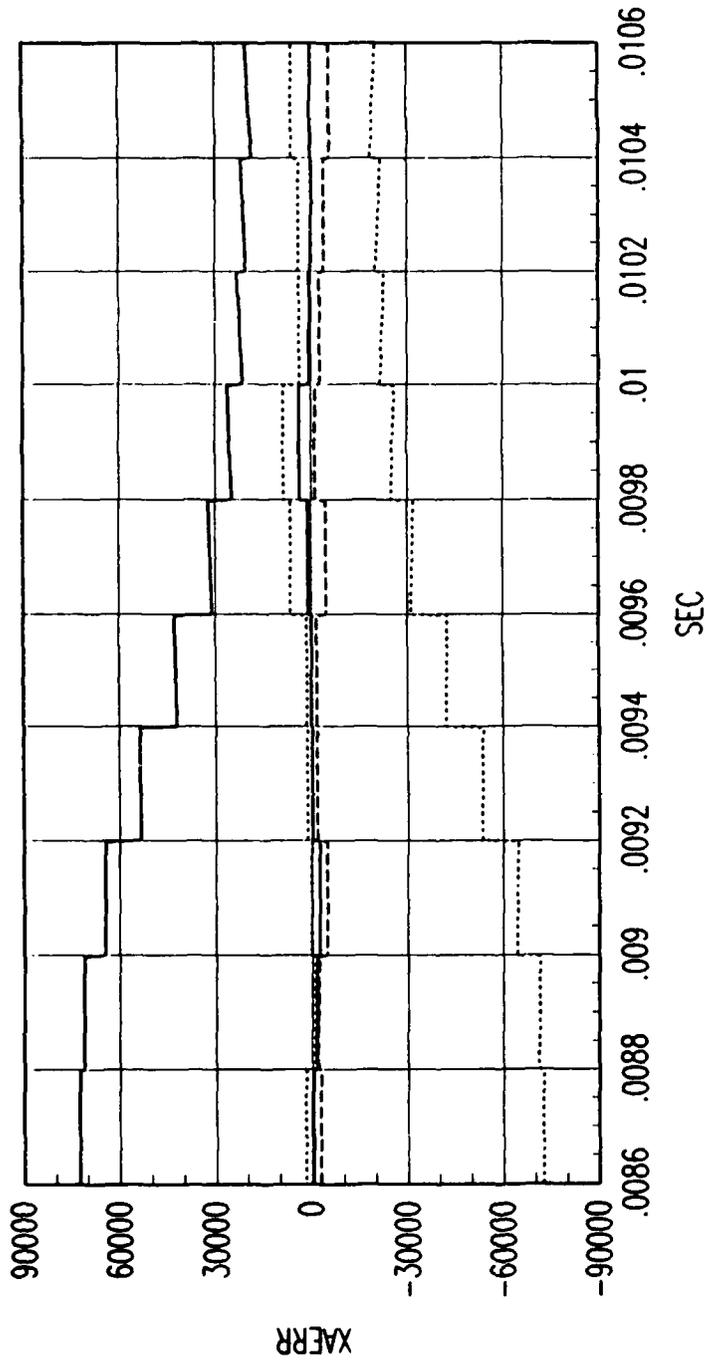


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.112. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.113. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

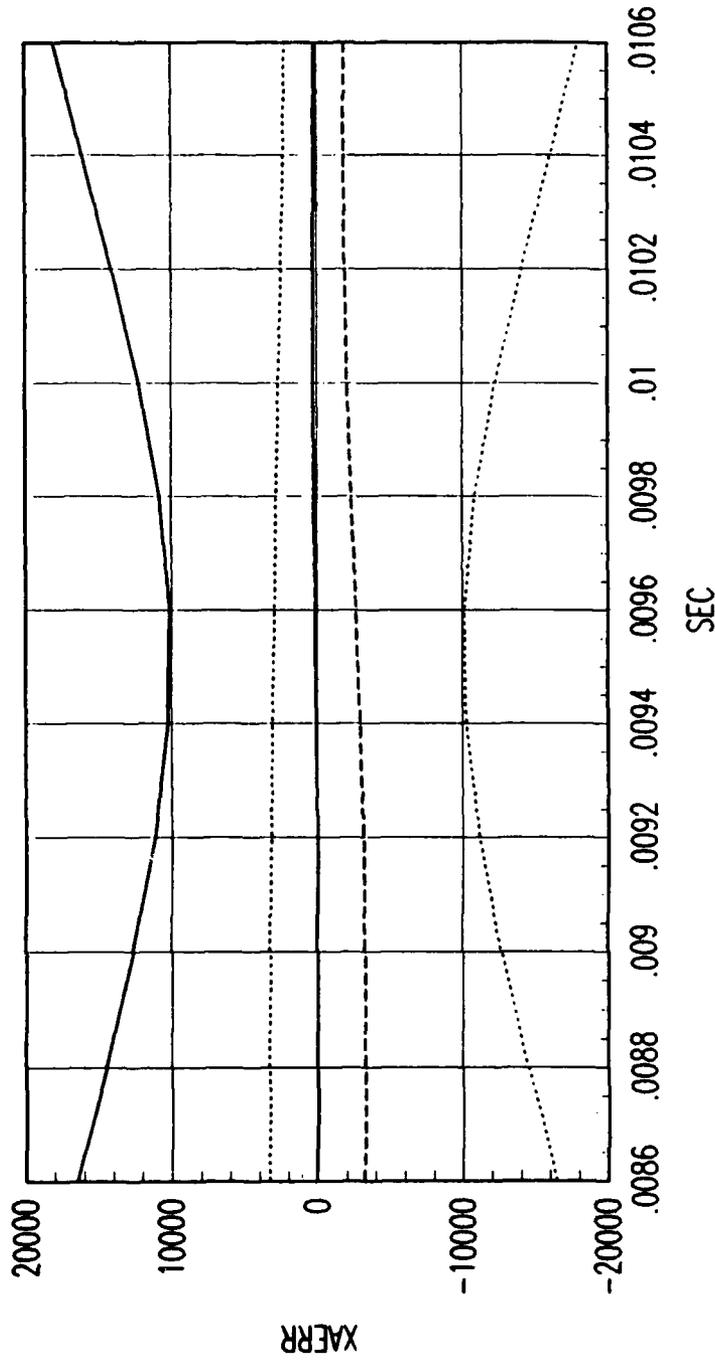
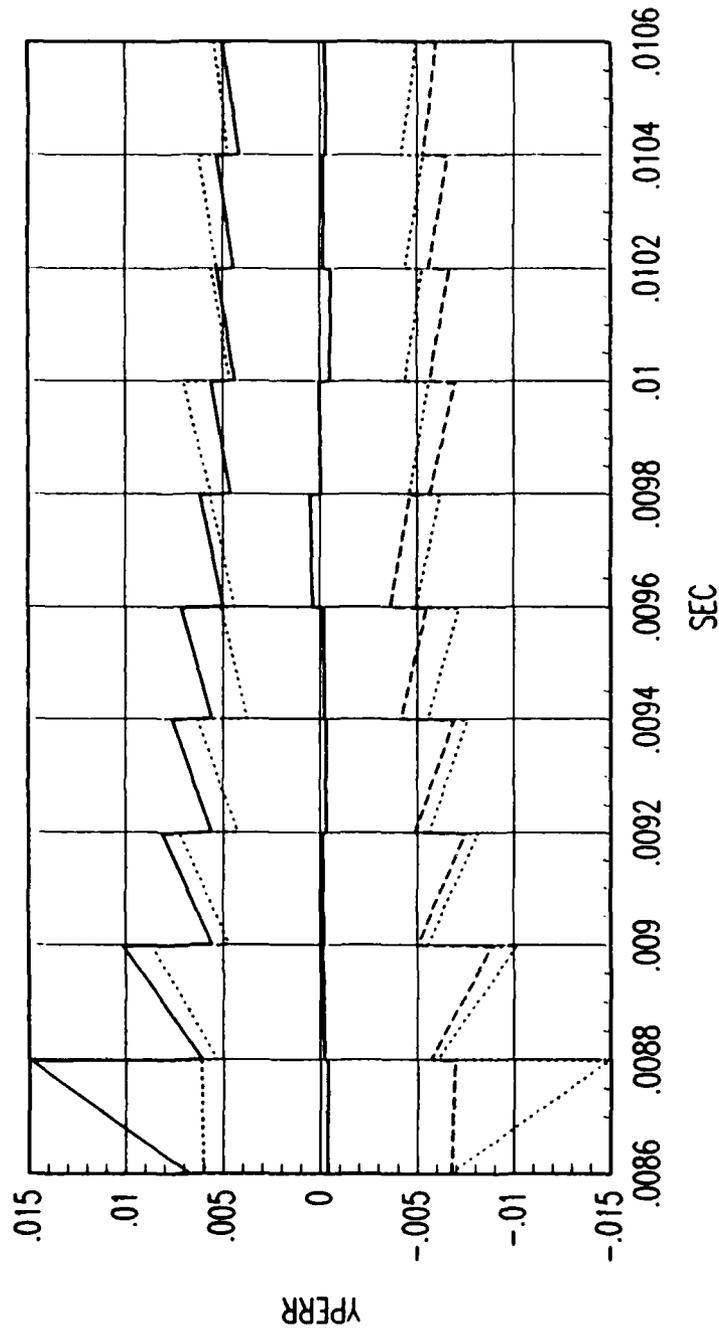


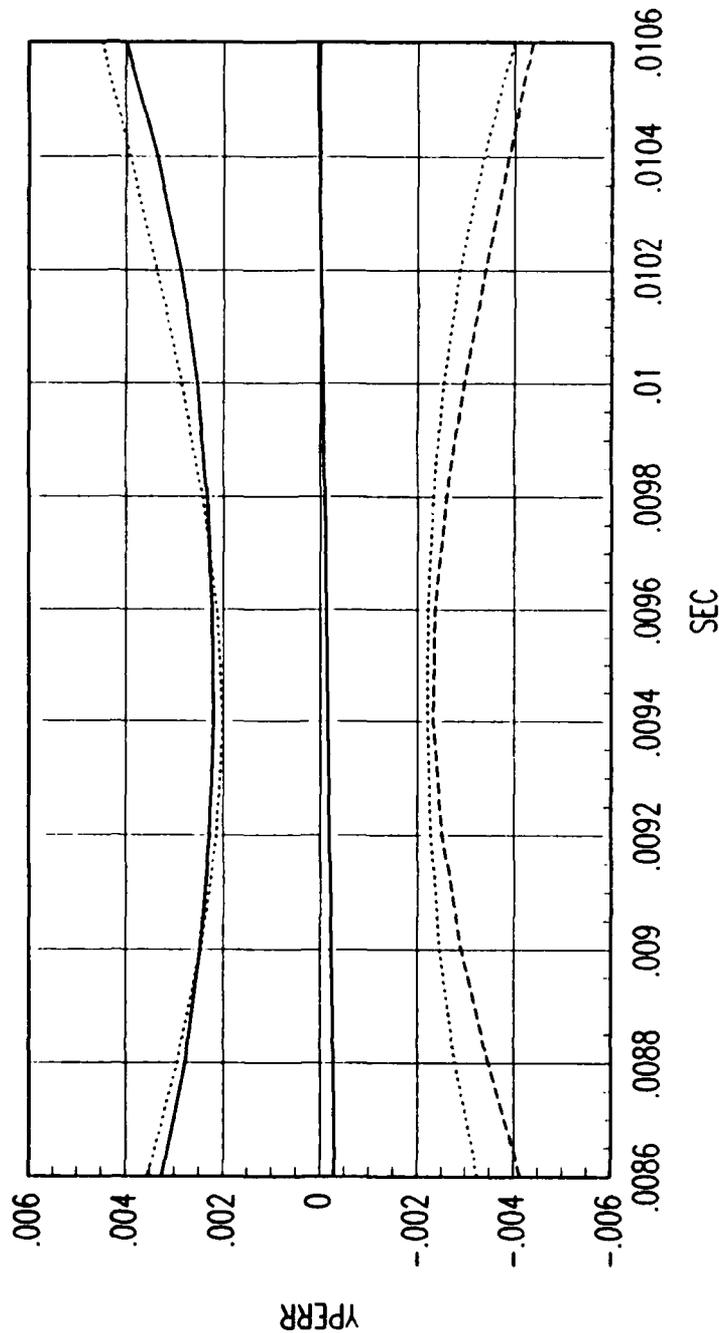
FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.114. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



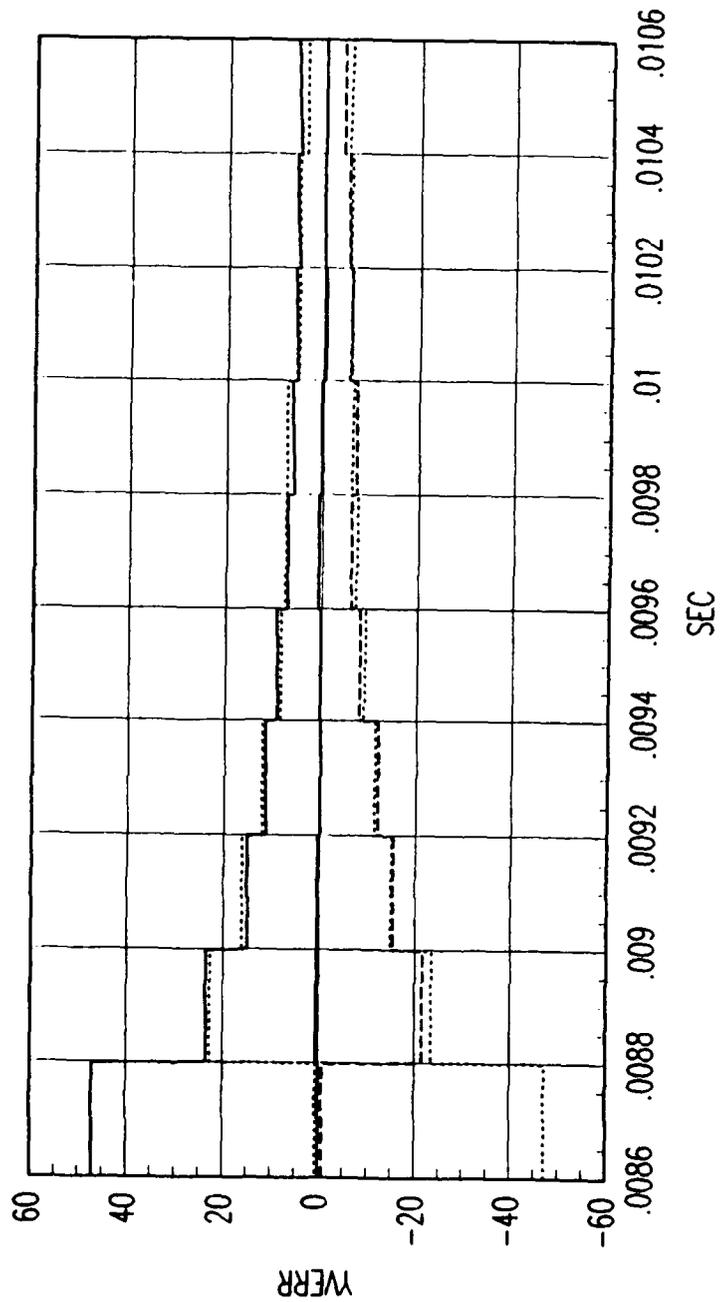
FRG. A FORWARD TIME EKF POSITION ERROR

Figure C.115. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



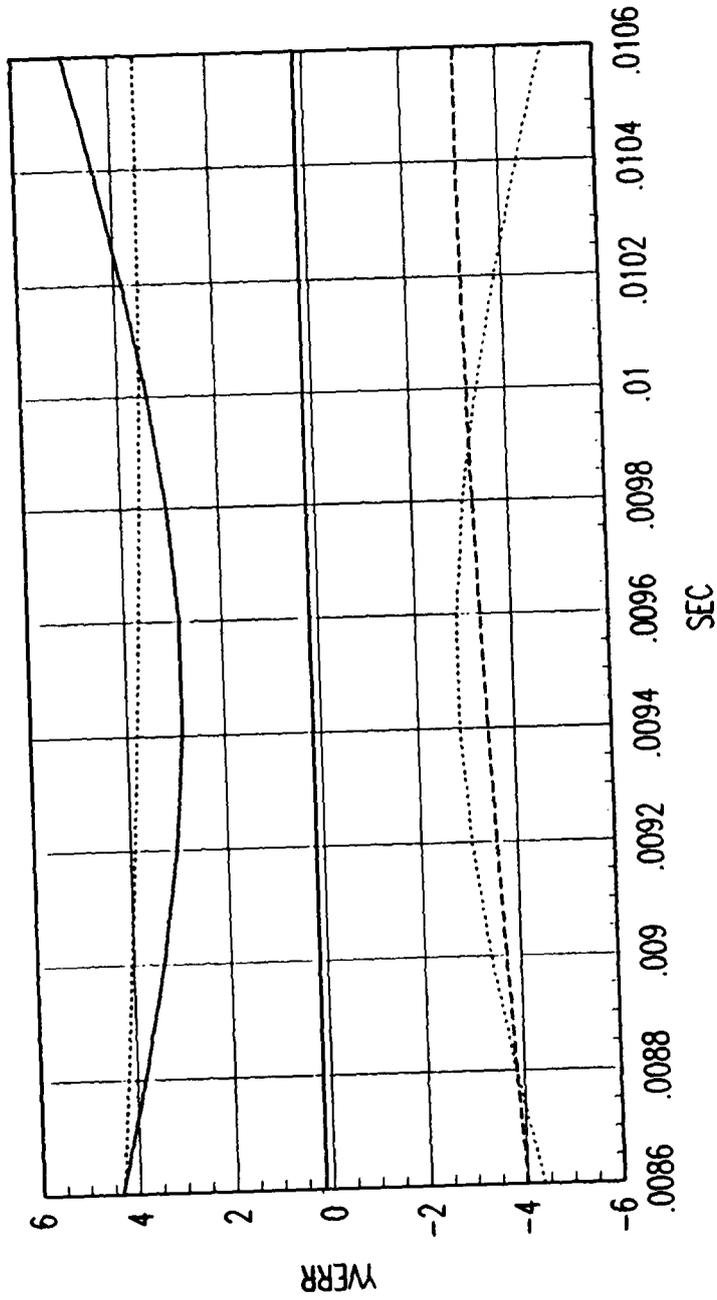
FRG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.116. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.117. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.118. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

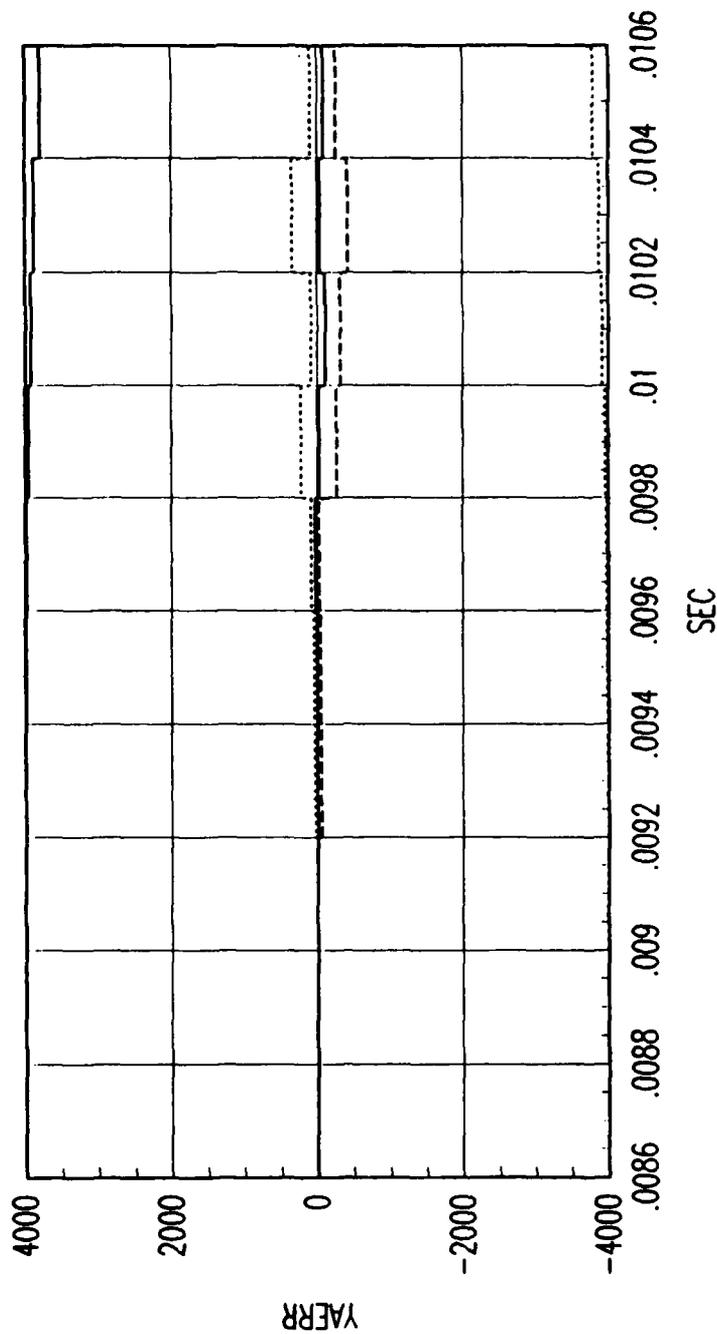
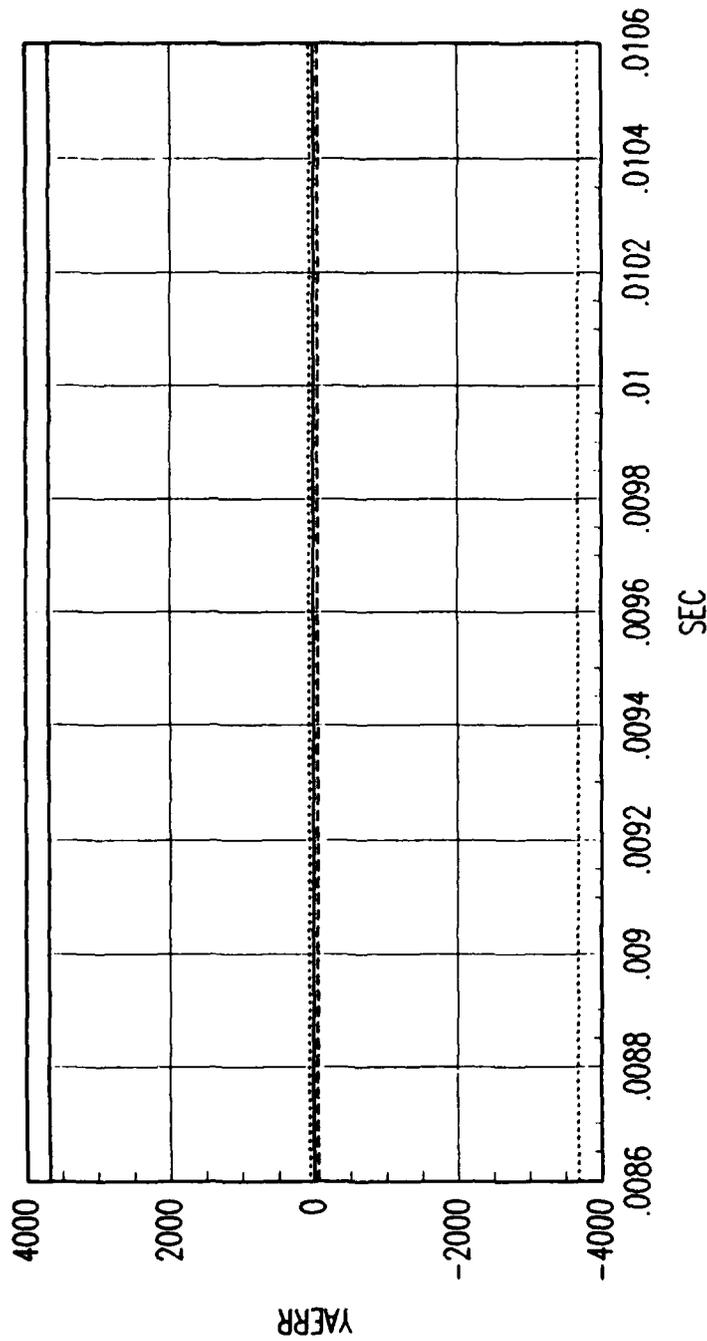


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.119. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.120. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

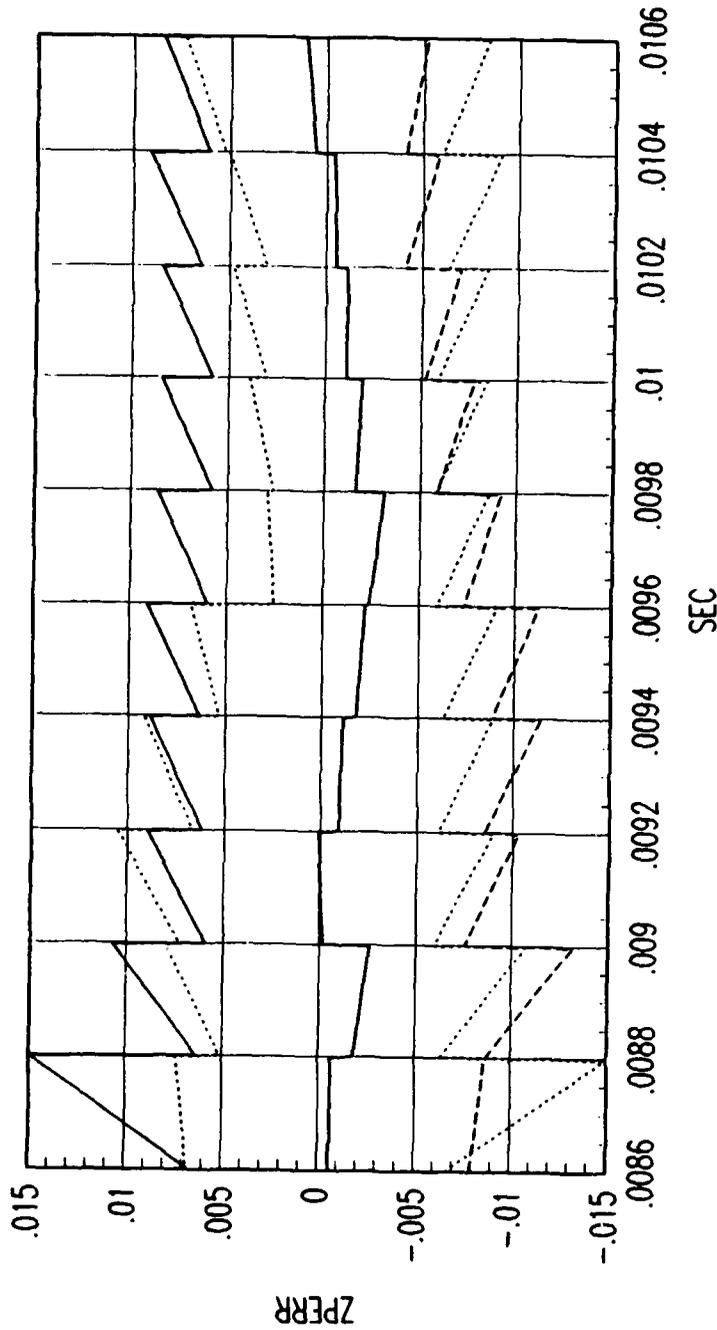
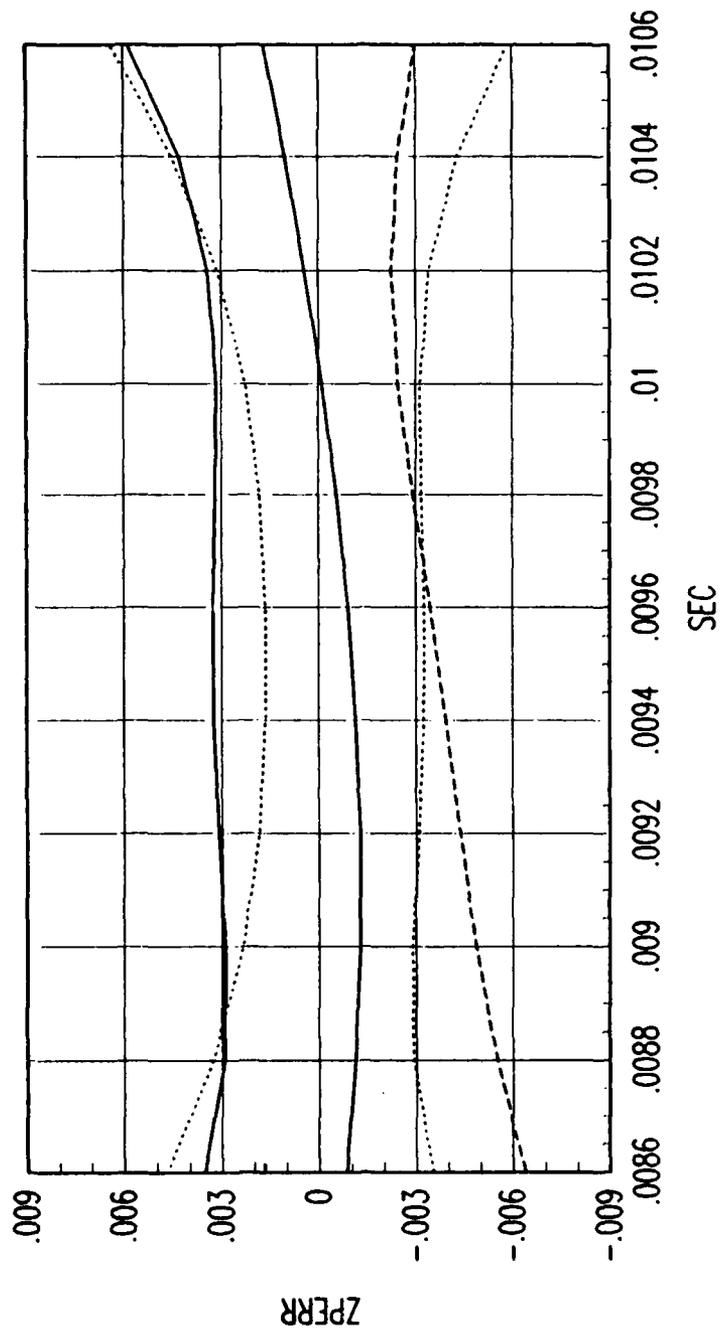


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.121. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.122. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

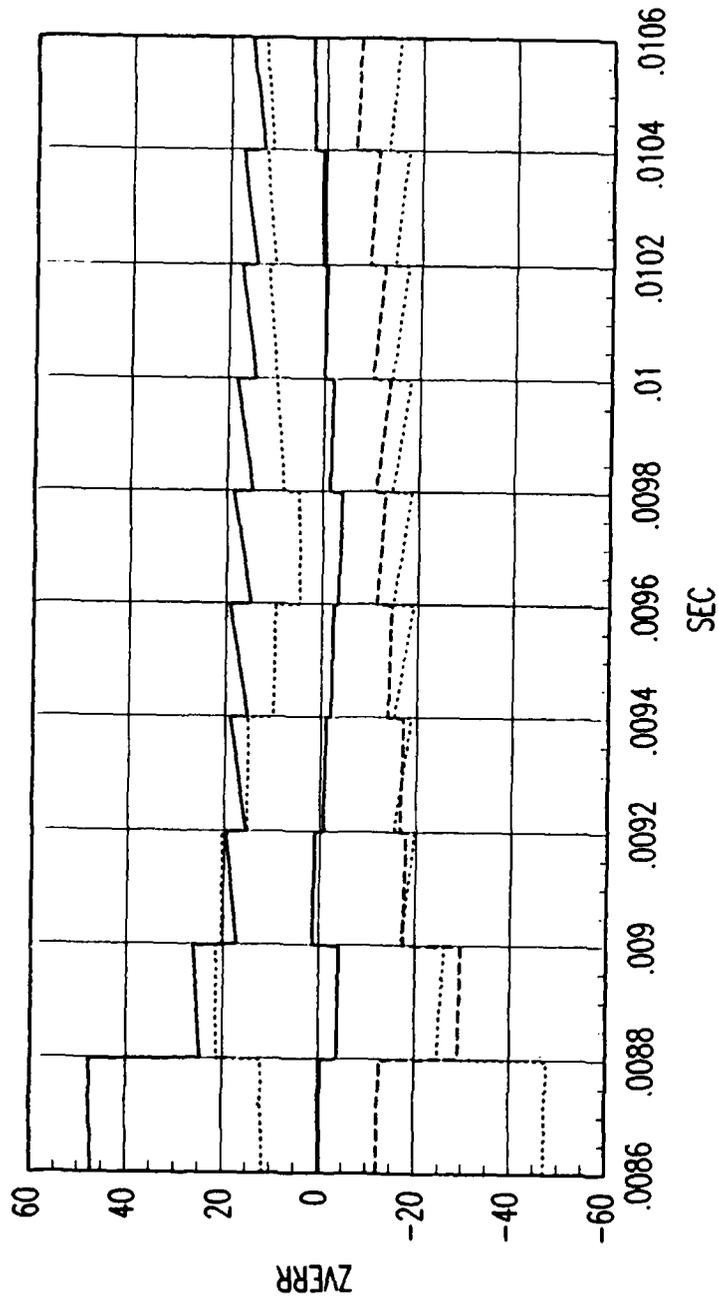
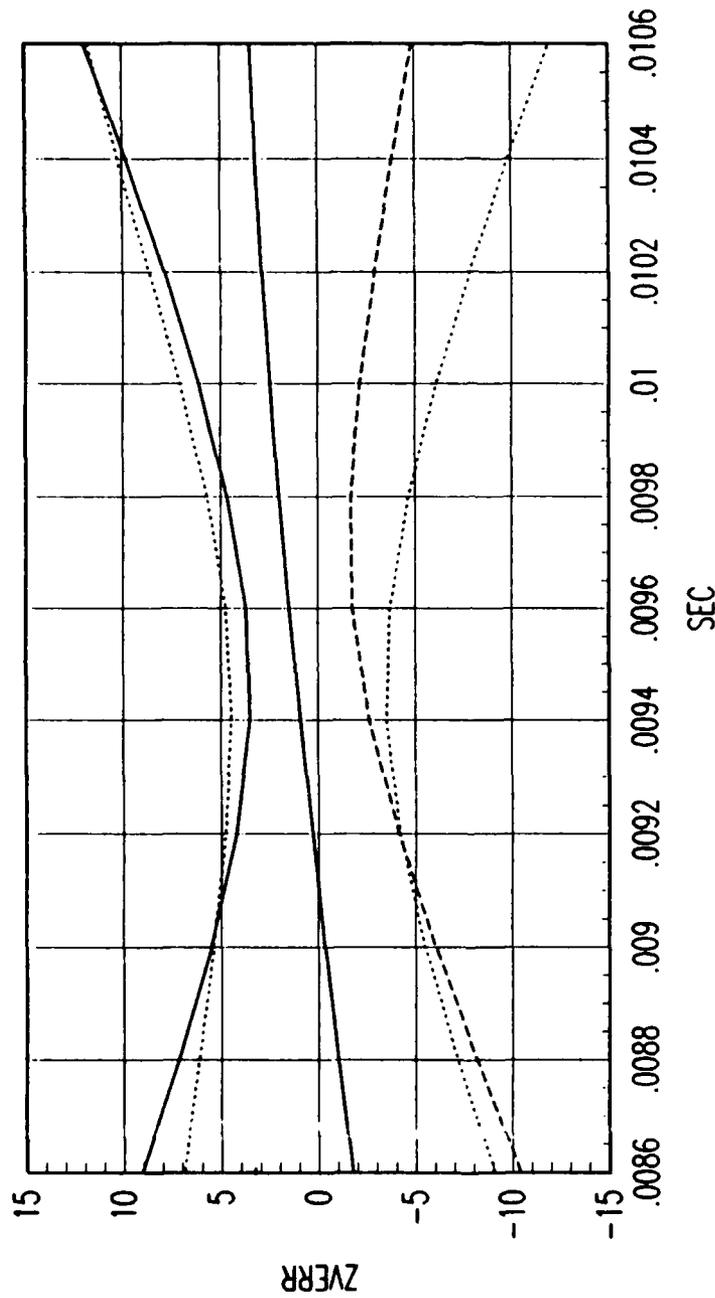


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.123. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.124. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

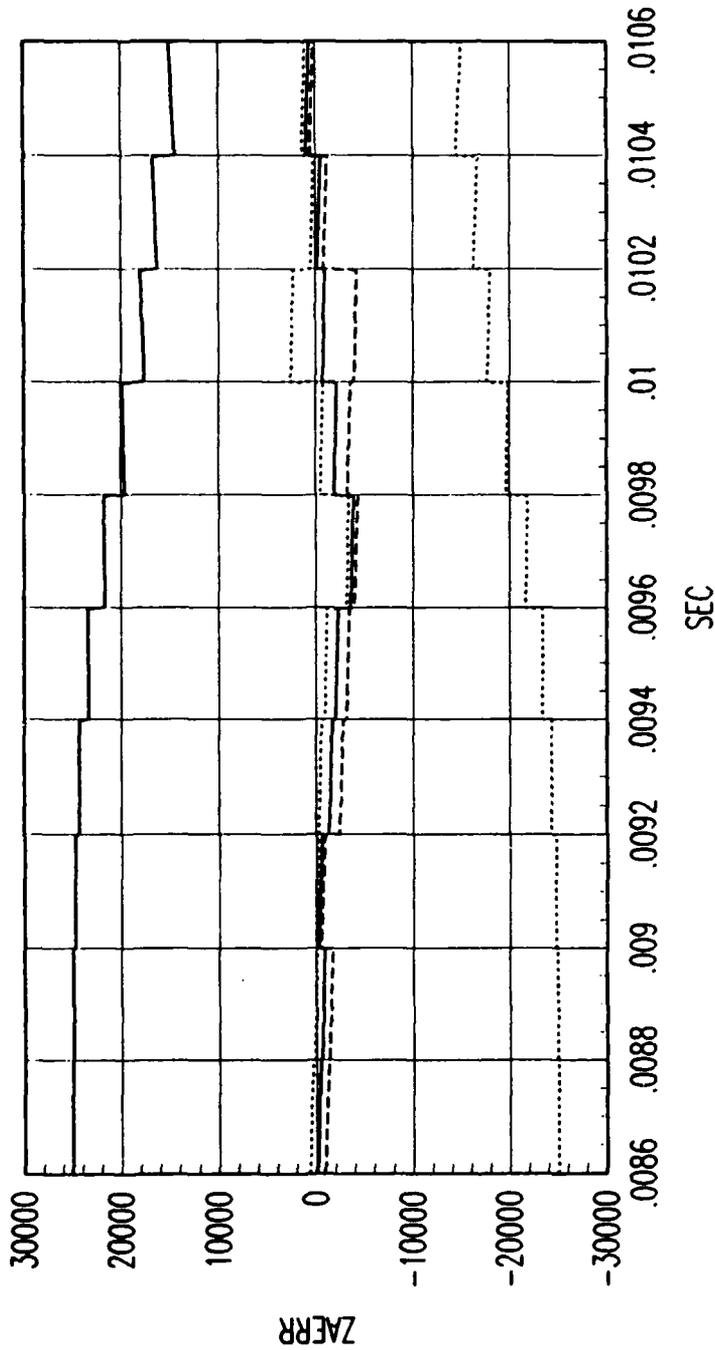


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.125. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

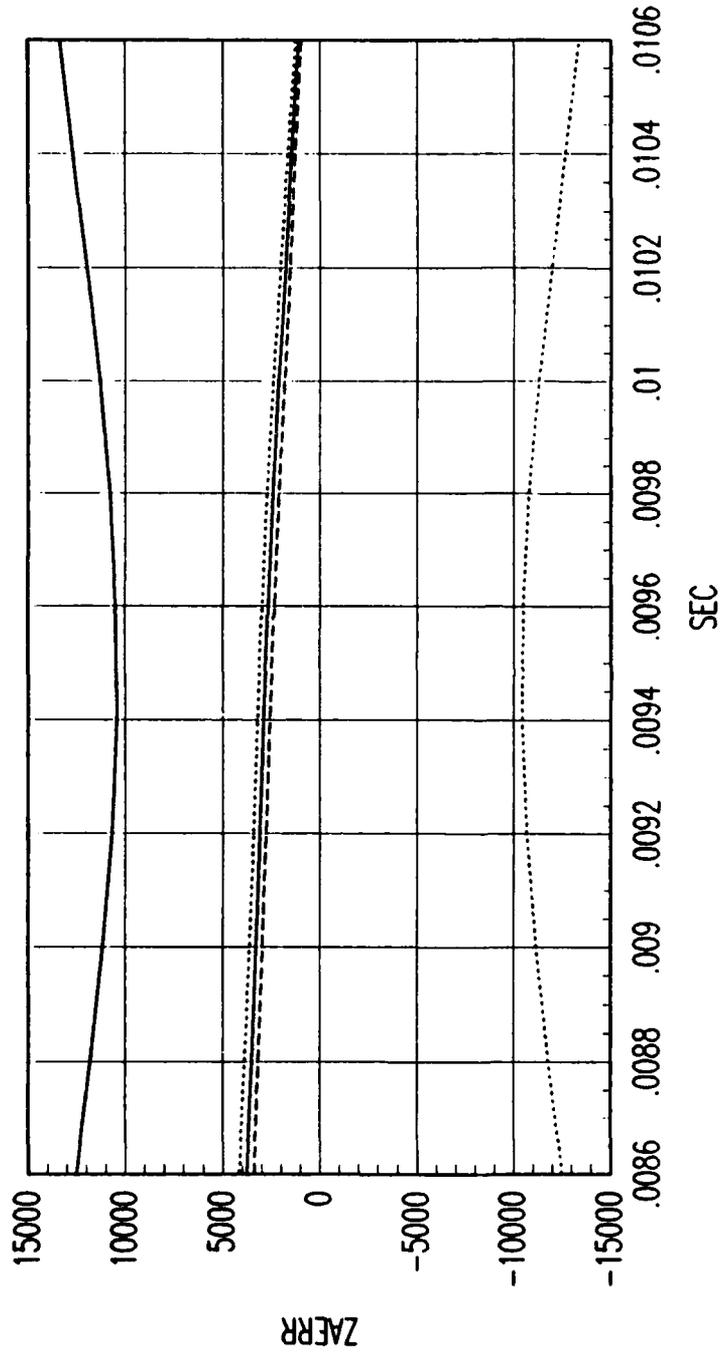


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.126. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

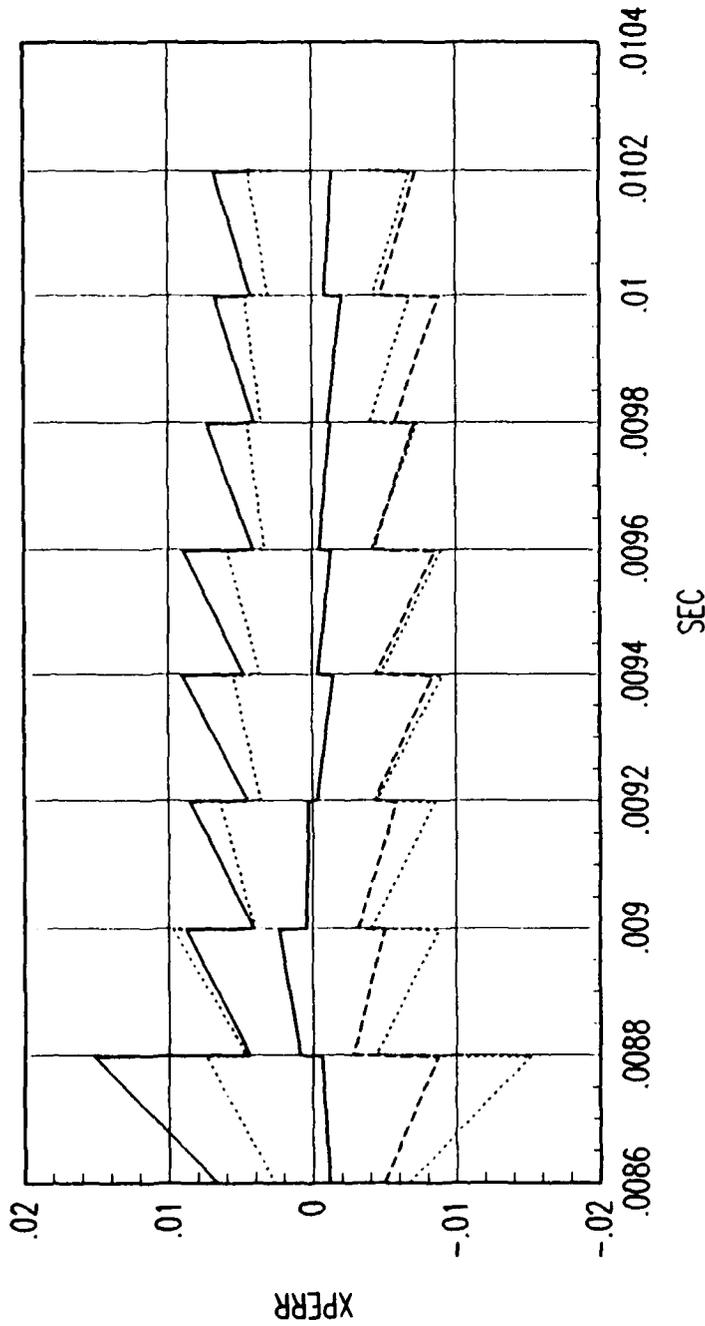
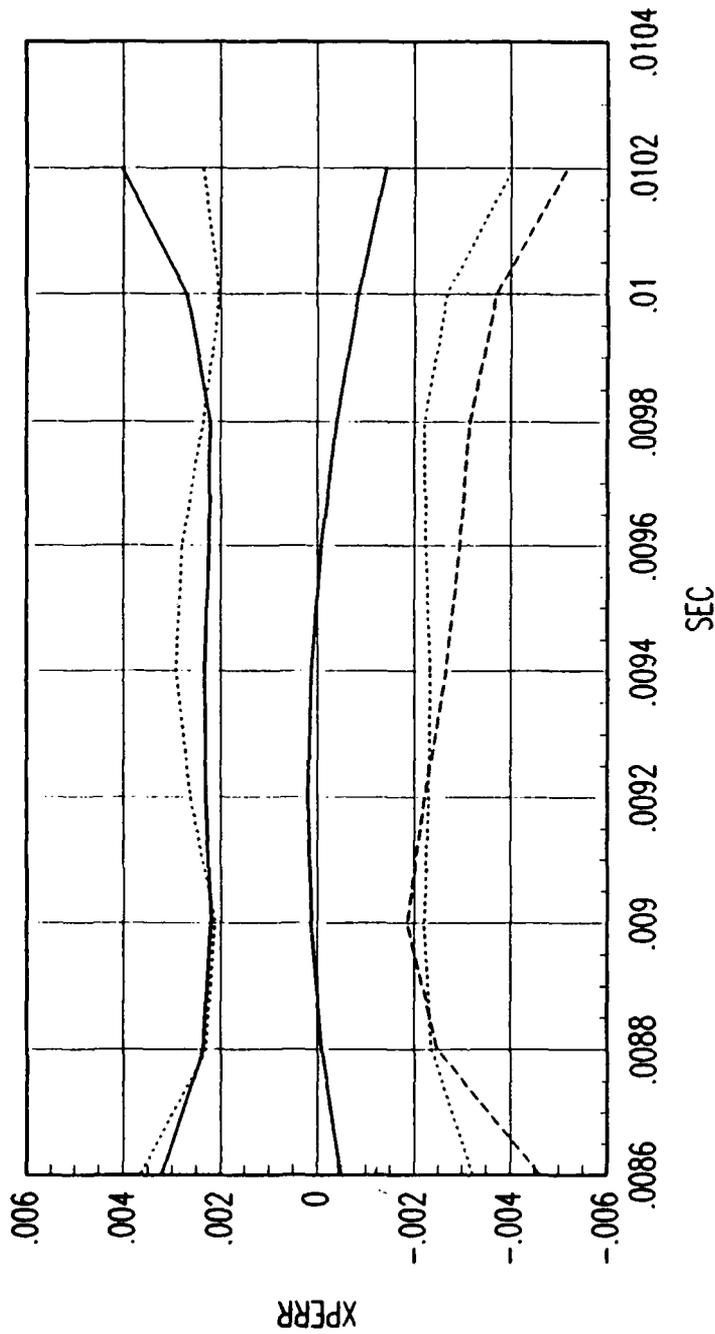


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.127. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.128. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

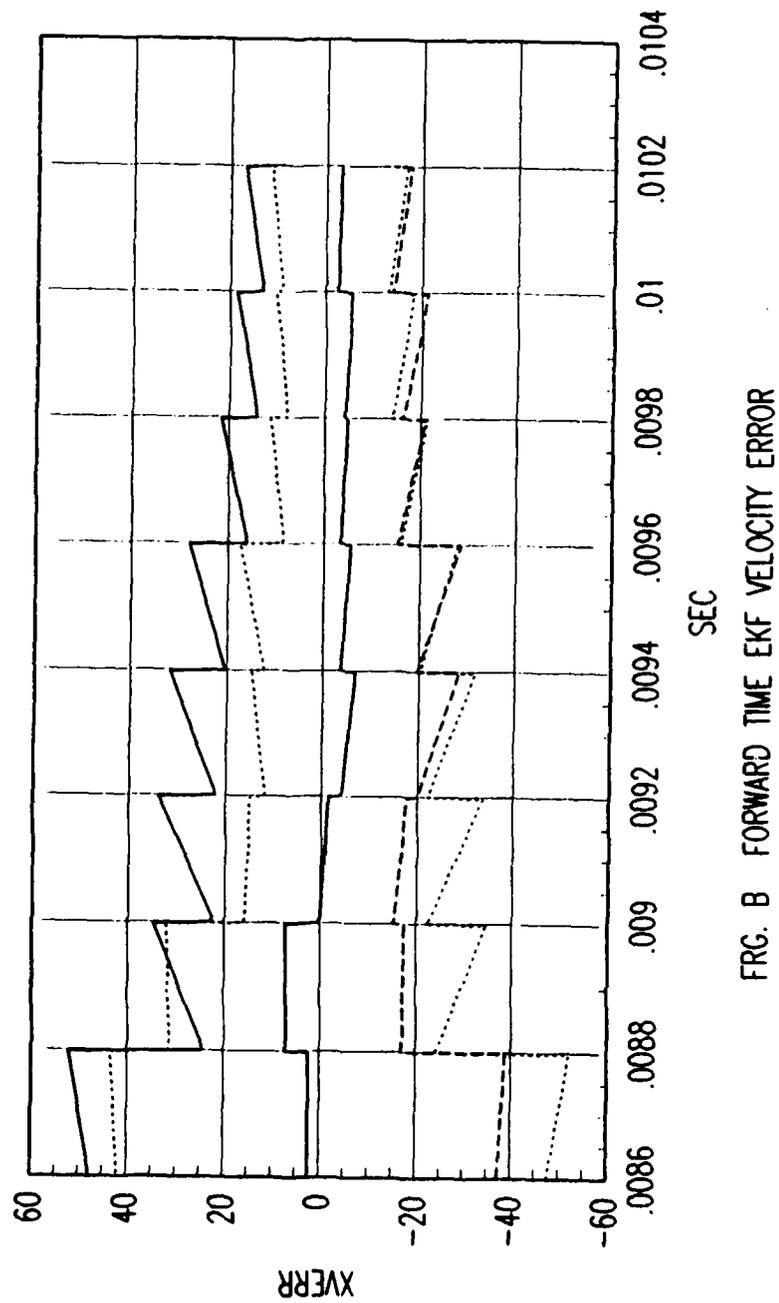
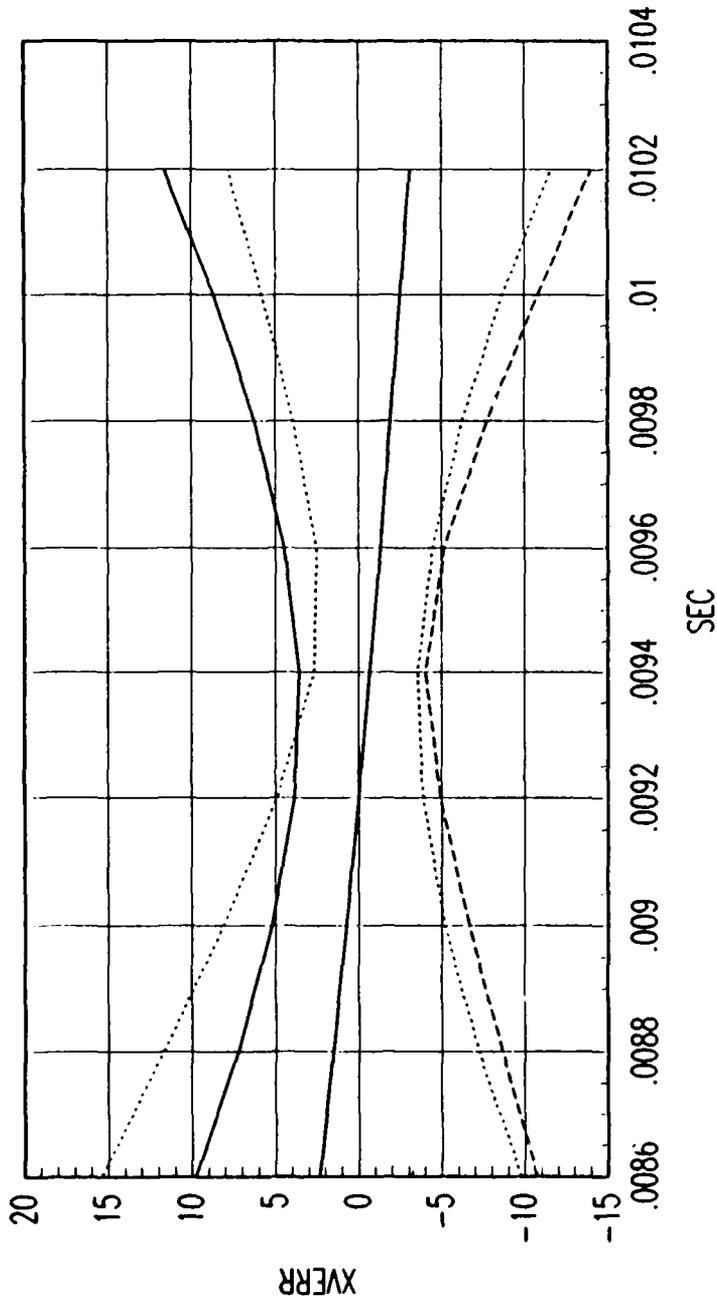


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.129. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.130. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

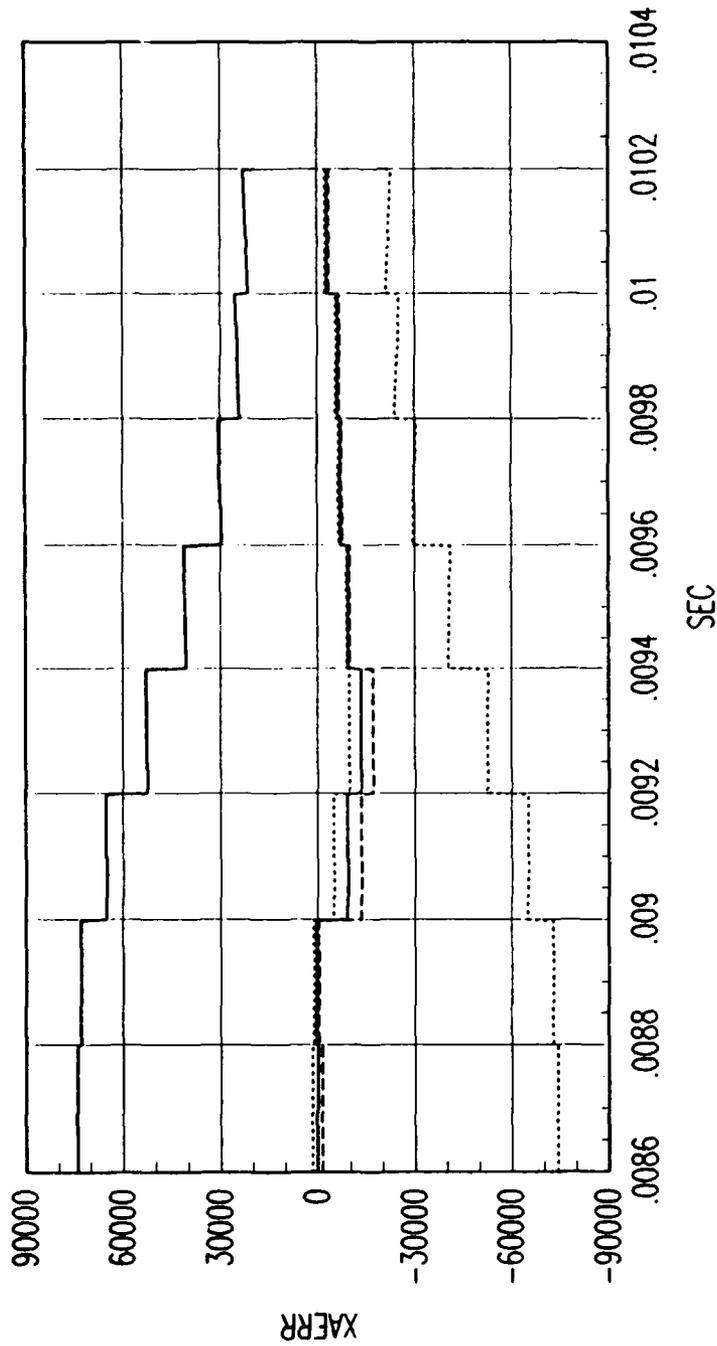
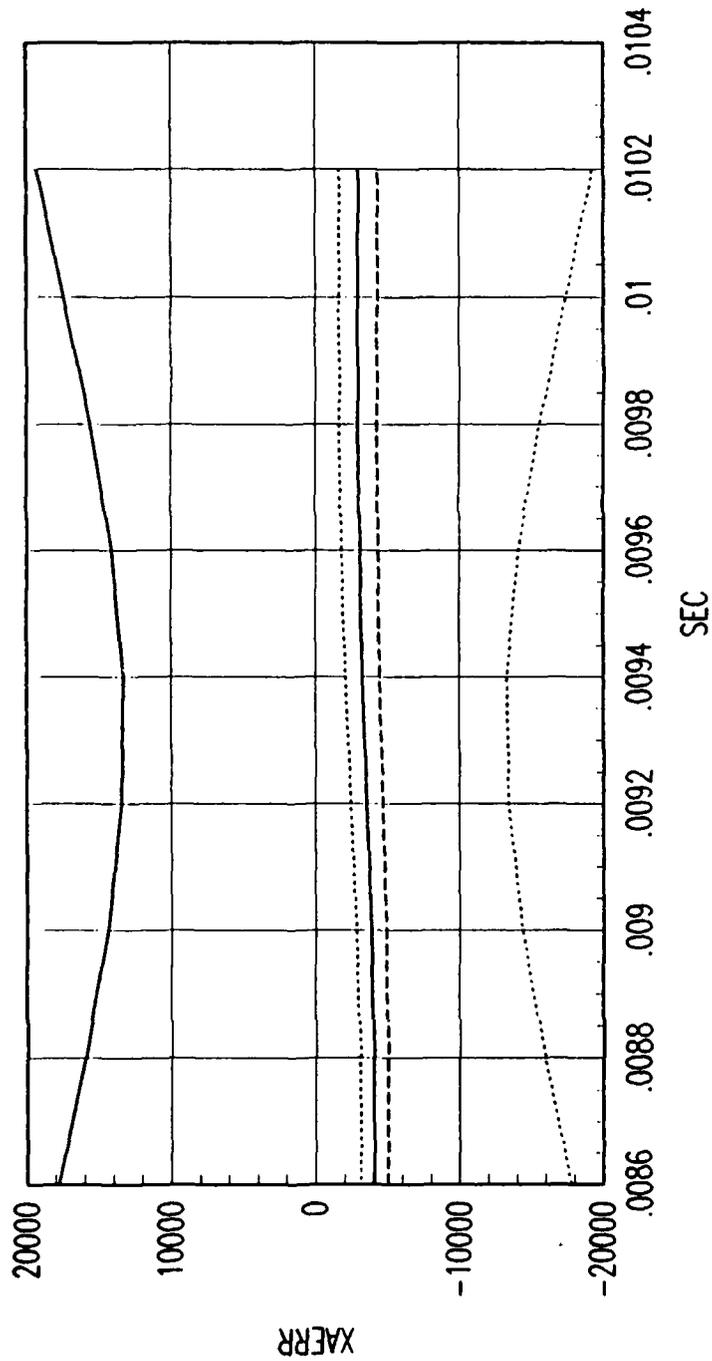


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.131. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.132. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

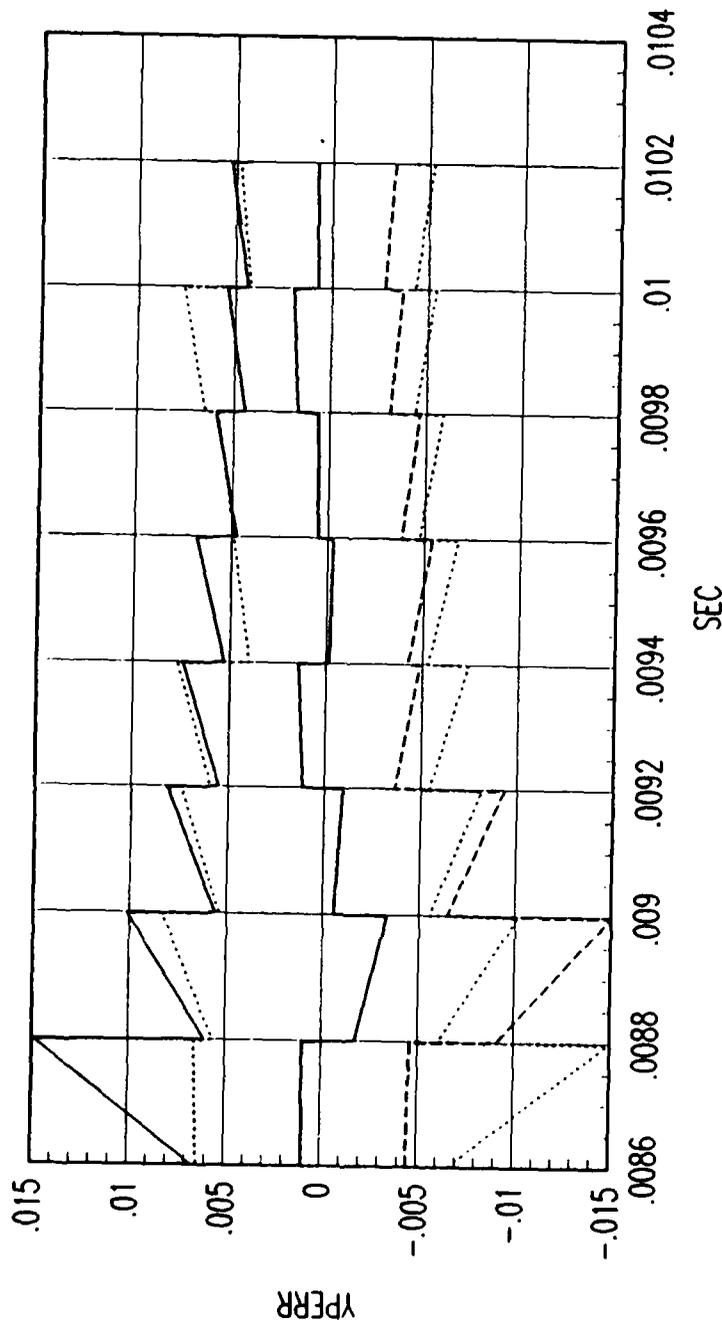
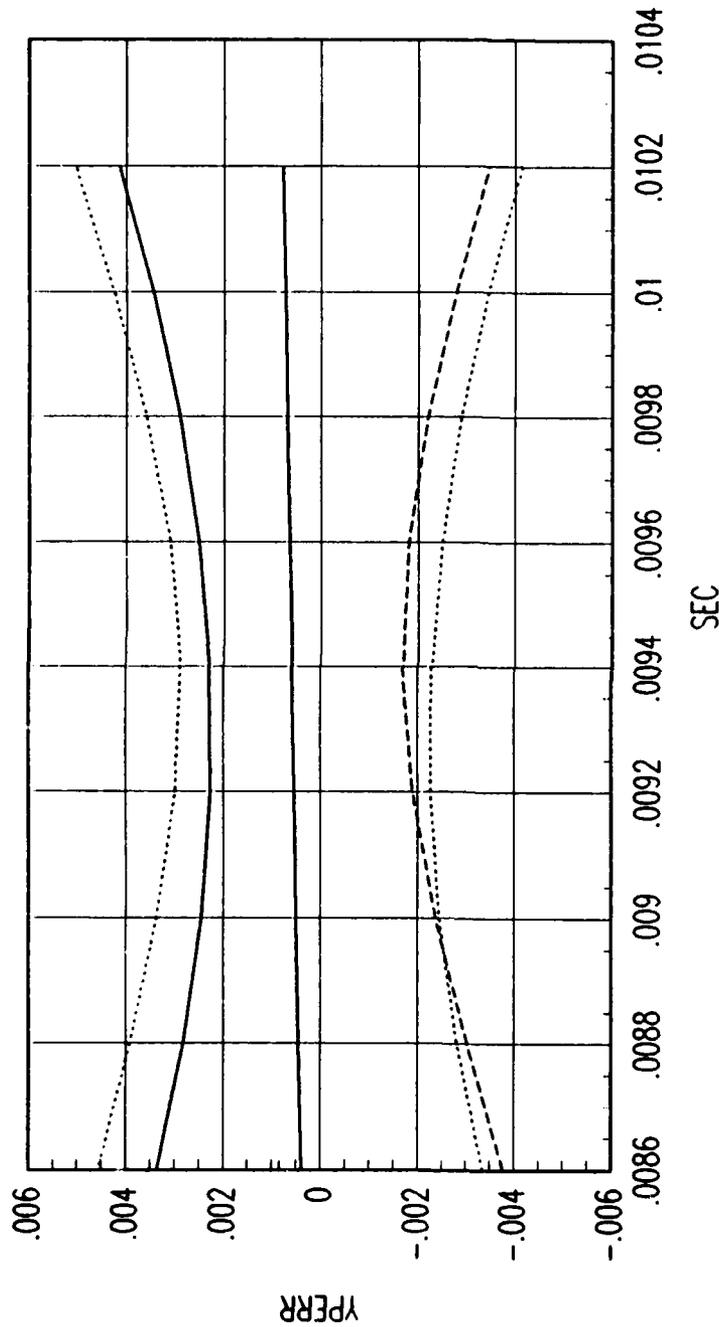


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.133. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.134. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

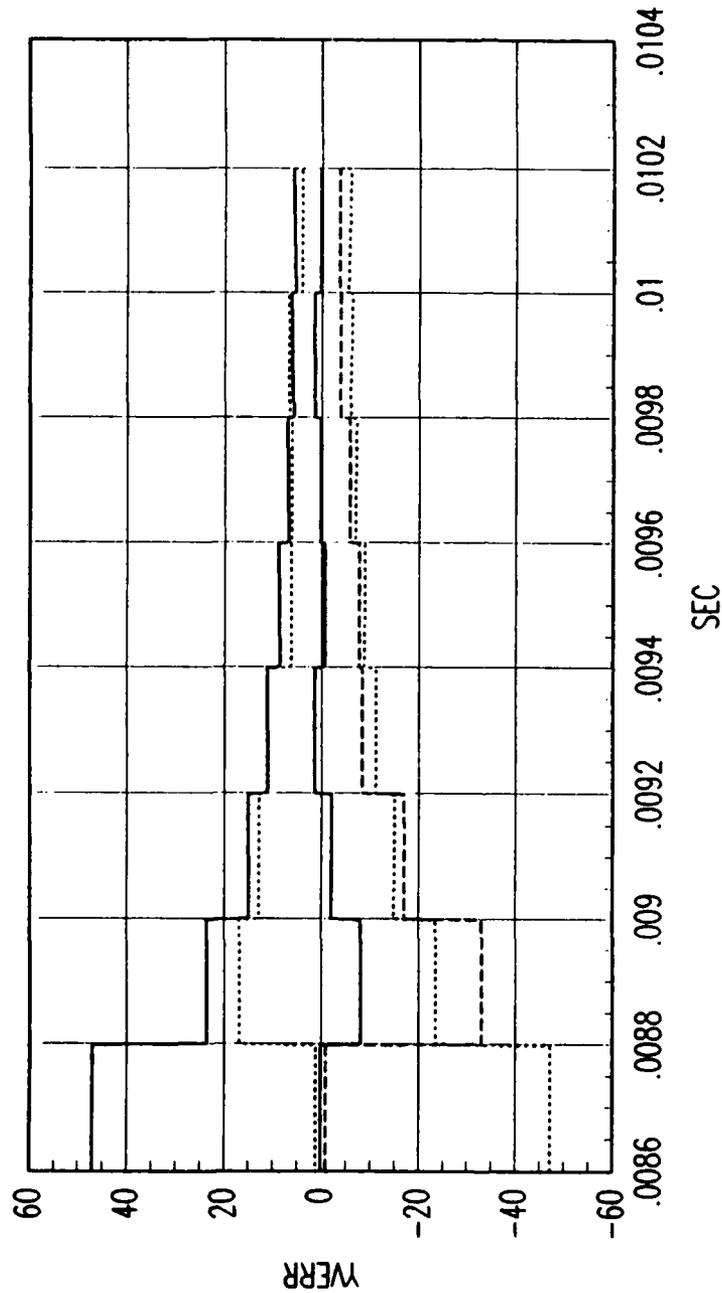
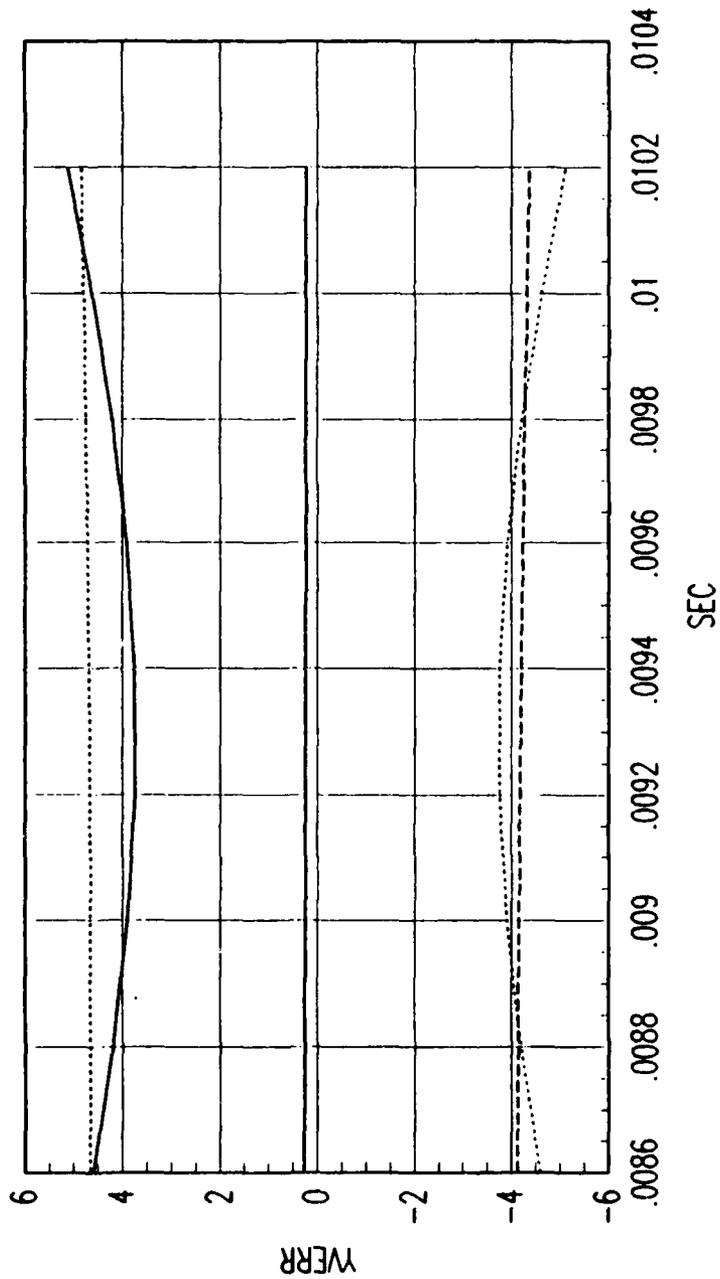


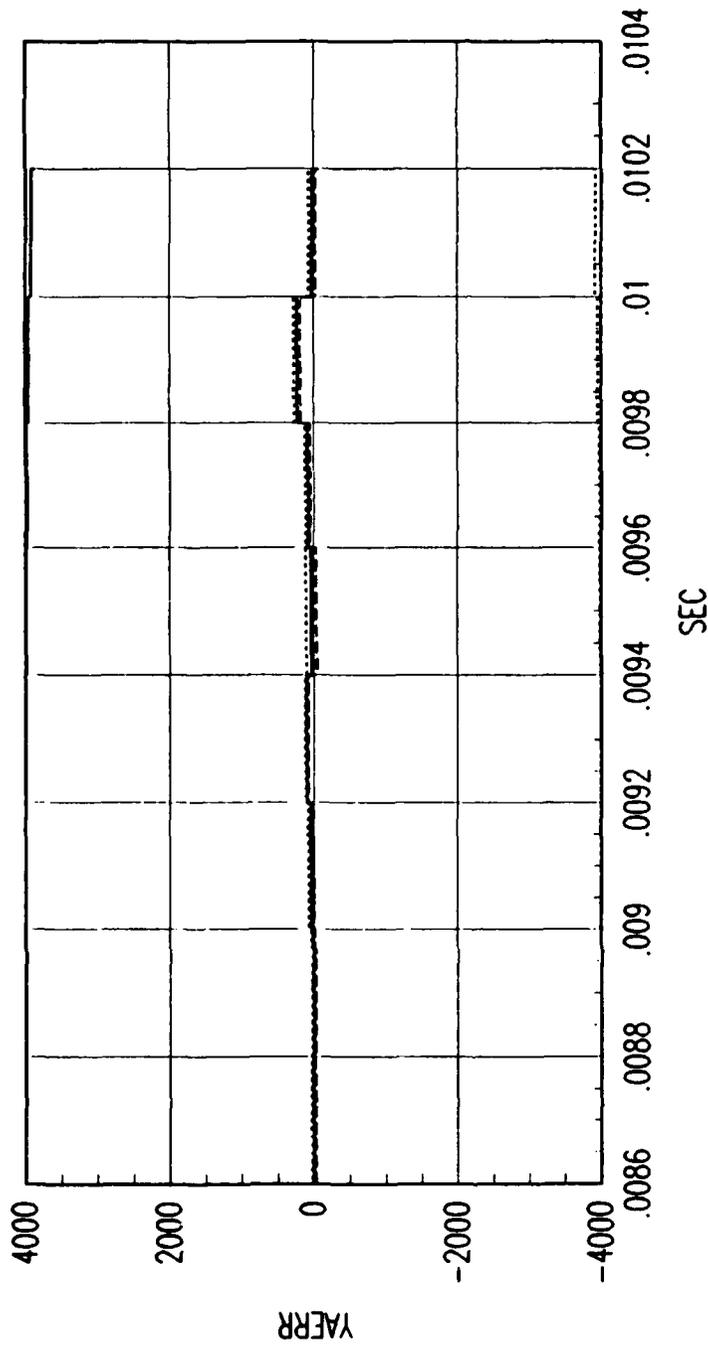
FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.135. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



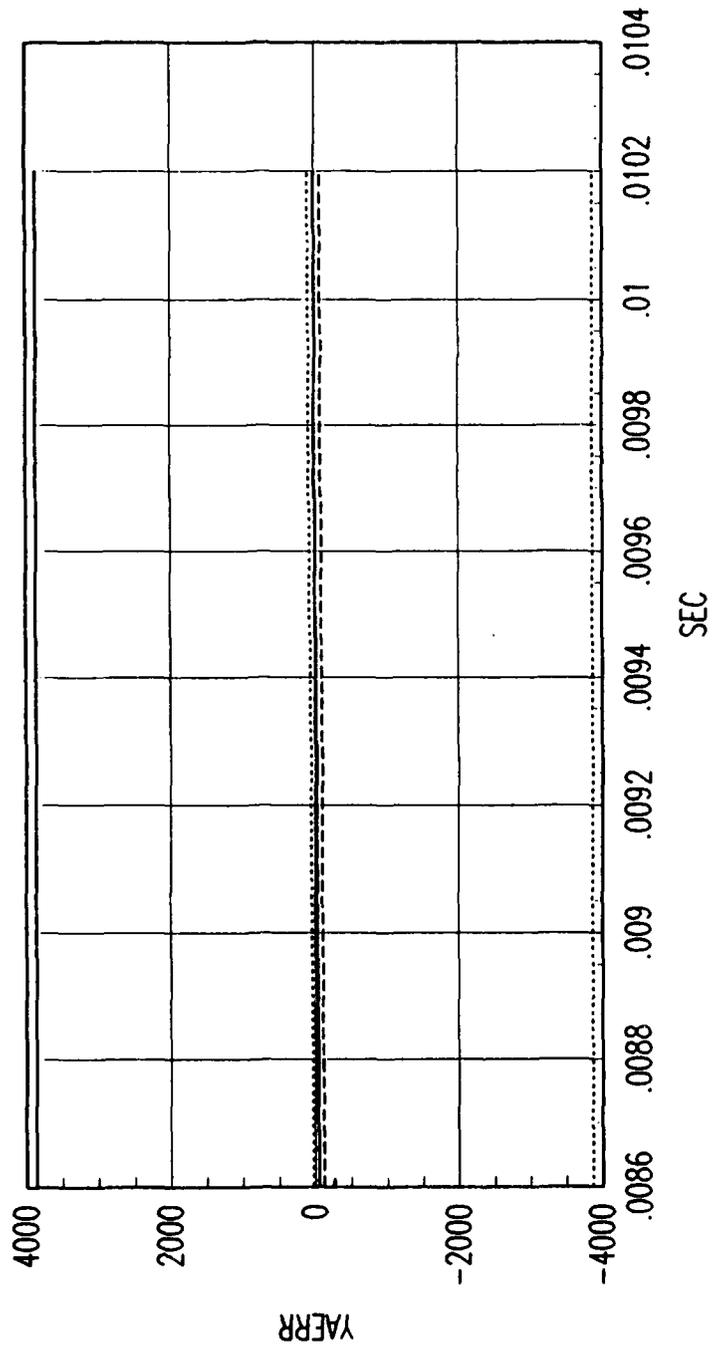
FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.136. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.137. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.138. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

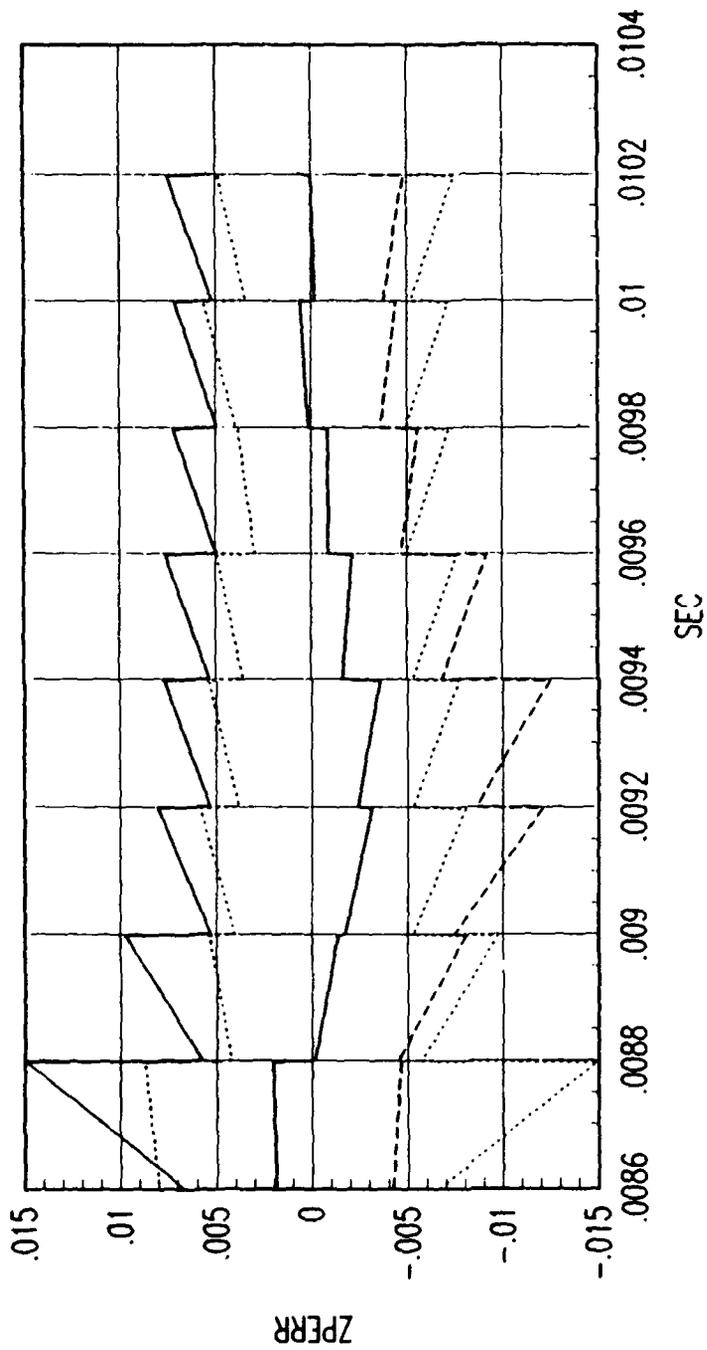
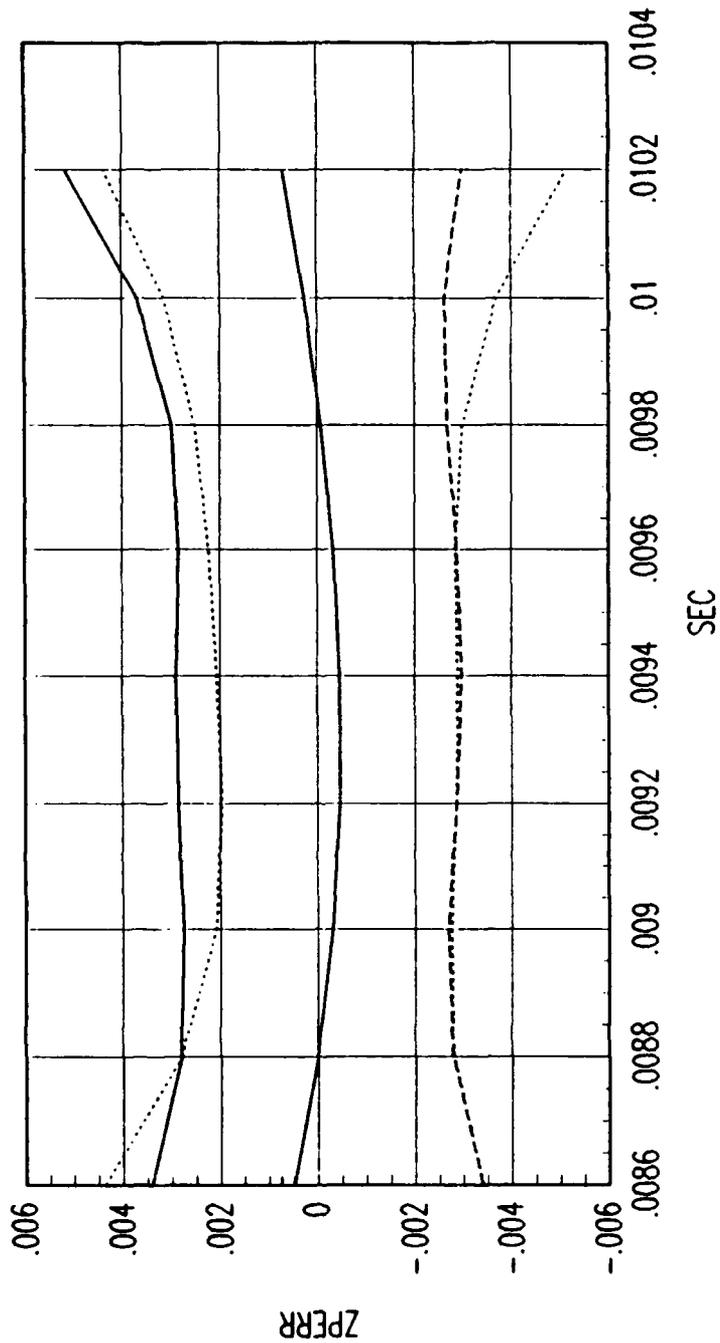


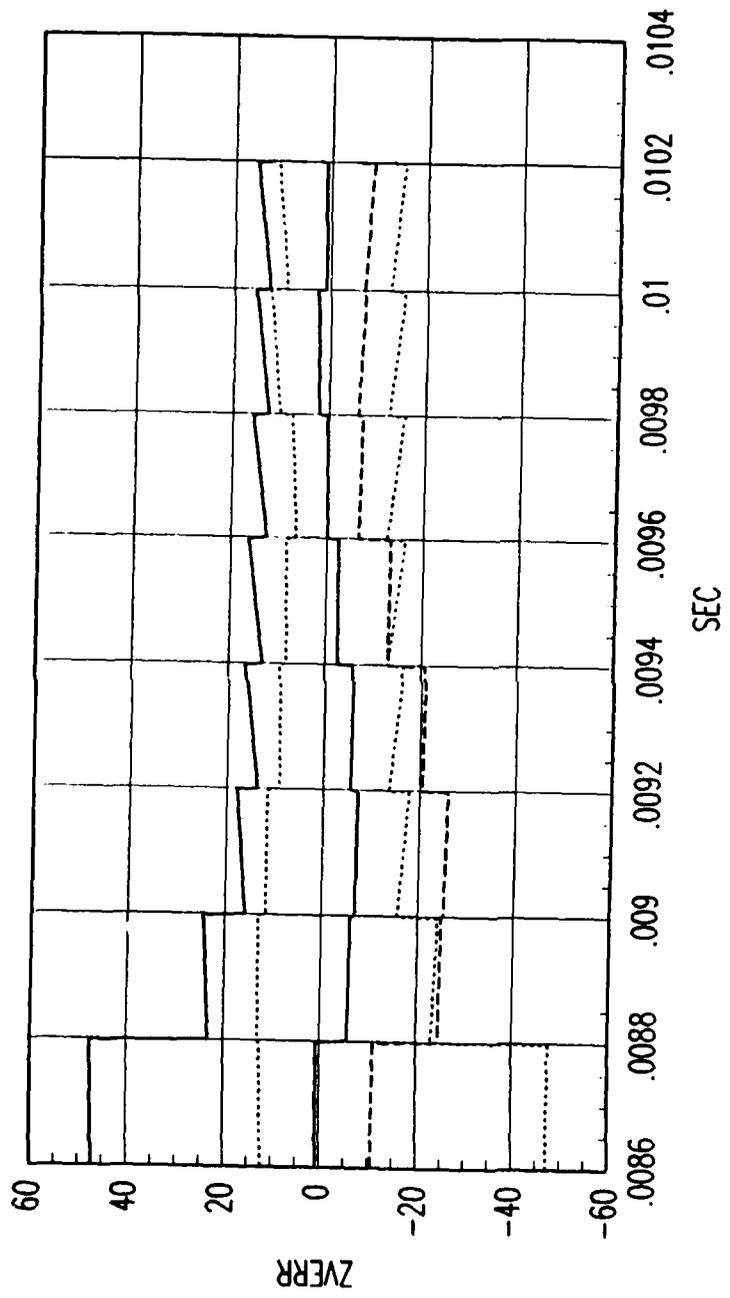
FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.139. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.140. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.141. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

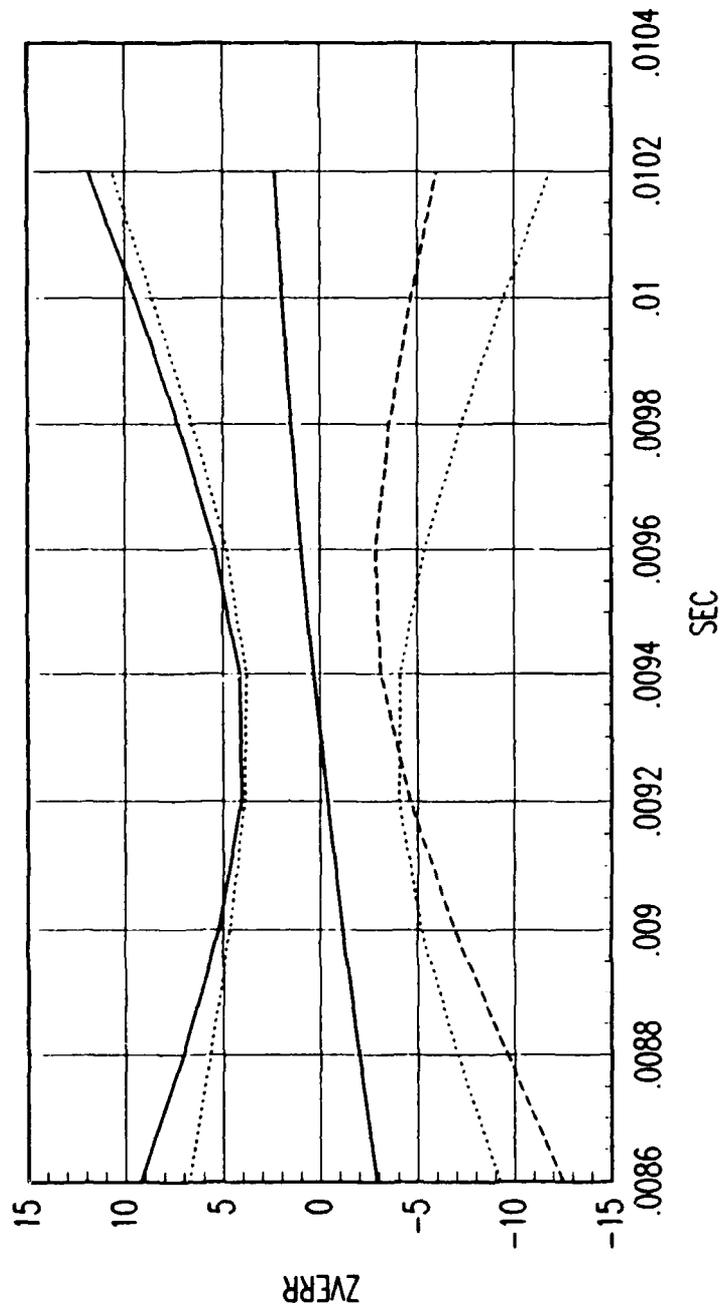
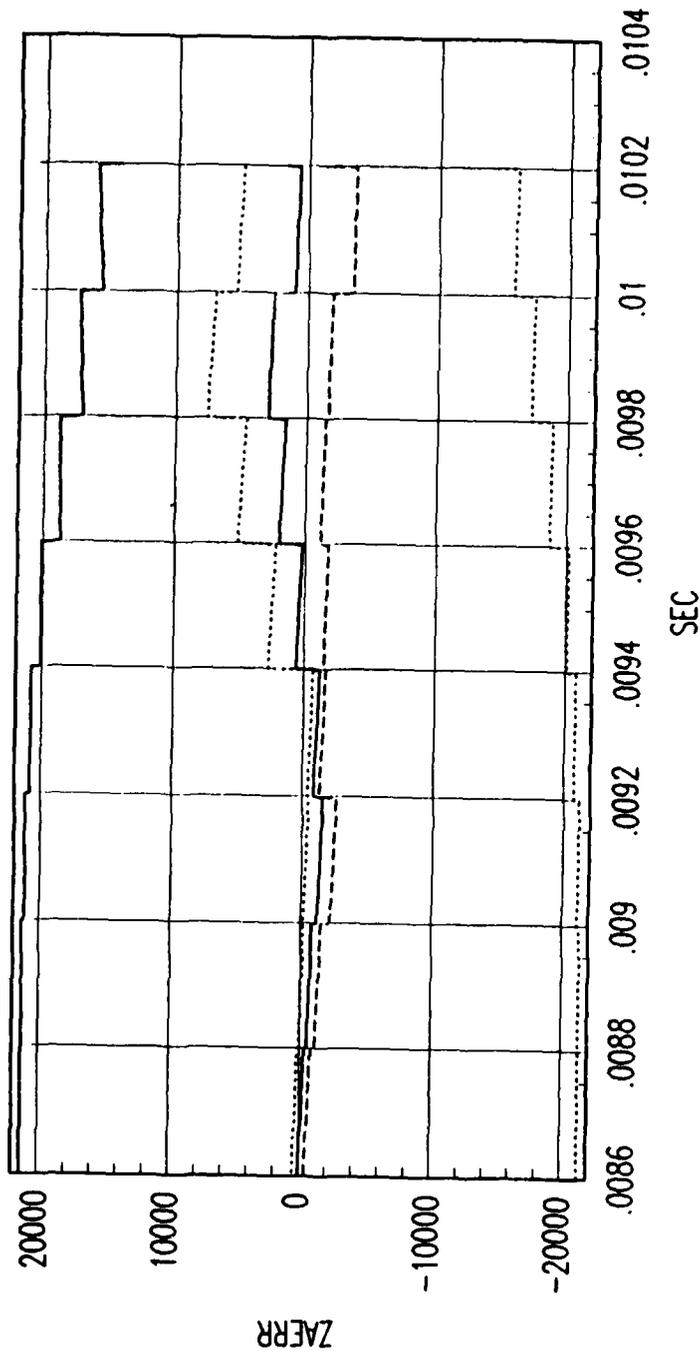


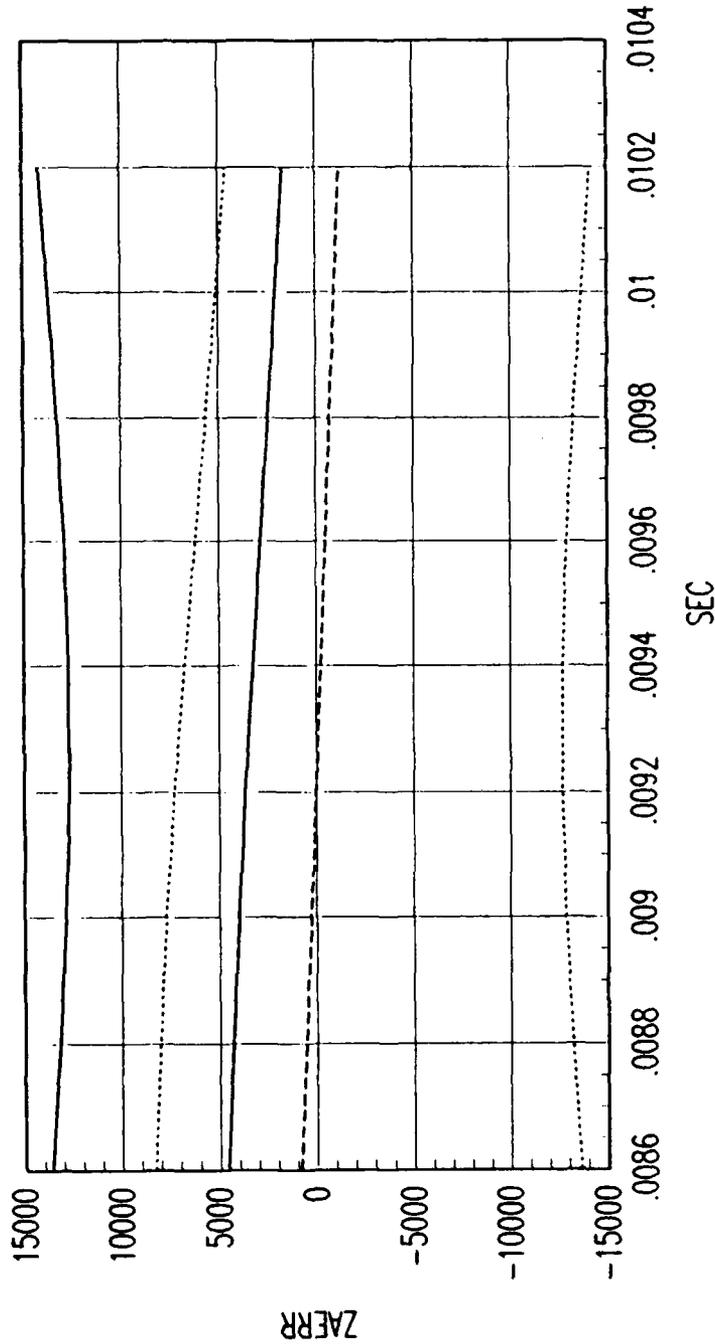
FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.142. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.143. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.144. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

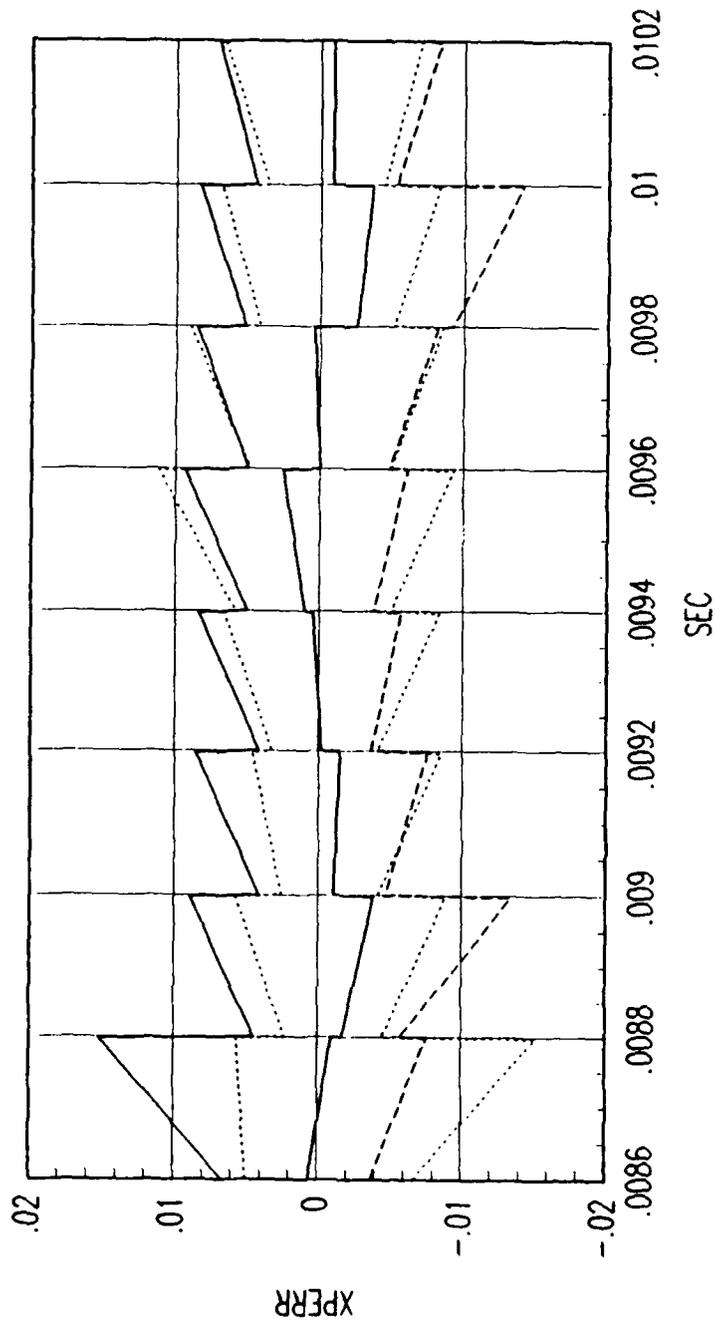


FIG. C FORWARD TIME EKf POSITION ERROR

Figure C.145. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

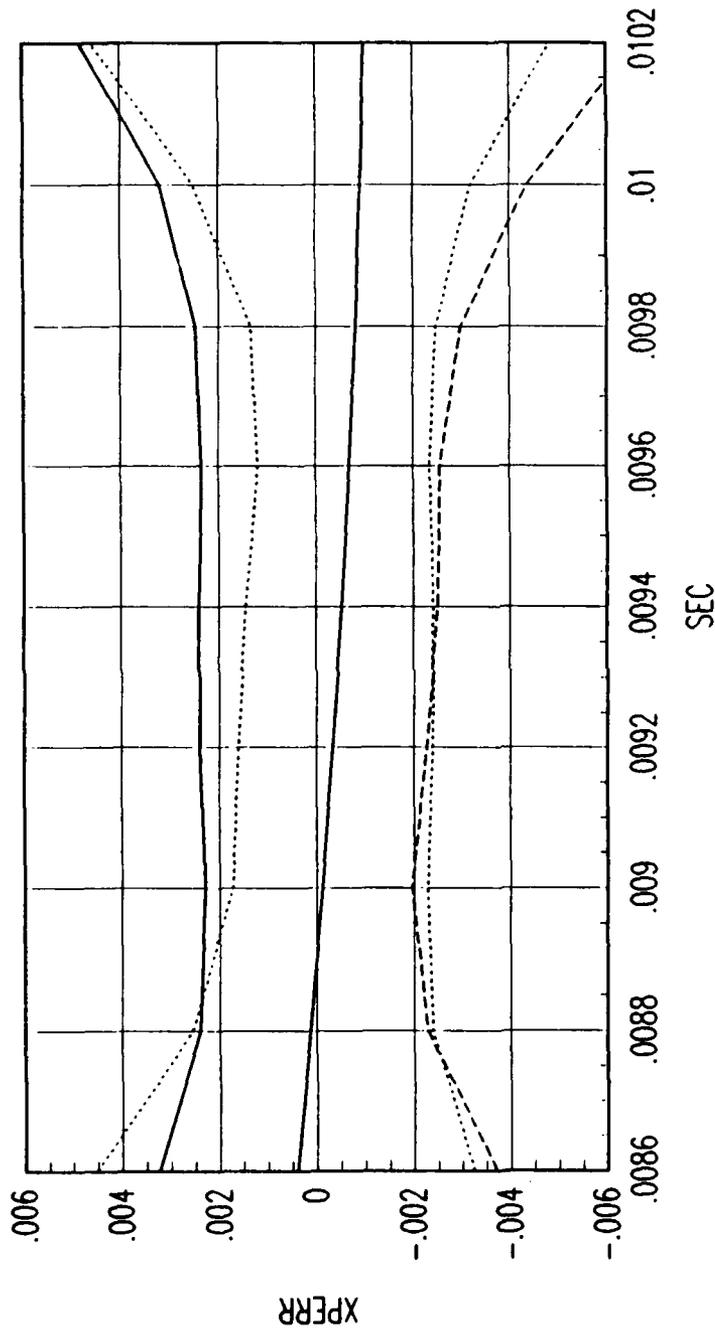
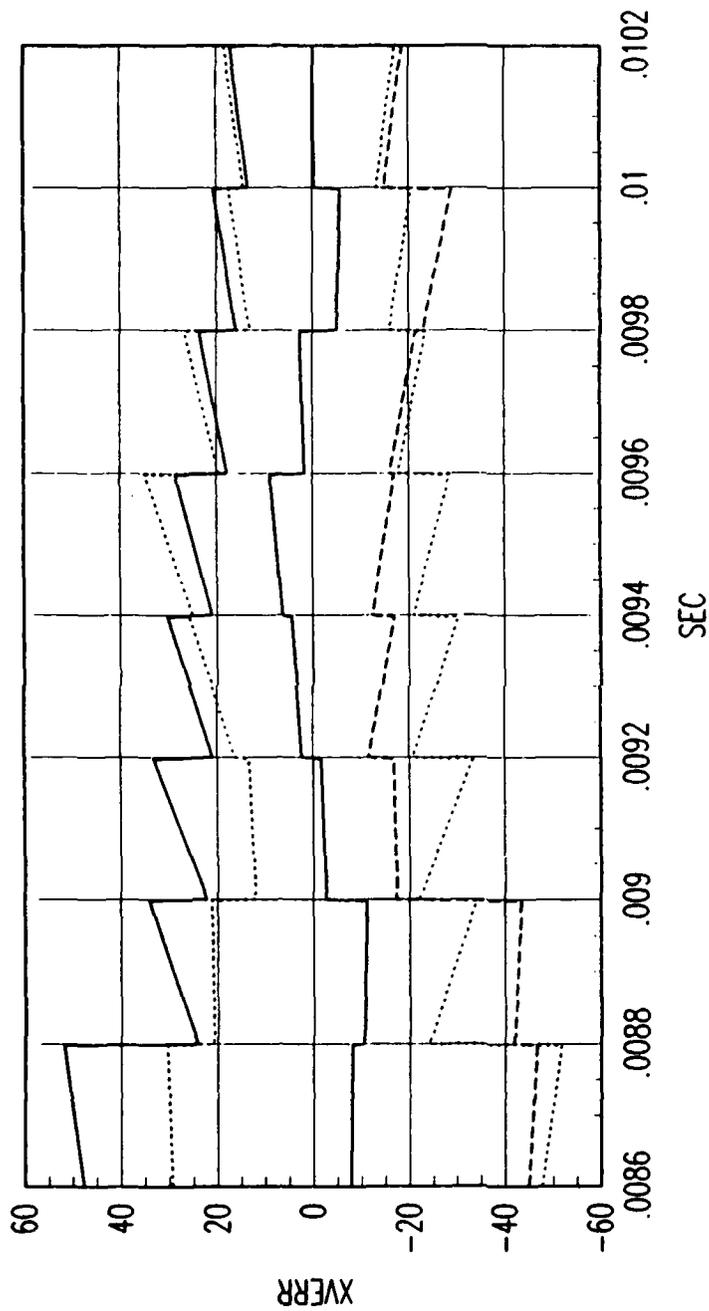


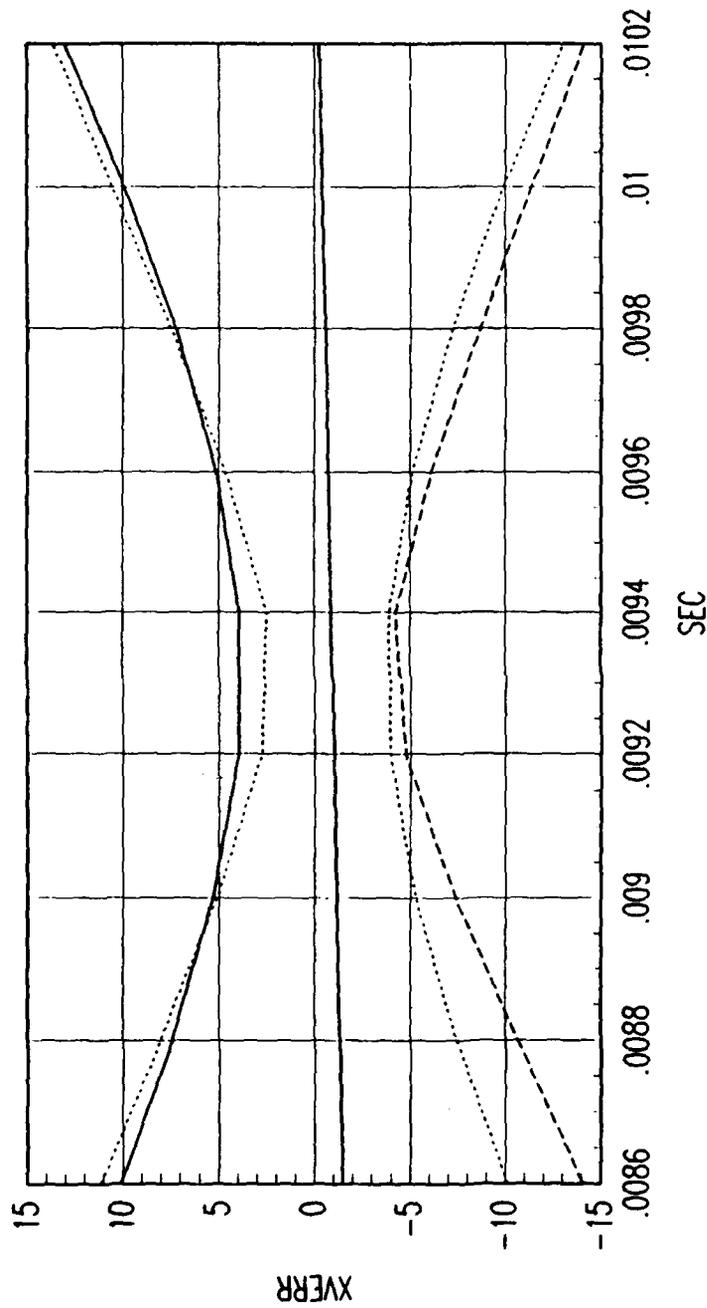
FIG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.146. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



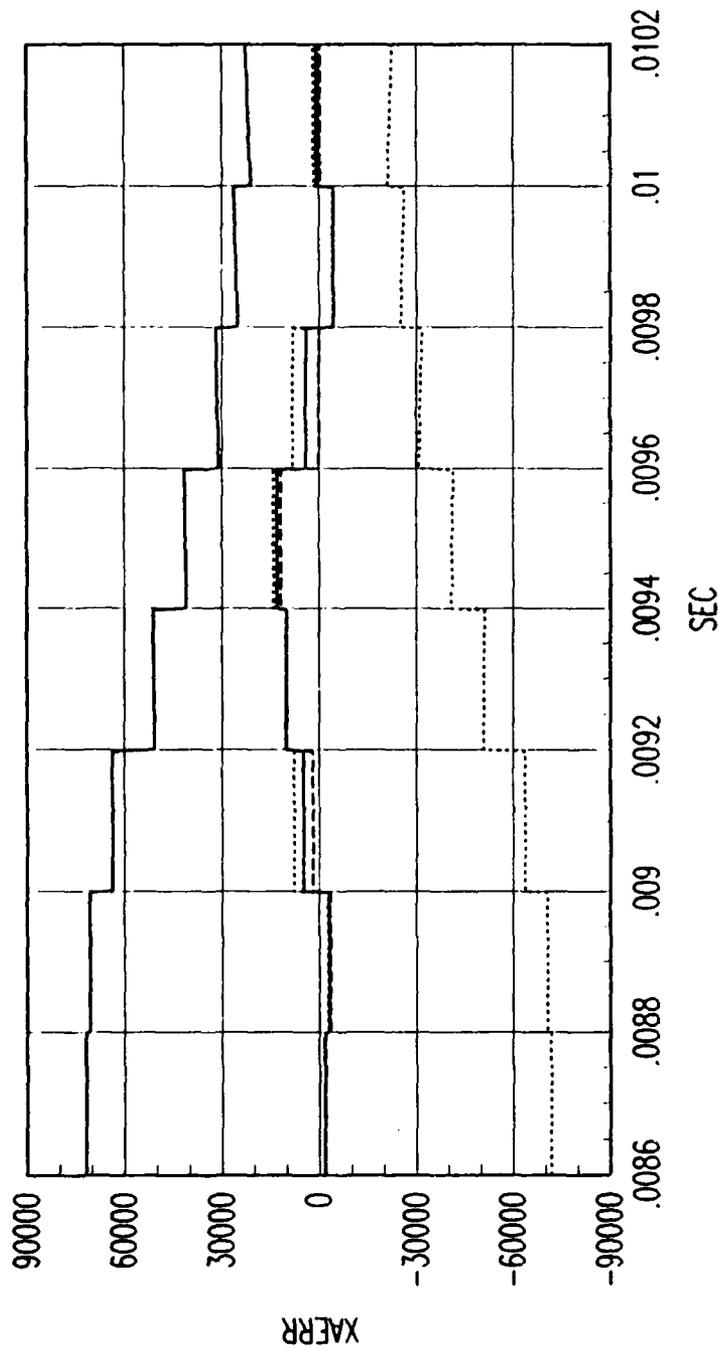
FRG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.147. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



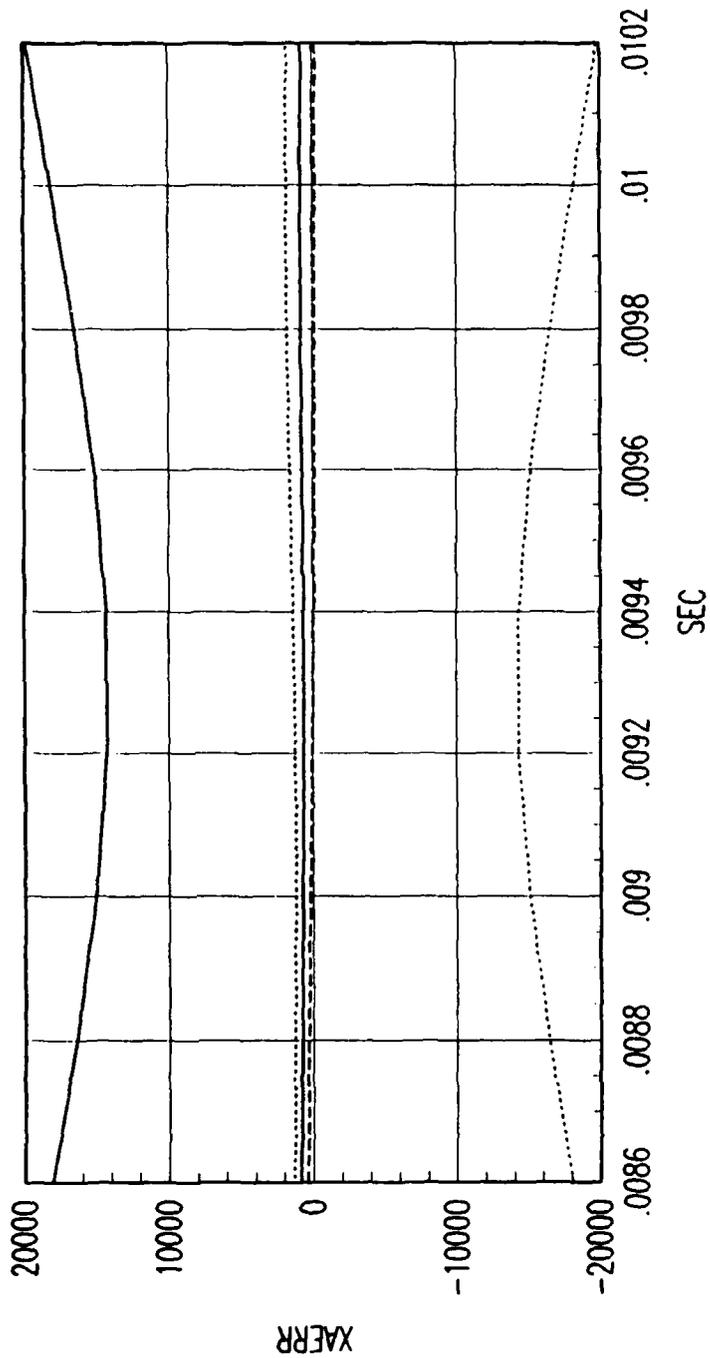
FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.148. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



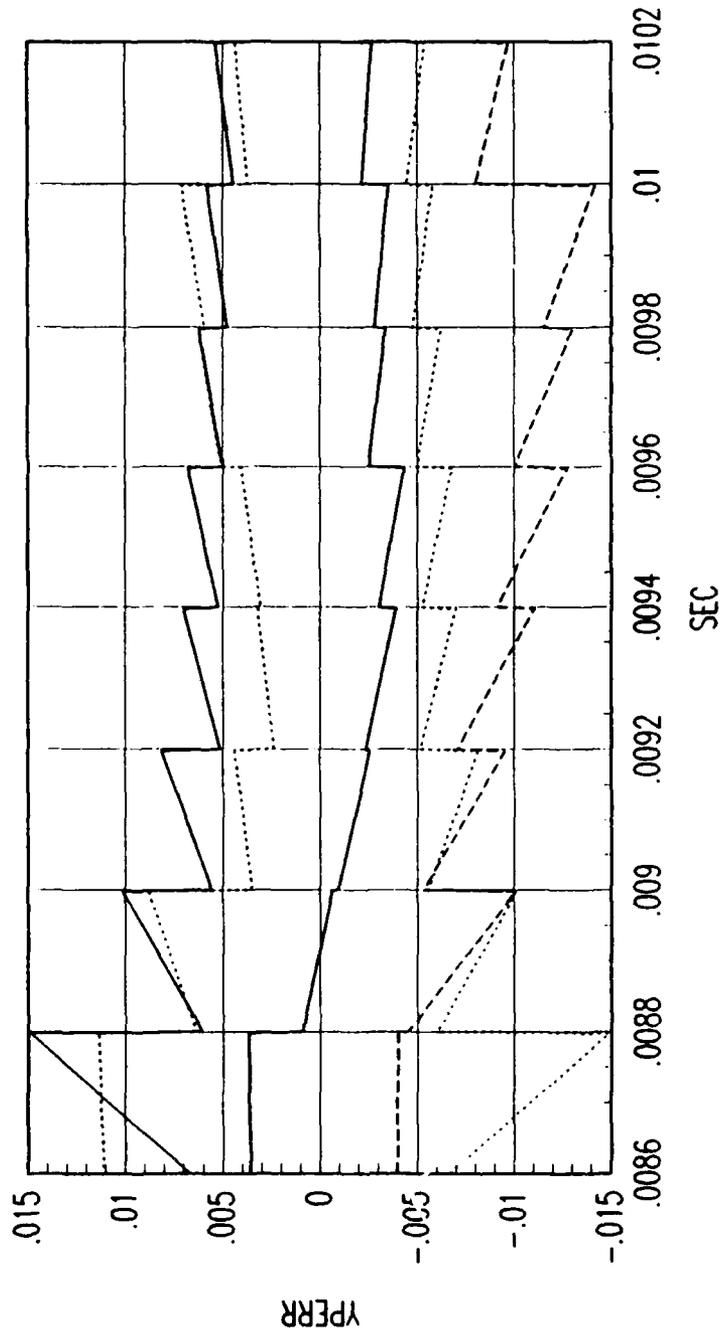
FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.149. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



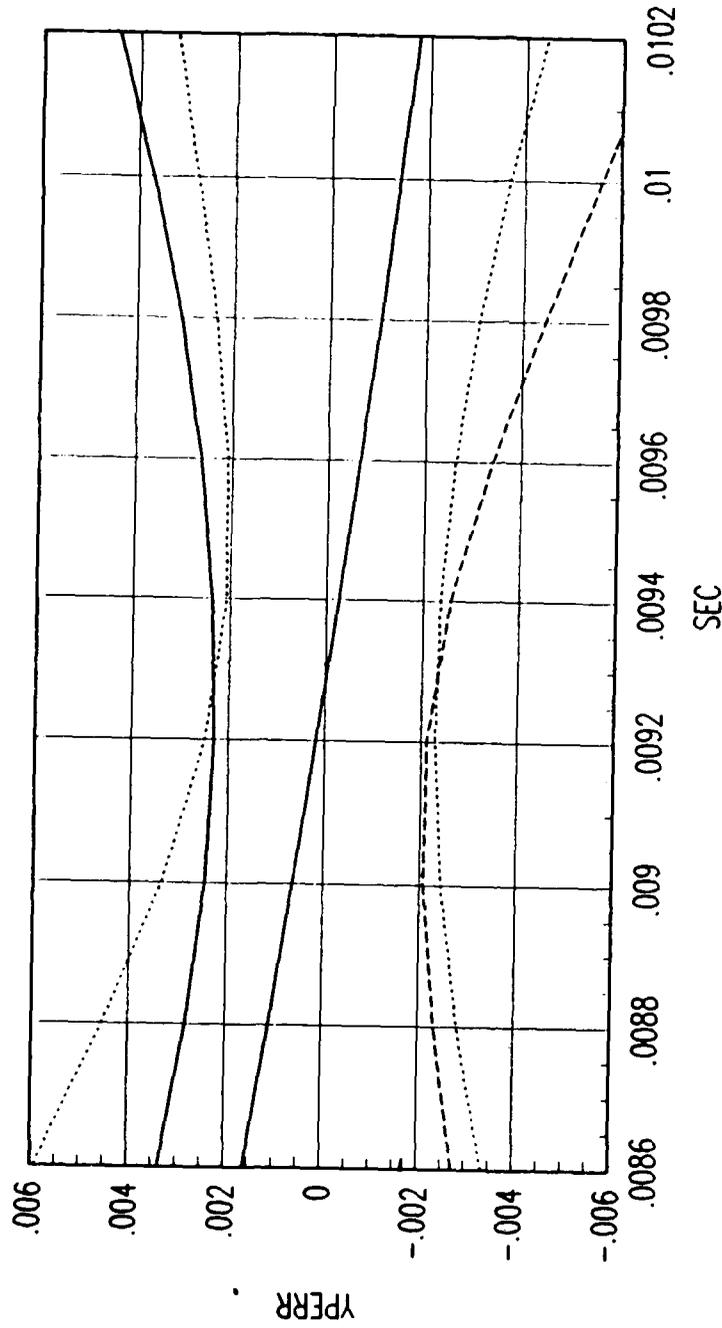
FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.150. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. C FORWARD TIME EKF POSITION ERROR

Figure C.151. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.152. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

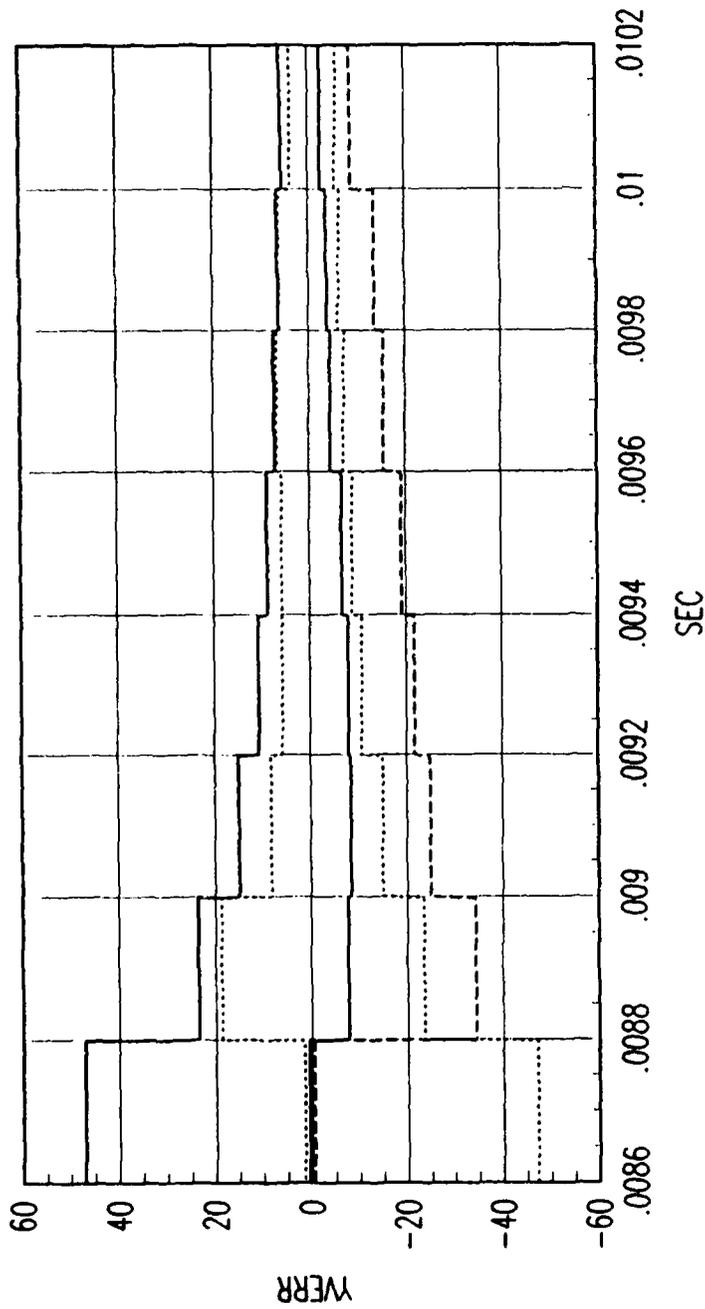
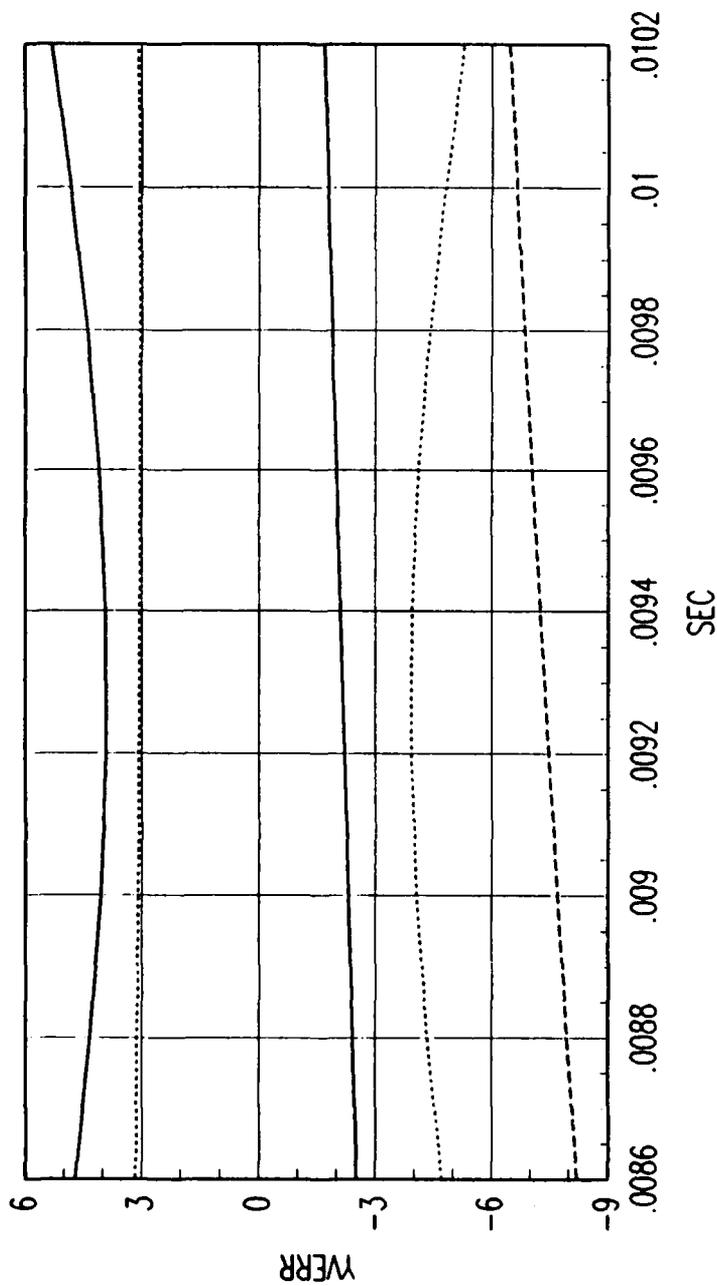


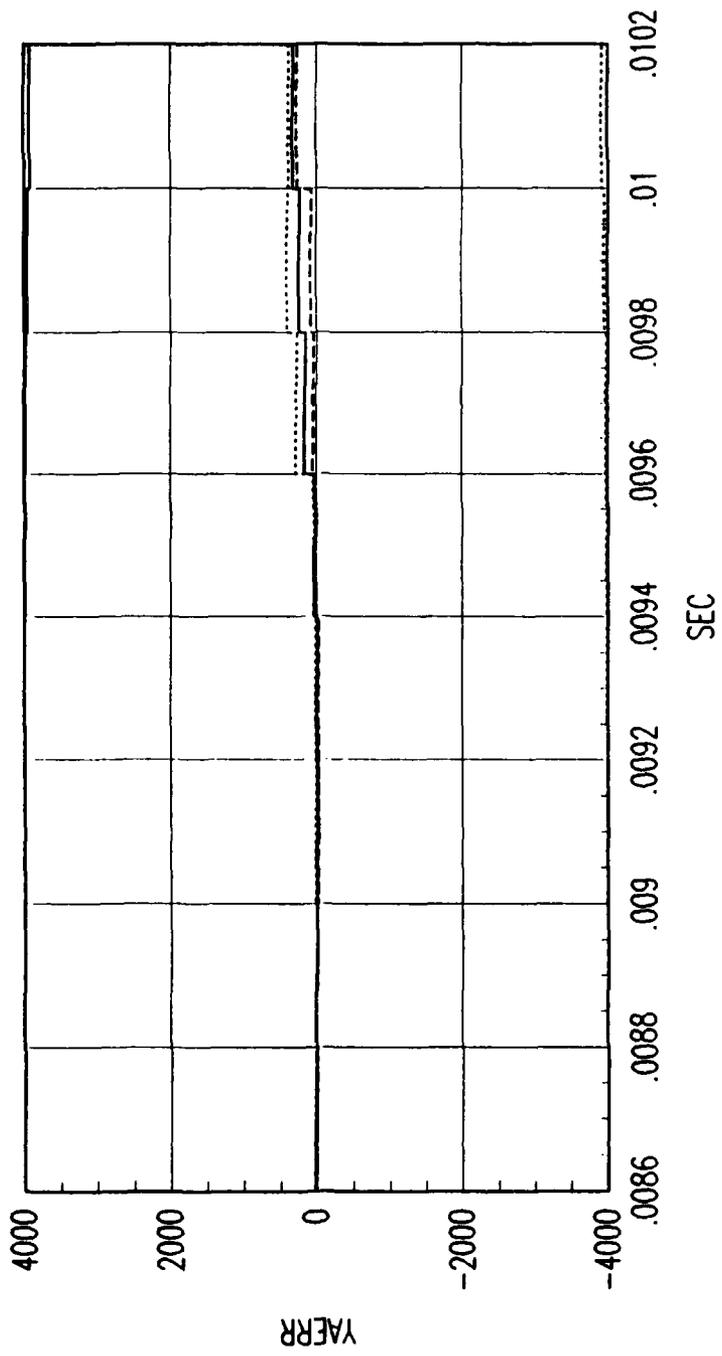
FIG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.153. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



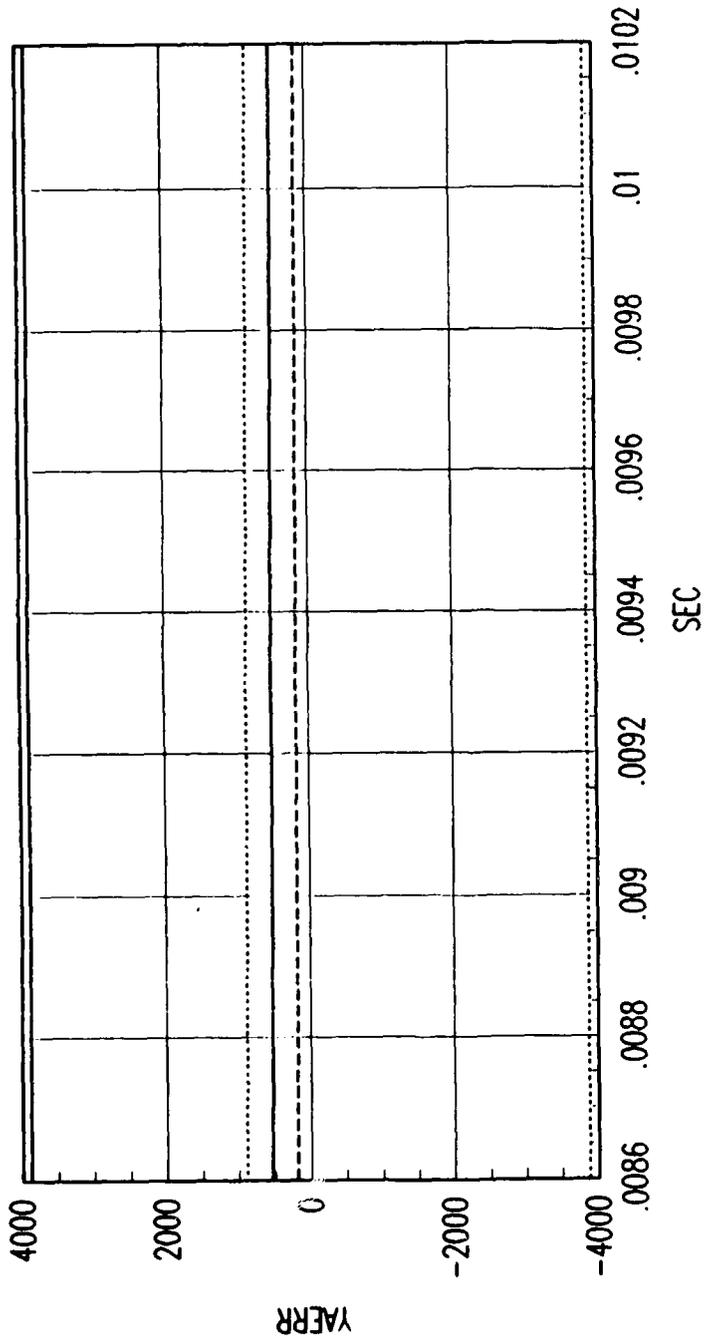
FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.154. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



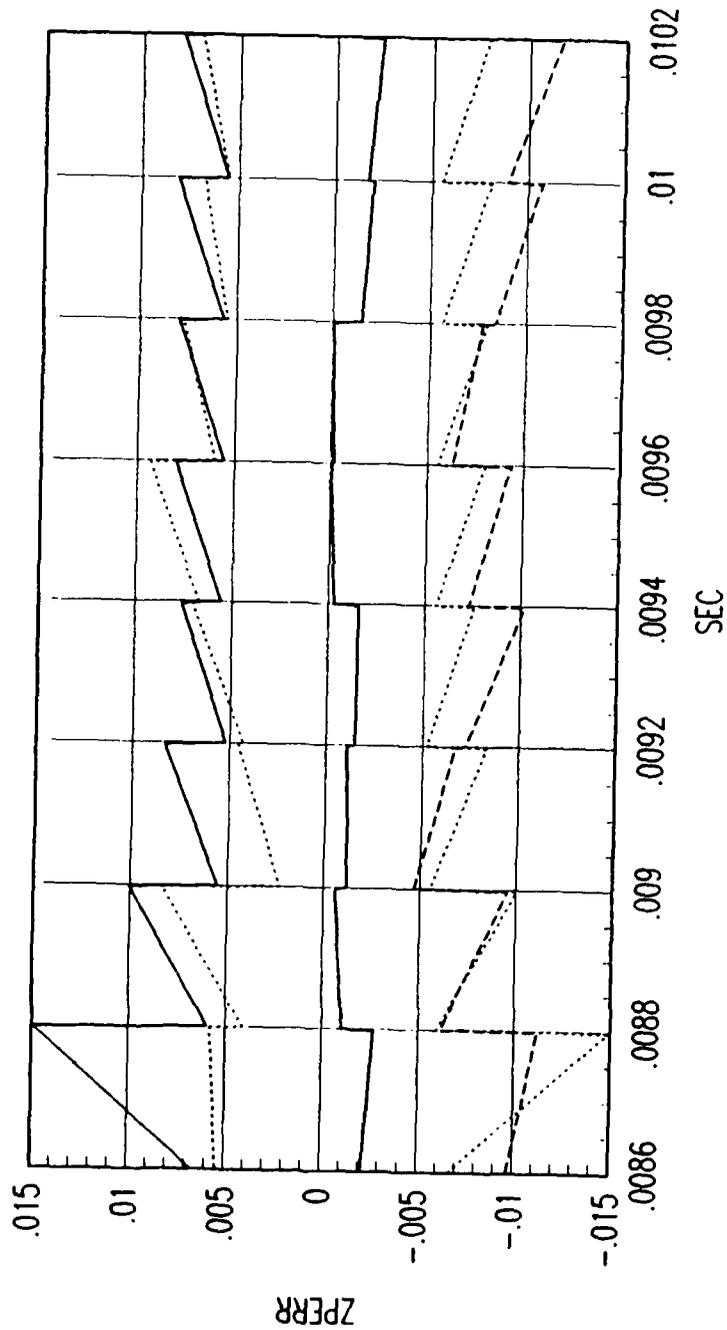
FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.155. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



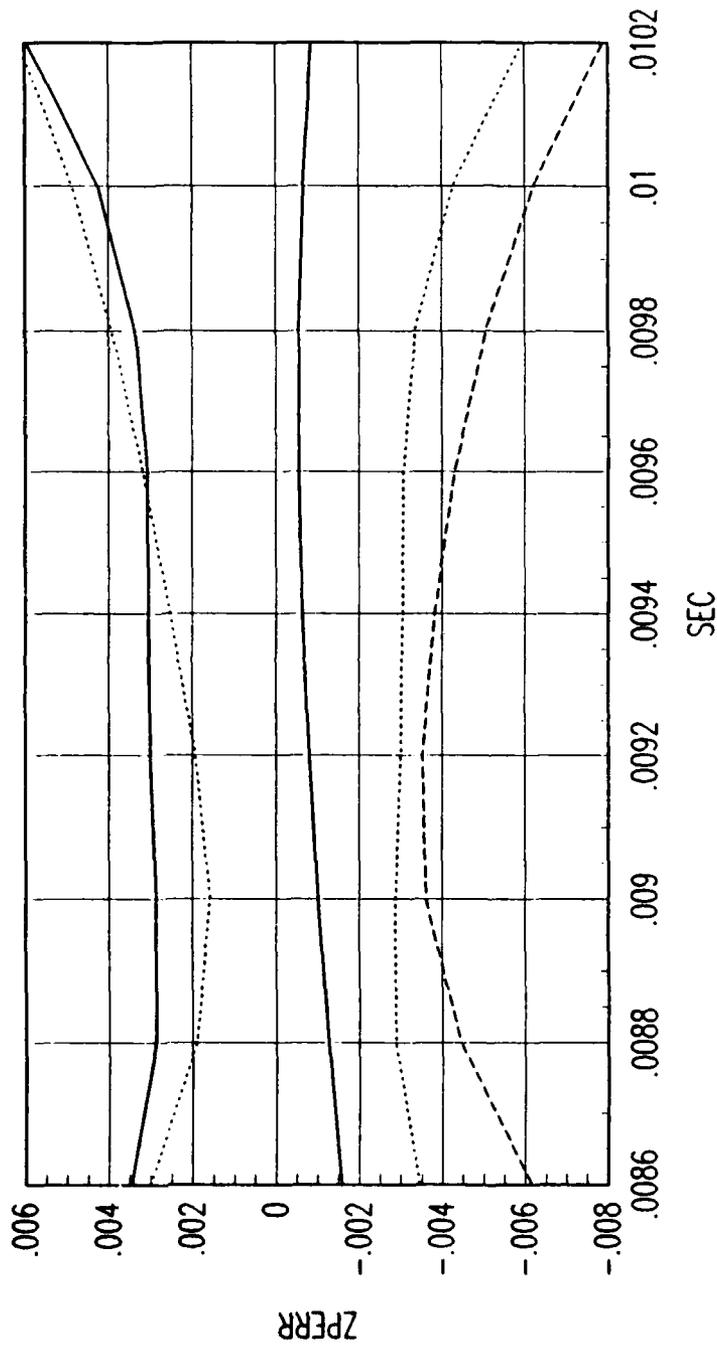
FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.156. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. C FORWARD TIME EKF POSITION ERROR

Figure C.157. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.15^a. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

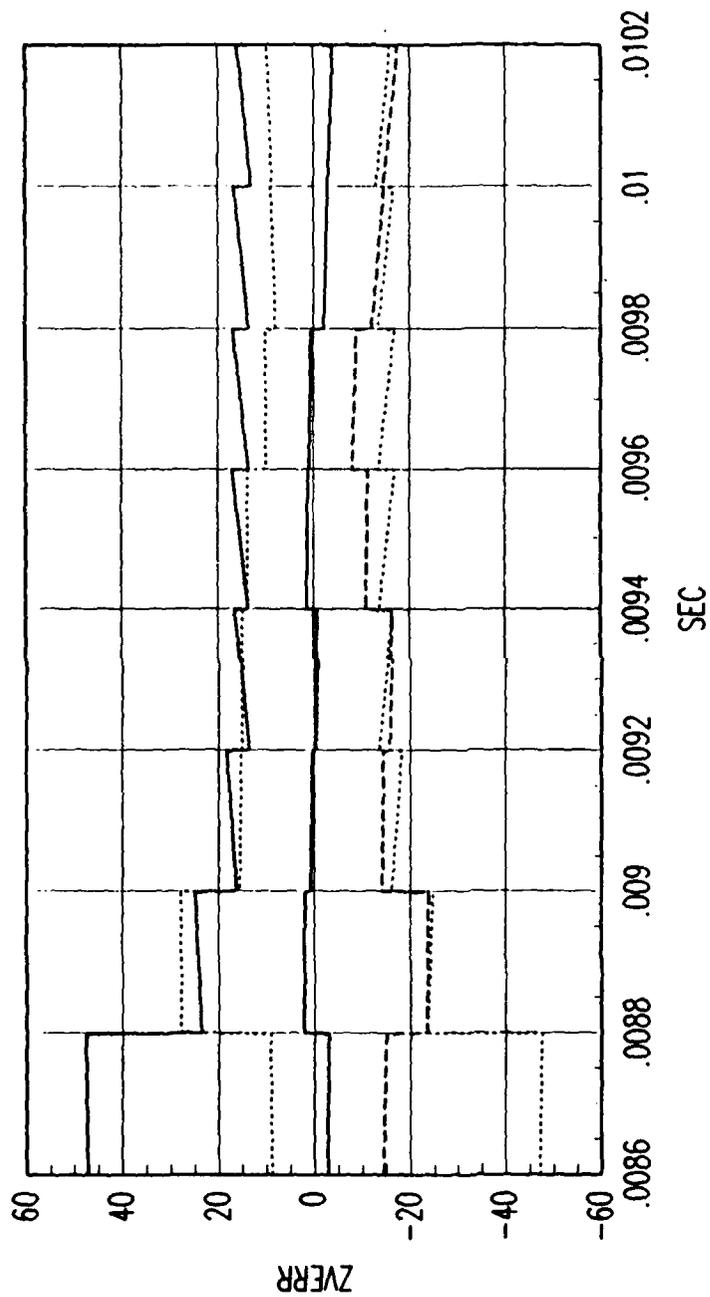


FIG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.159. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

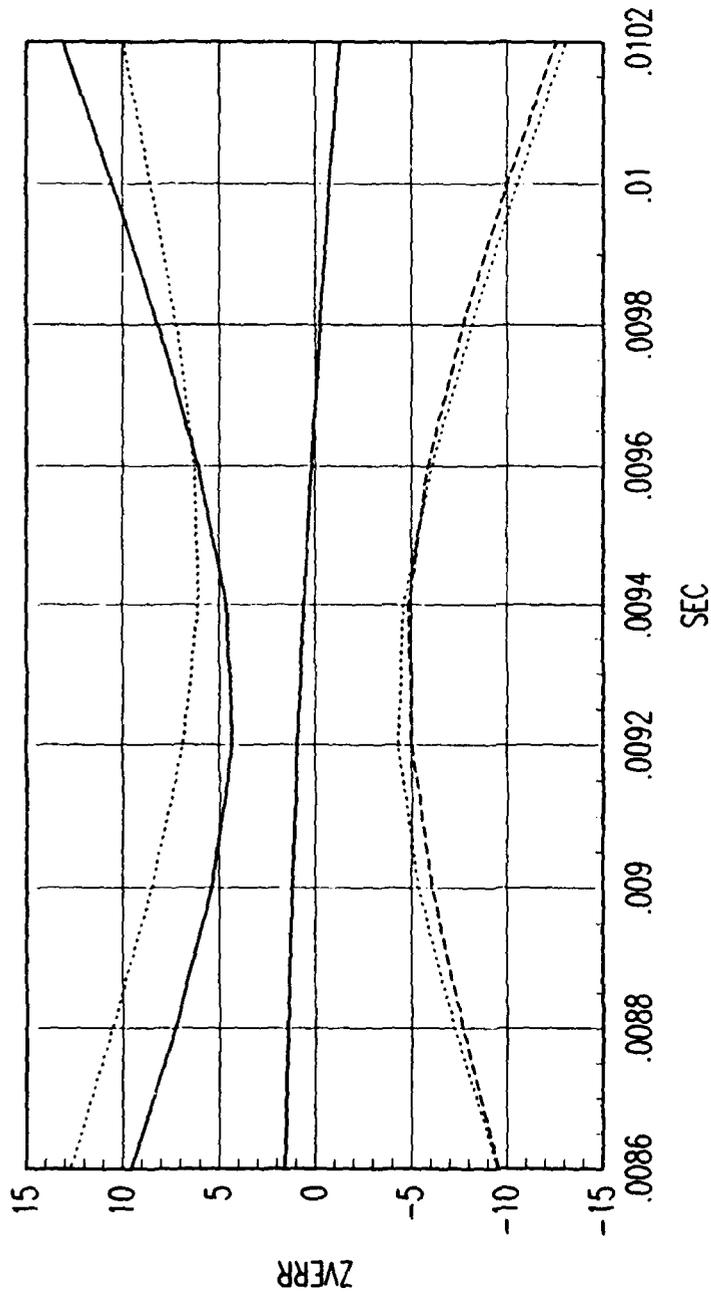
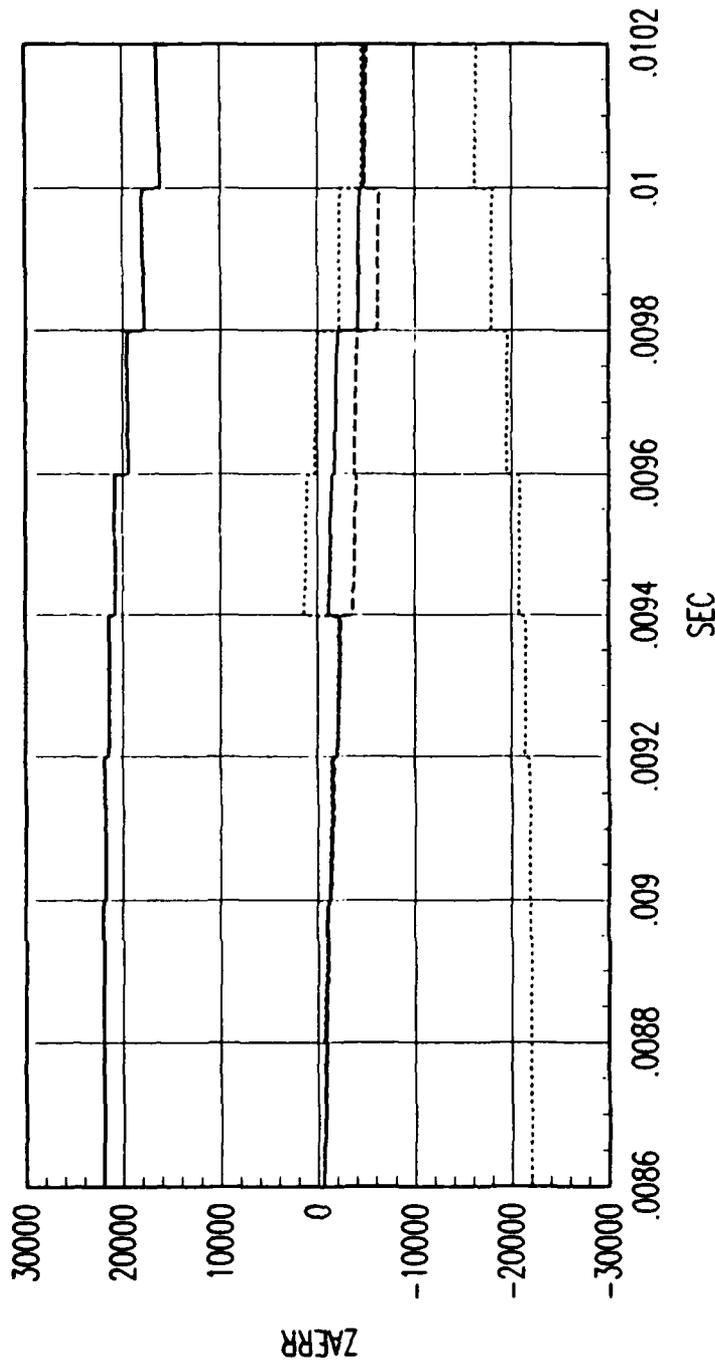


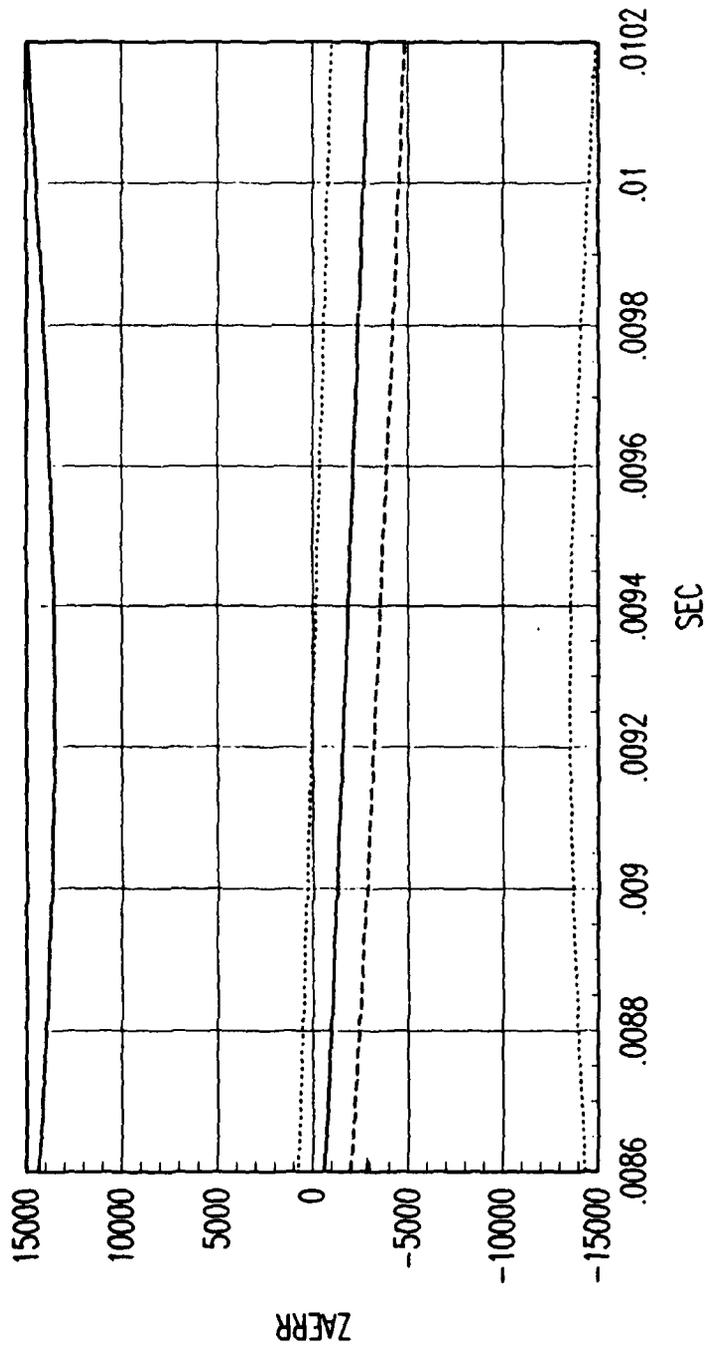
FIG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.160. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.161. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.162. Tracking Error Plot, Category 2, $v(t_0) = 6000$ ft/sec.

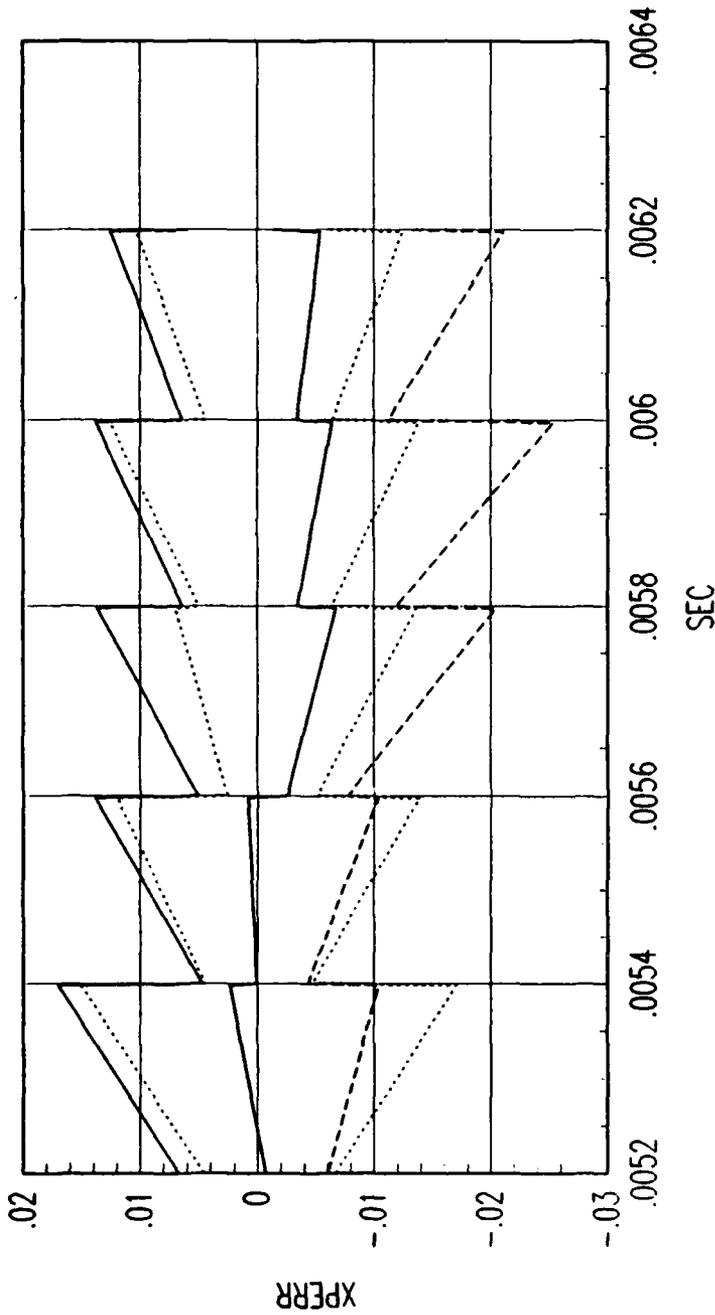


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.163. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

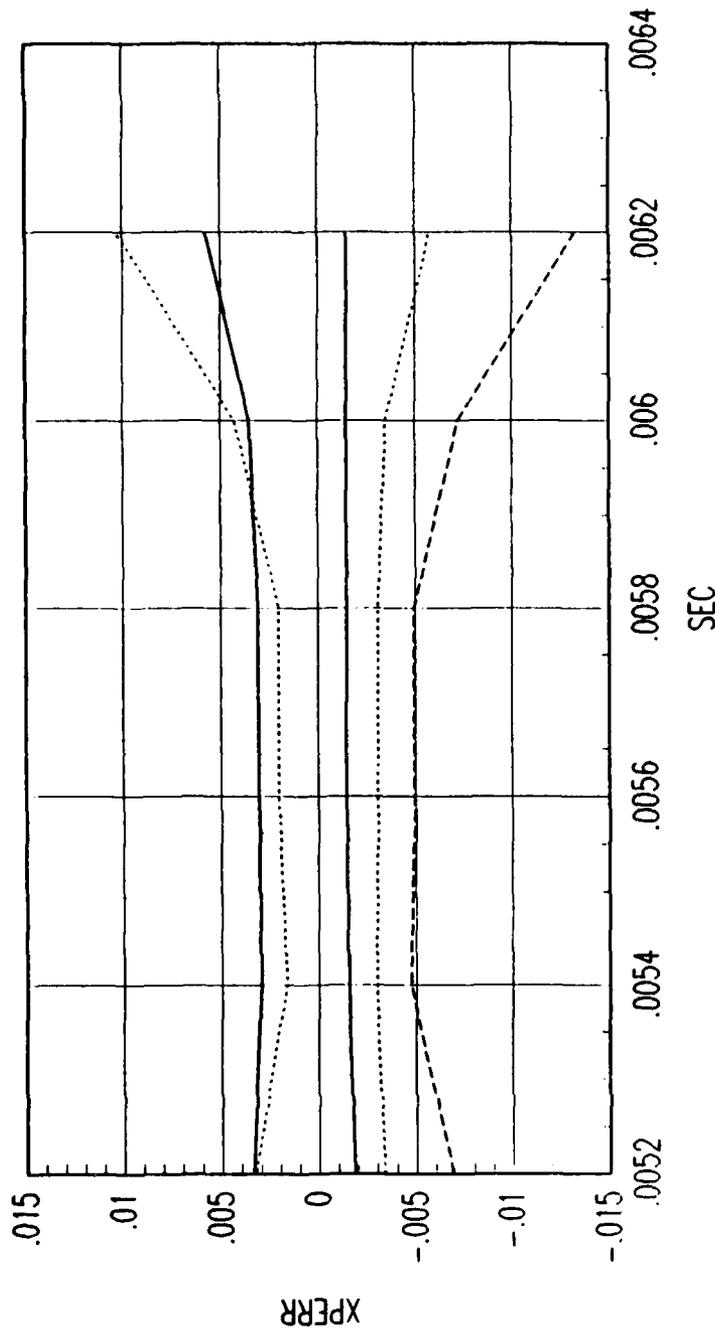


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.164. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

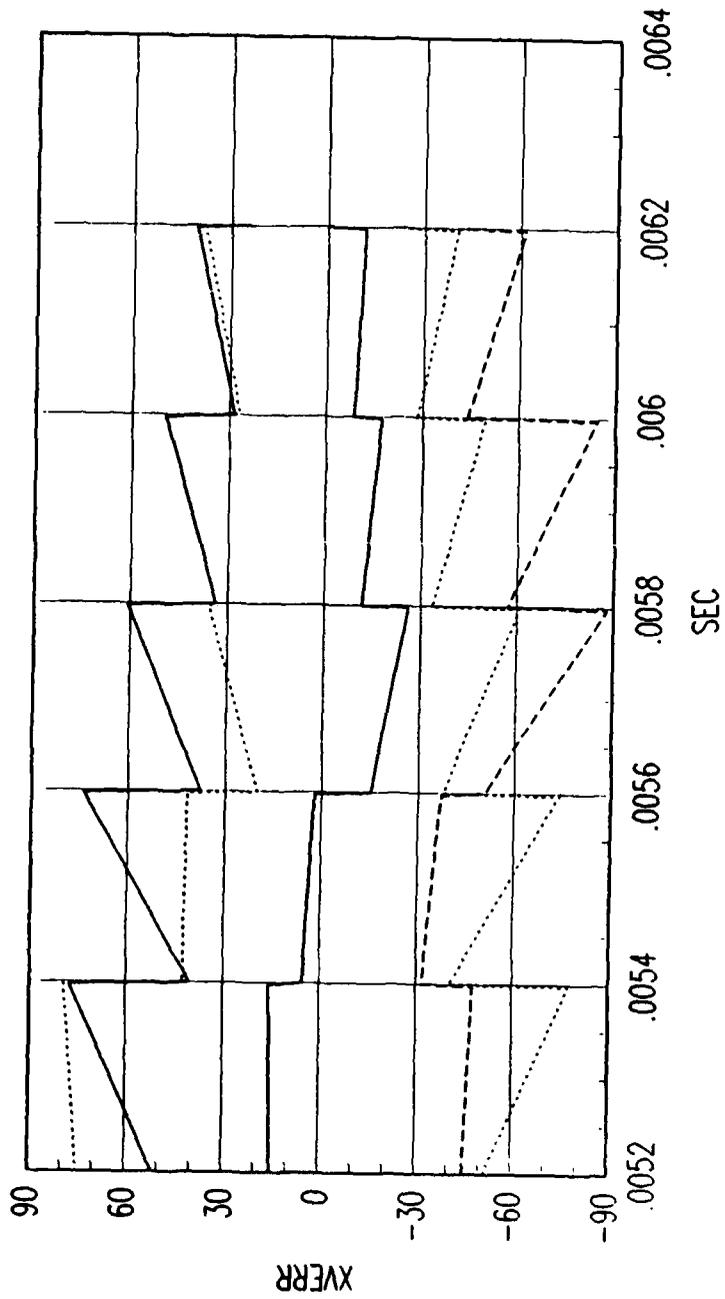
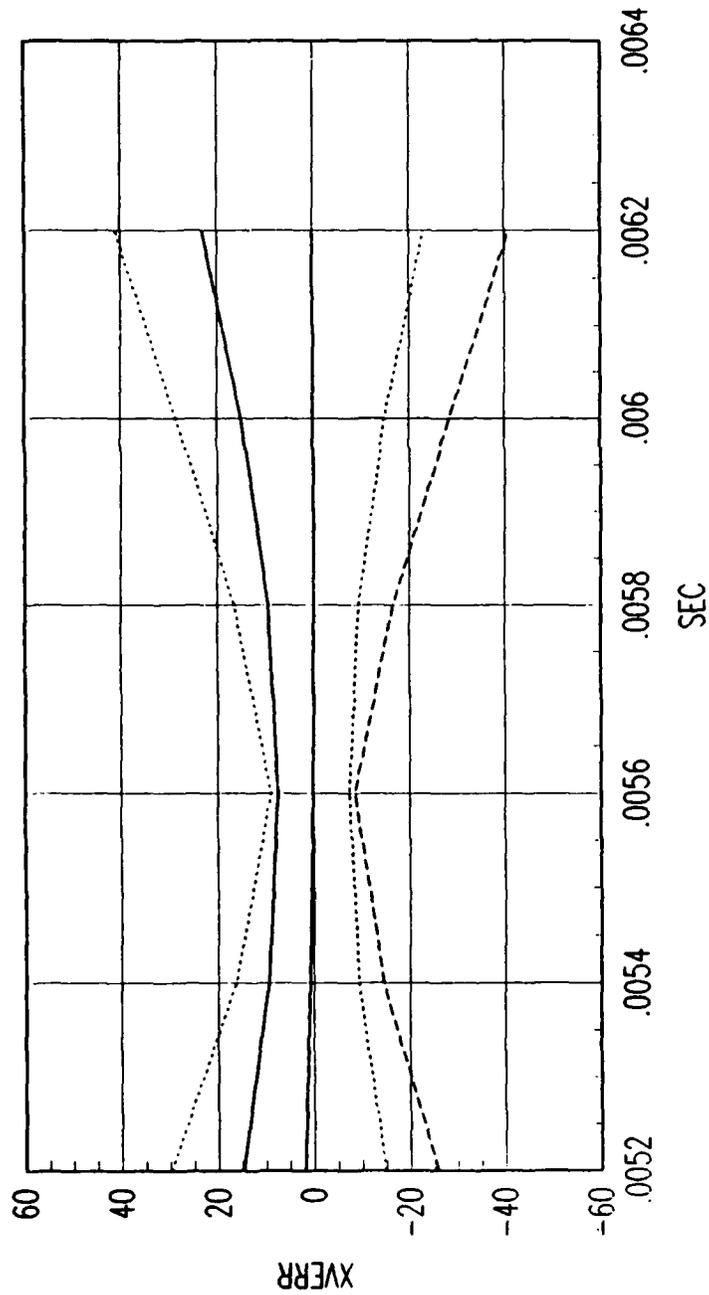


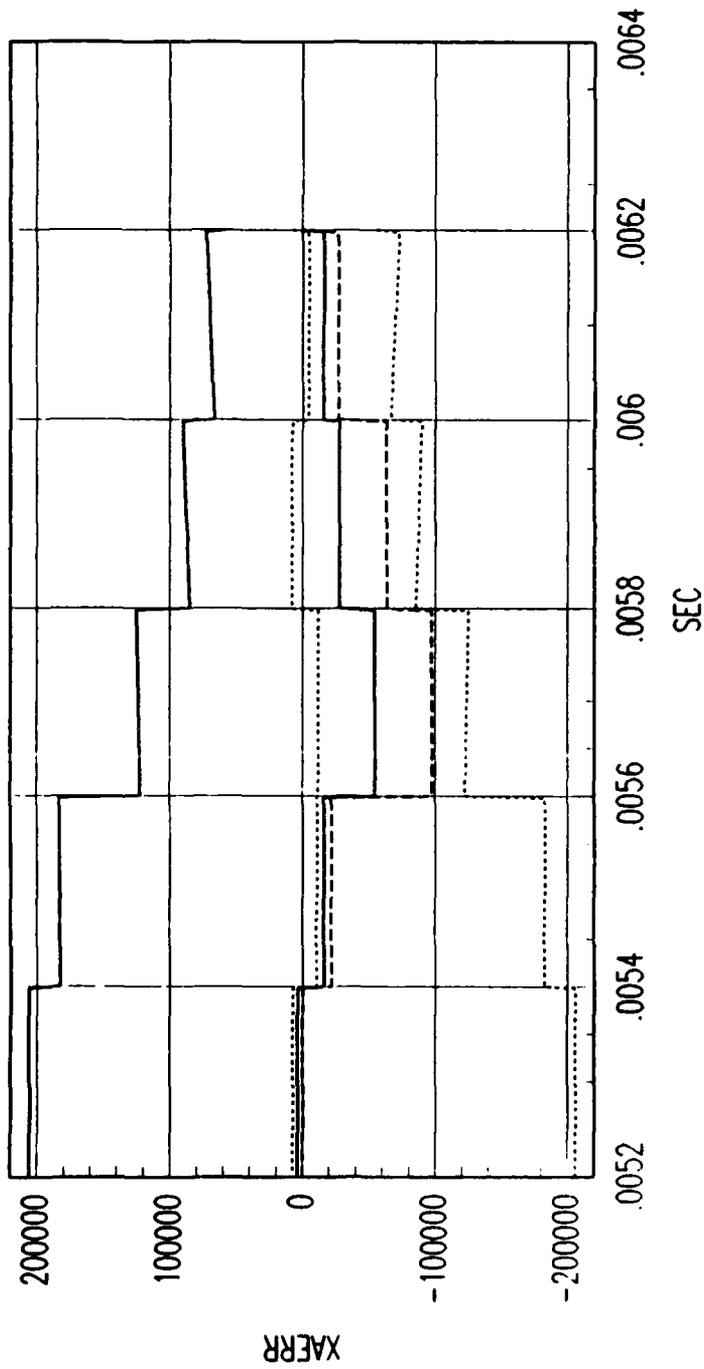
FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.165. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.166. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.167. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

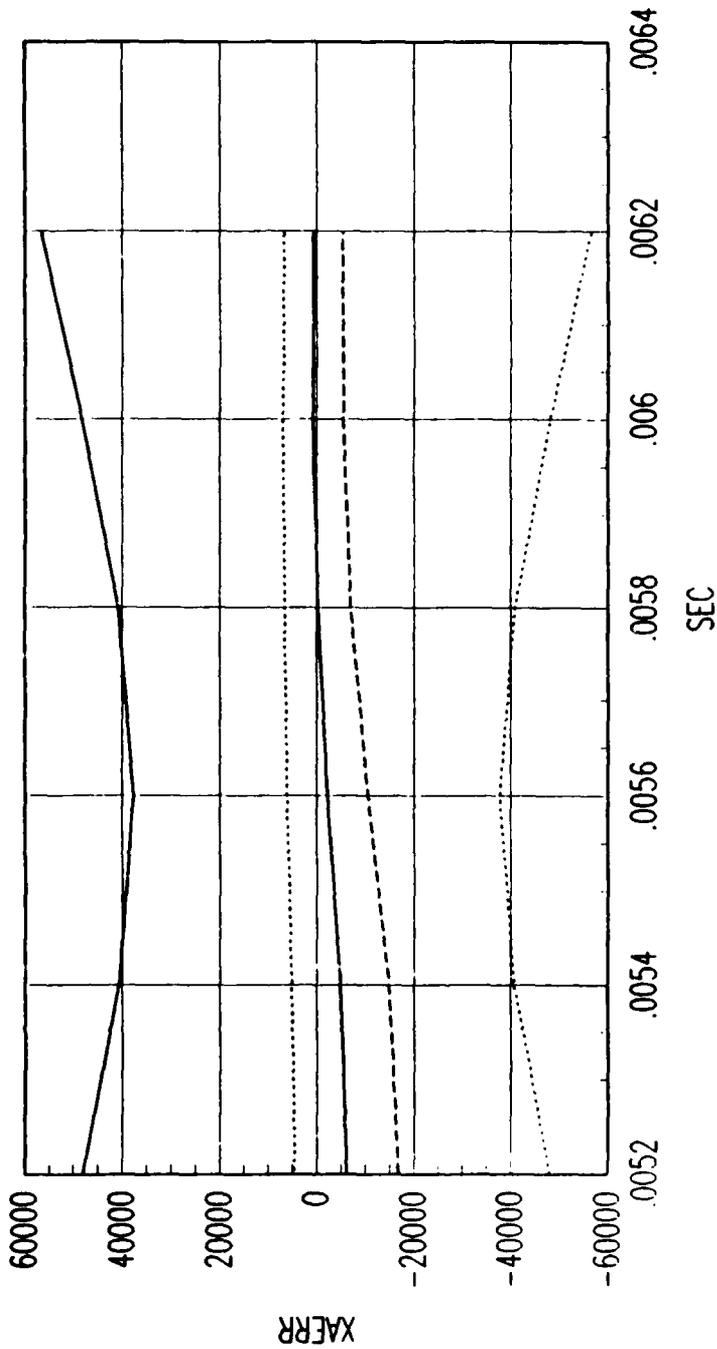


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.168. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

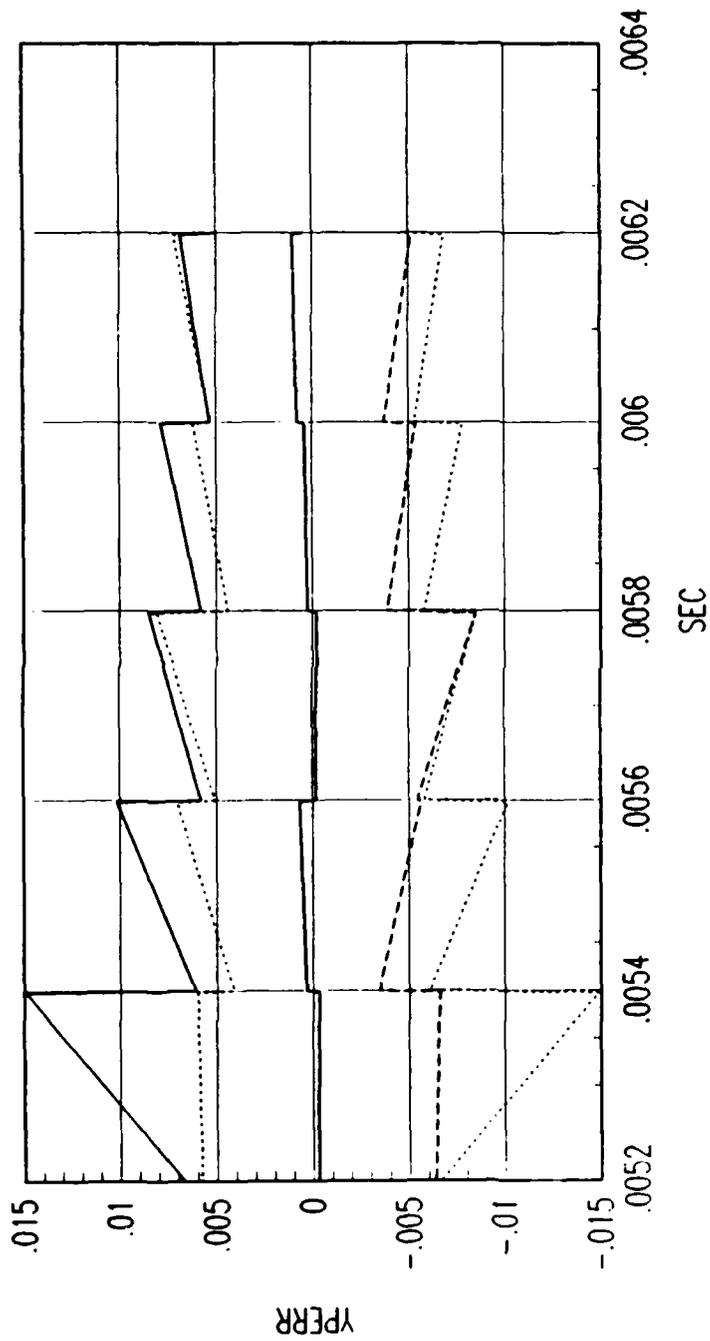


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.169. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

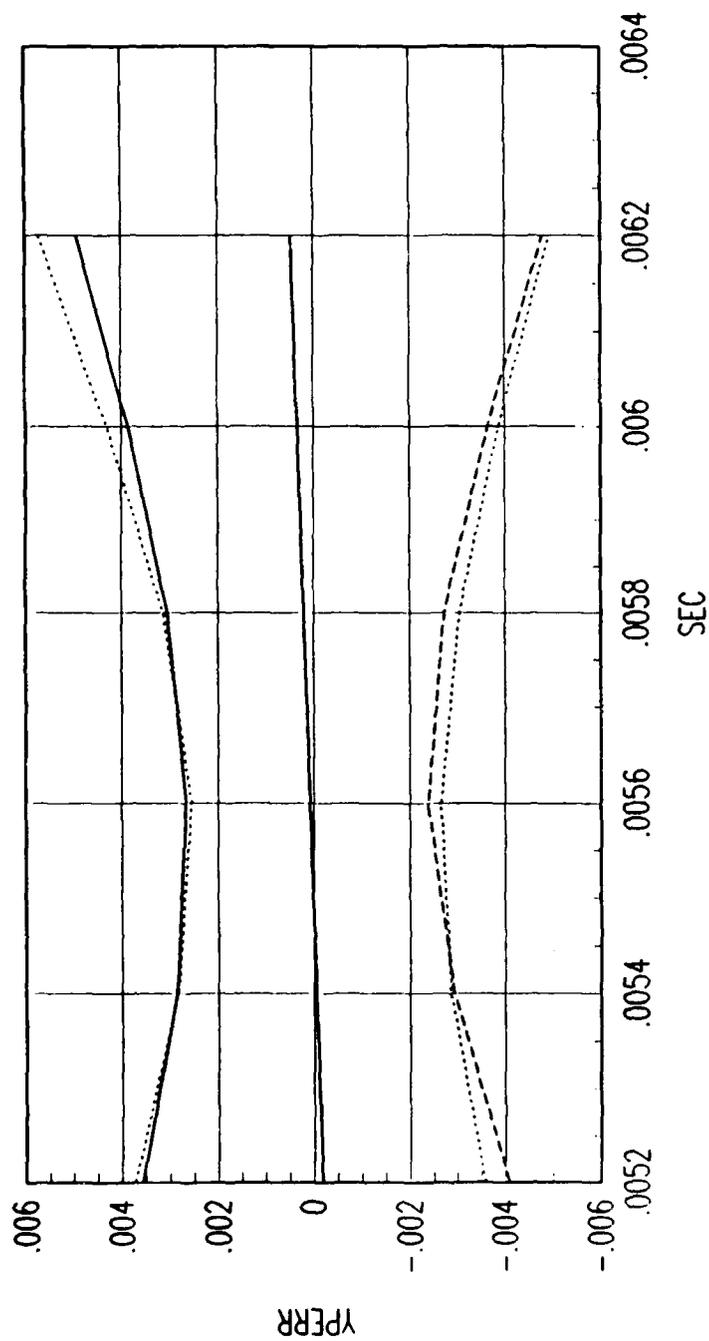


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.170. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

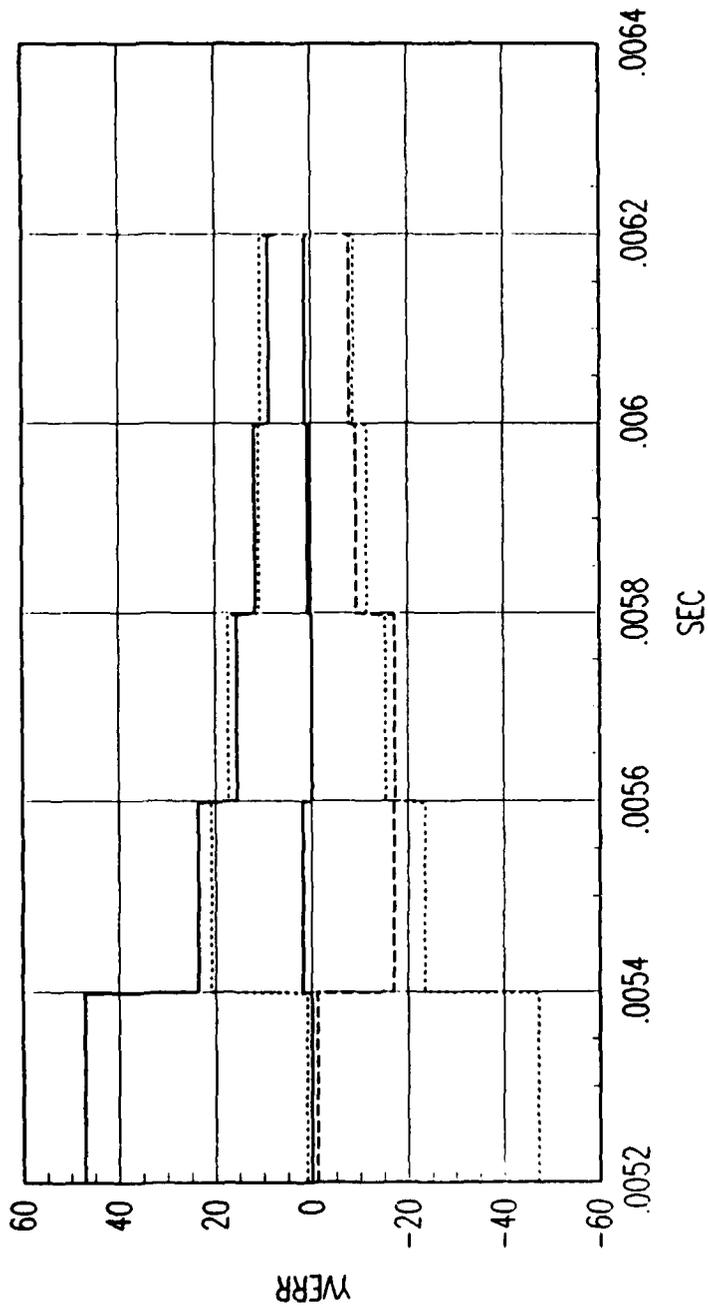


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.171. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

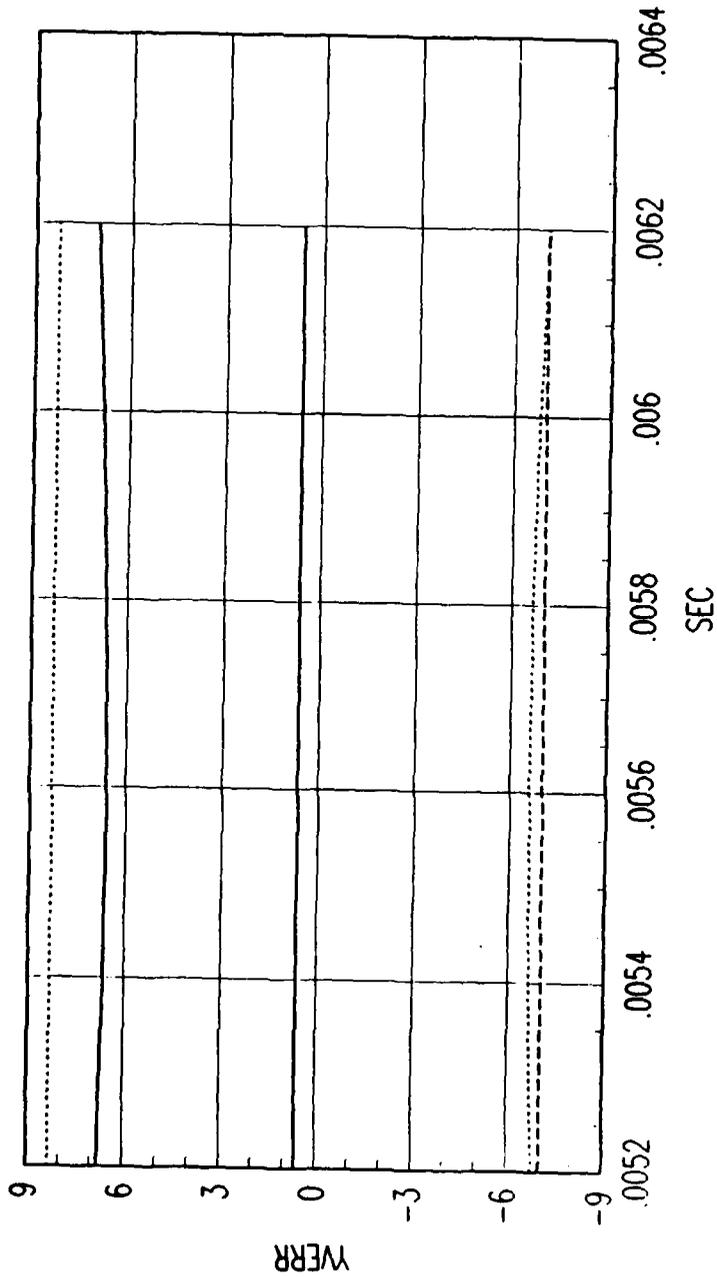
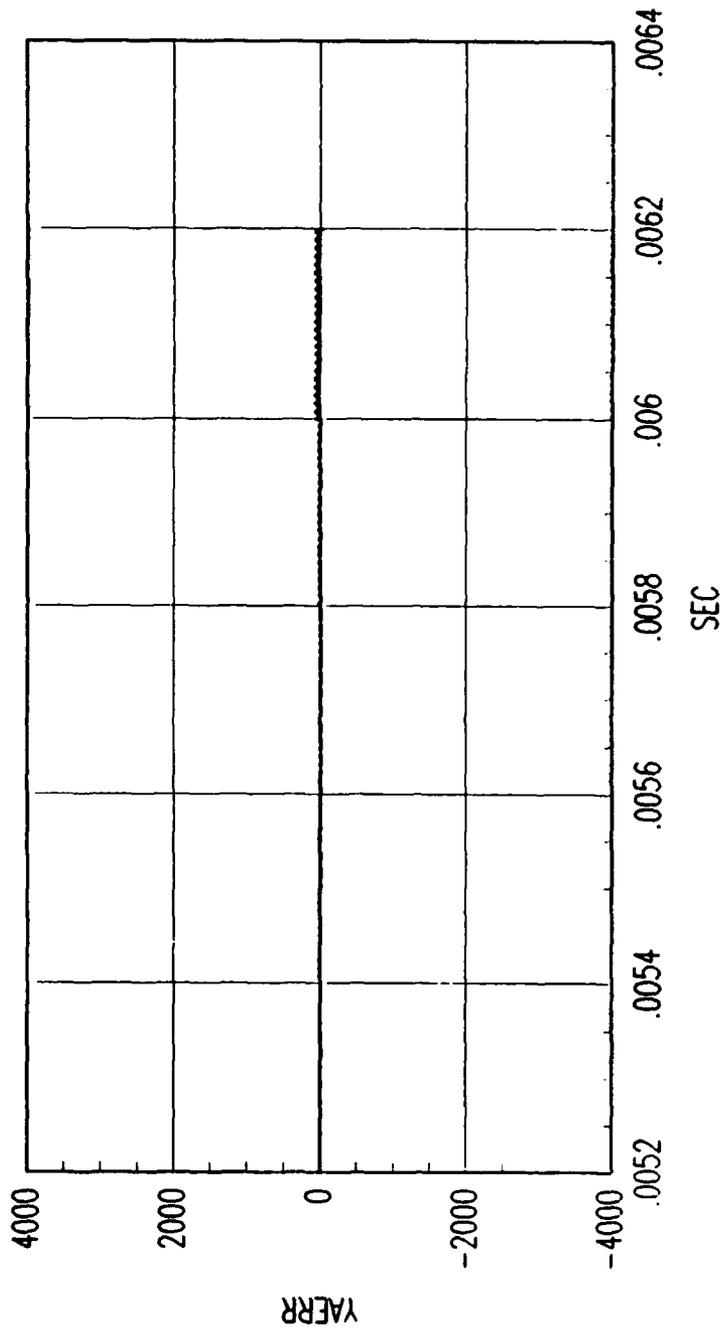


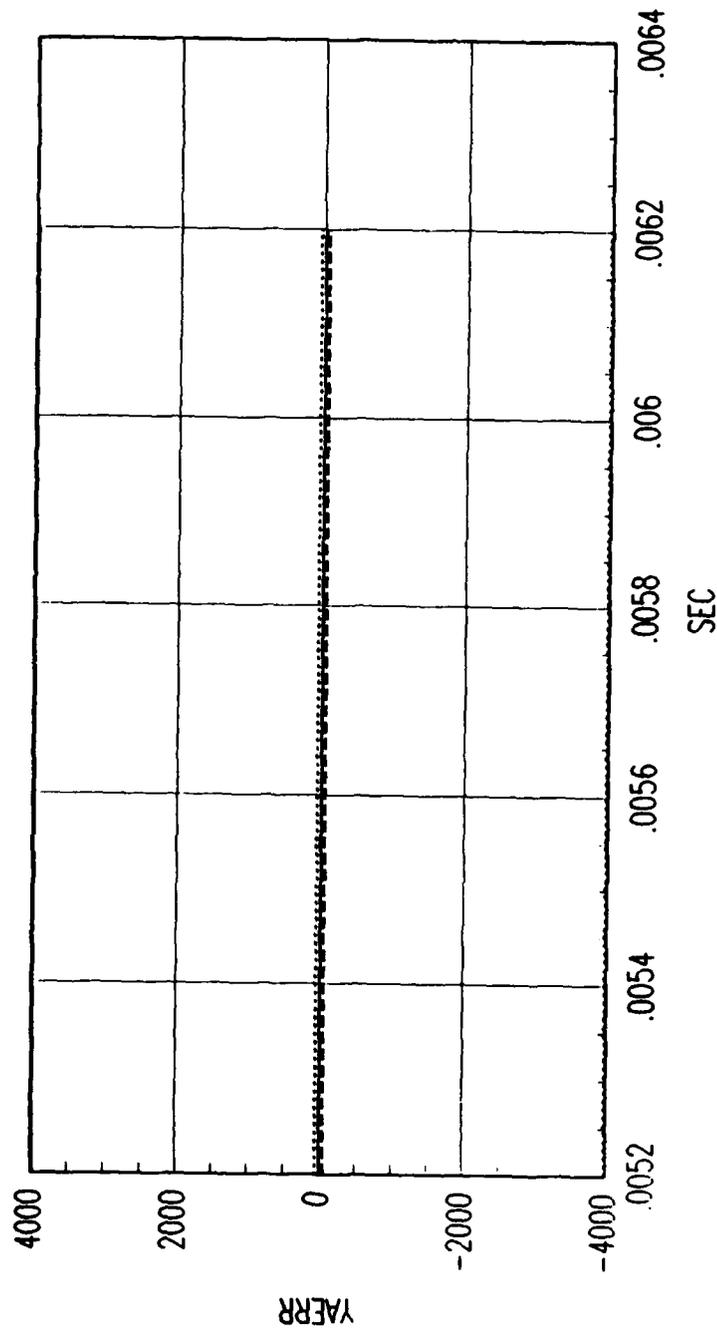
FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.172. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



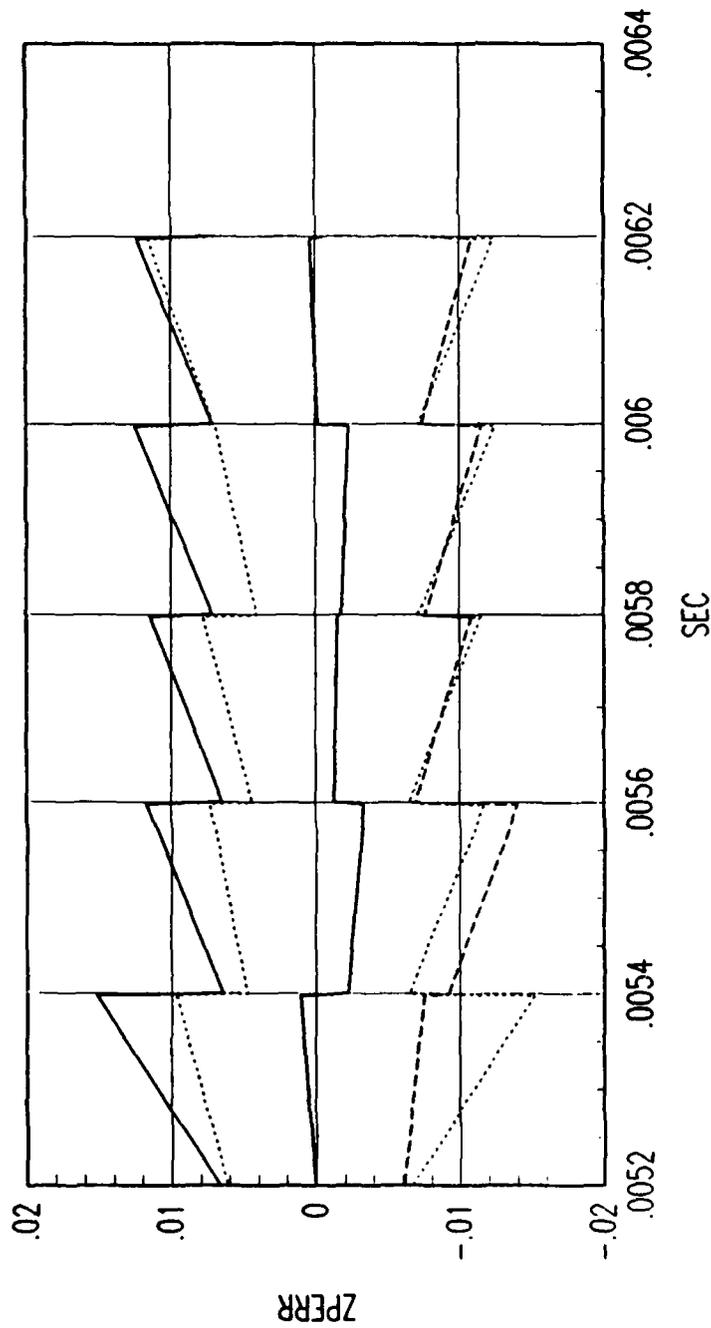
FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.173. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.174. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. A FORWARD TIME EKF POSITION ERROR

Figure C.175. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

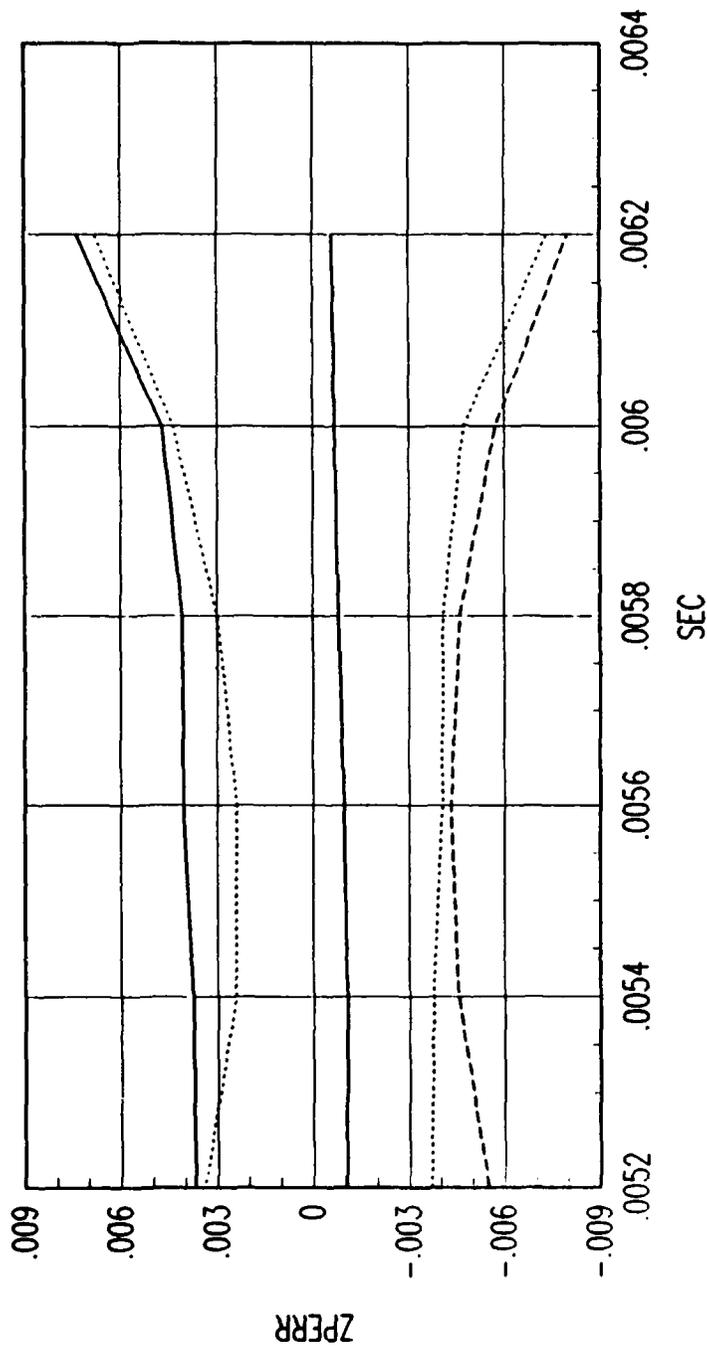


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.176. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

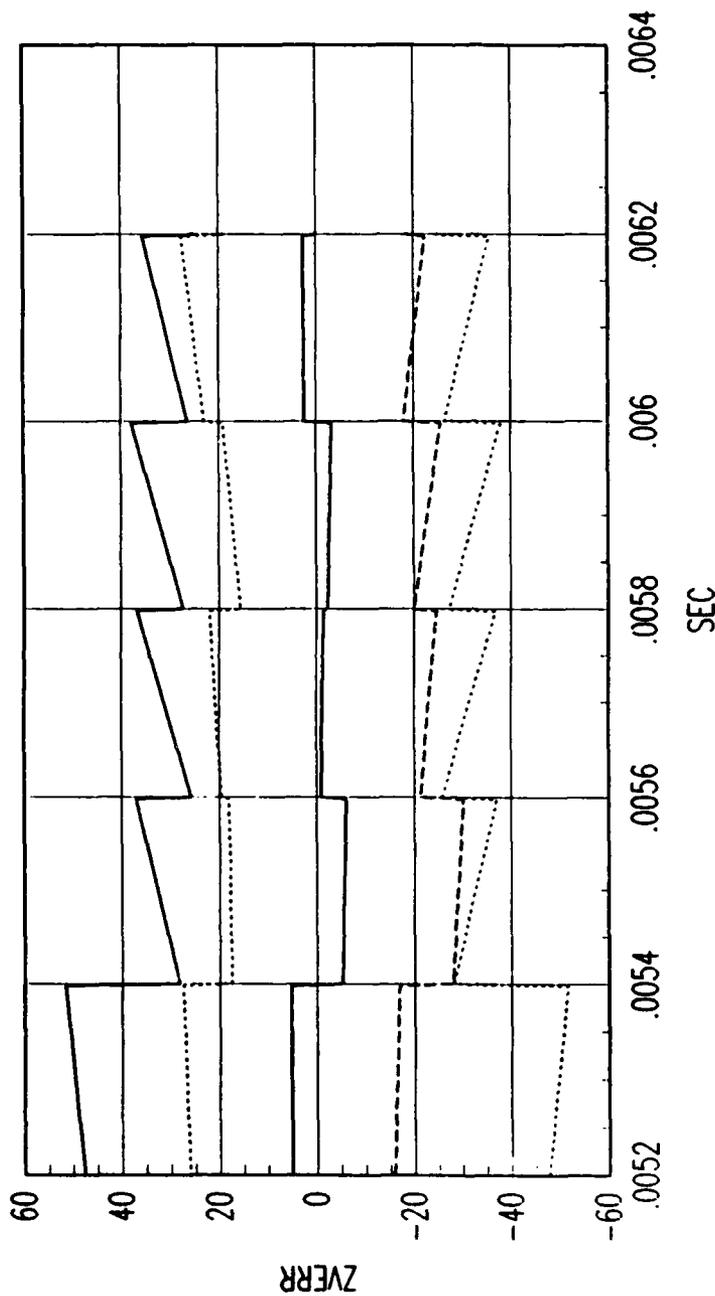
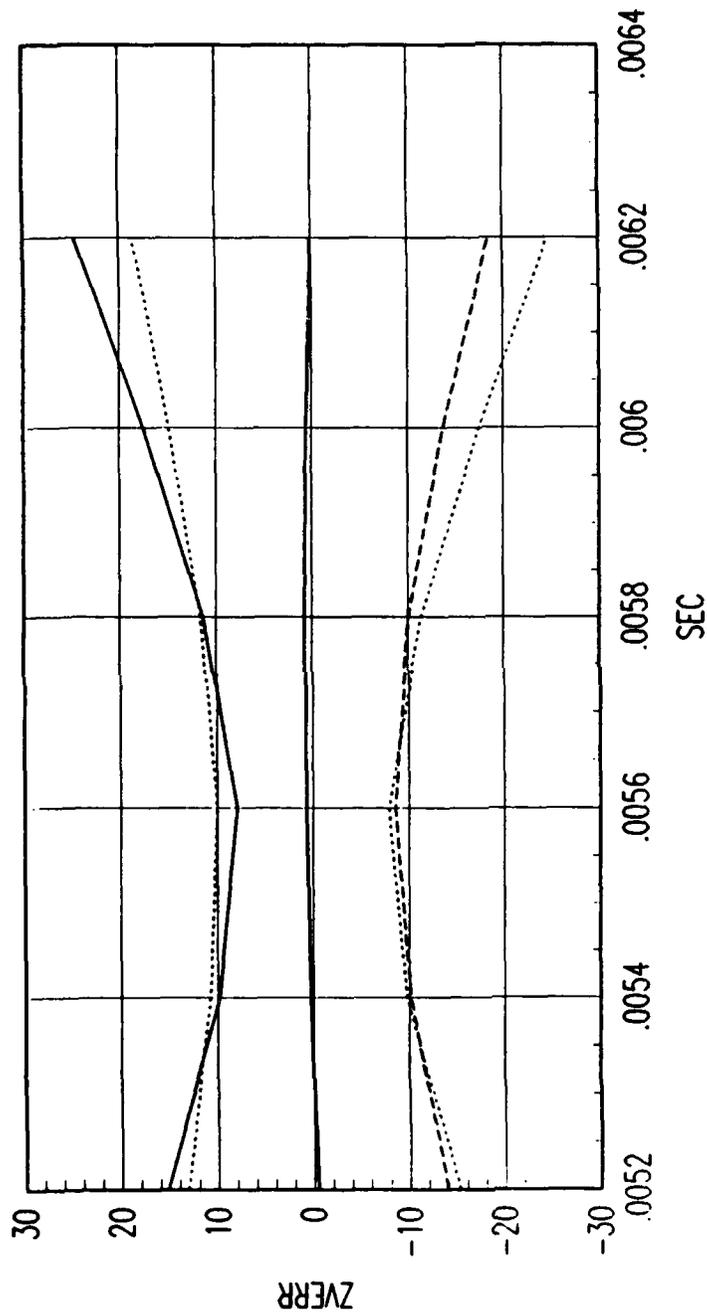


FIG. A FORWARD TIME EKf VELOCITY ERROR

Figure C.177. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.178. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

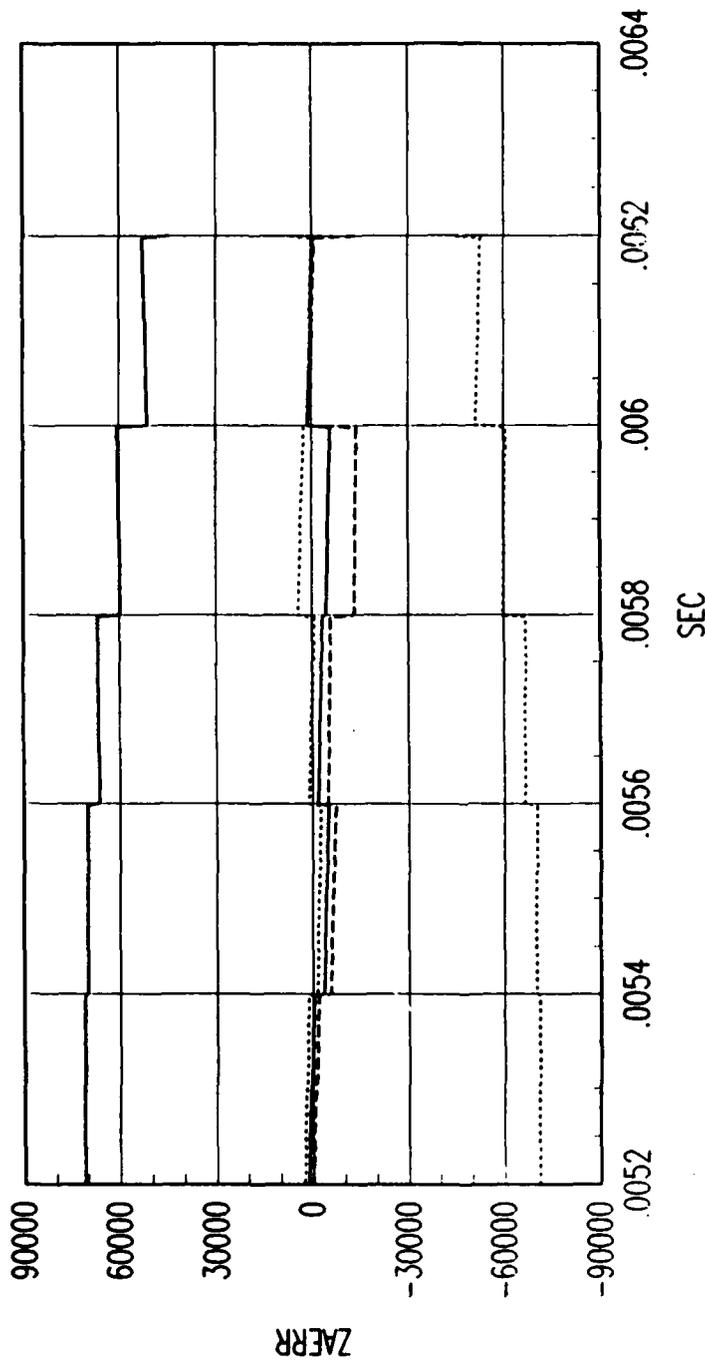


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.179. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

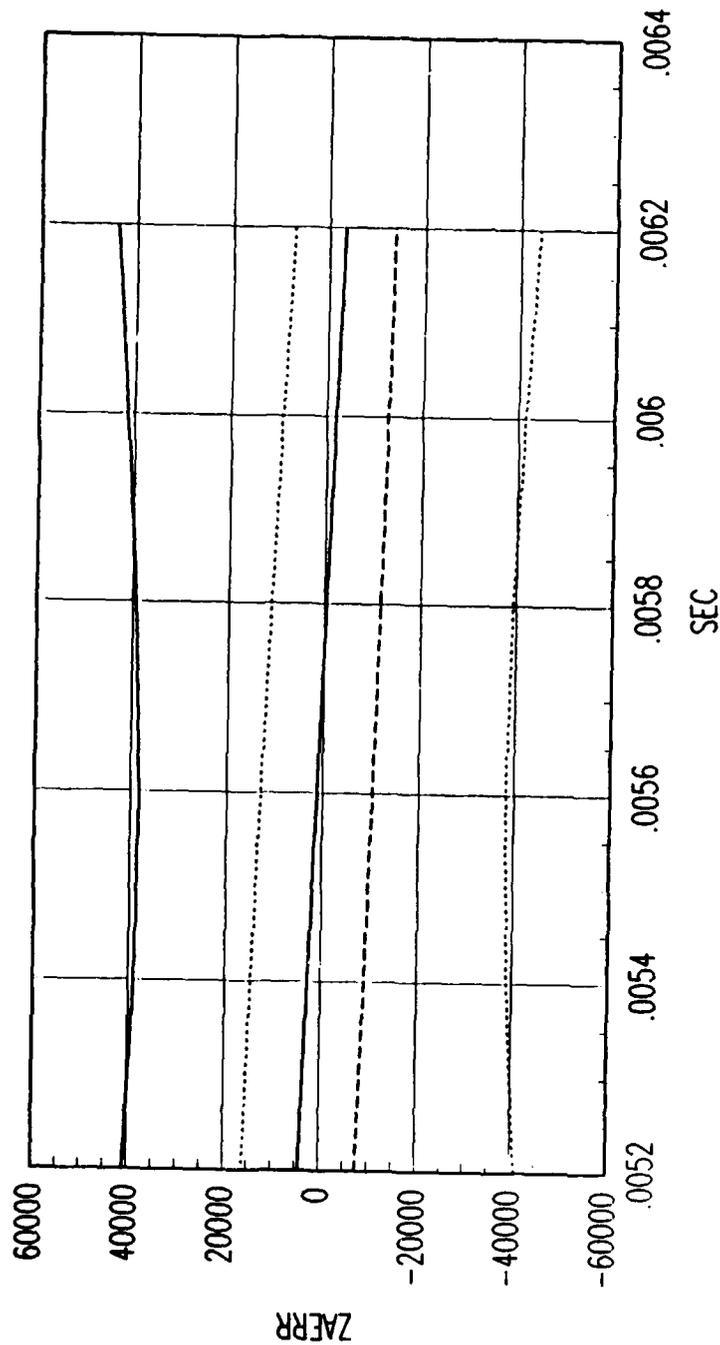


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.180. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

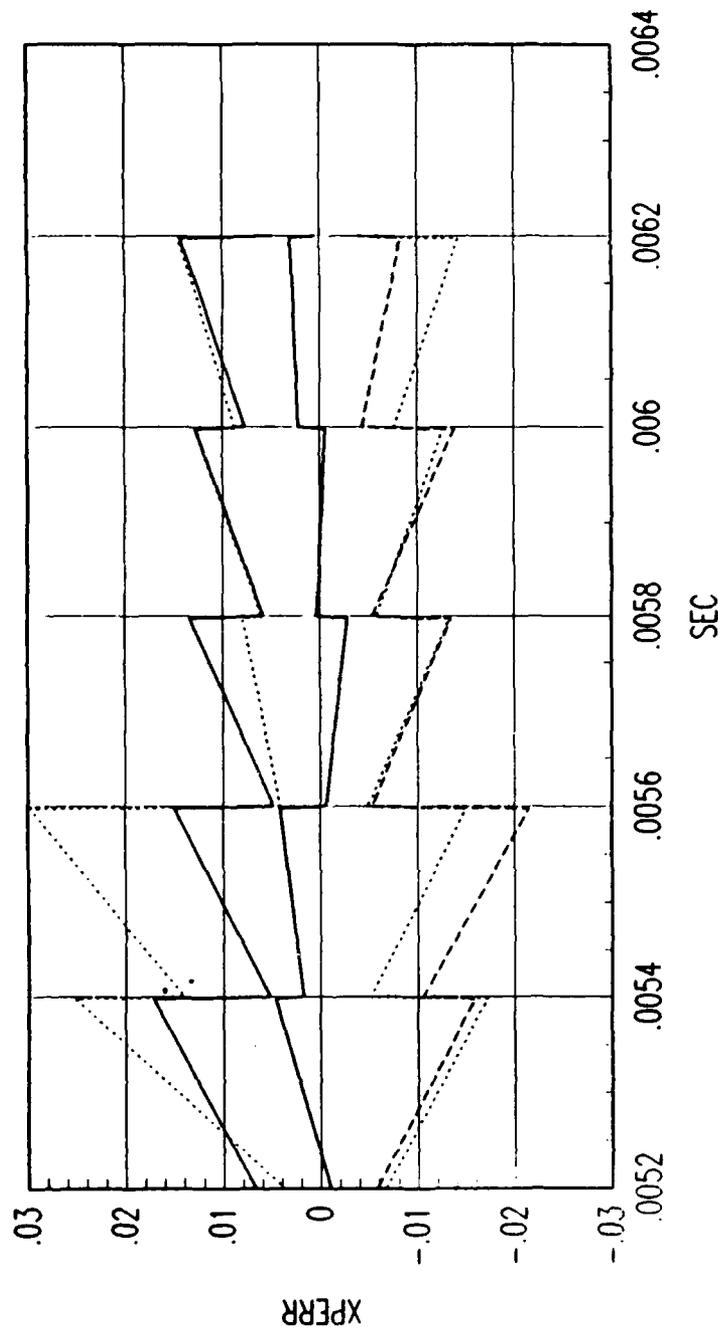


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.181. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

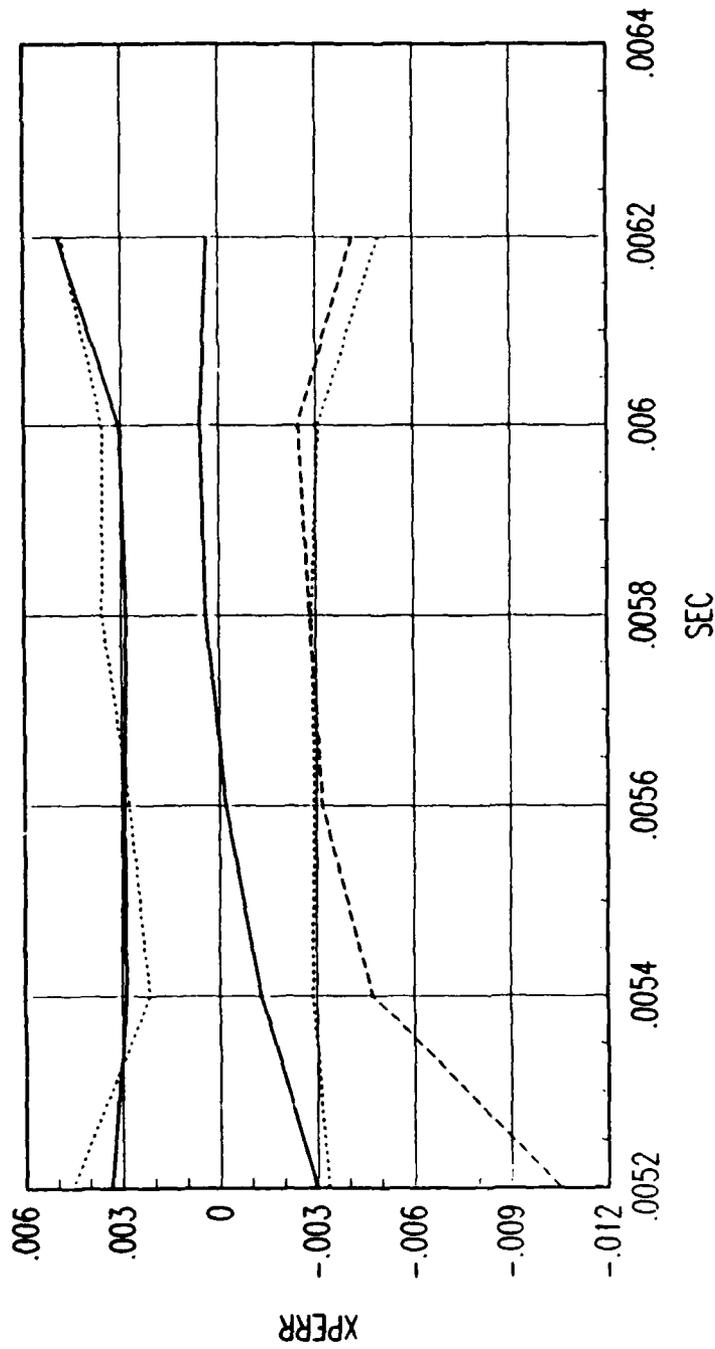


FIG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.182. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

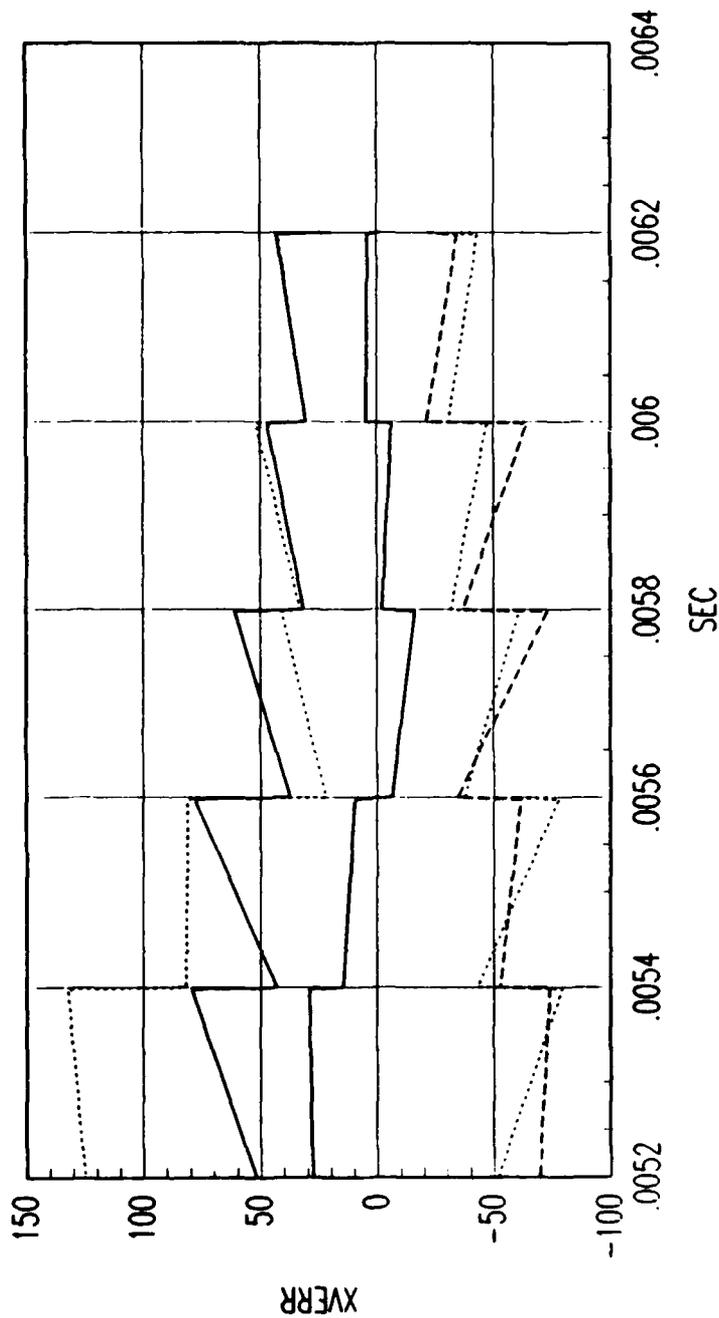


FIG. B FORWARD TIME VELOCITY ERROR

Figure C.183. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

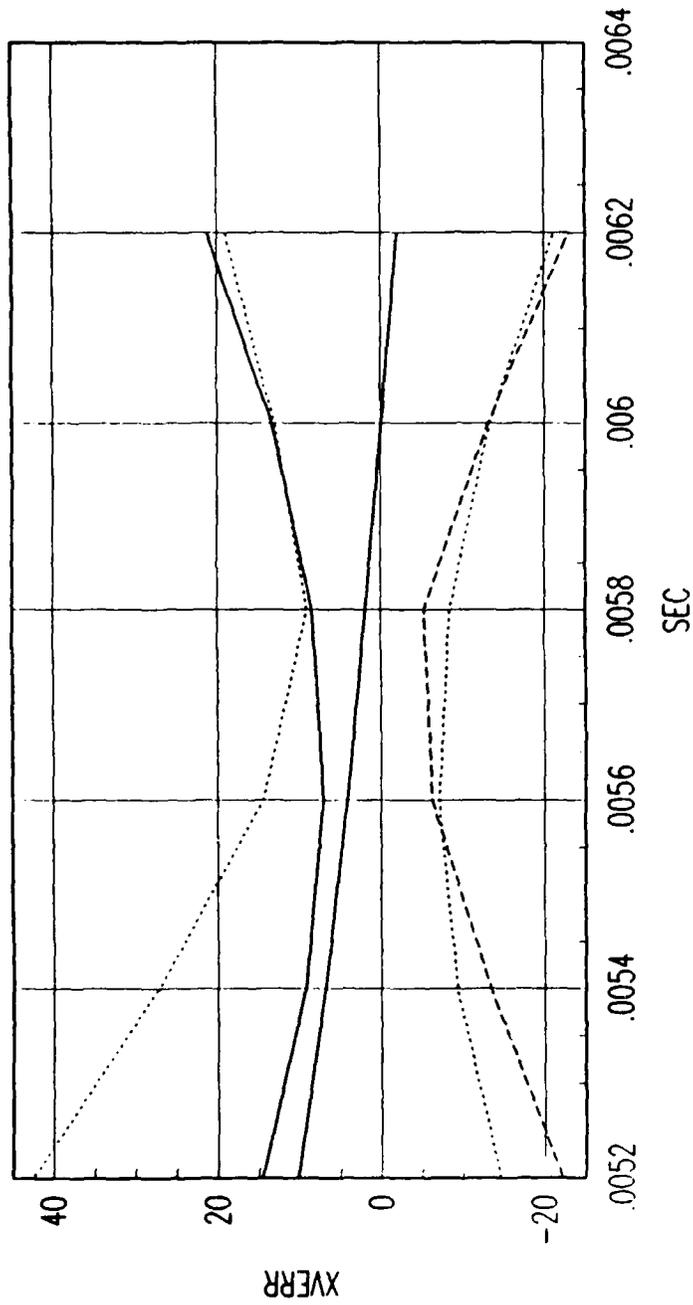
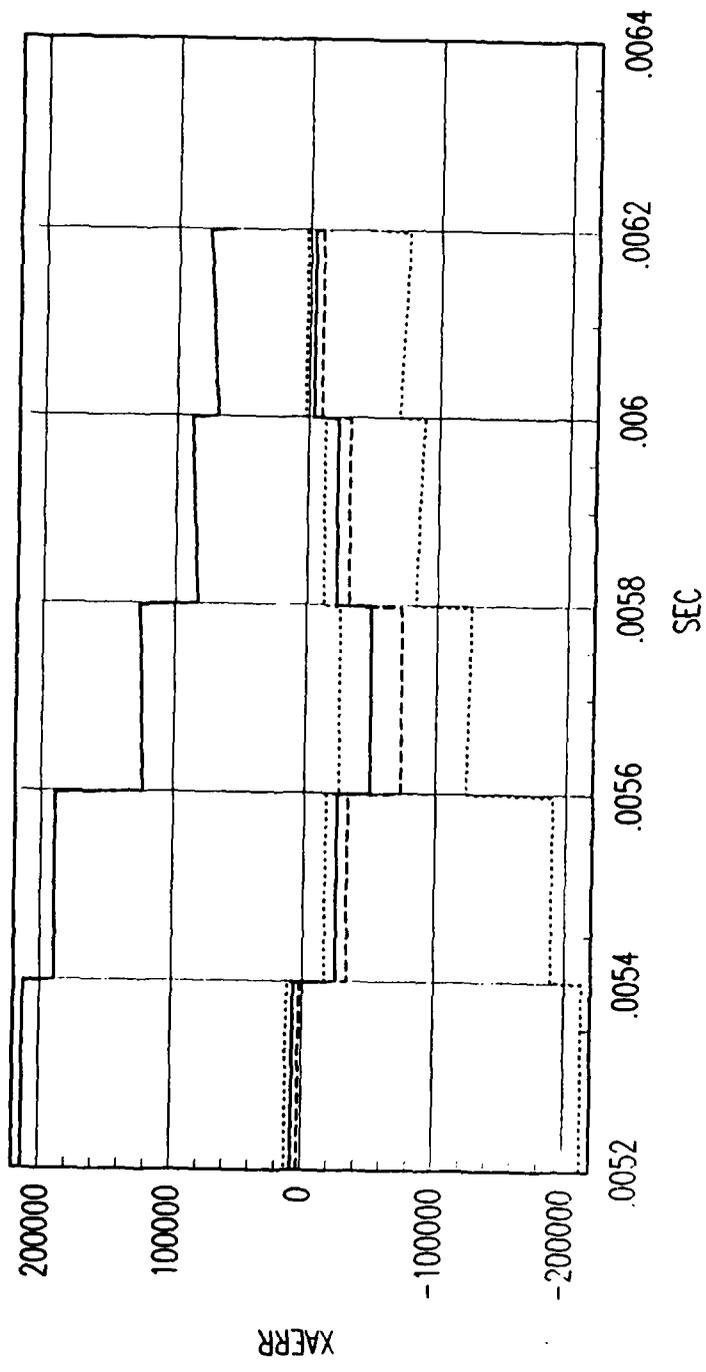


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.184. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.185. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

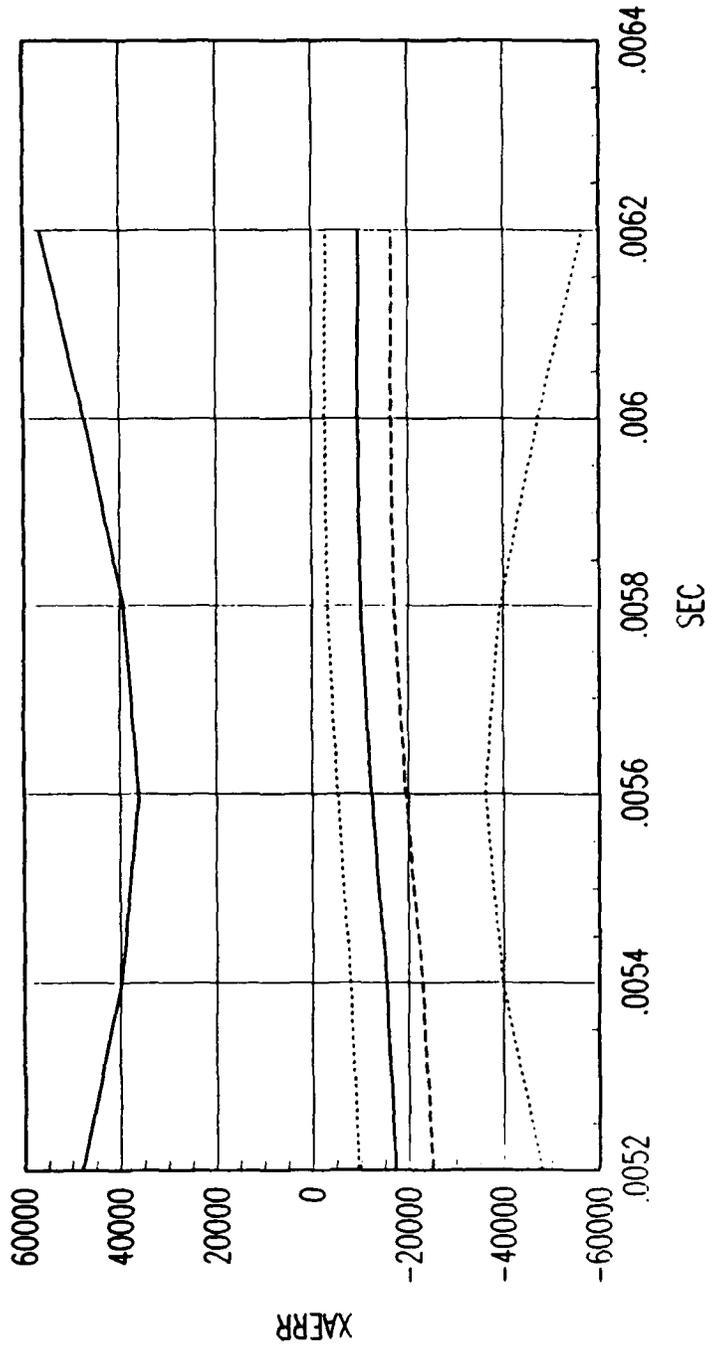
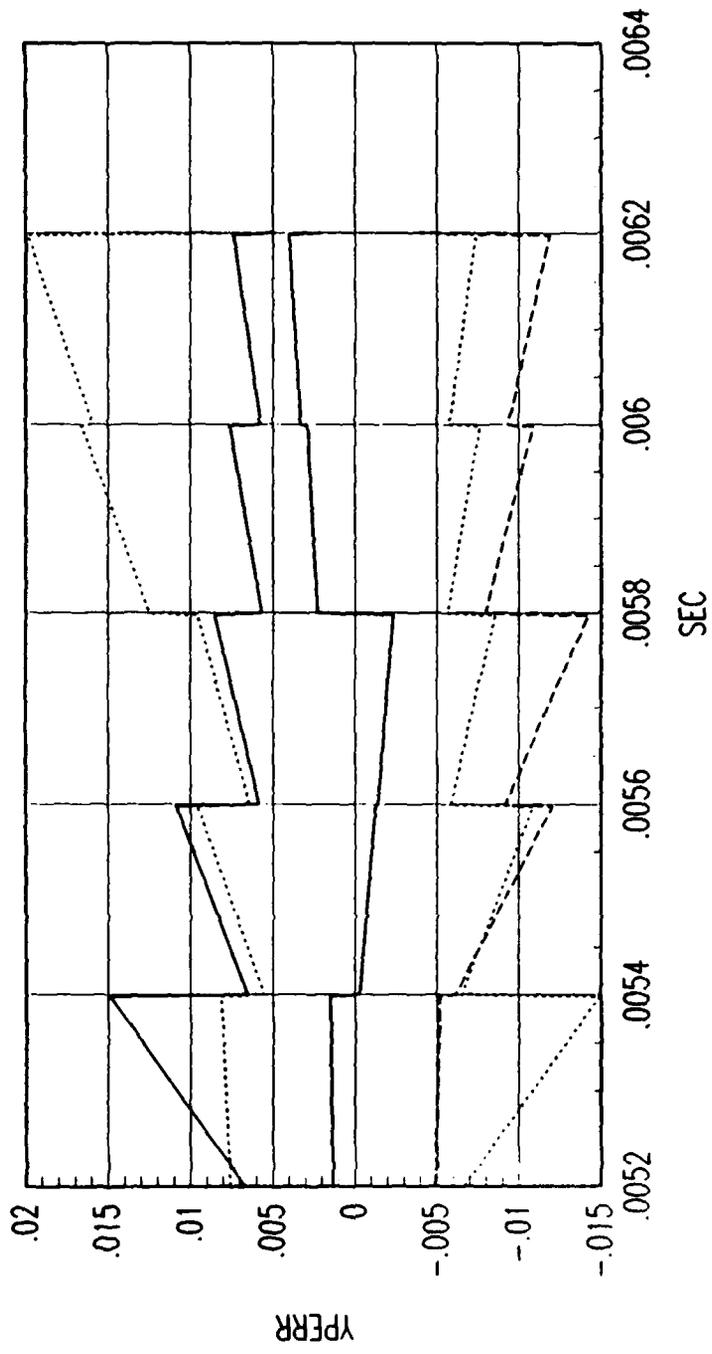


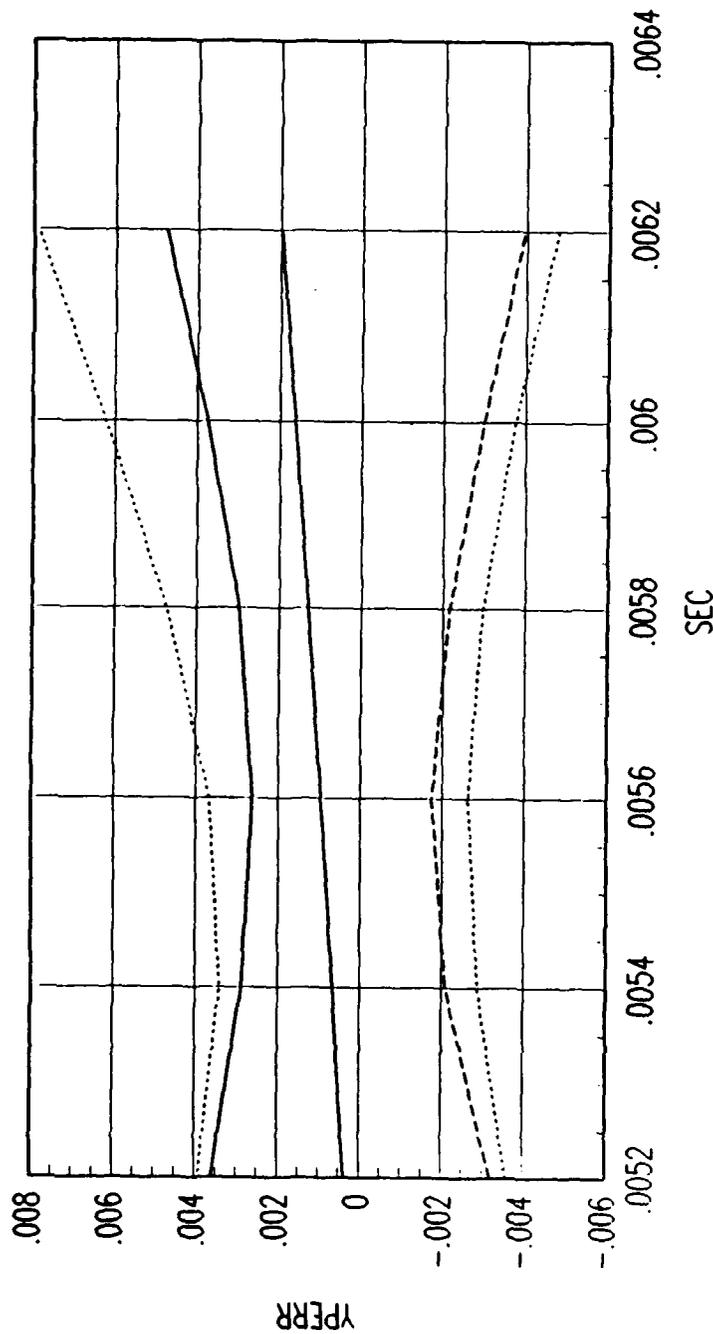
FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.186. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF POSITION ERROR

Figure C.187. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.188. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

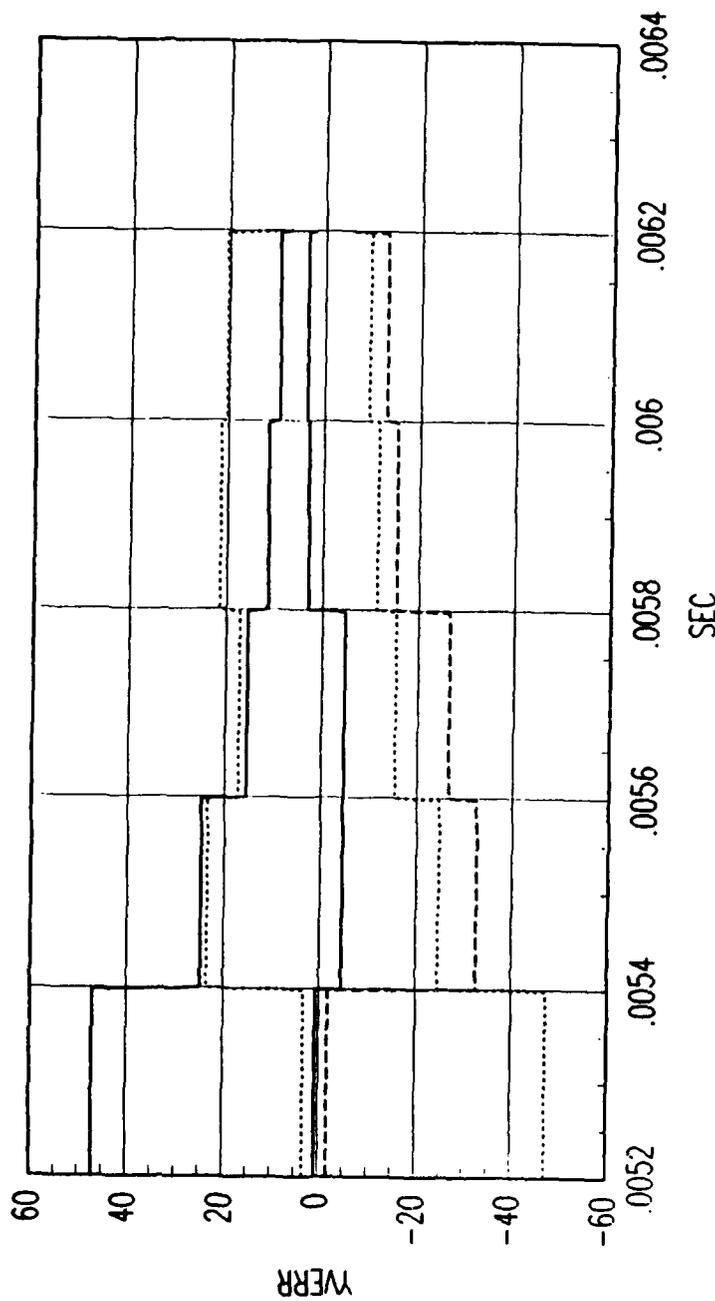
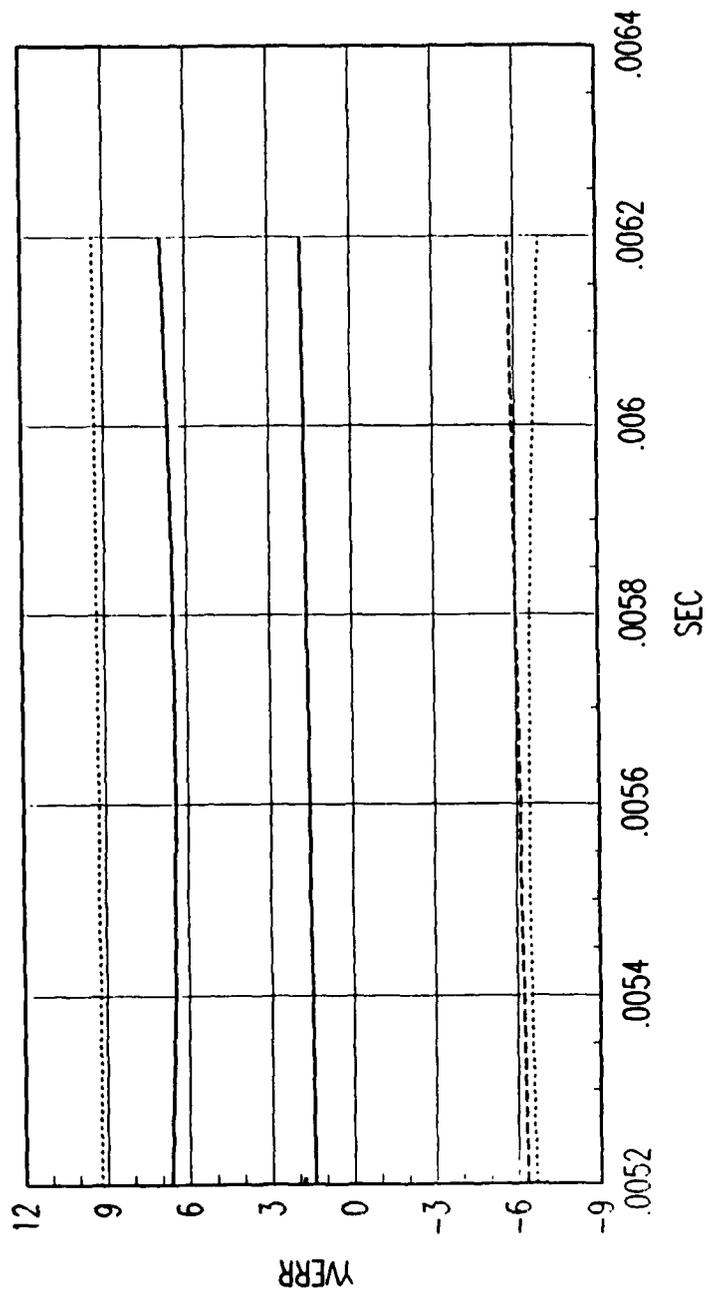


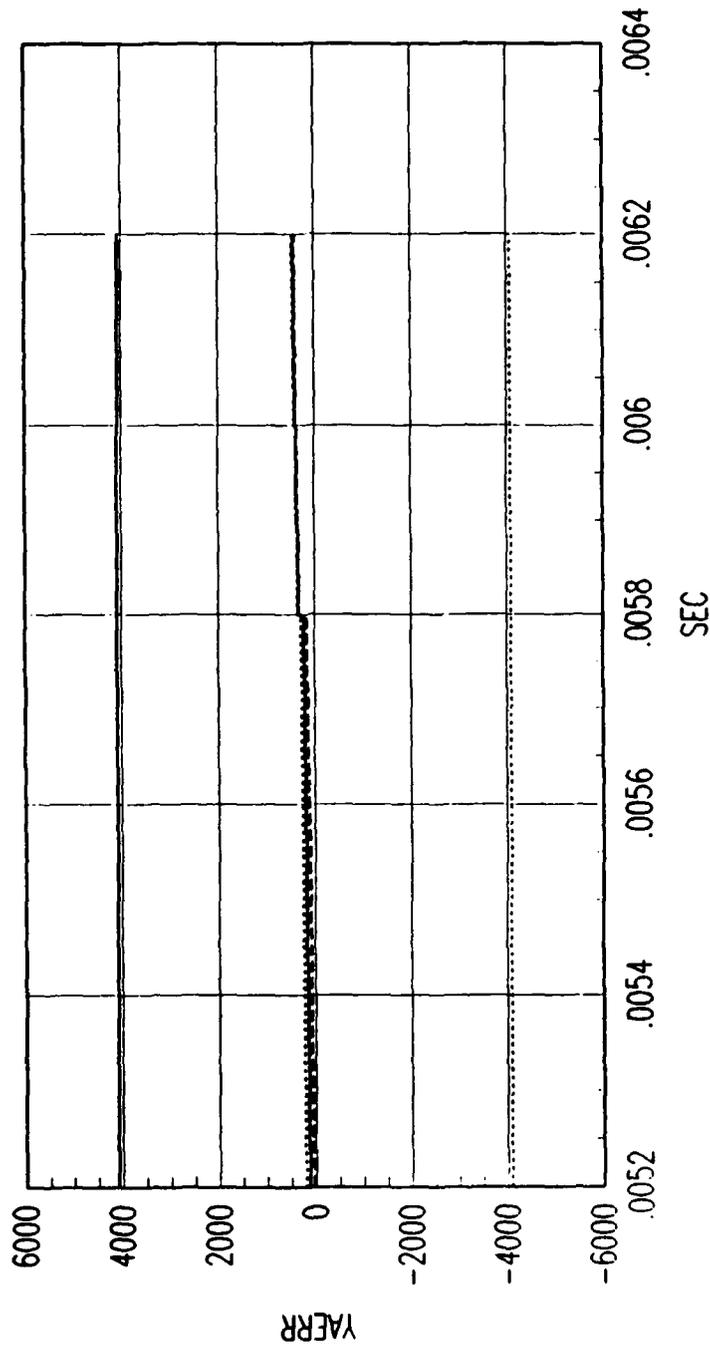
FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.189. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



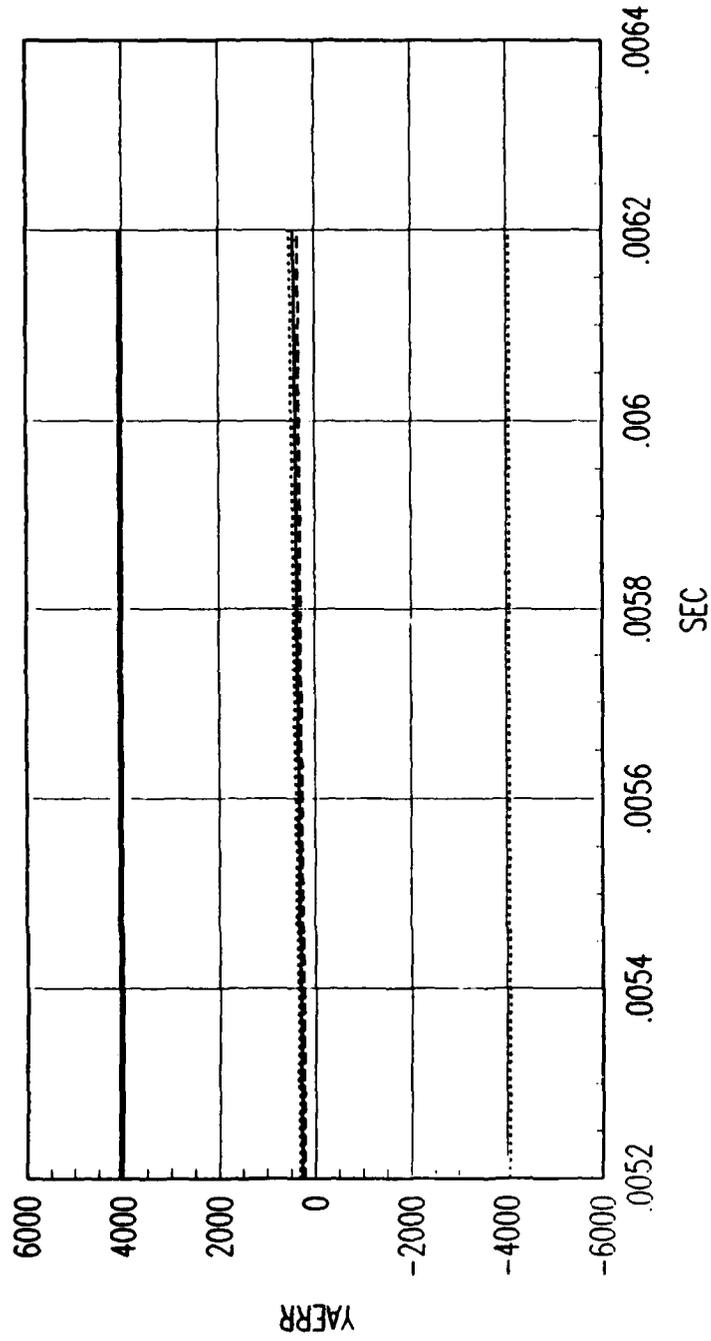
FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.190. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.191. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.192. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

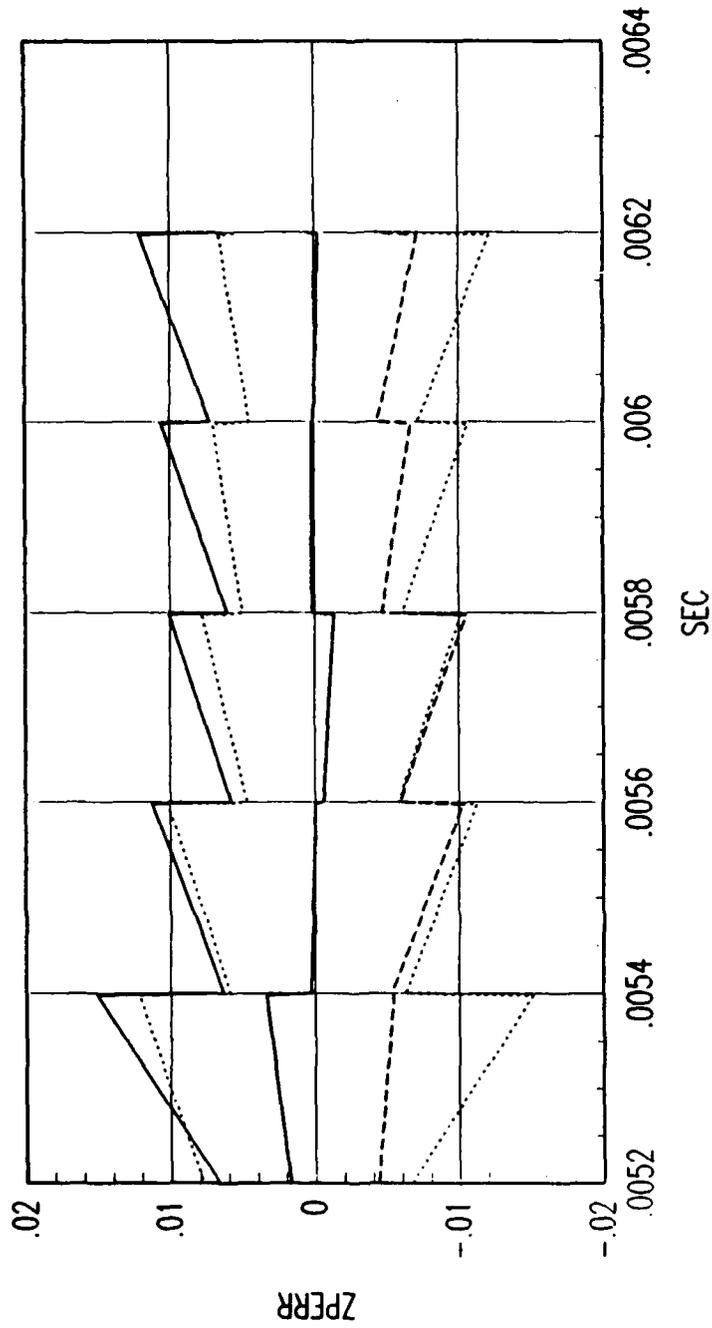
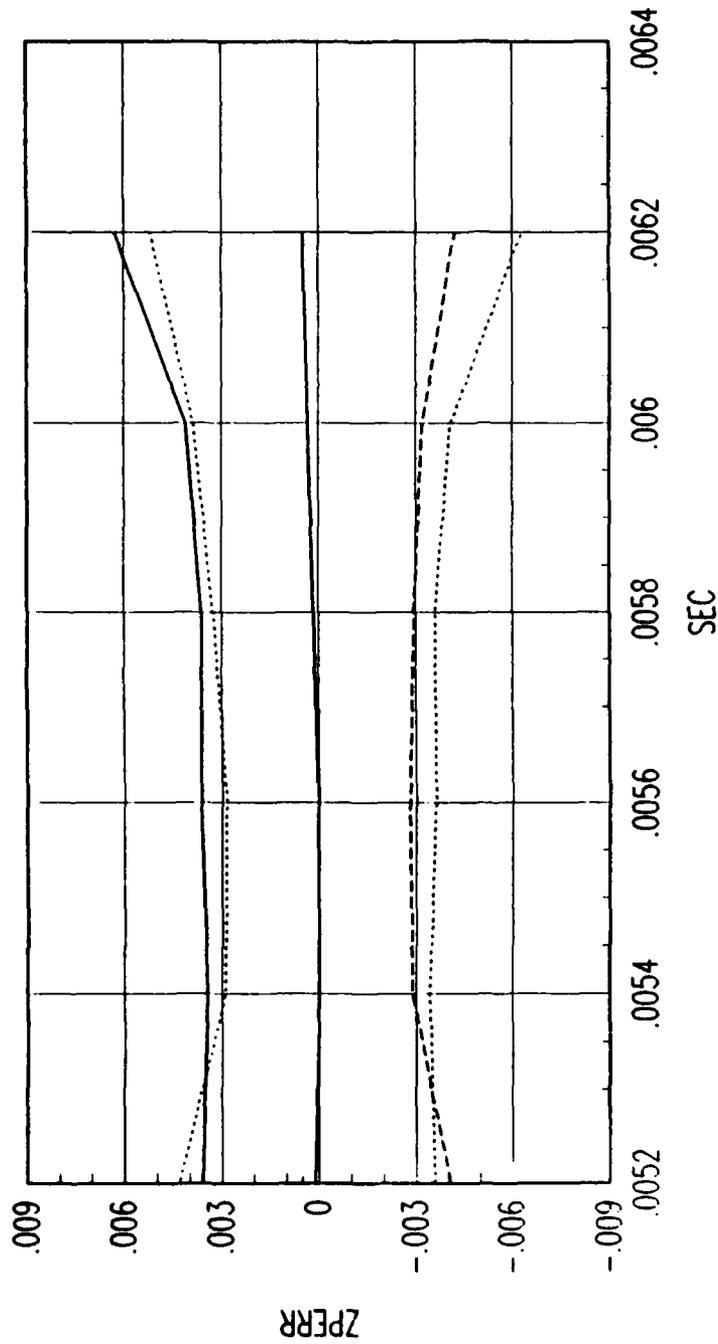


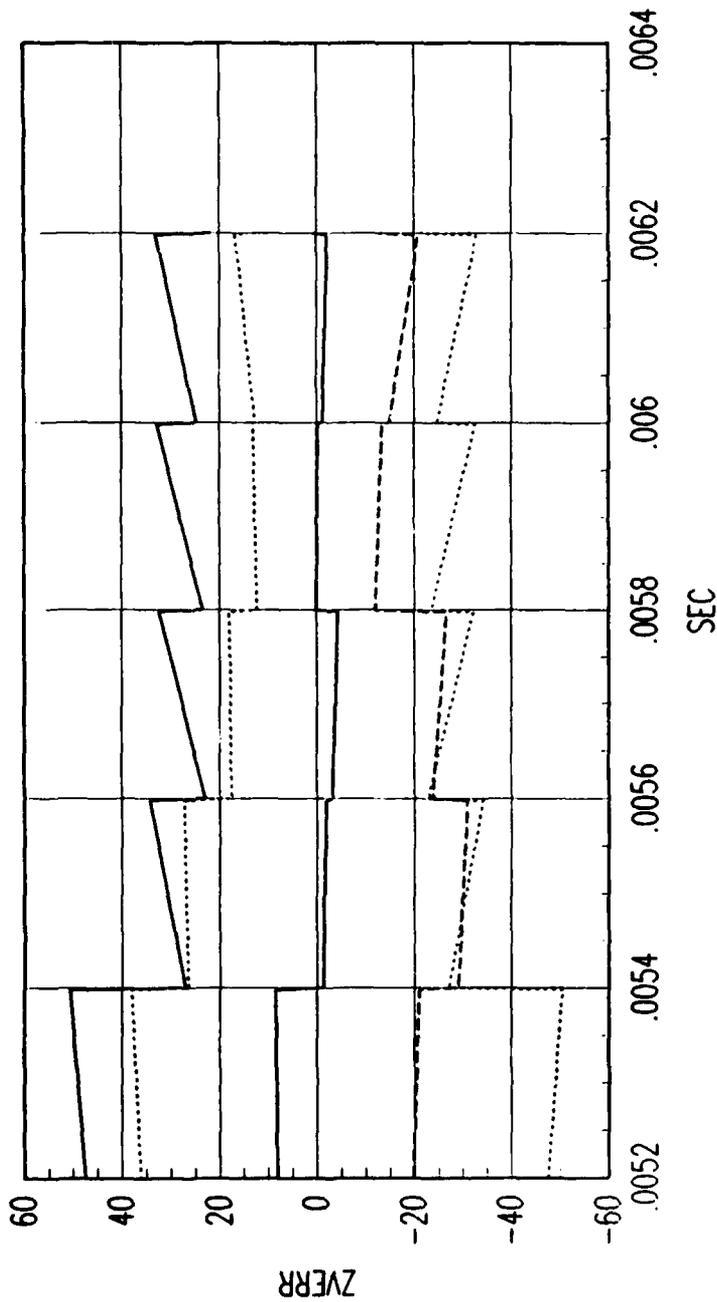
FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.193. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.194. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.195. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

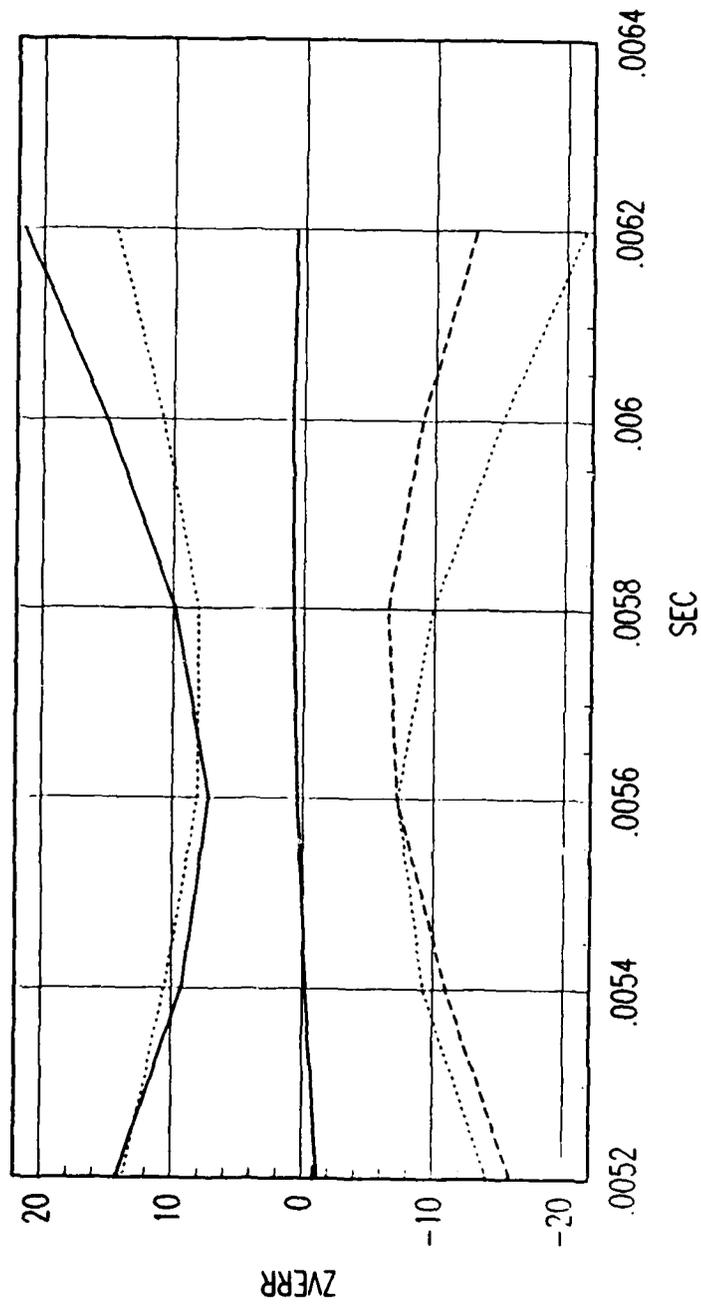


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.196. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

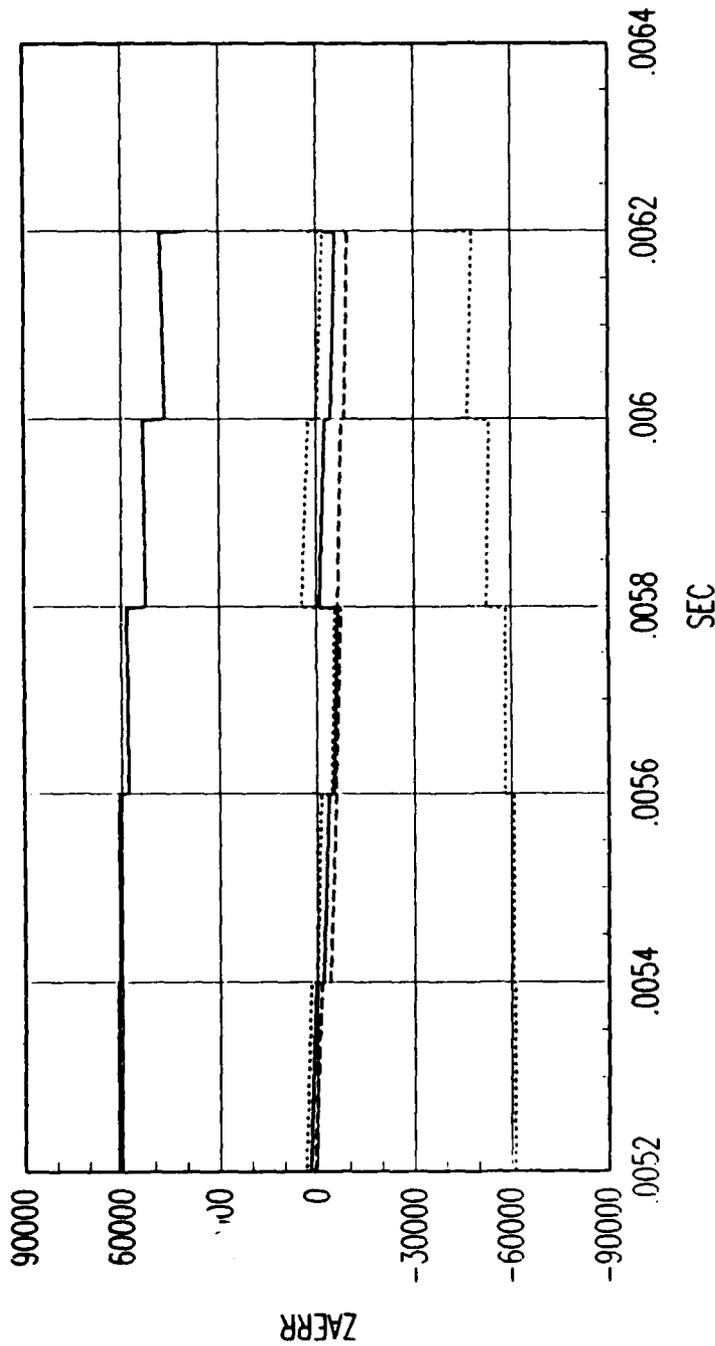


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.197. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

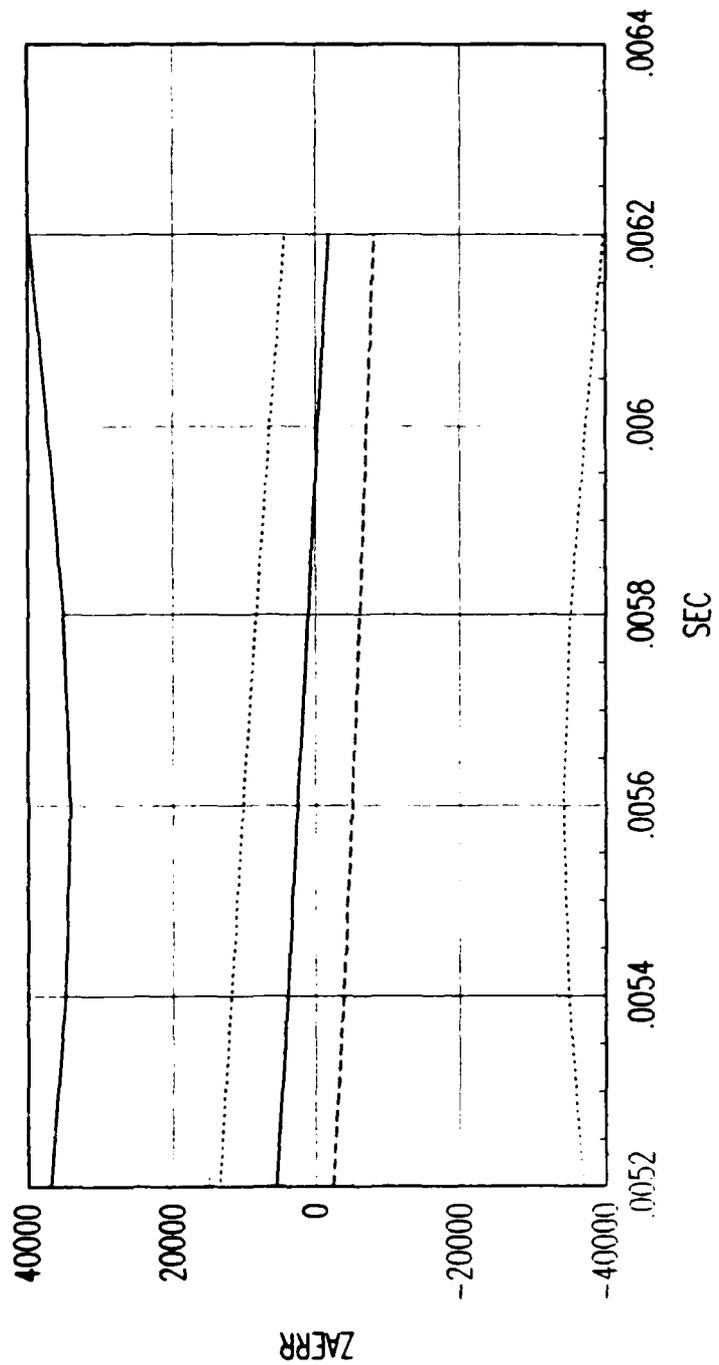


FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.198. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

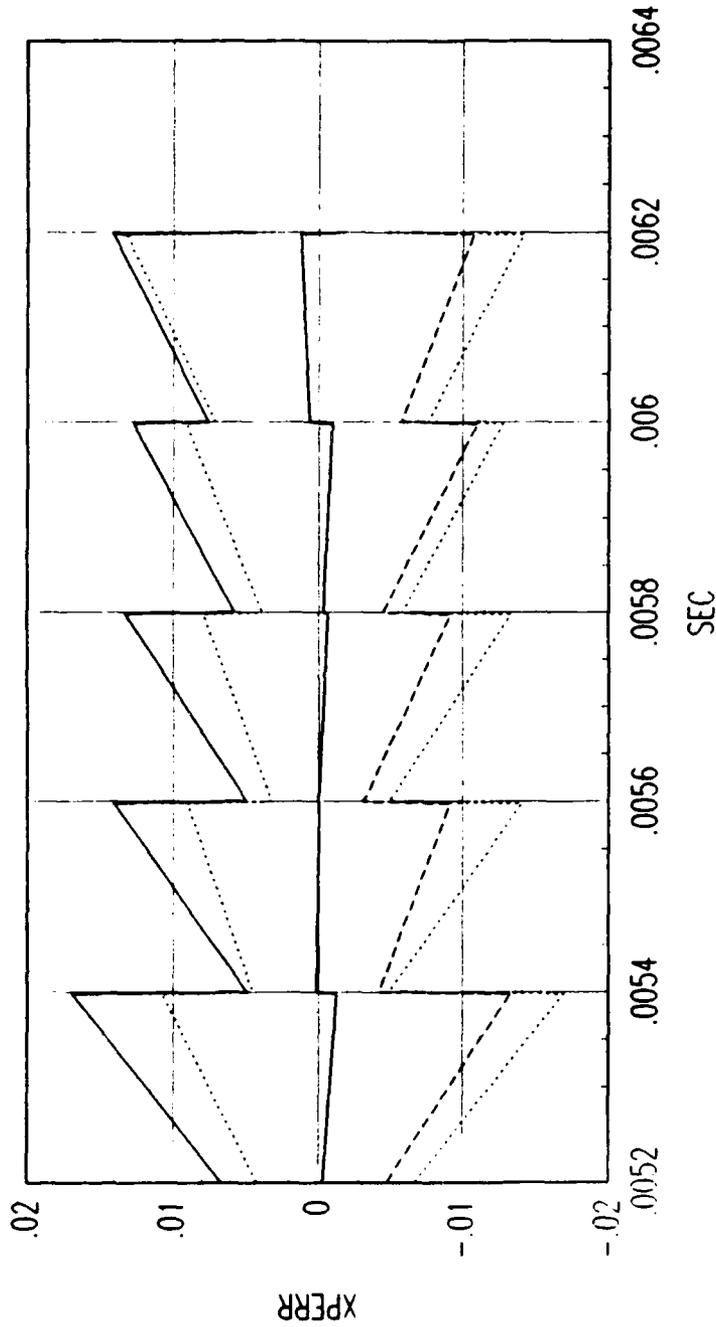
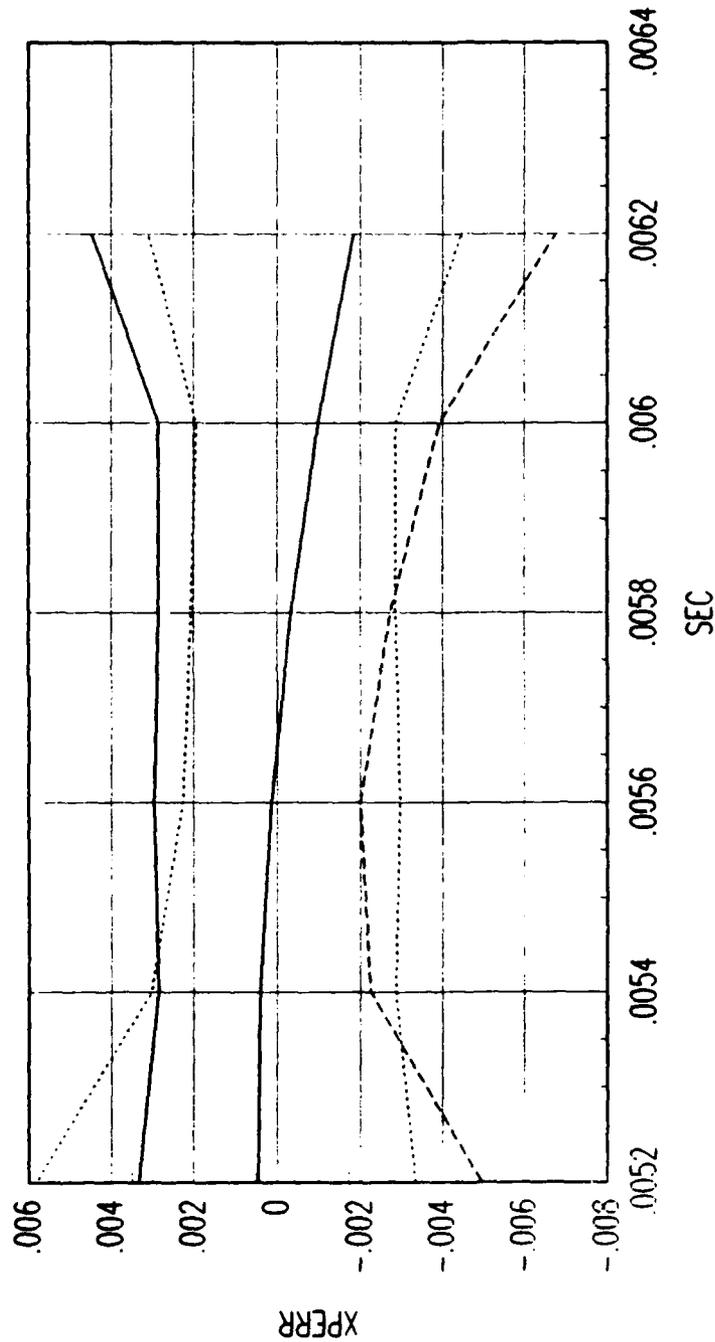


FIG. C FORWARD TIME EKF POSITION ERROR

Figure C.199. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.200. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

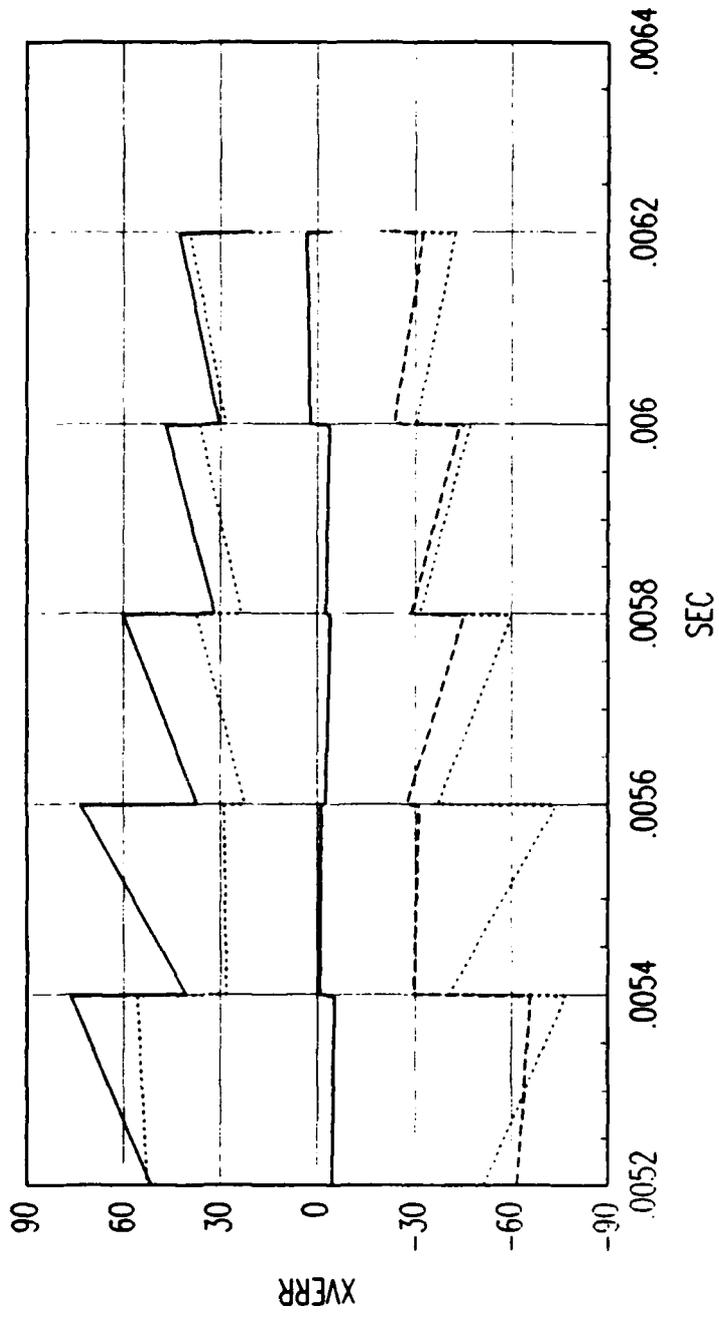
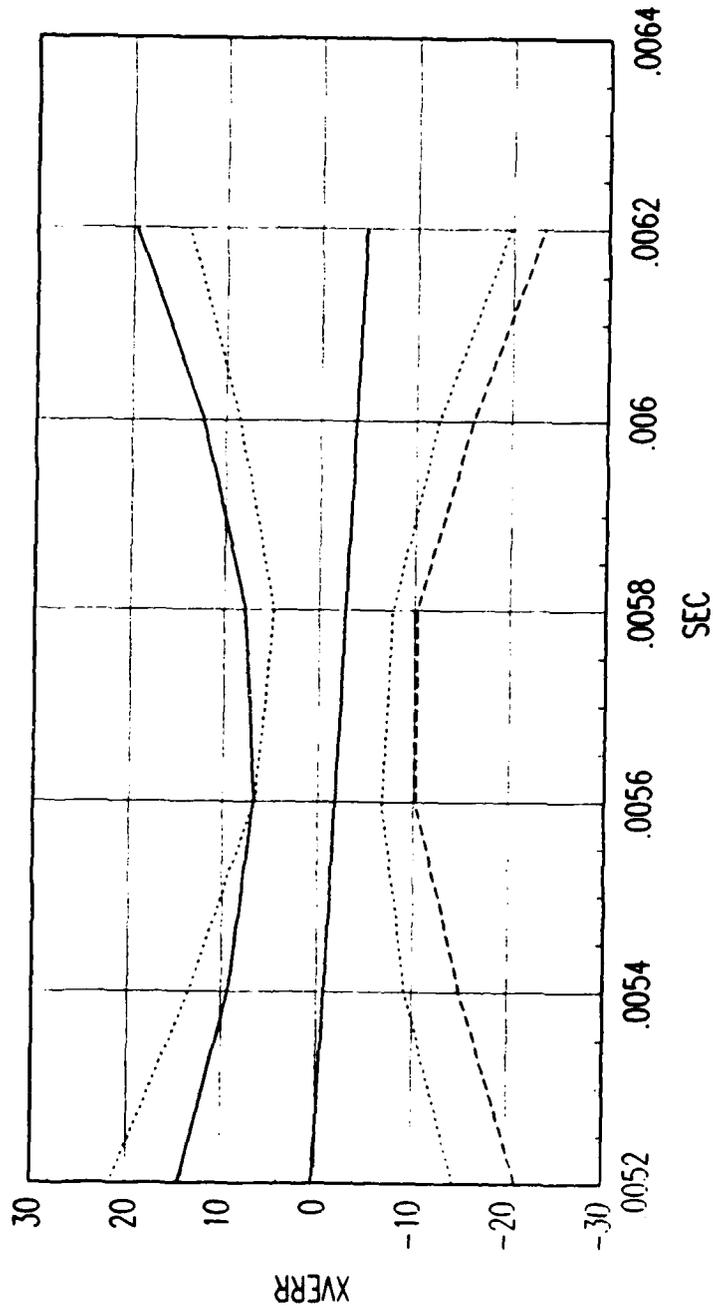


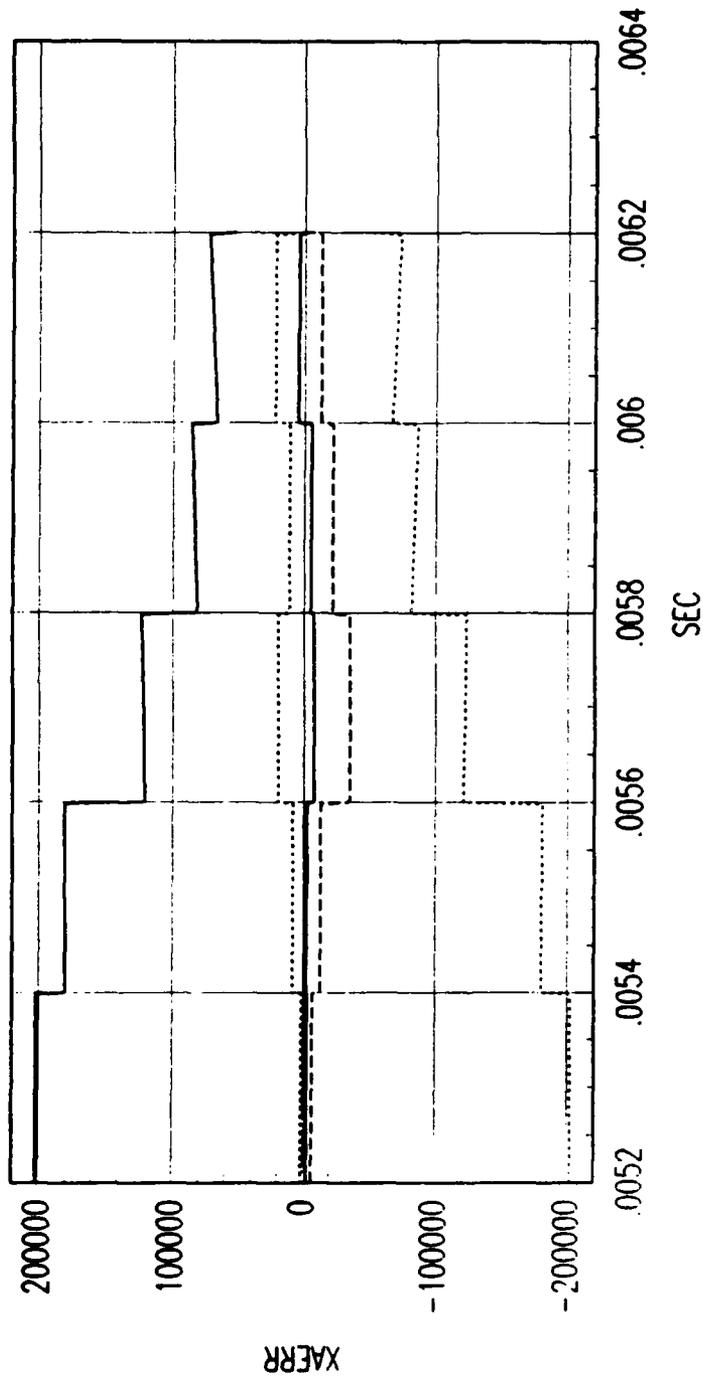
FIG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.201. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



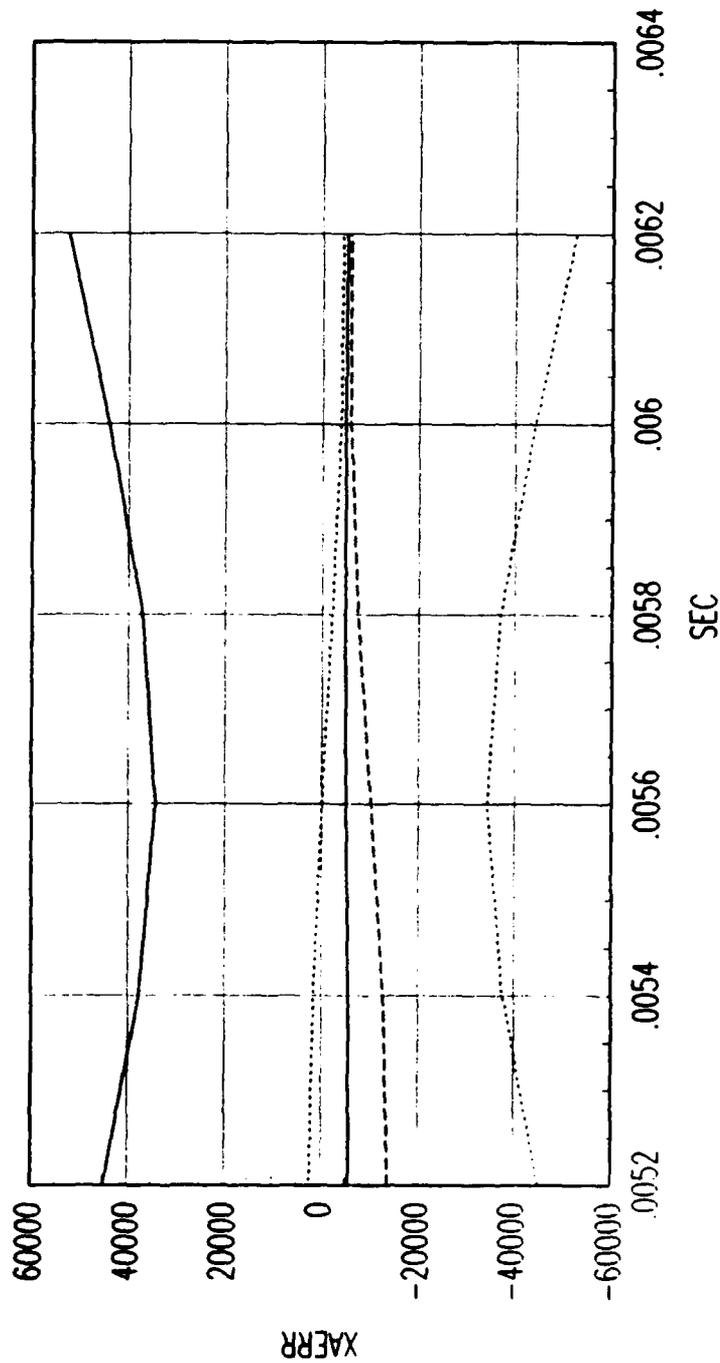
FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.202. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.203. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.204. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

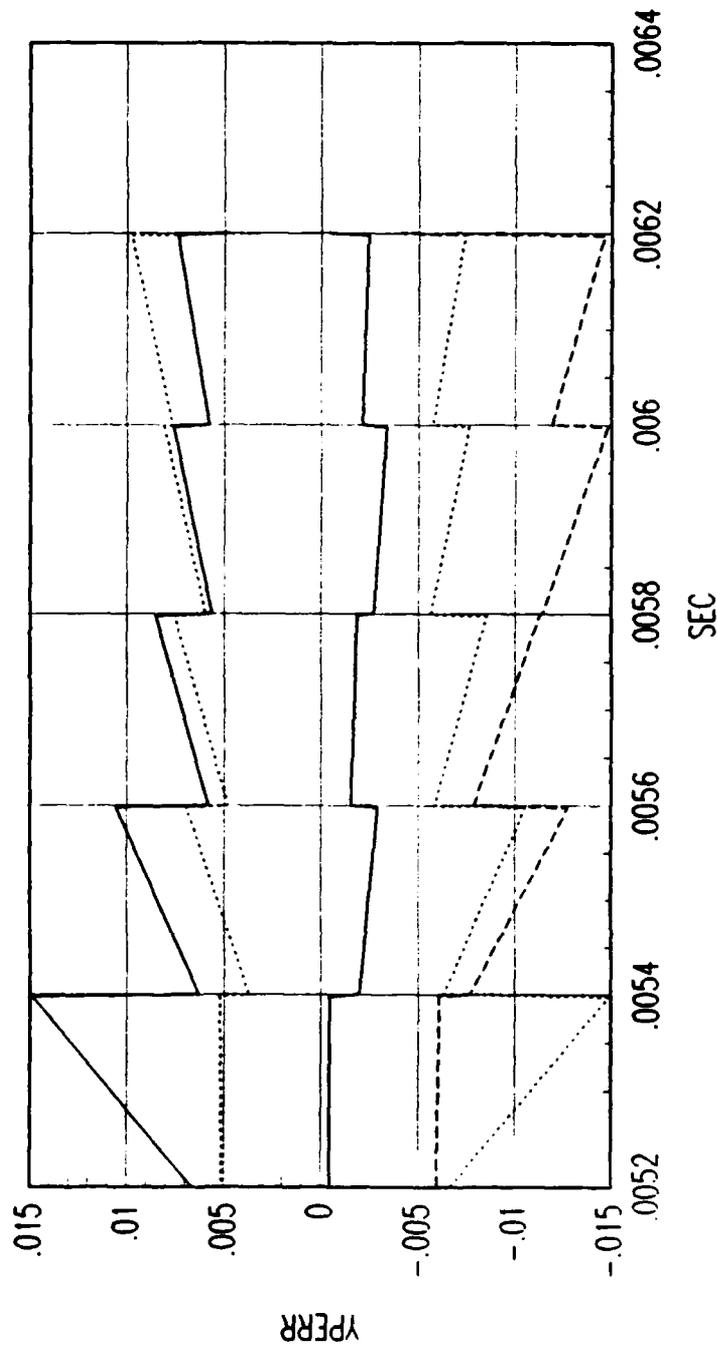


FIG. C FORWARD TIME EKF POSITION ERROR

Figure C.205. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

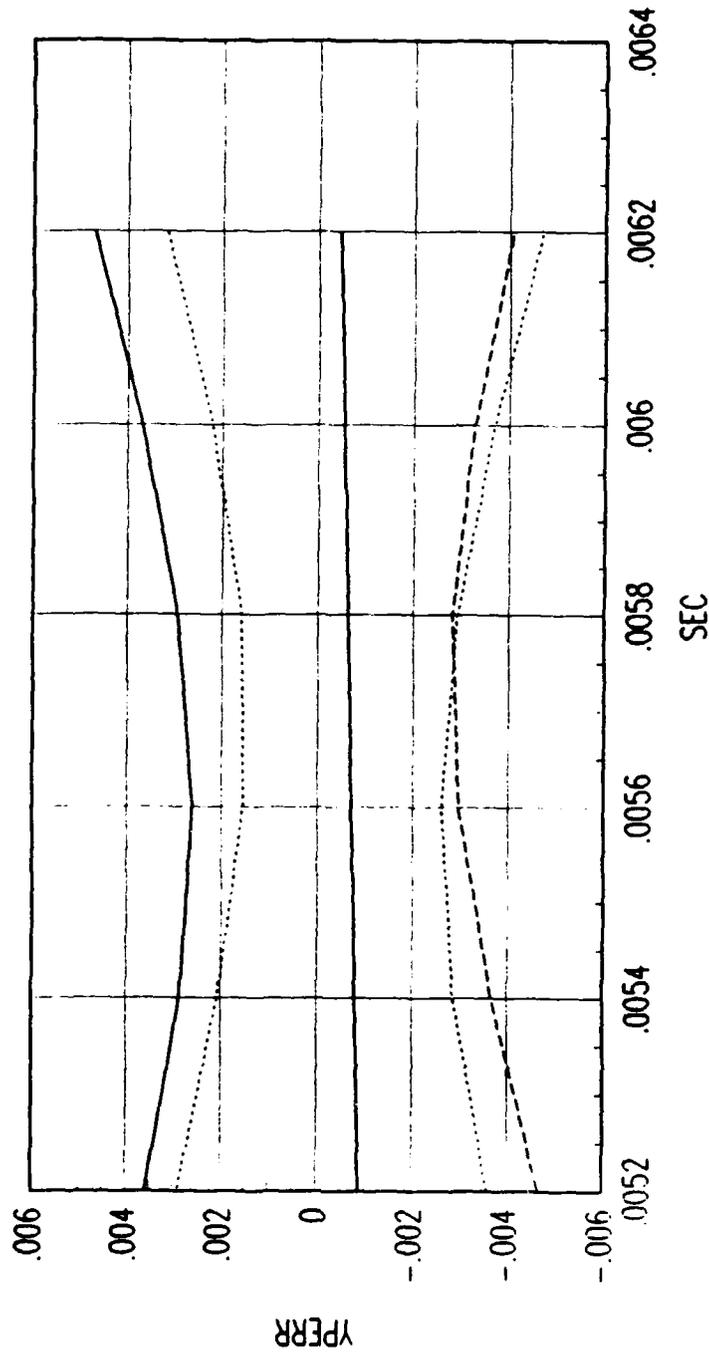
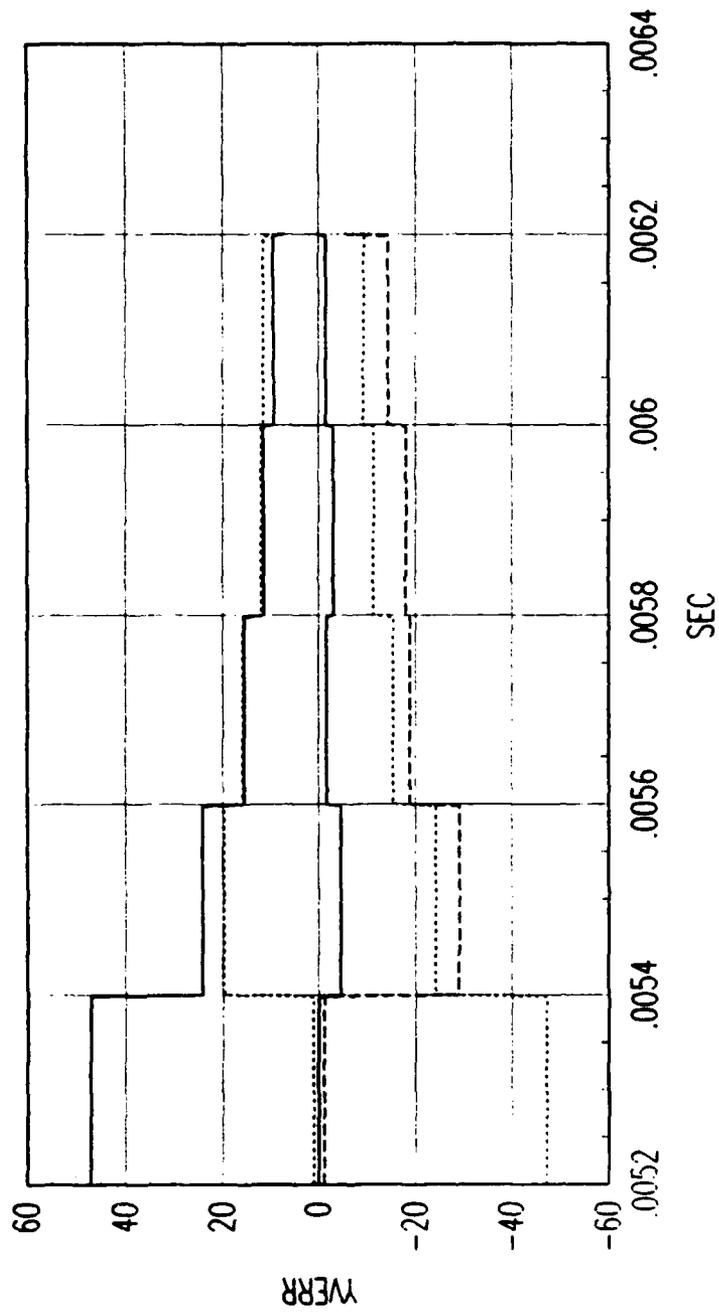


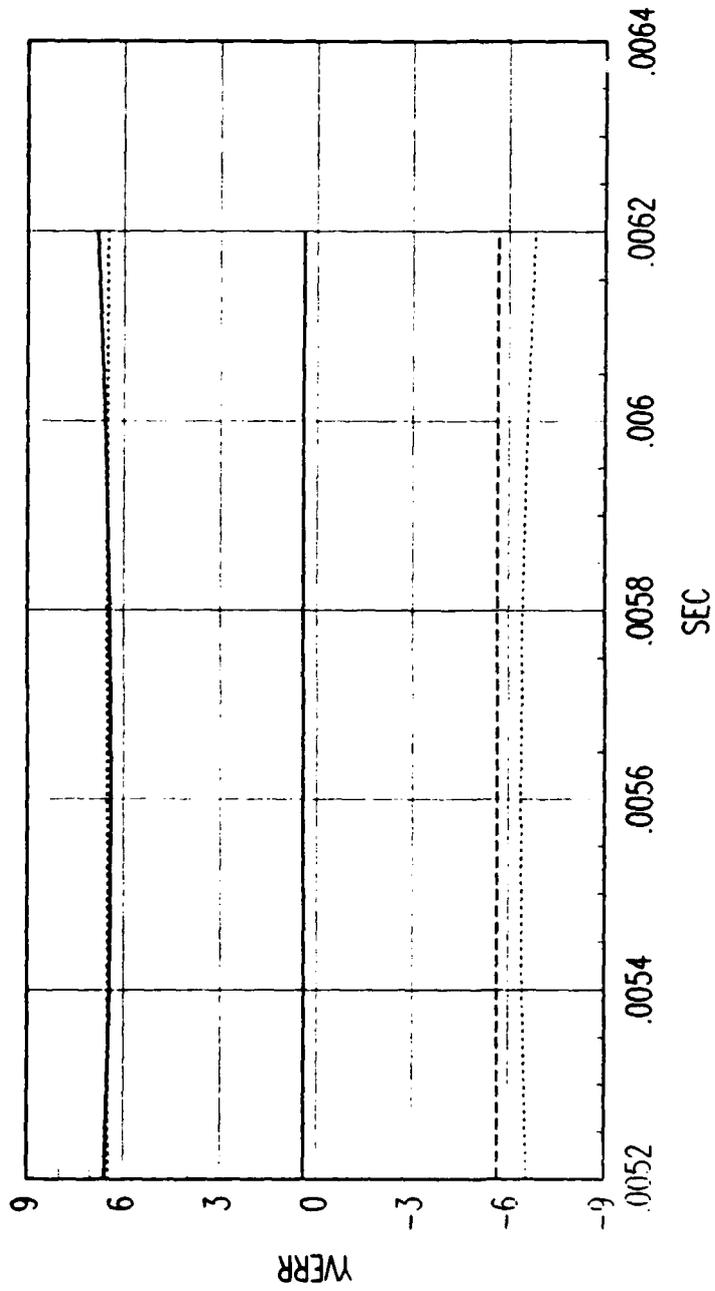
FIG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.206. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



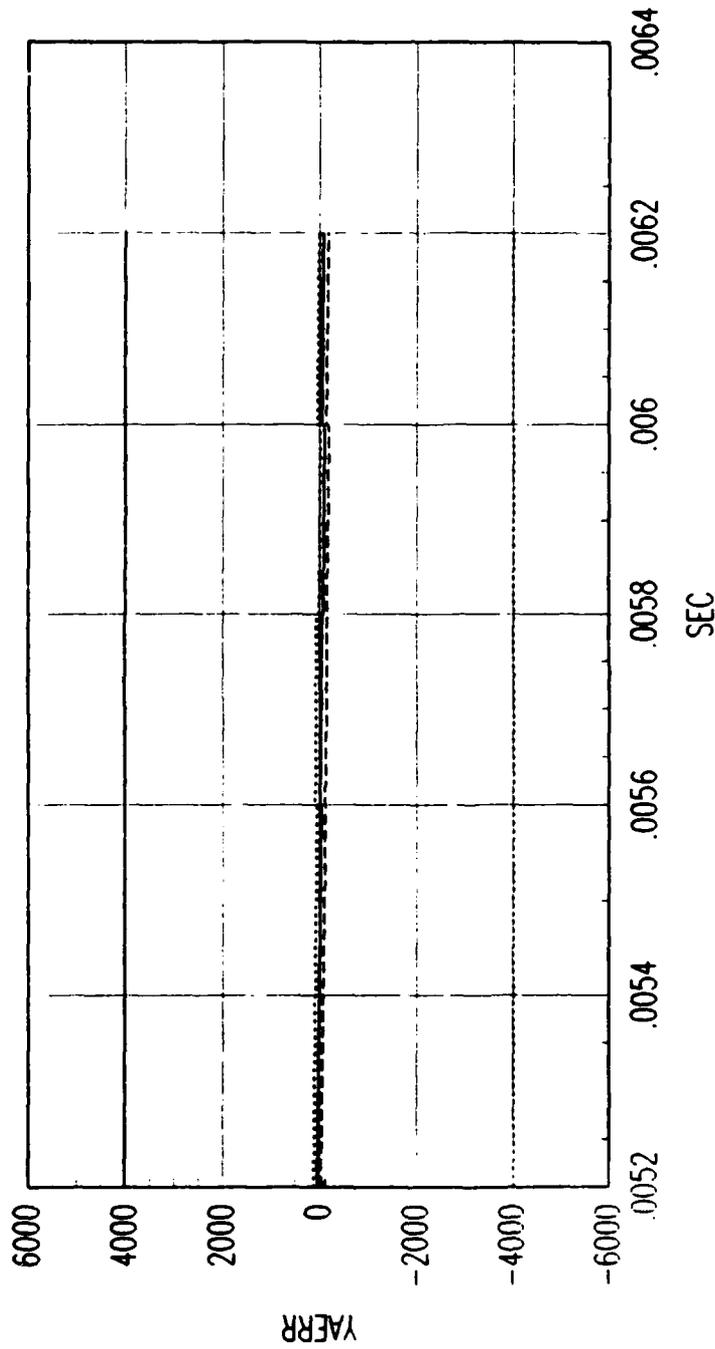
FRG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.207. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



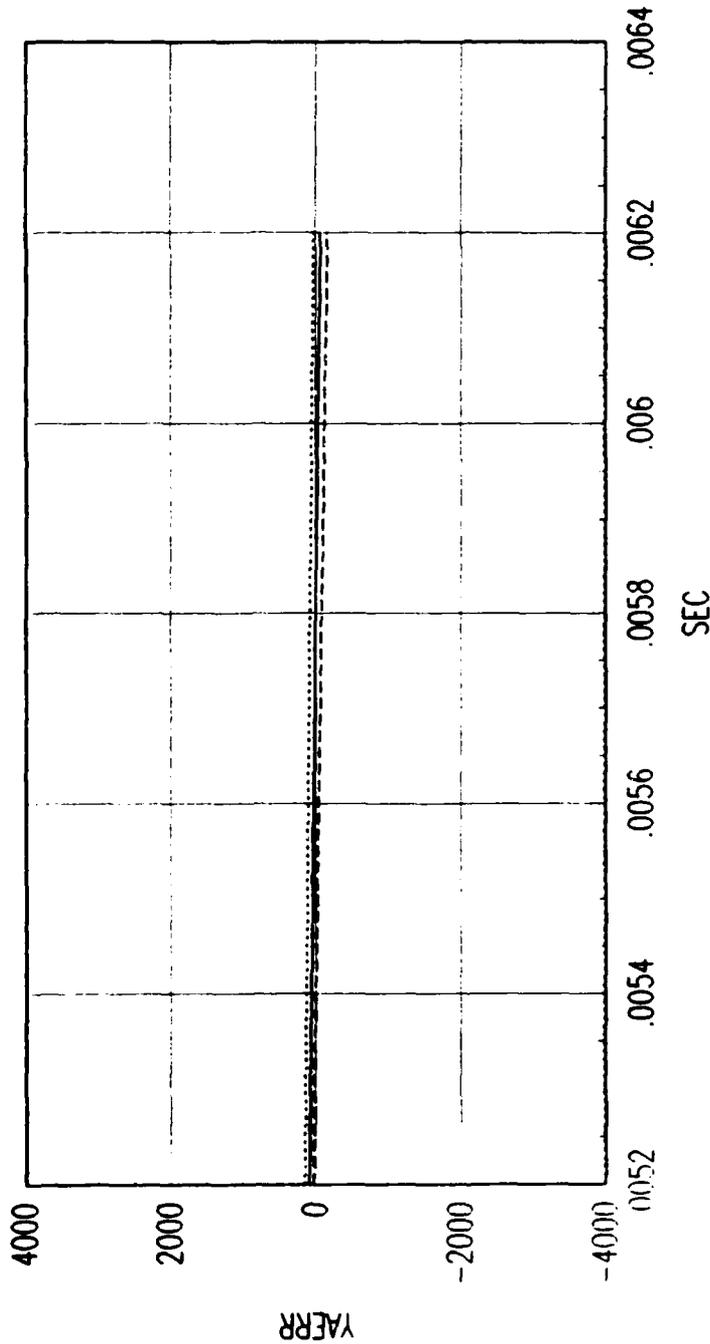
FRG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.208. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.209. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.210. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

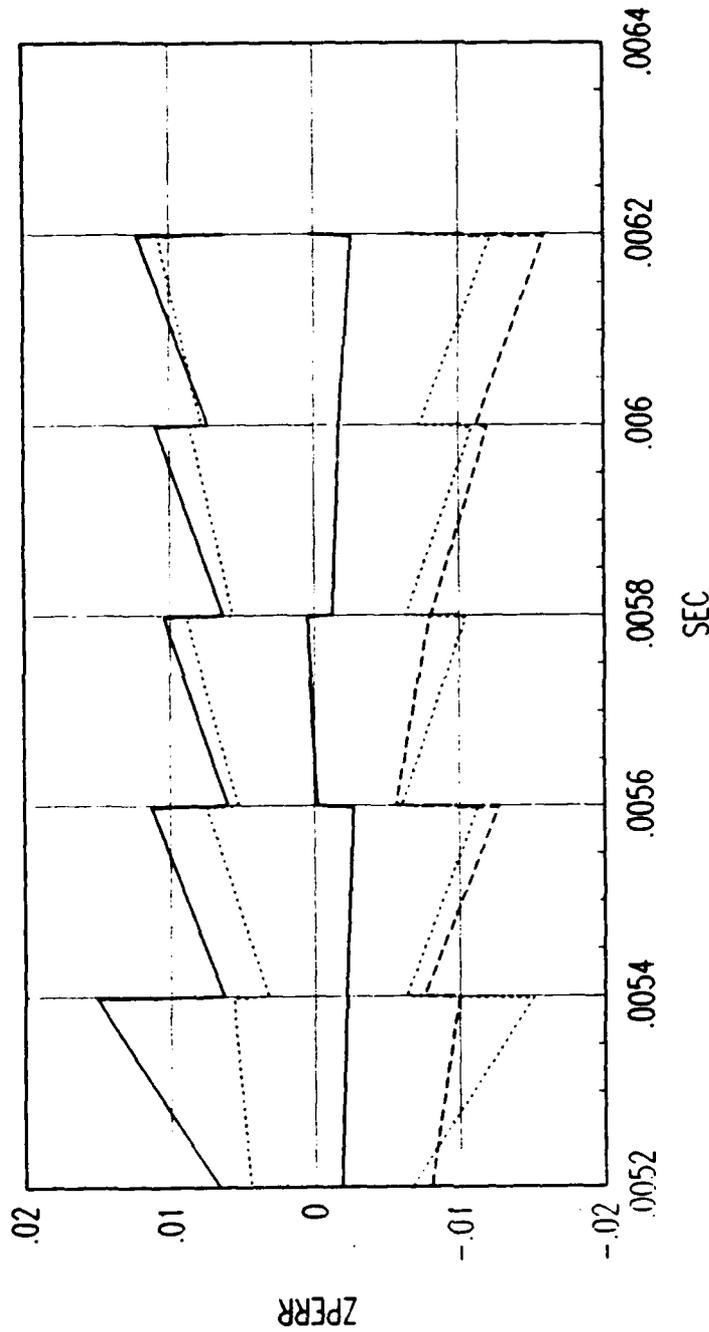


FIG. C FORWARD TIME EKF POSITION ERROR

Figure C.211. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

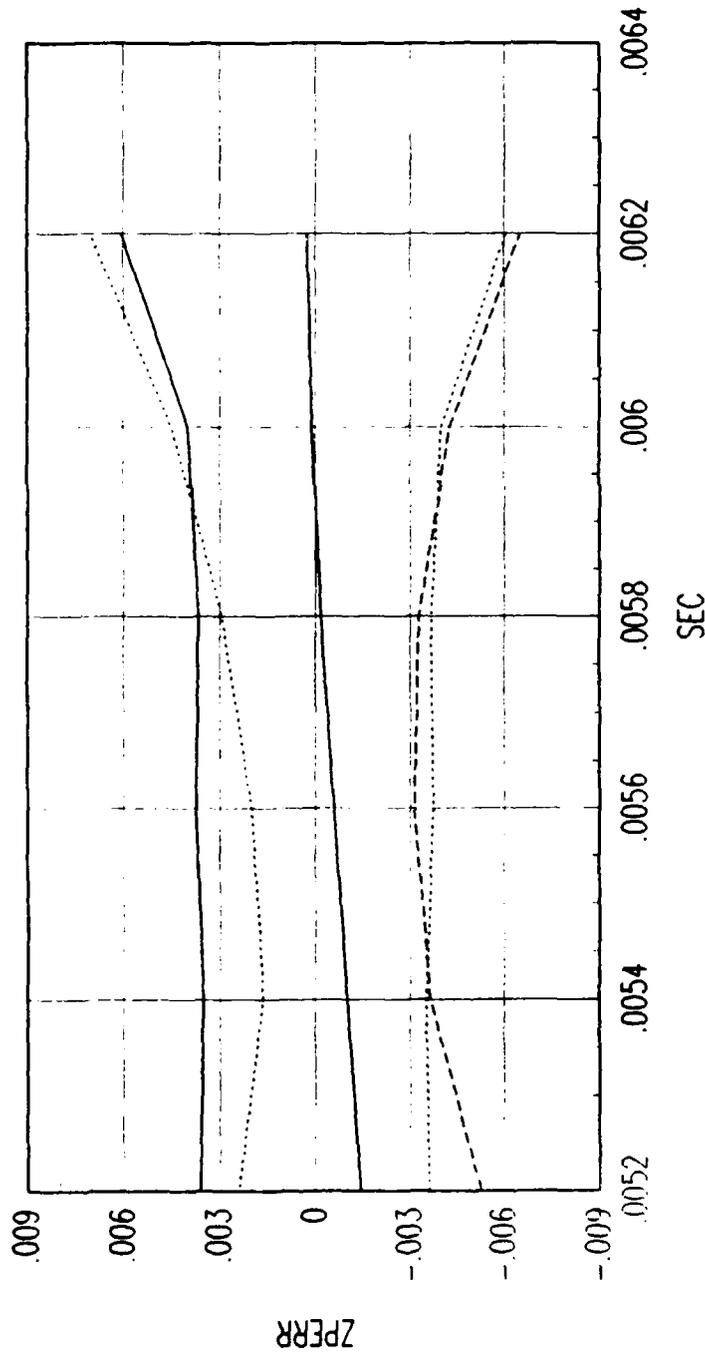
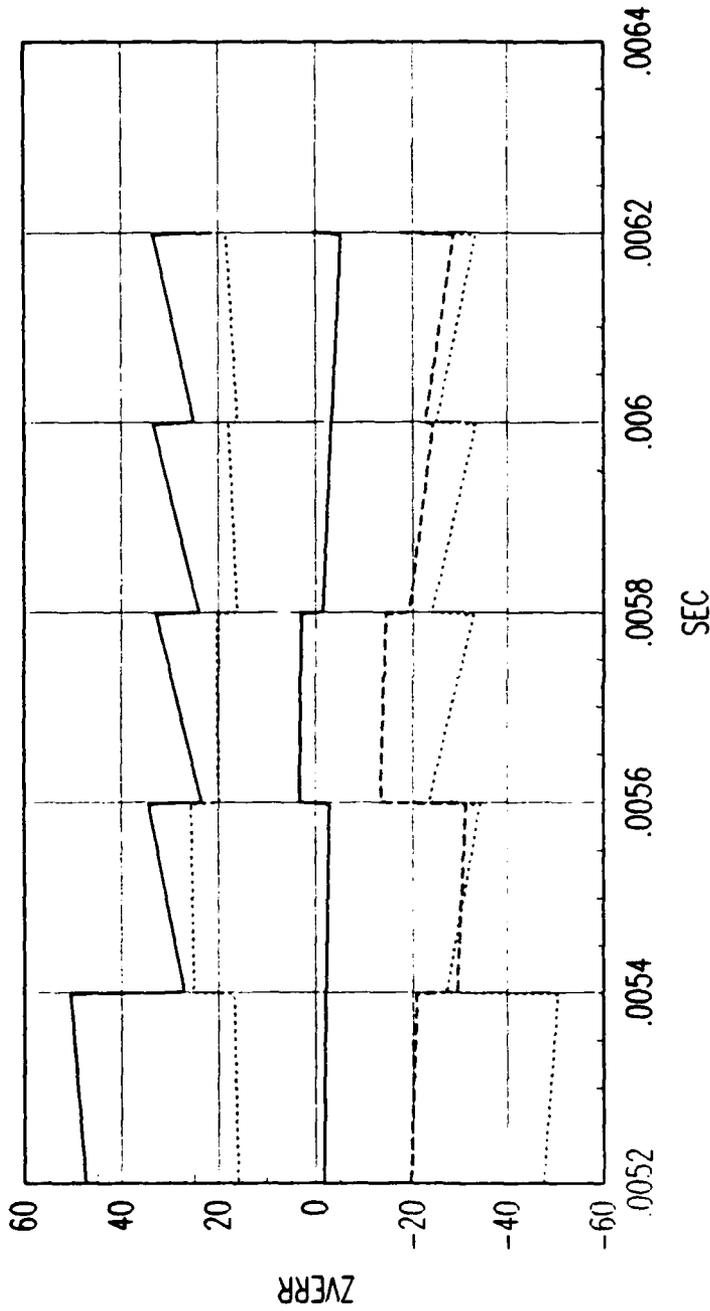


FIG. C REVERSE TIME SMOOTHER POSITION ERROR

Figure C.212. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C FORWARD TIME EKF VELOCITY ERROR

Figure C.213. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

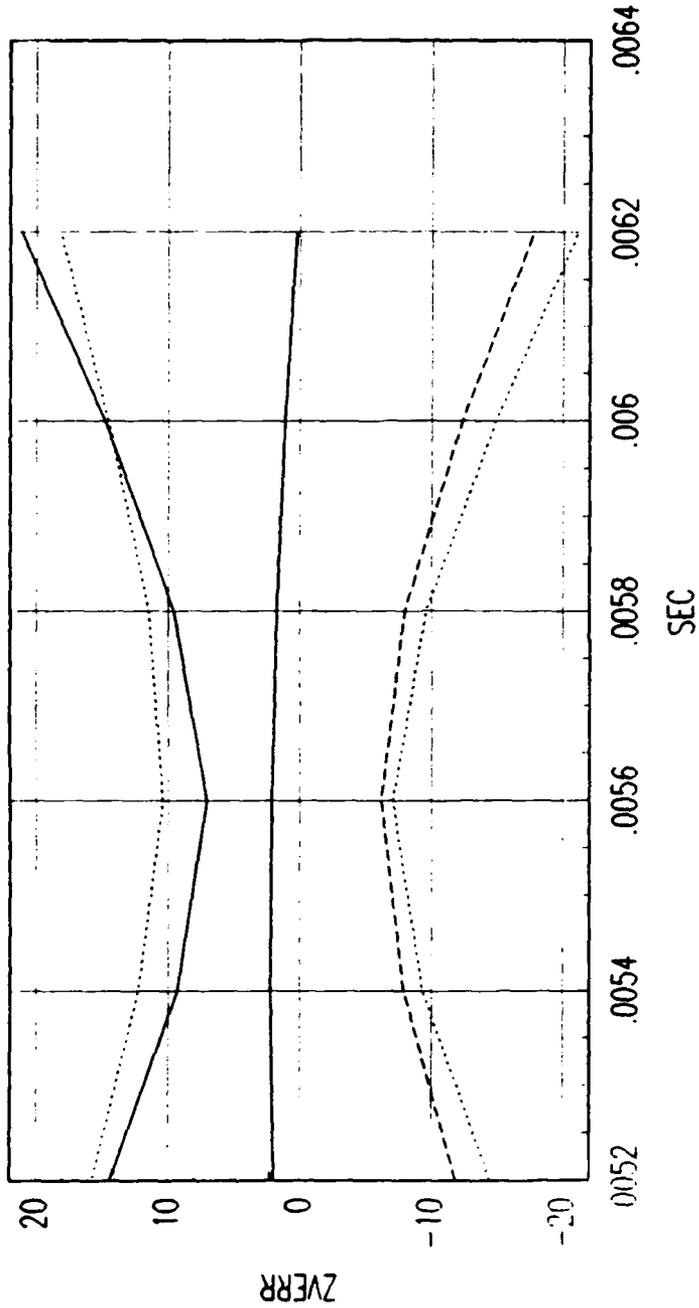
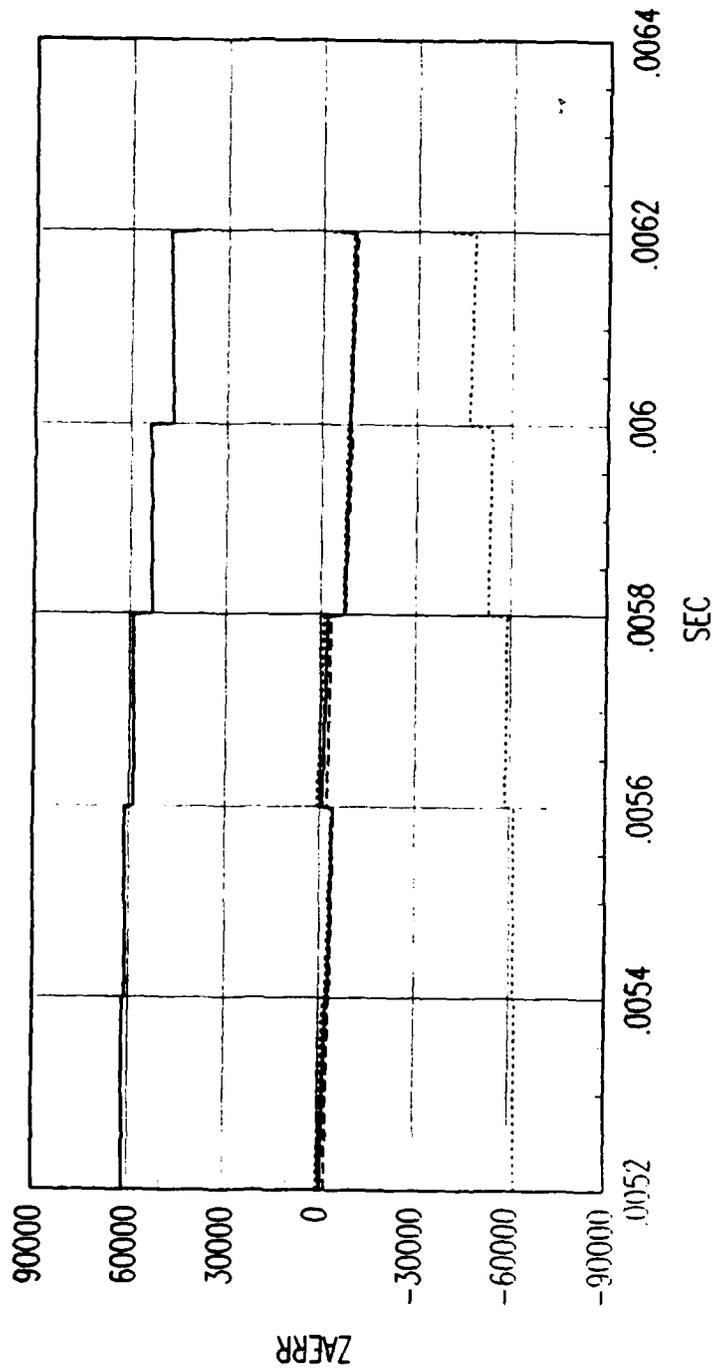


FIG. C REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.214. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.



FRG. C FORWARD TIME EKF ACCELERATION ERROR

Figure C.215. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

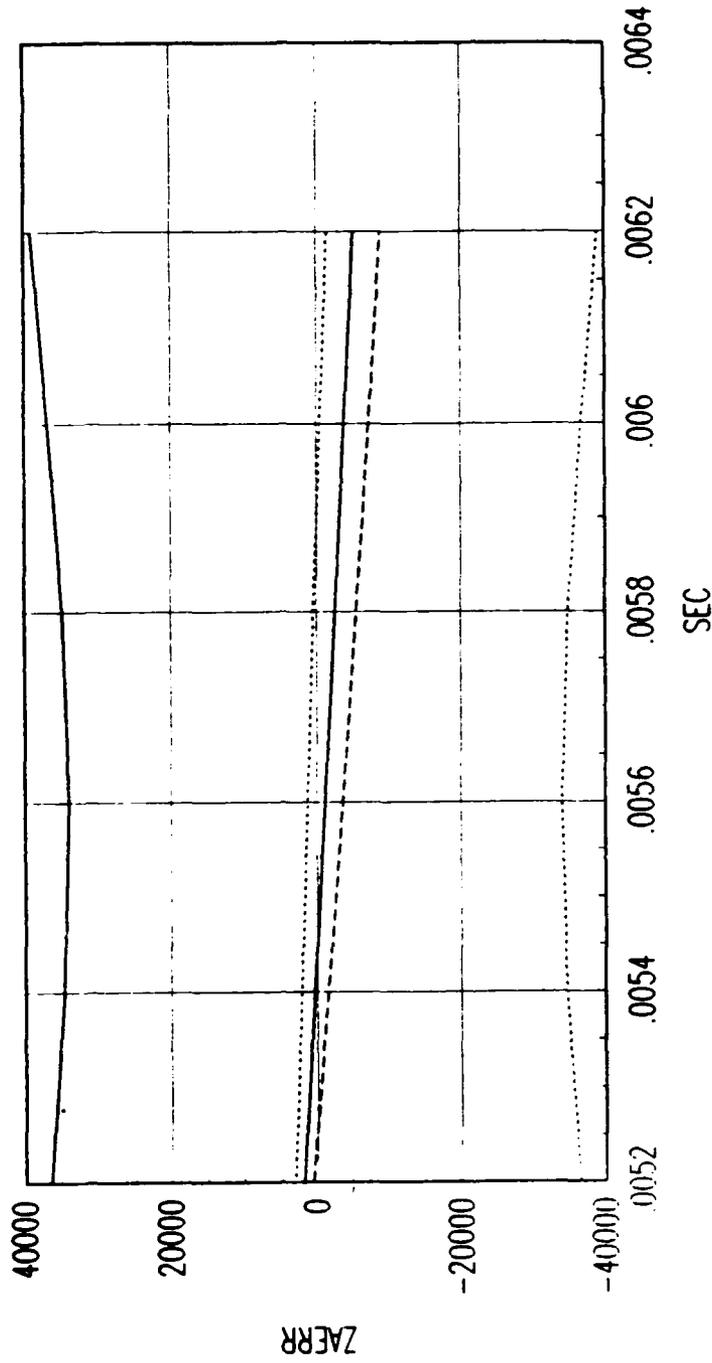


FIG. C REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.216. Tracking Error Plot, Category 2, $v(t_0) = 10000$ ft/sec.

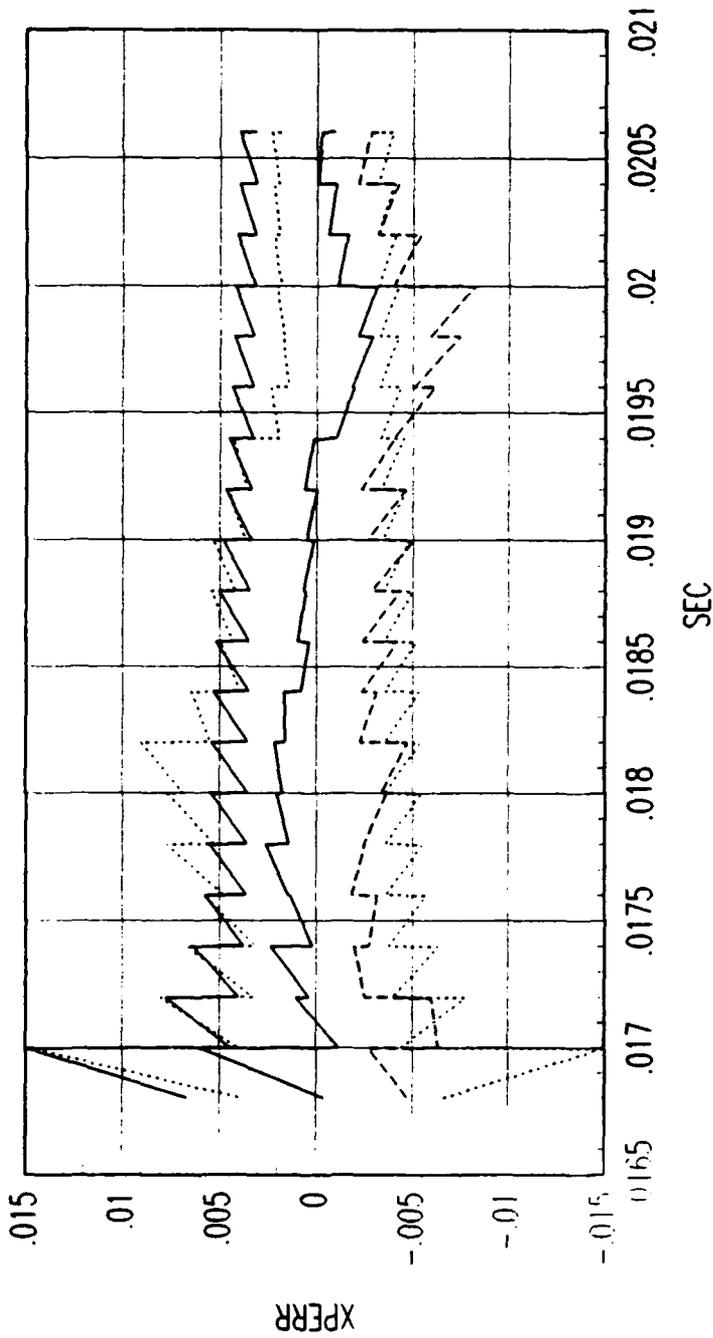
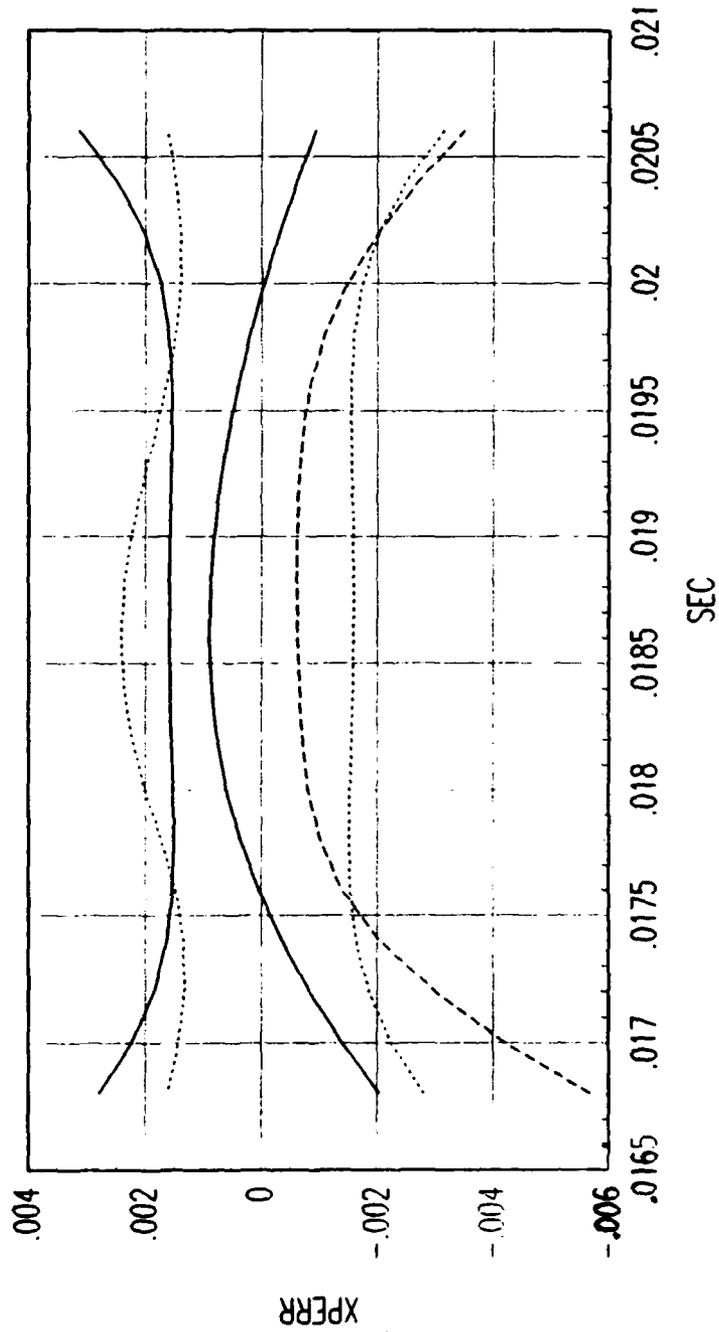


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.217. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.218. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

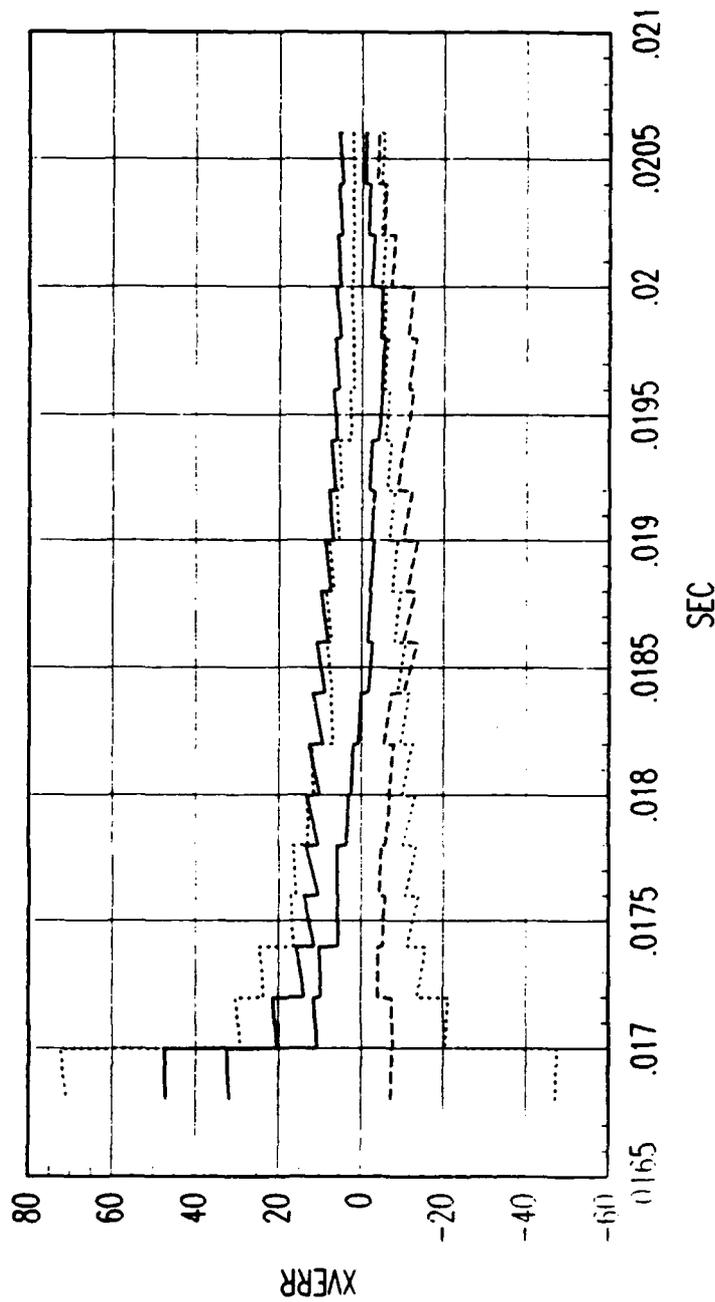


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.219. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

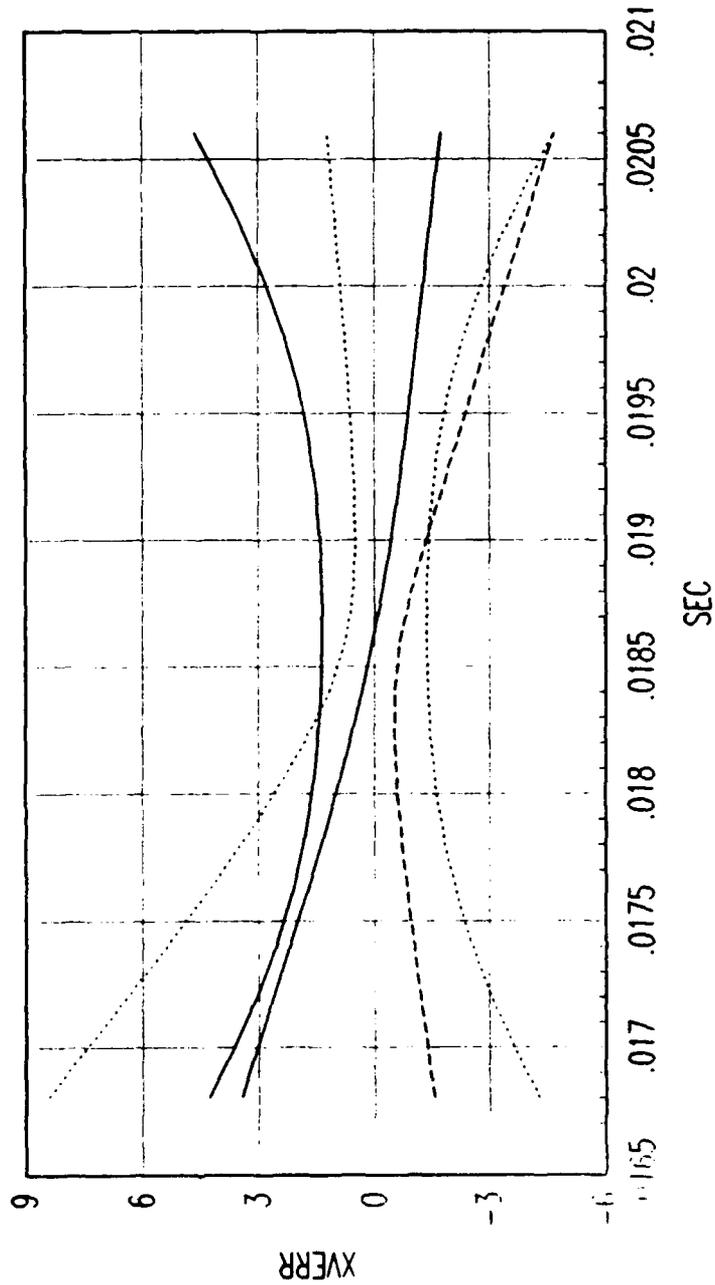
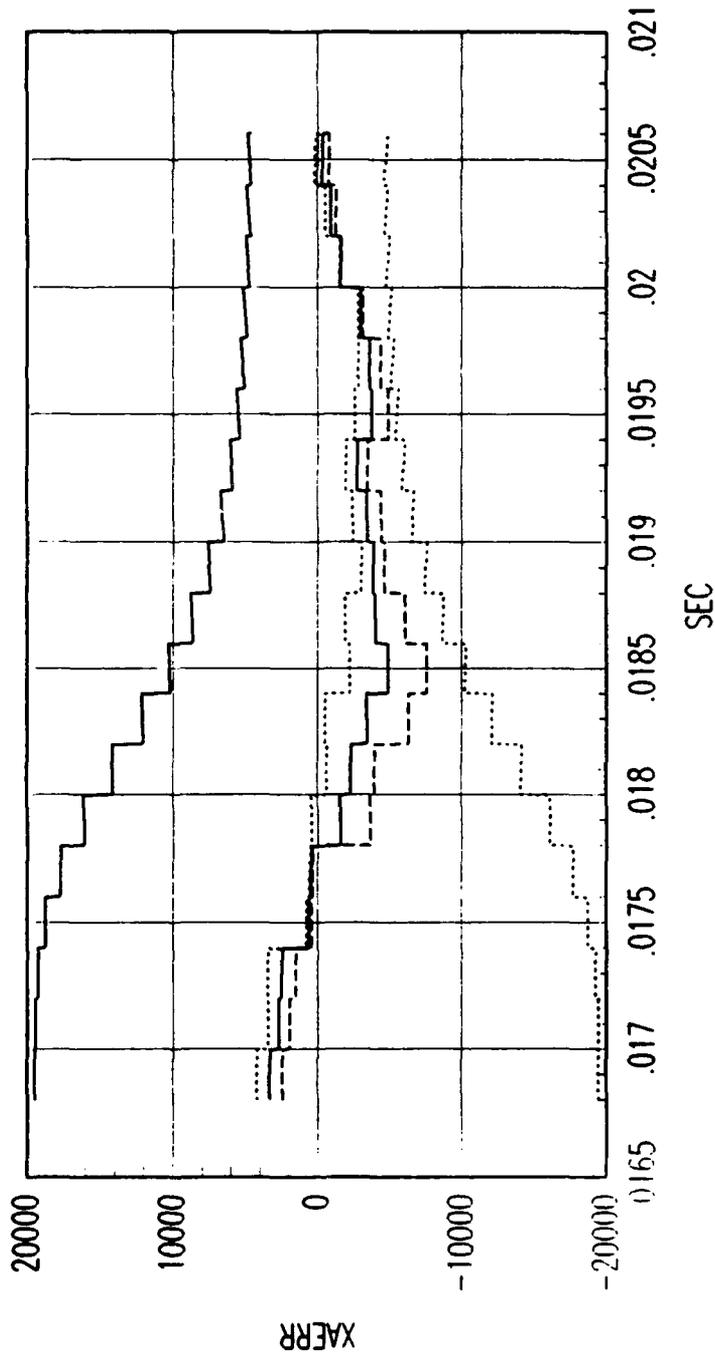


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.220. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.221. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

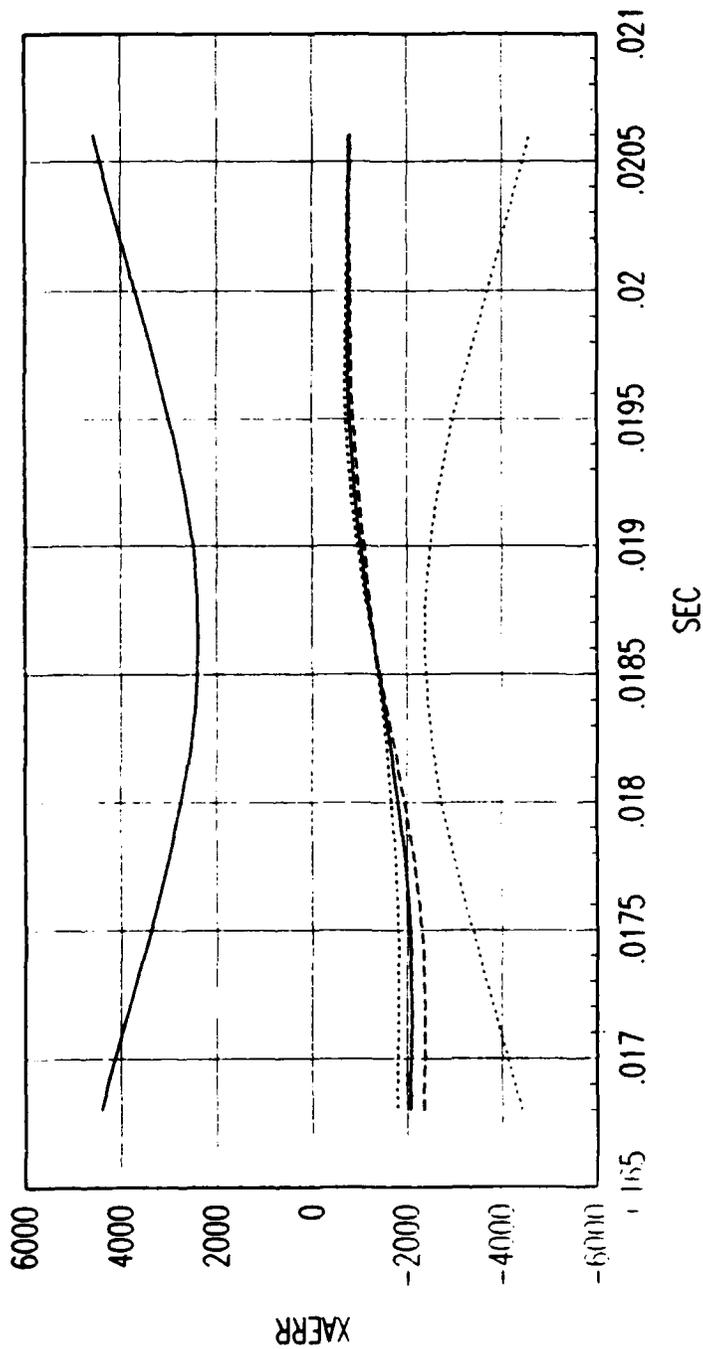


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.222. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

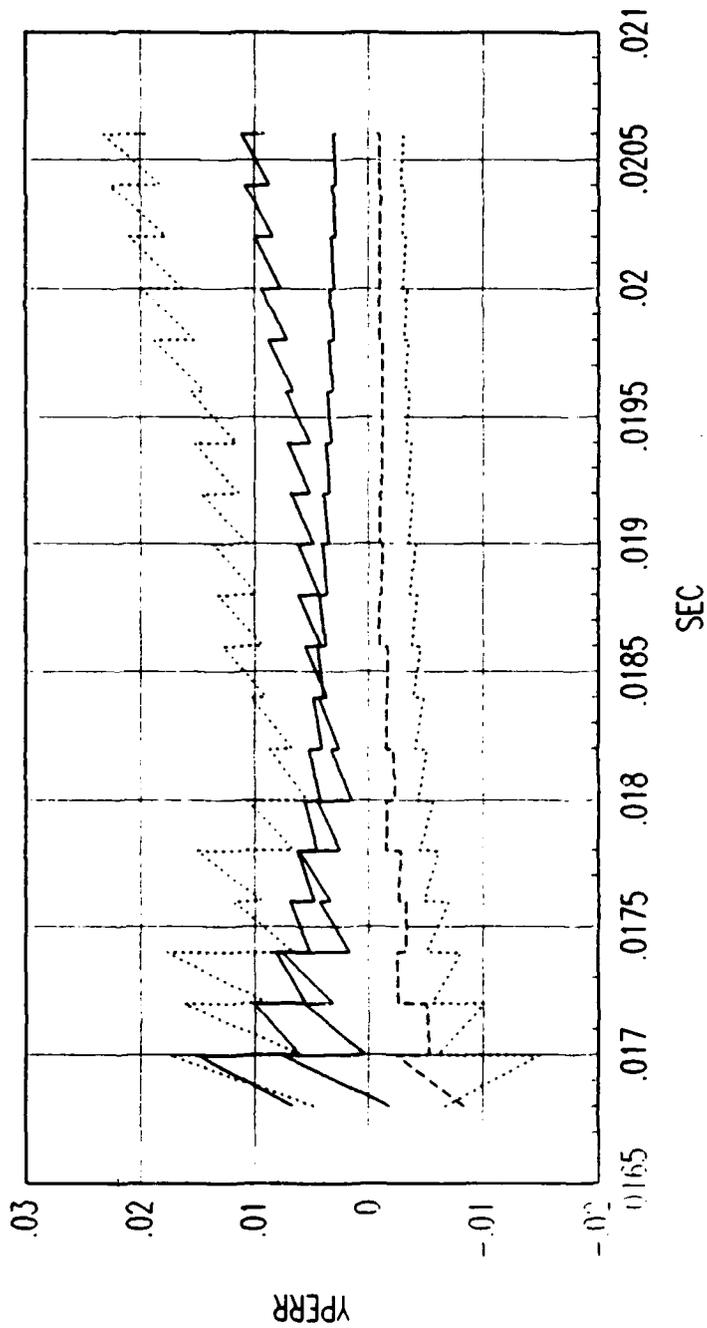


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.223. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

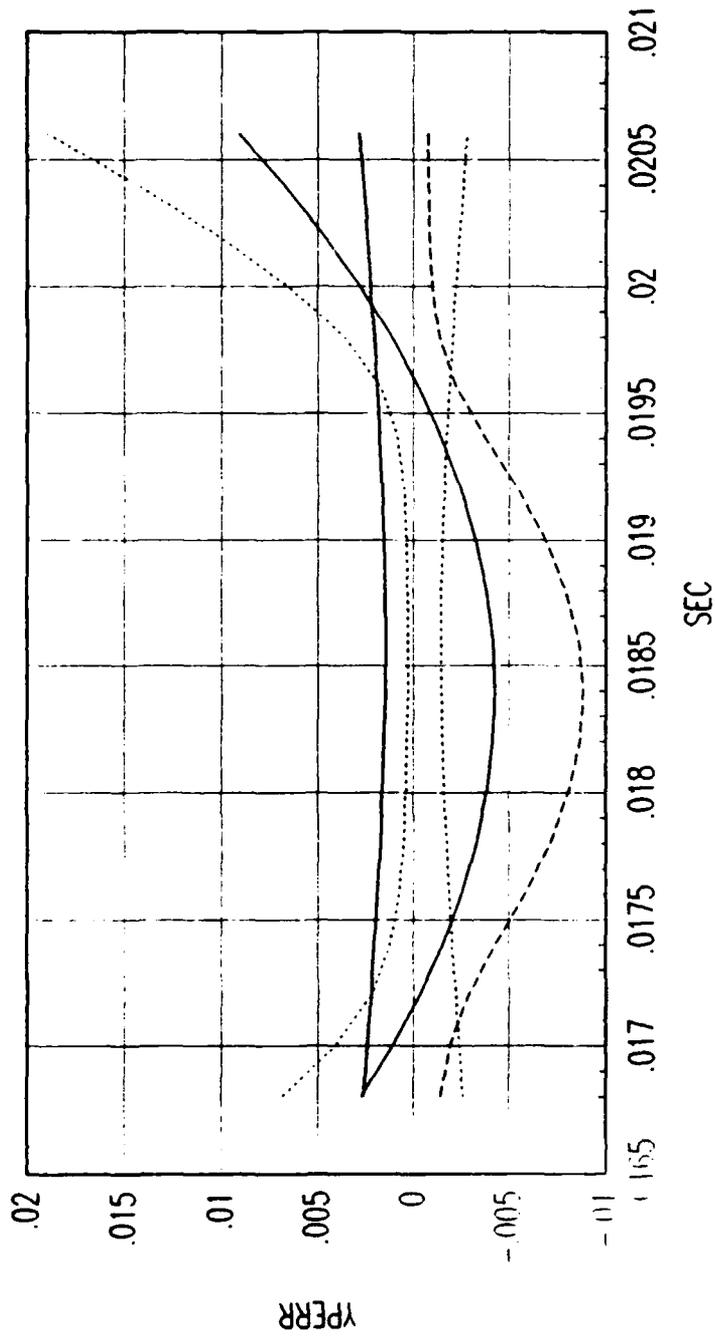


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.224. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

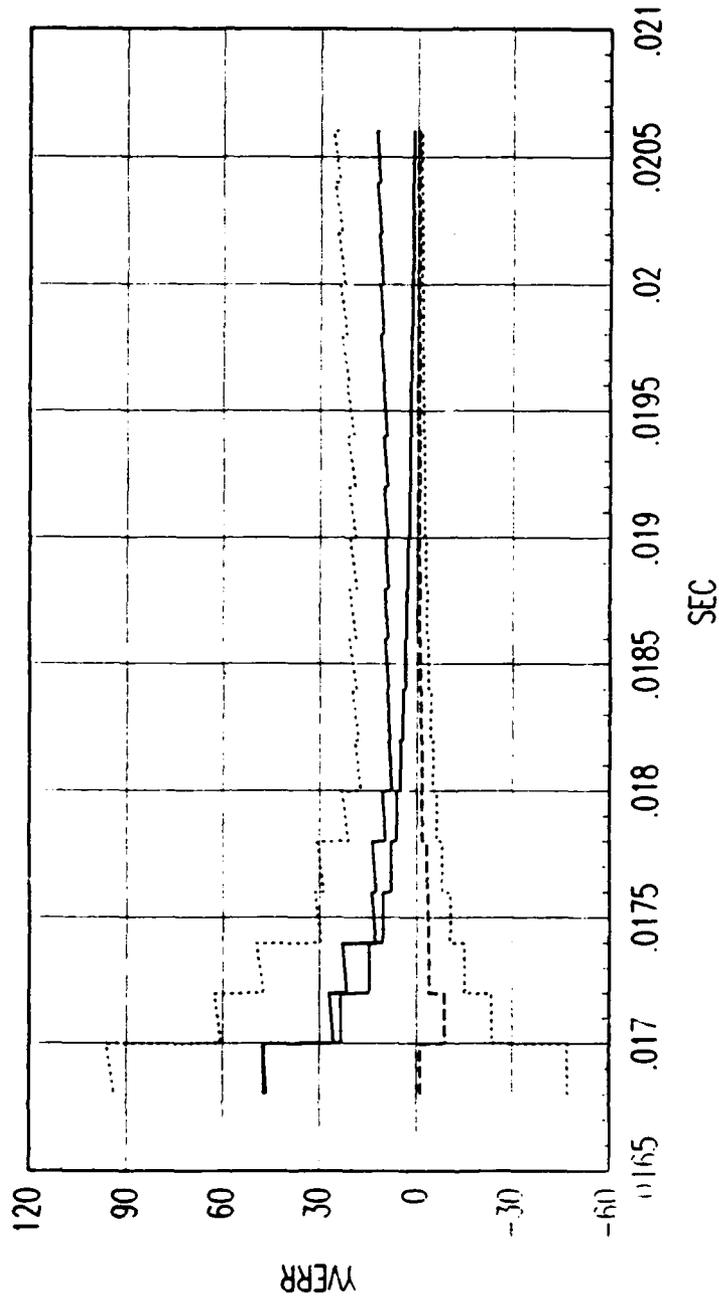


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.225. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

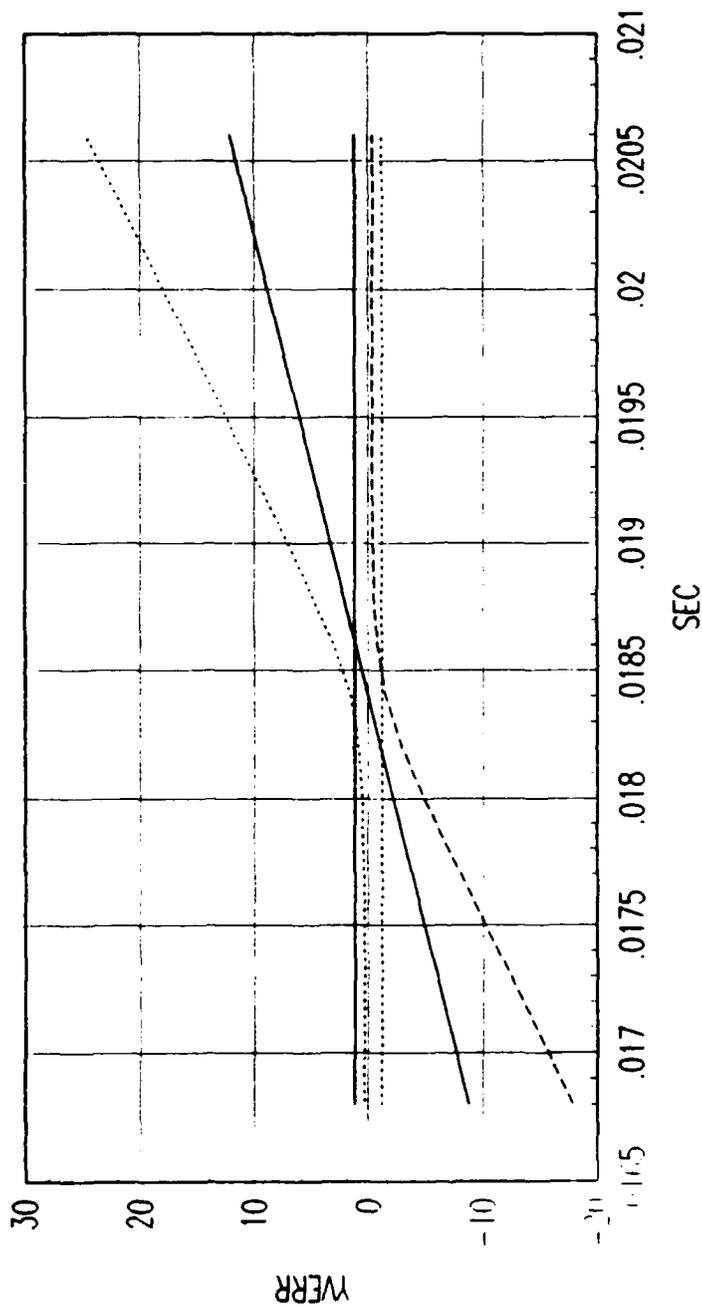


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.226. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

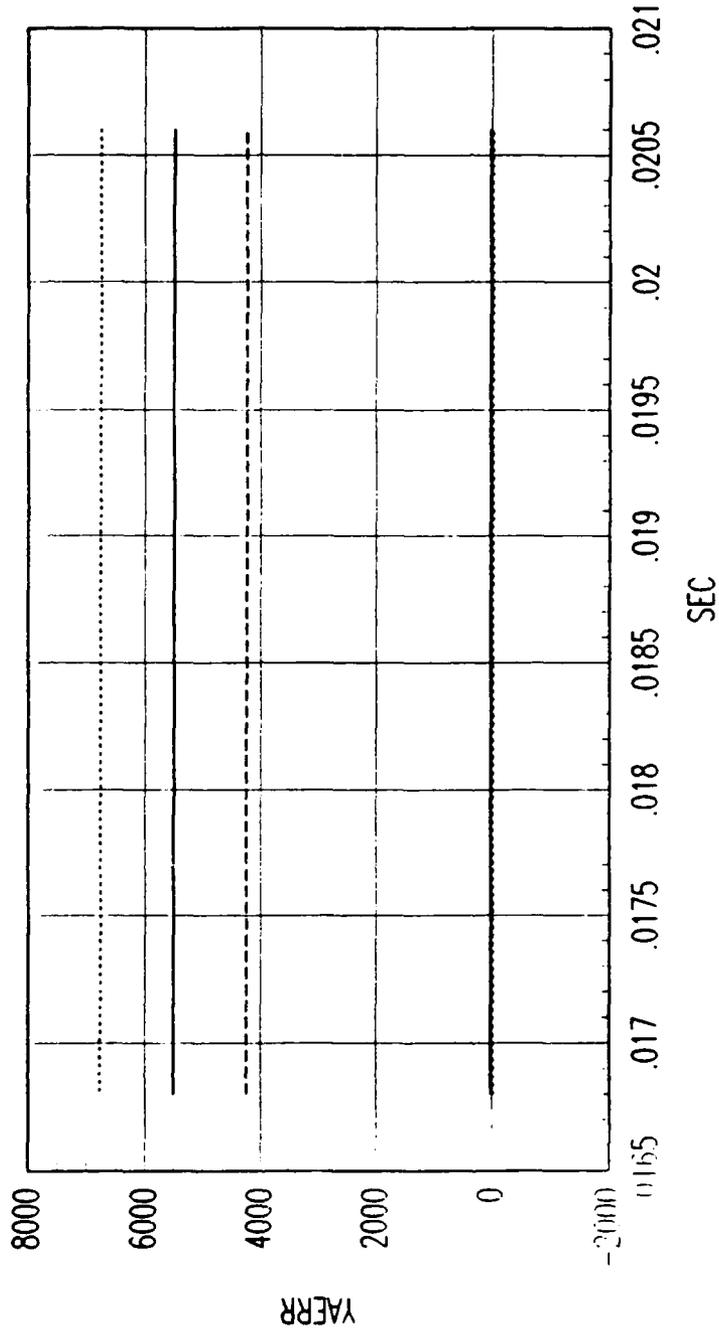
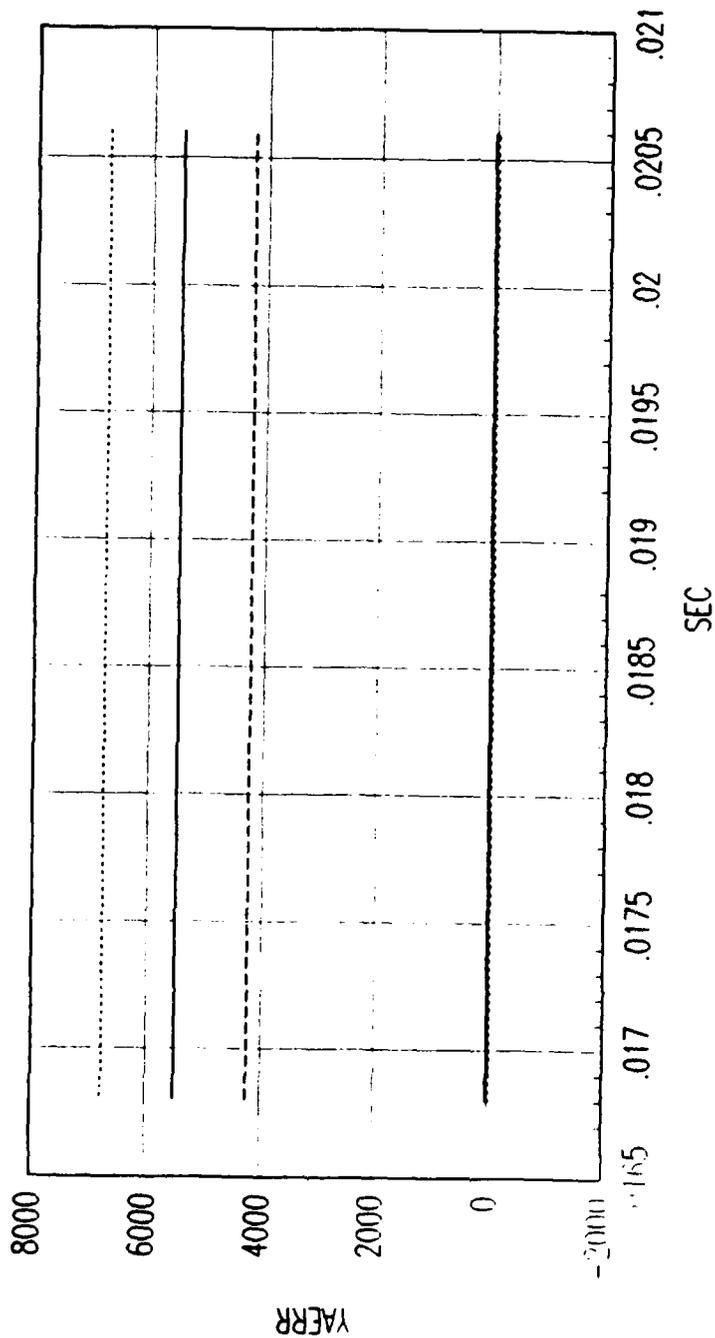


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.227. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.228. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

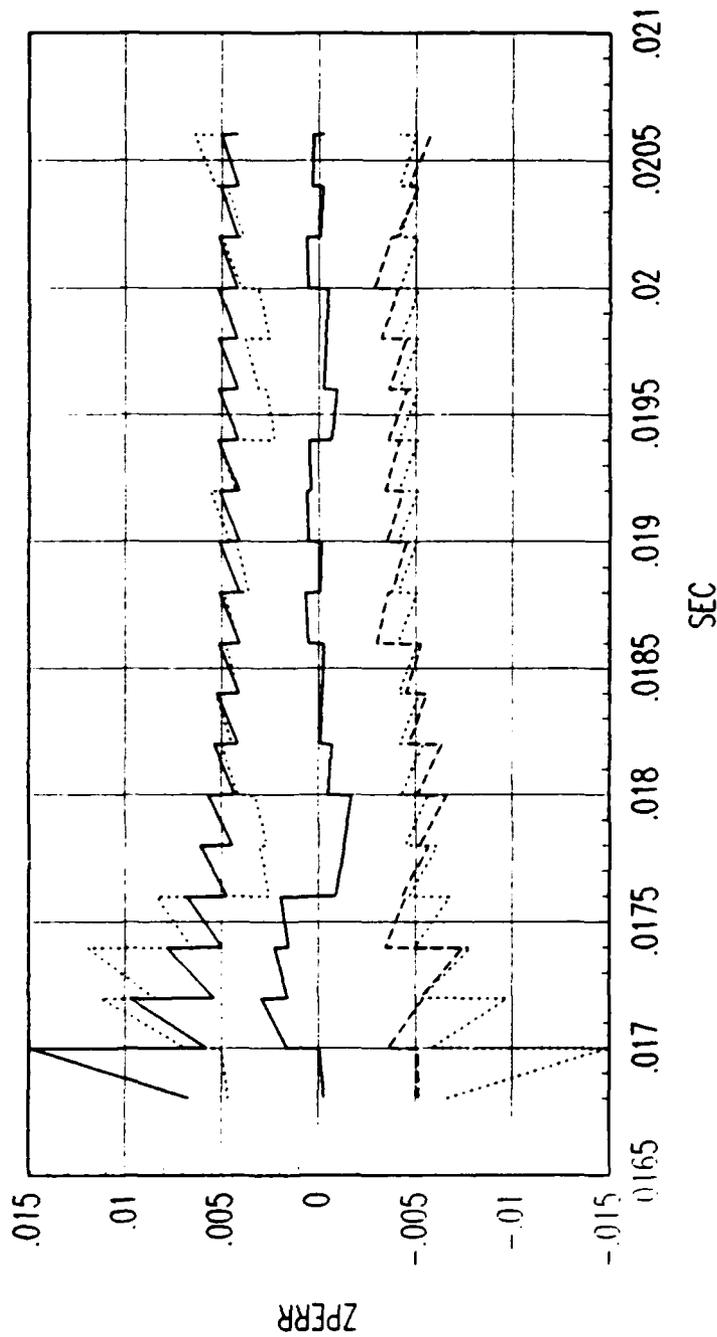
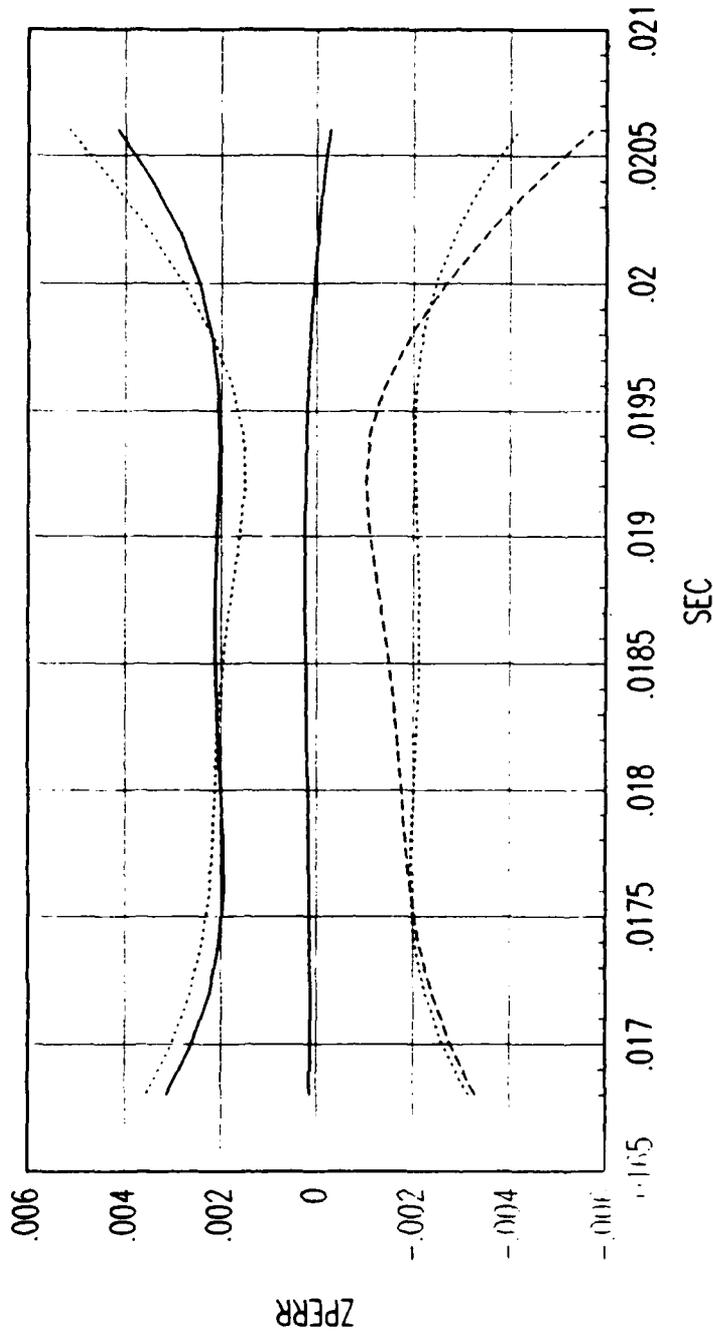


FIG. A FORWARD TIME EKf POSITION ERROR

Figure C.229. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.230. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

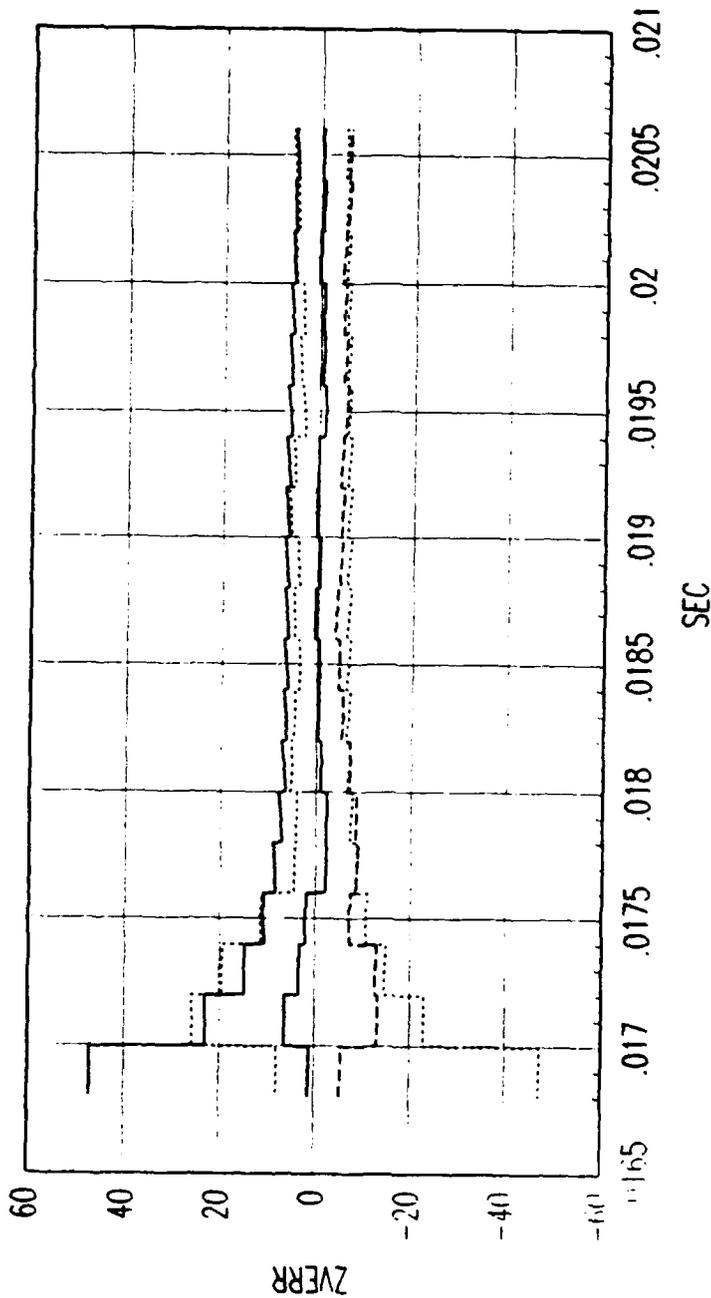


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.231. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

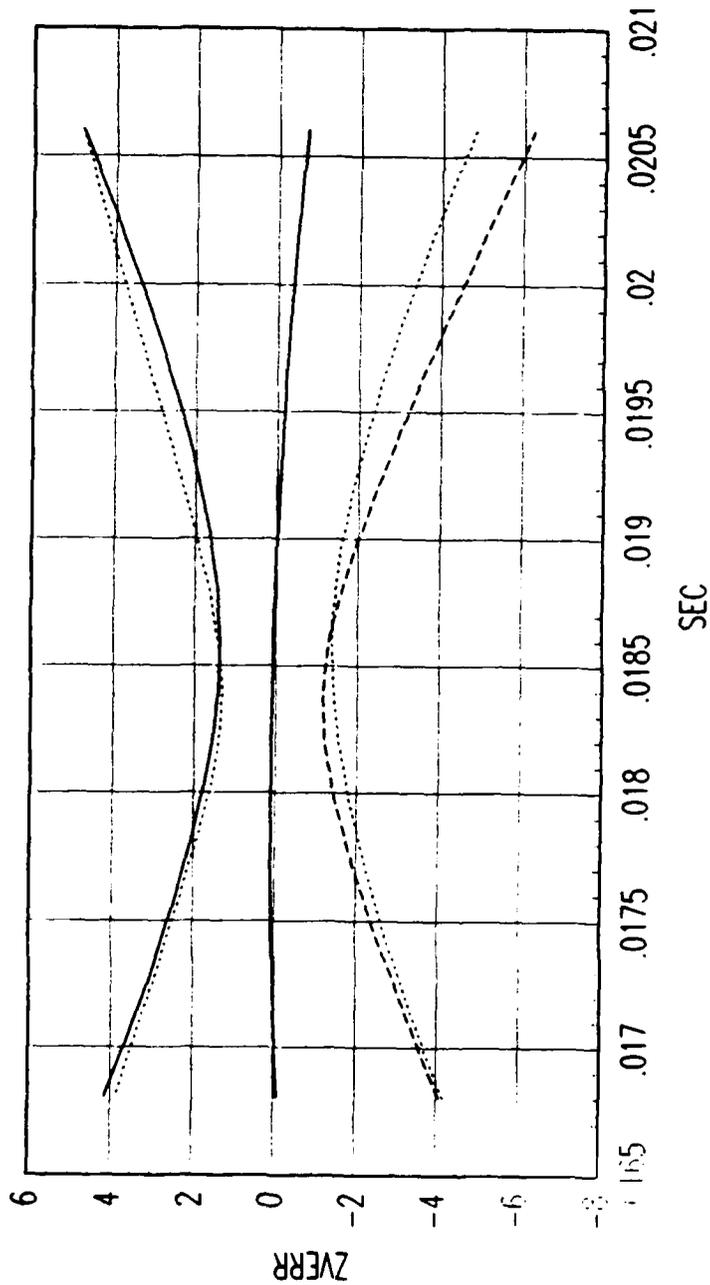
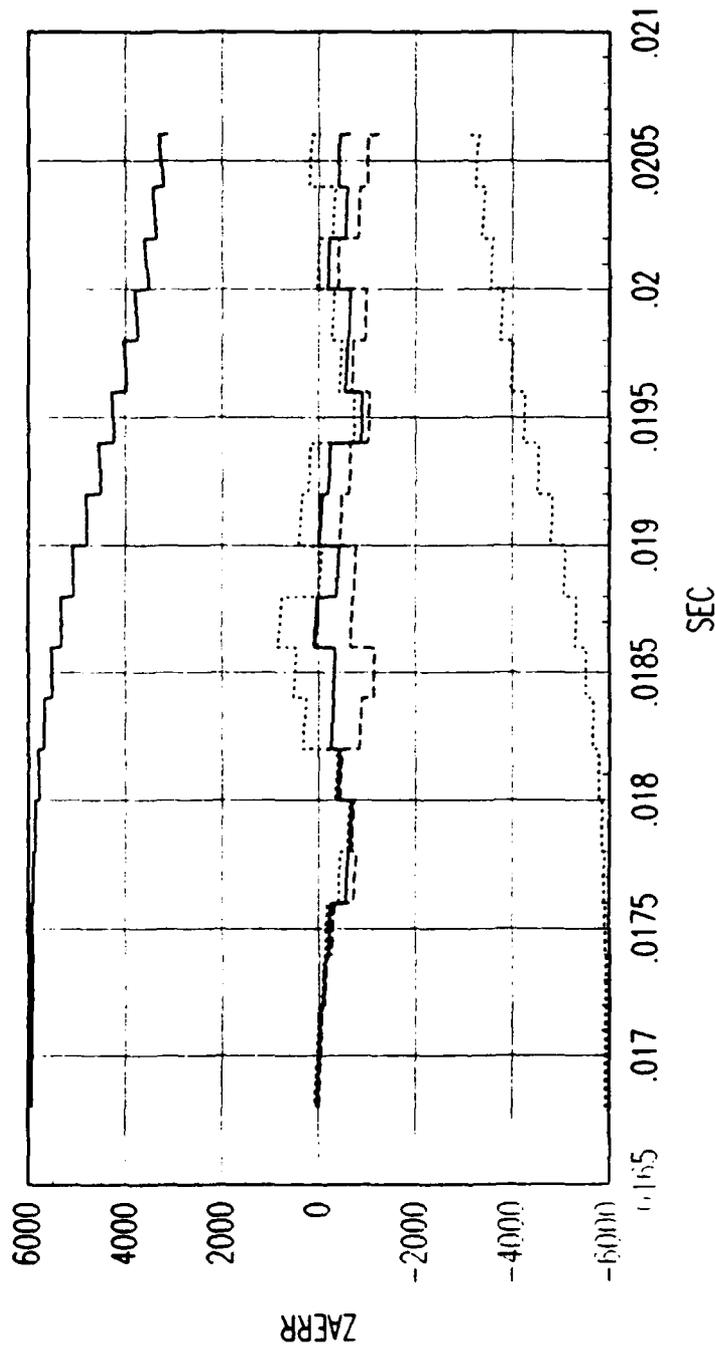


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.232. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.233. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

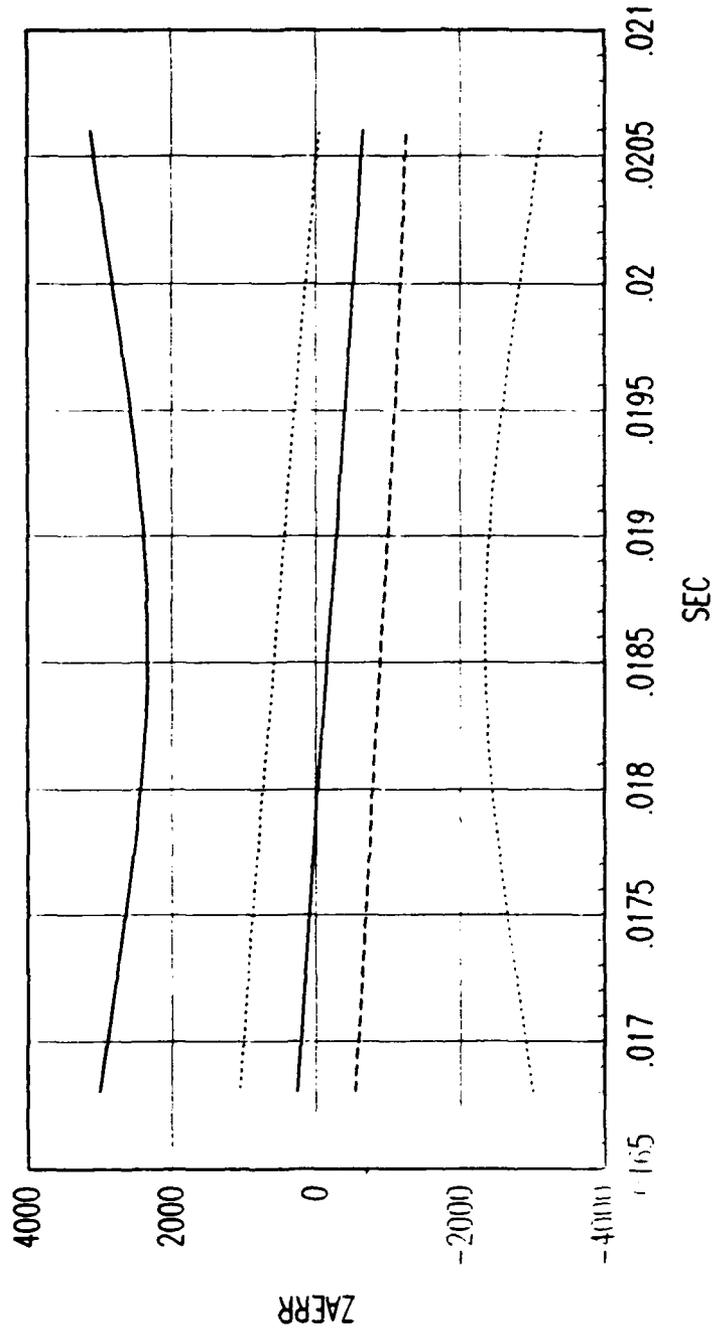


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.234. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

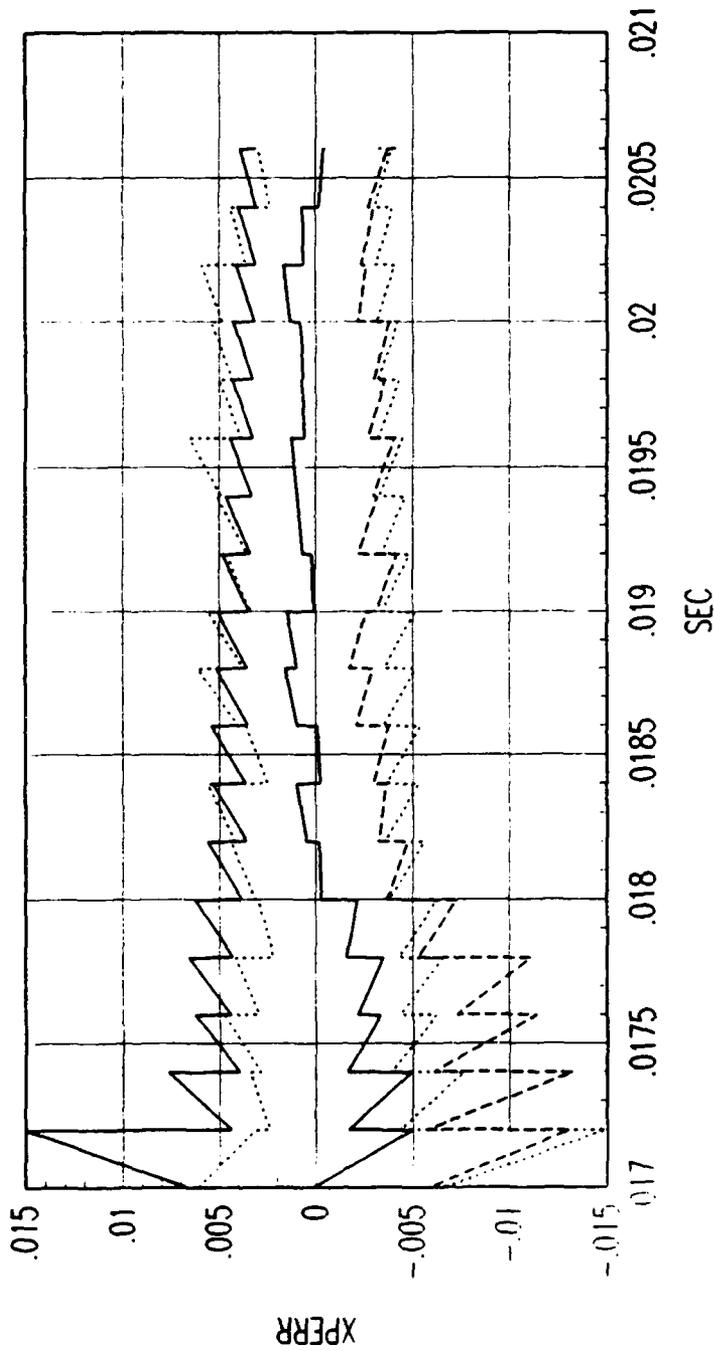
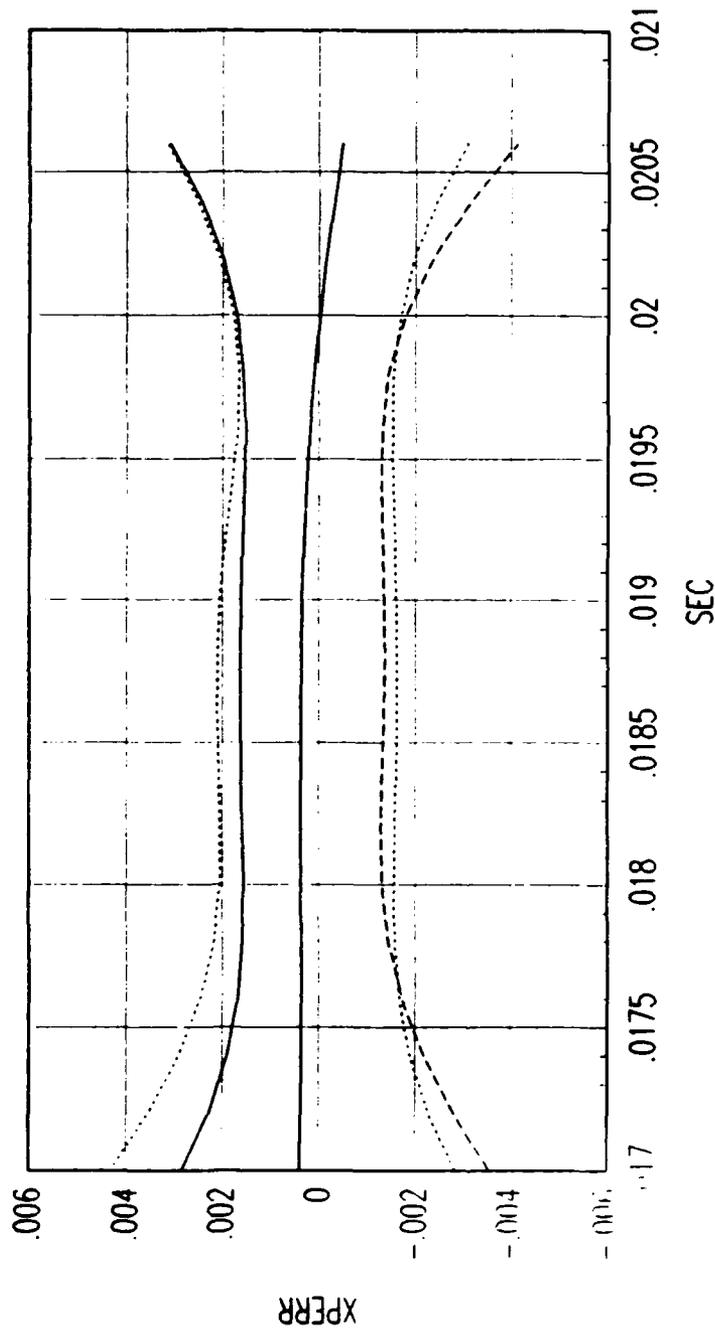


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.235. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.236. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

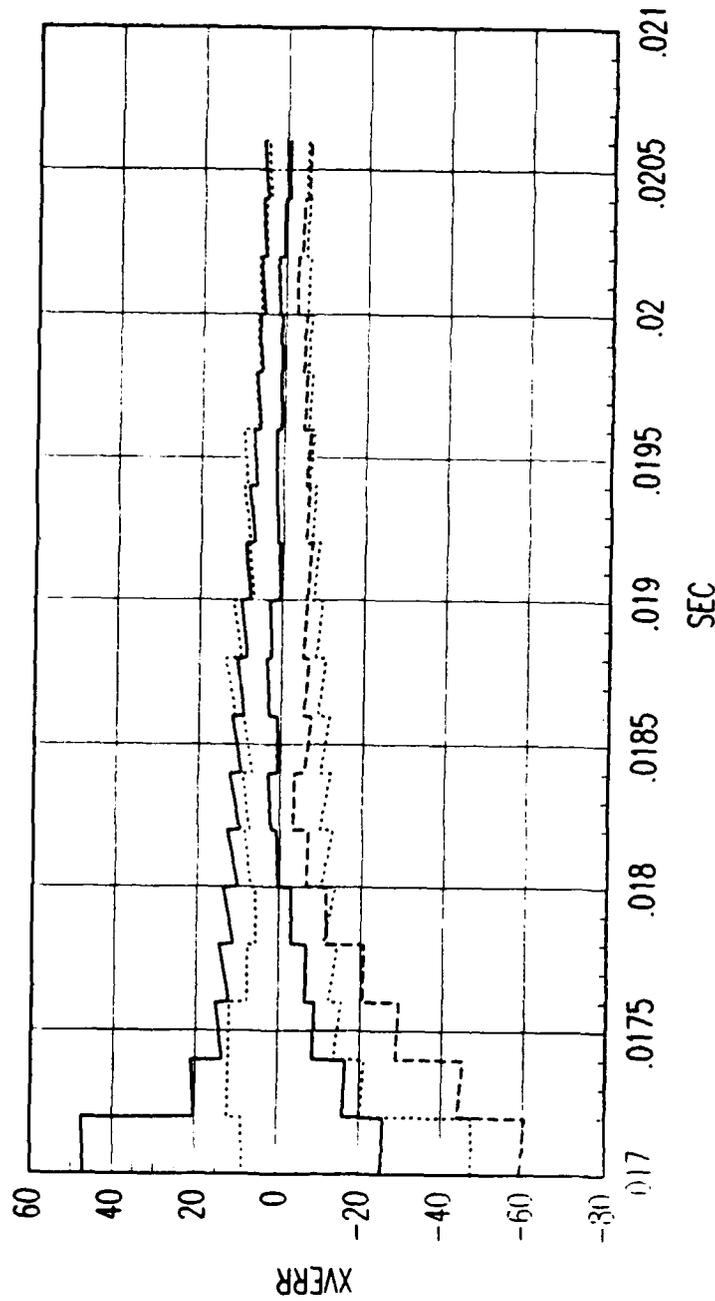
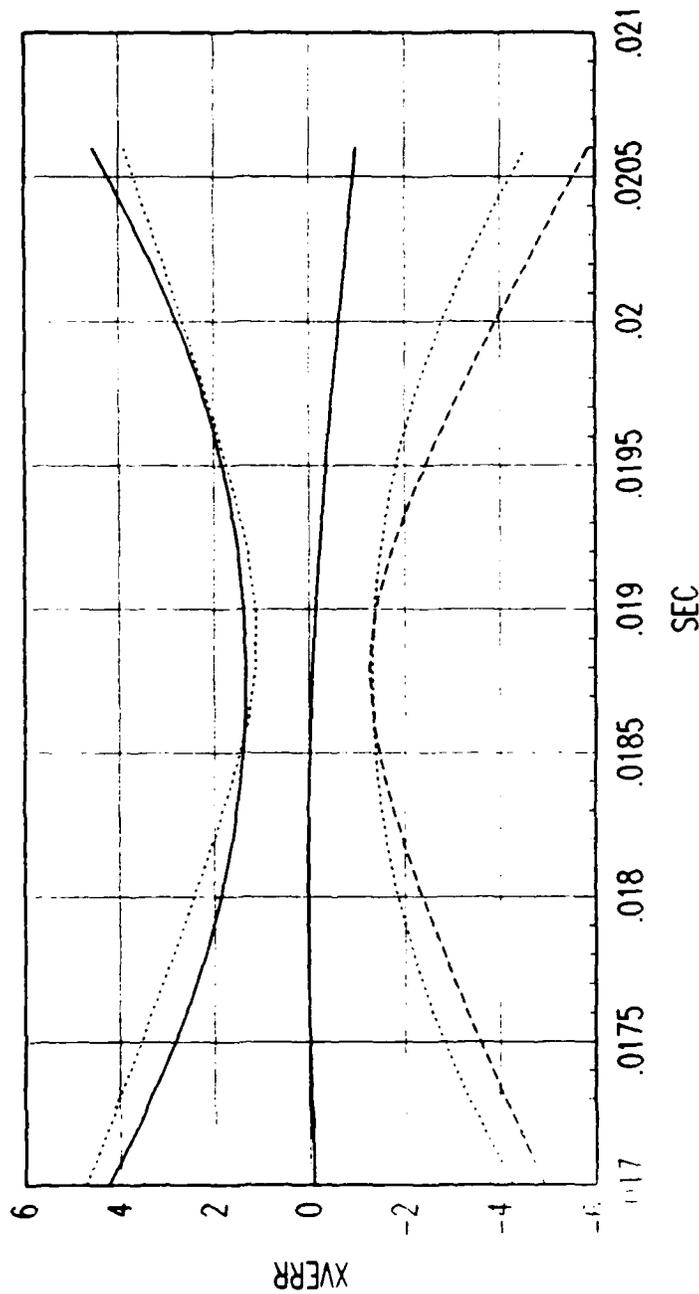


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.237. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.238. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

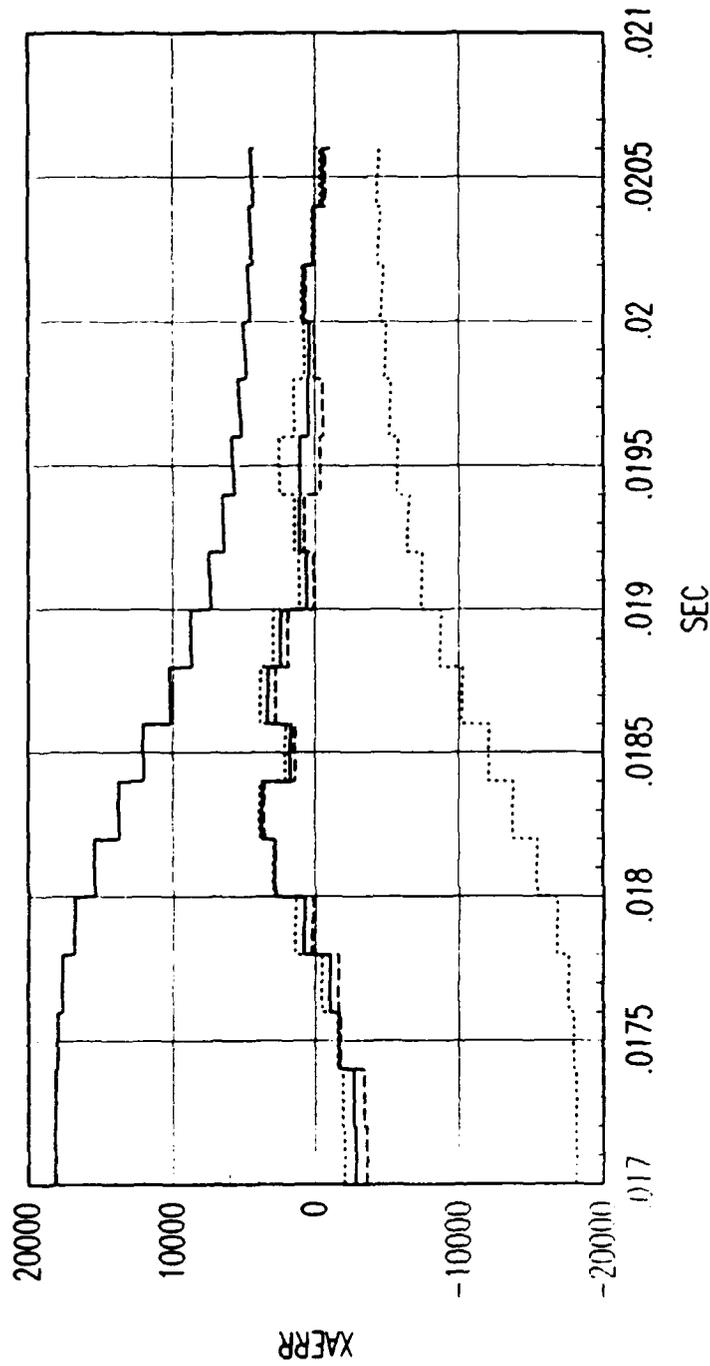


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.239. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

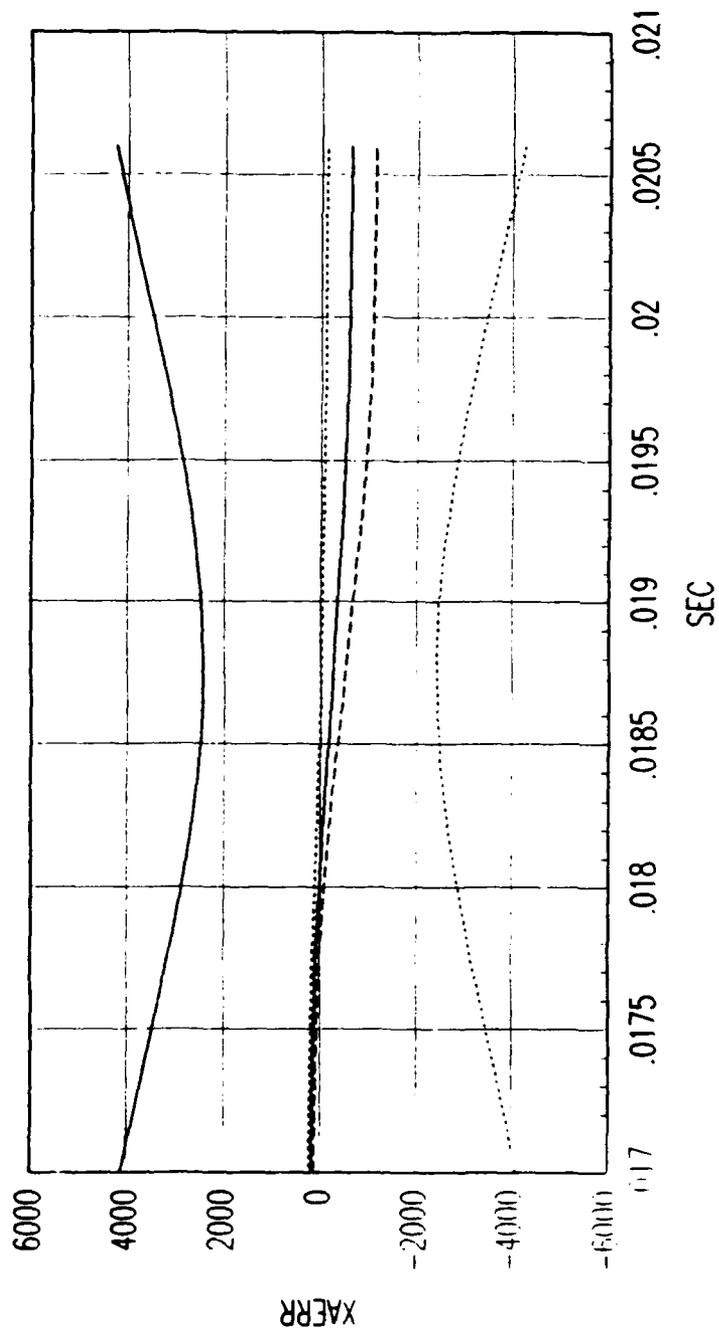
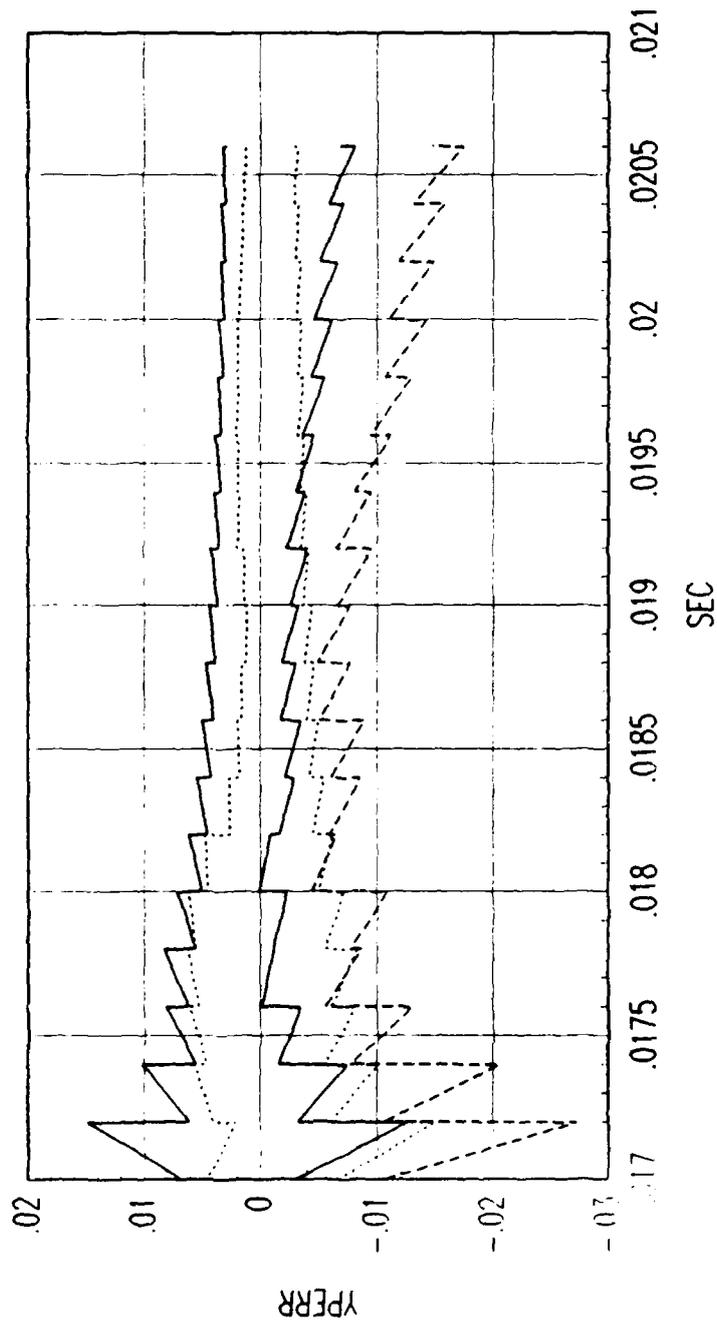


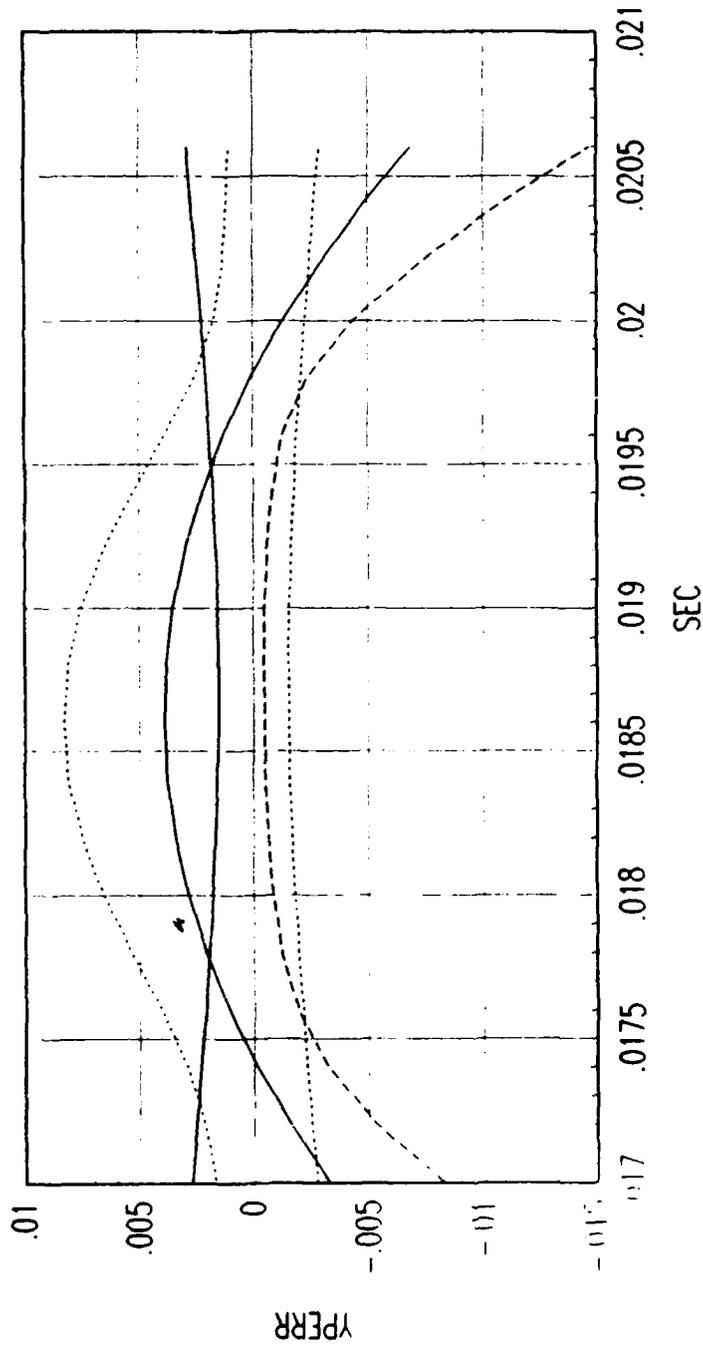
FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.240. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B FORWARD TIME EKF POSITION ERROR

Figure C.2*1. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.242. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

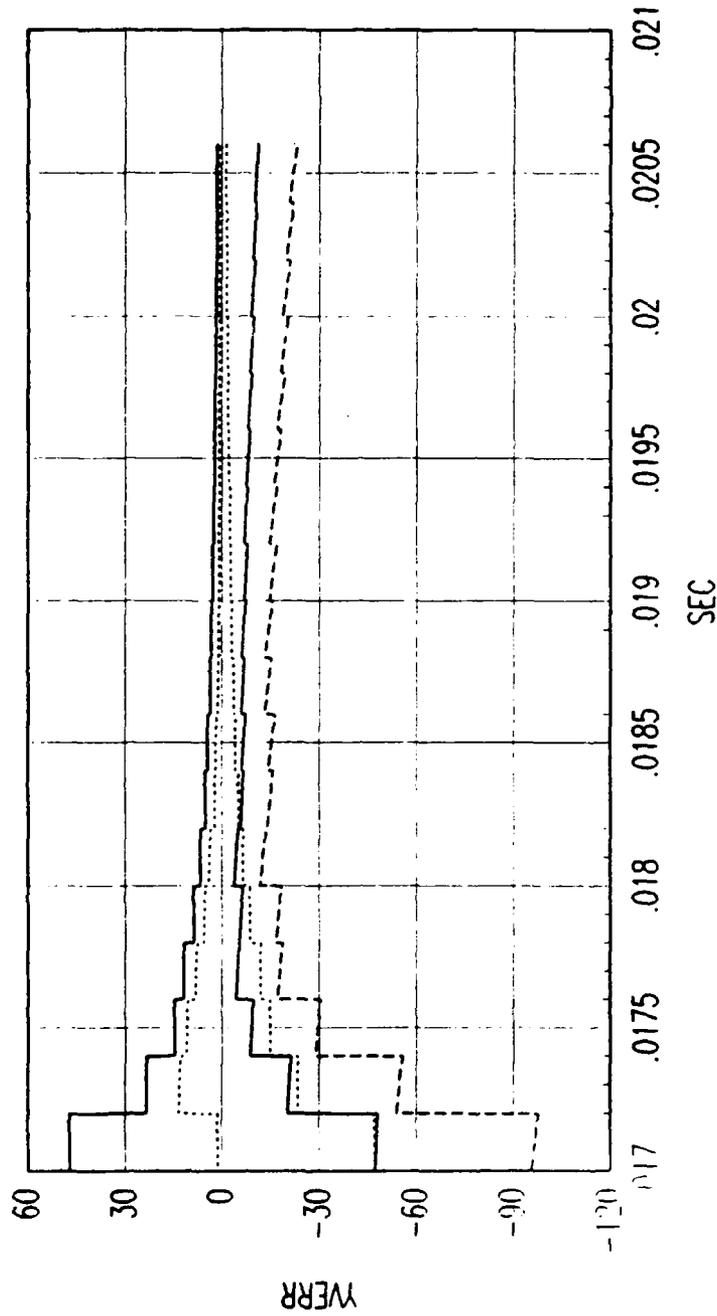
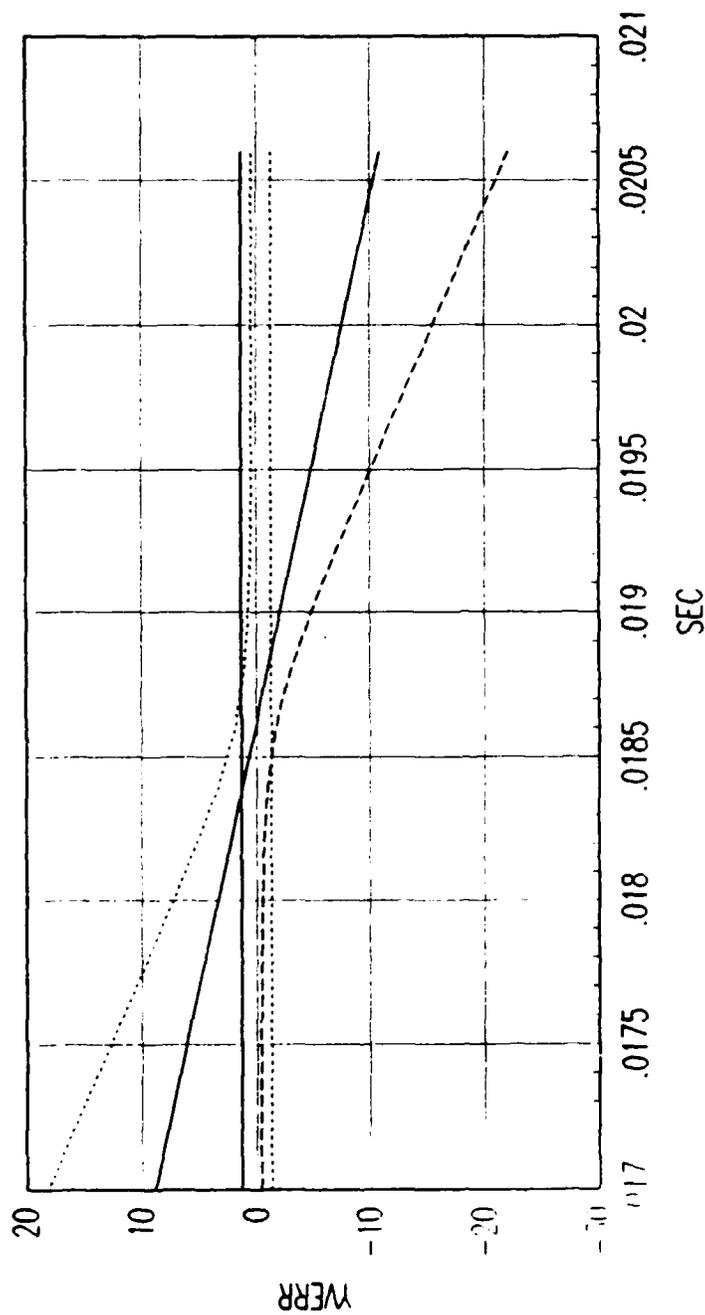


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.243. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.244. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

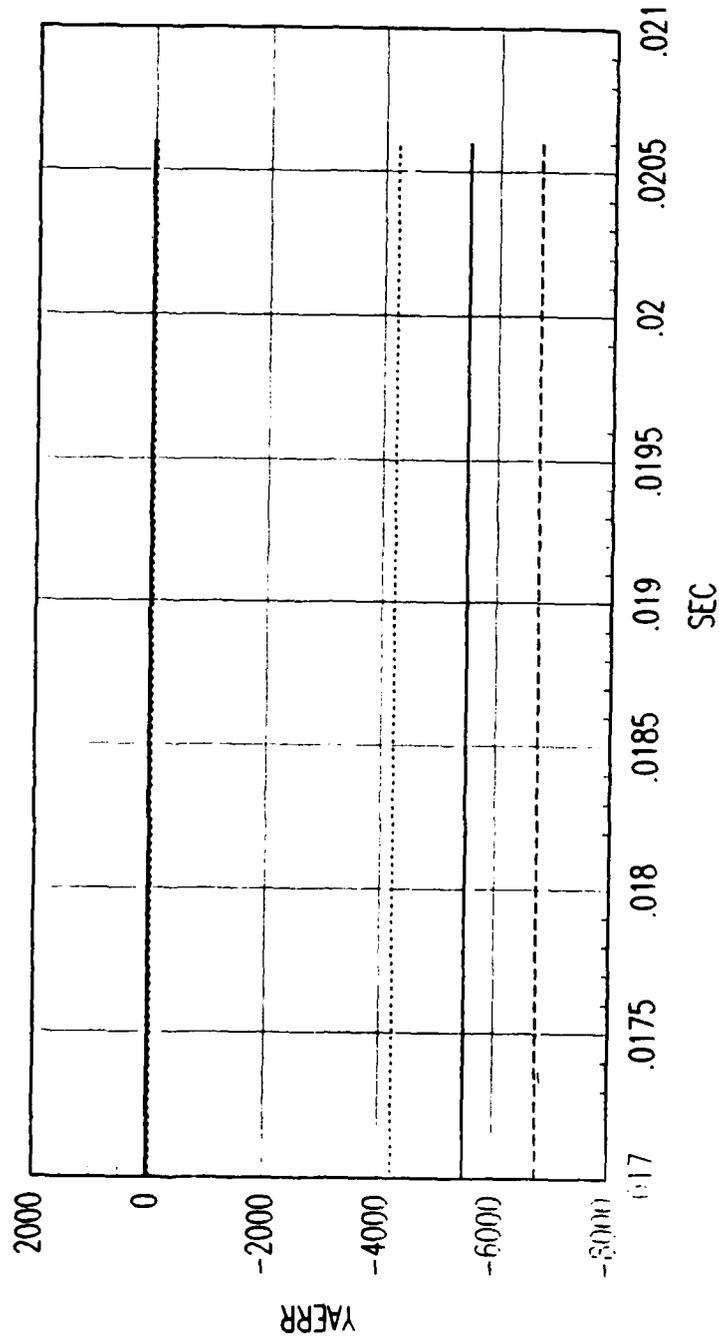


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.245. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

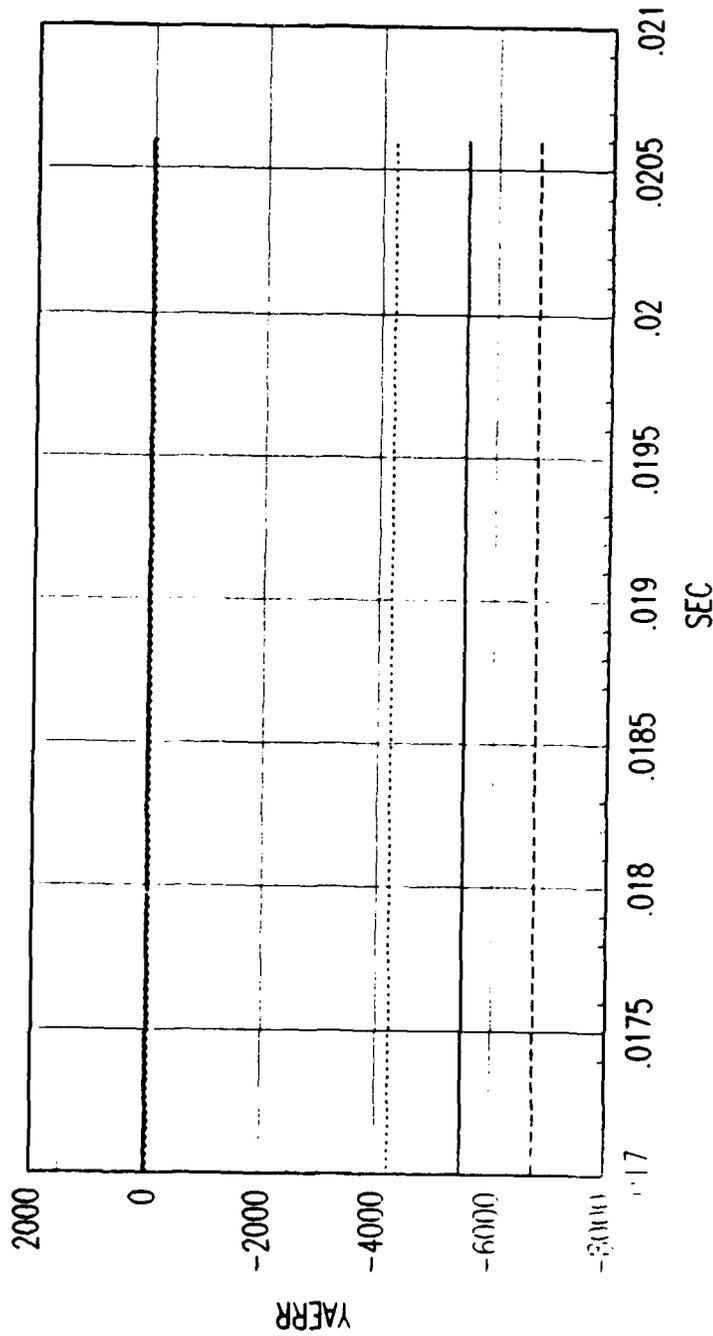


FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.246. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

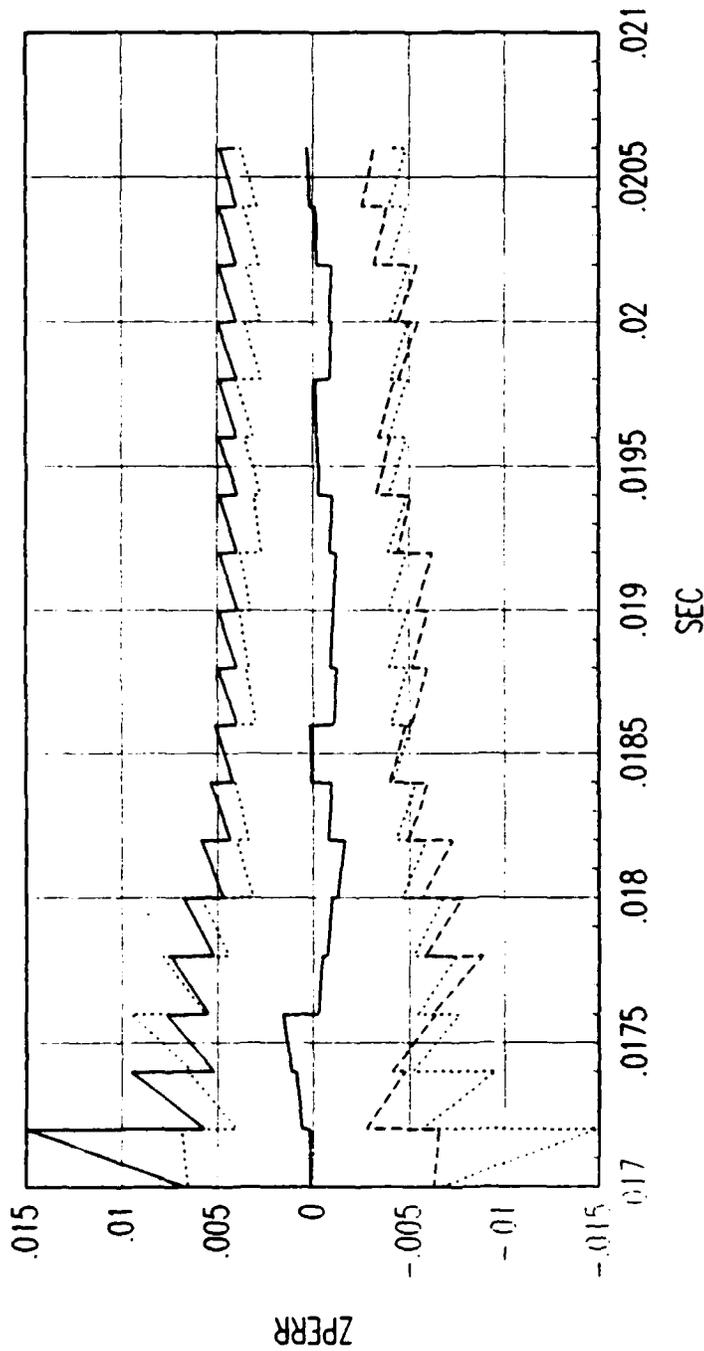
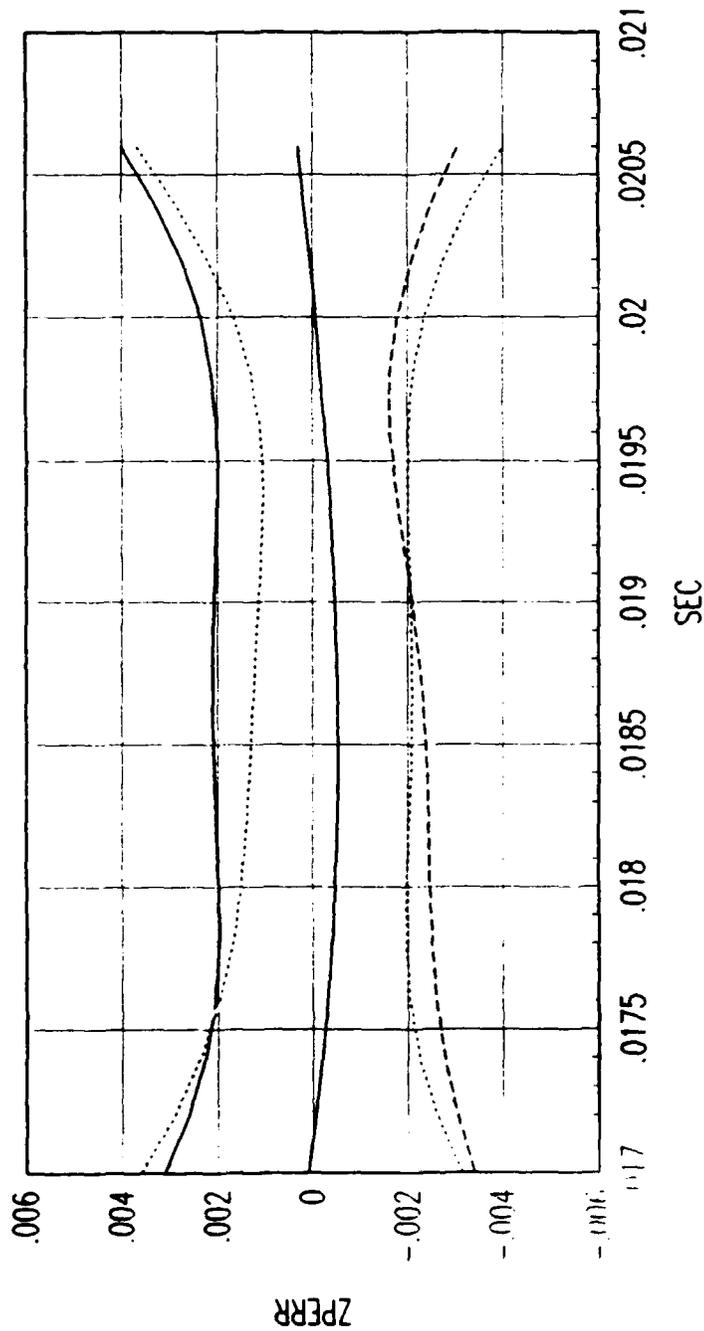


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.247. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.248. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

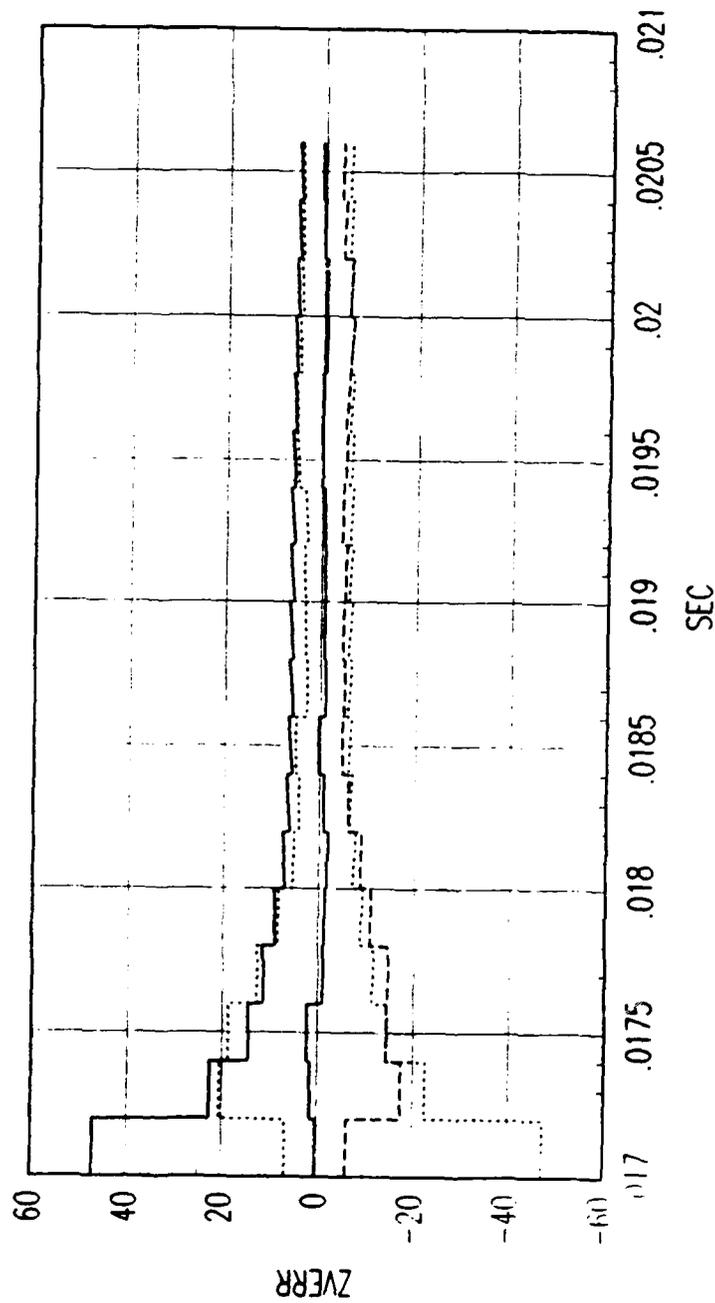


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.249. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

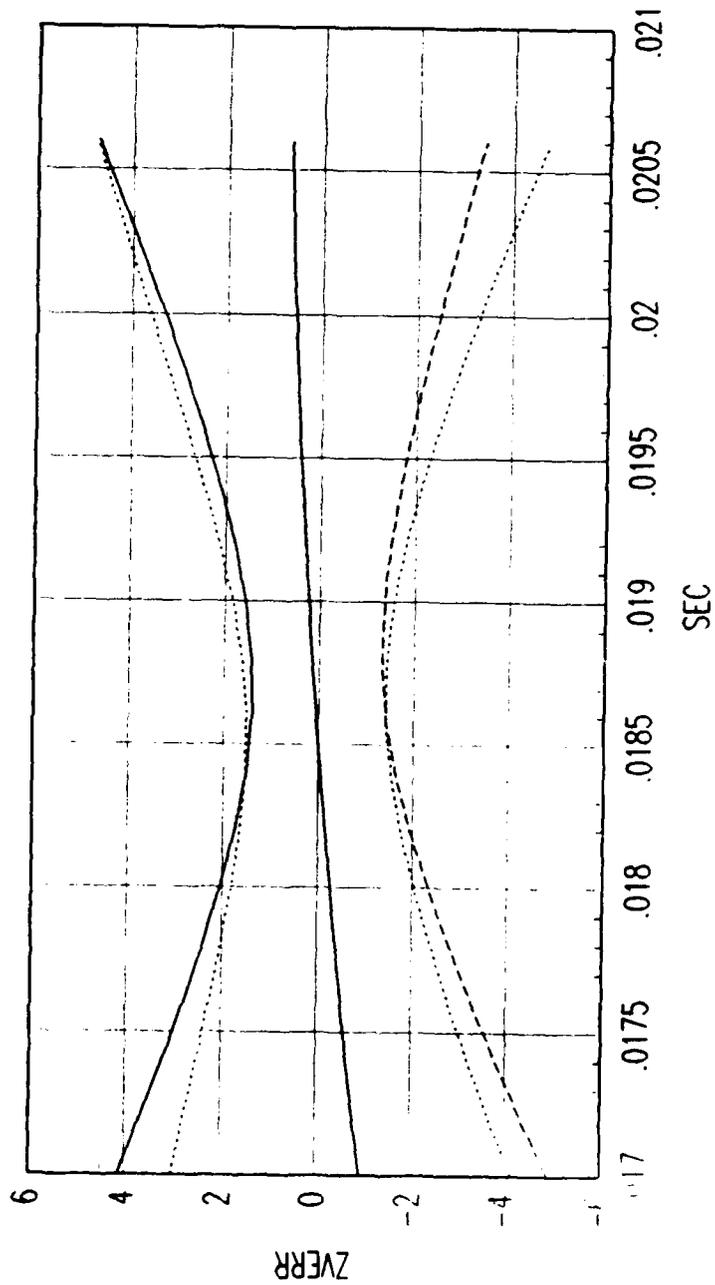
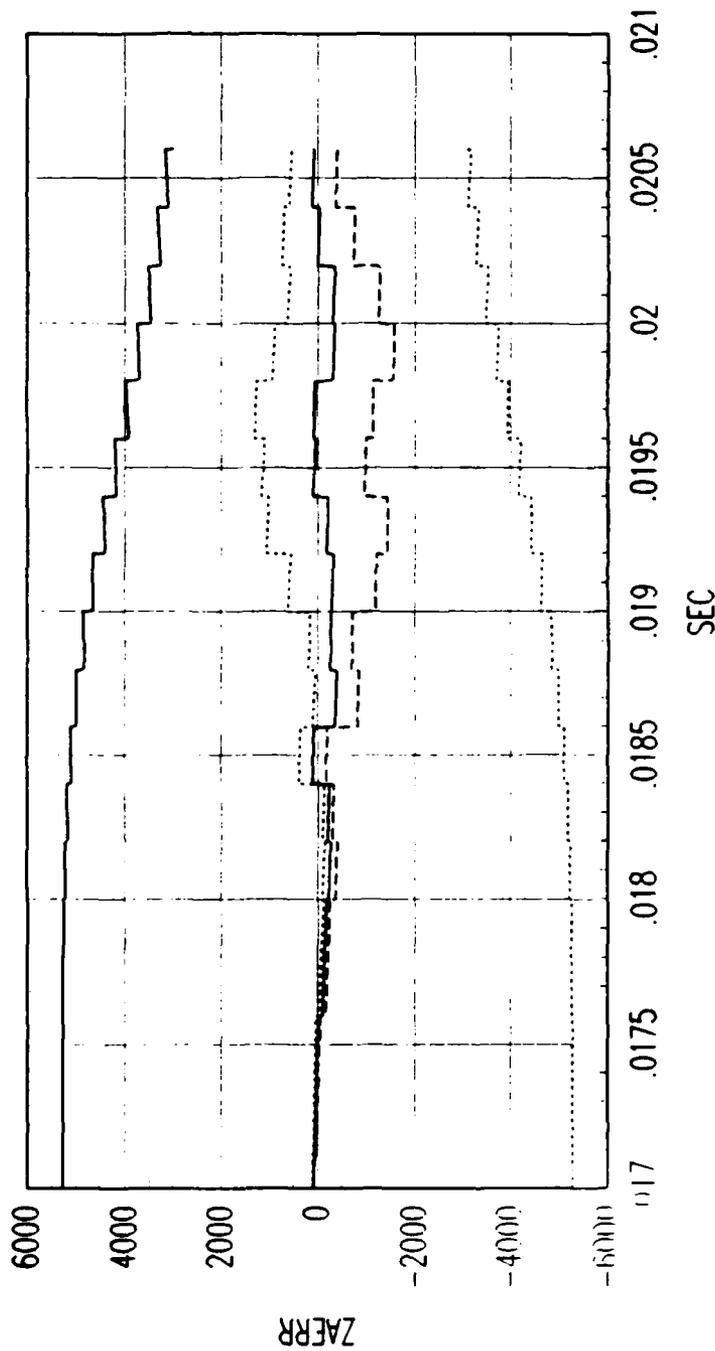


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.250. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.251. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

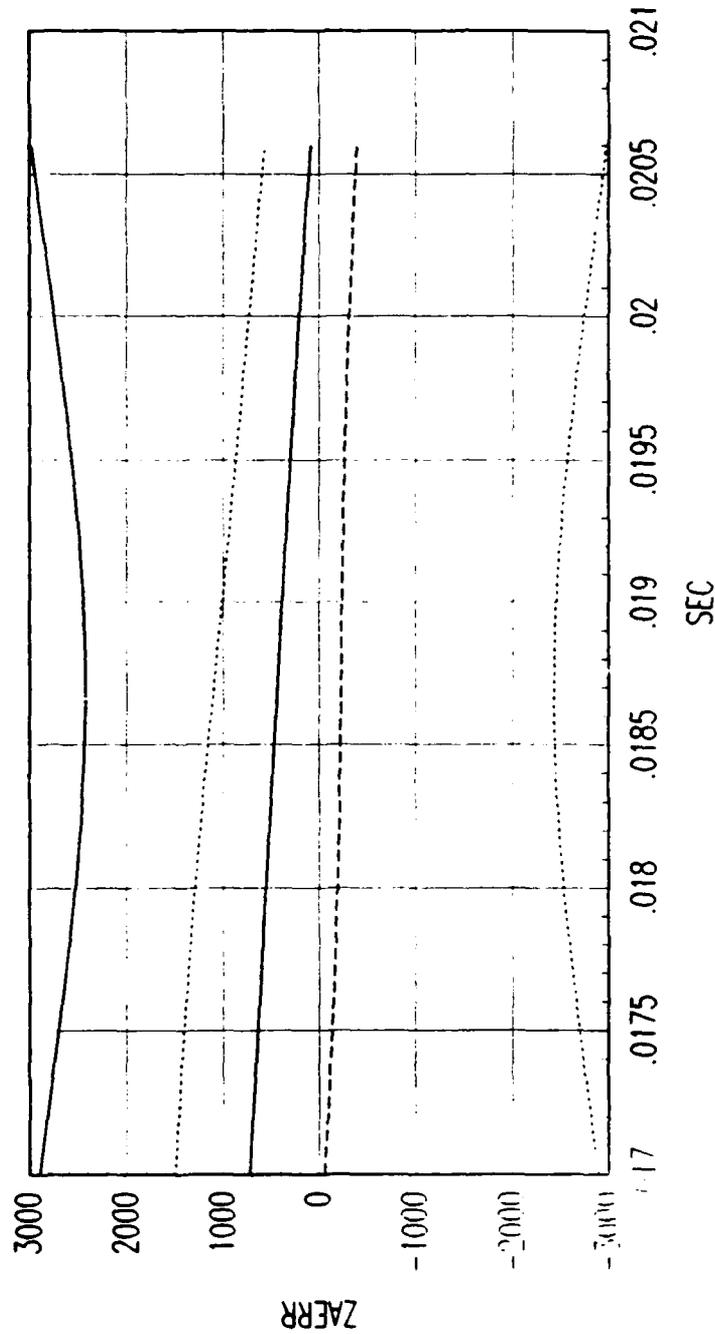


FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.252. Tracking Error Plot, Category 3, $v(t_0) = 3000$ ft/sec.

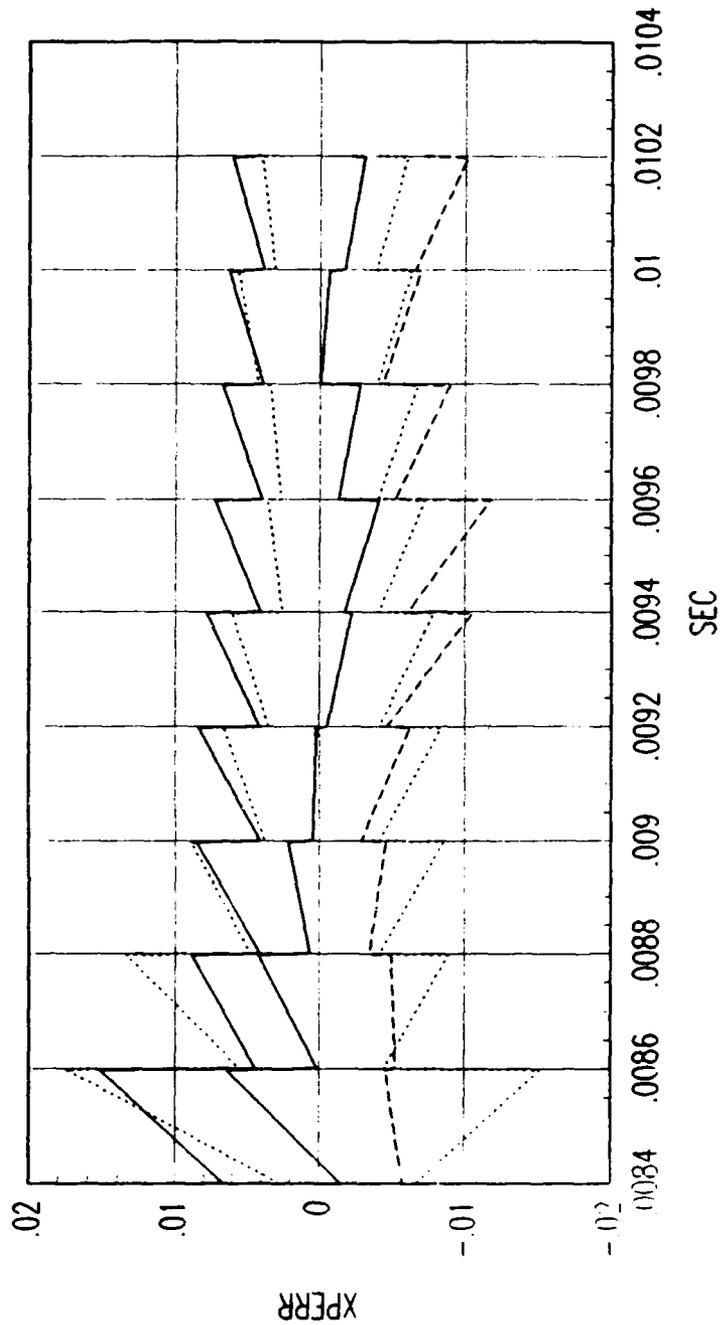


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.253. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

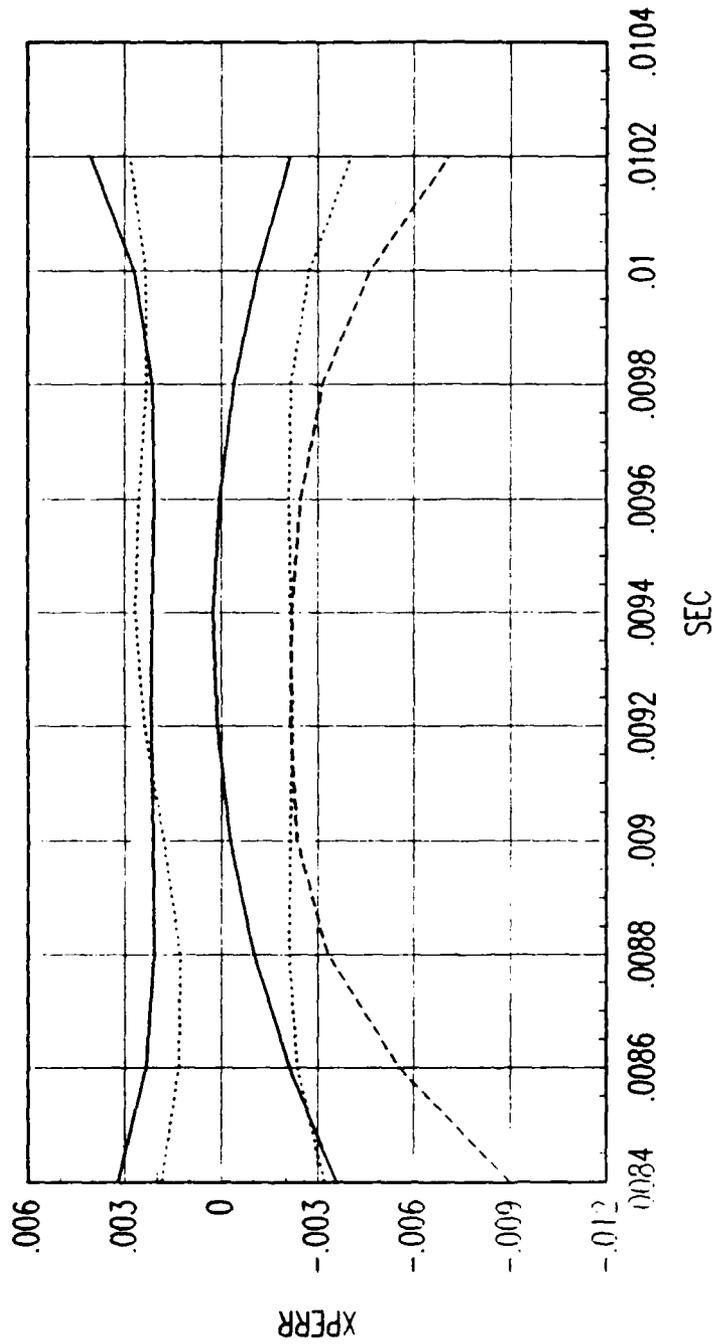
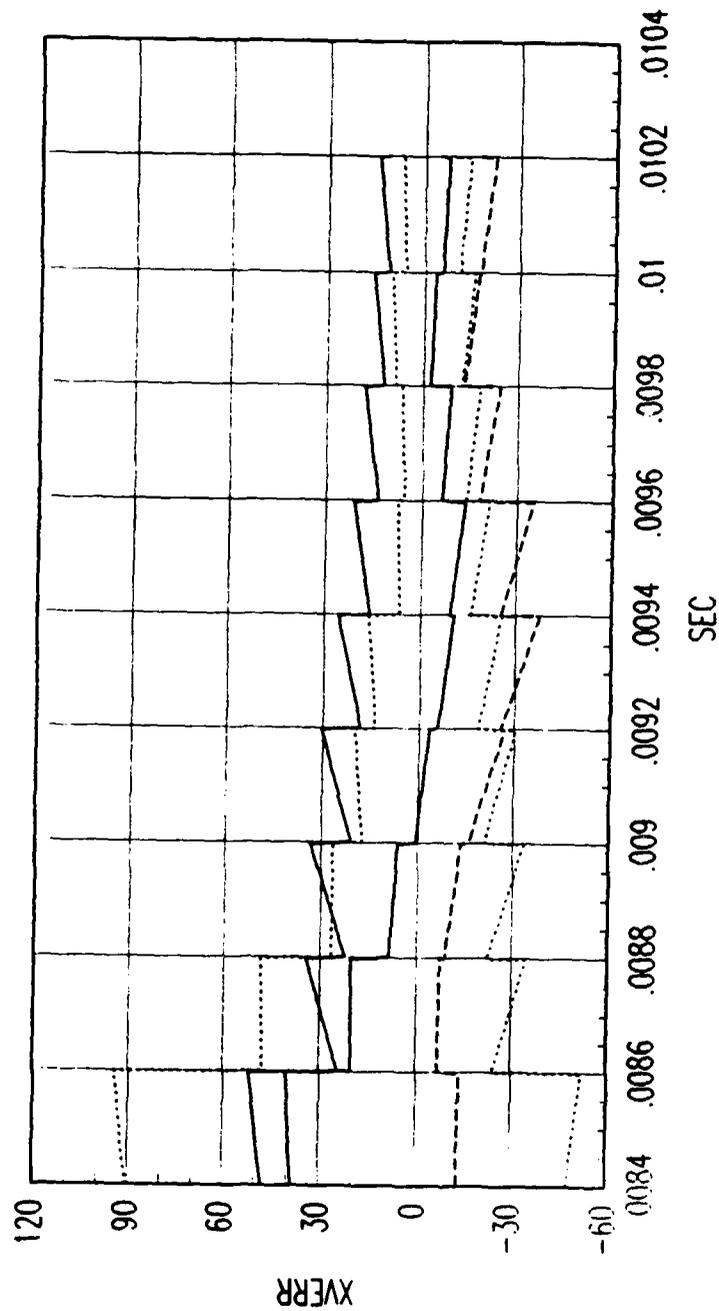


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.254. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.255. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

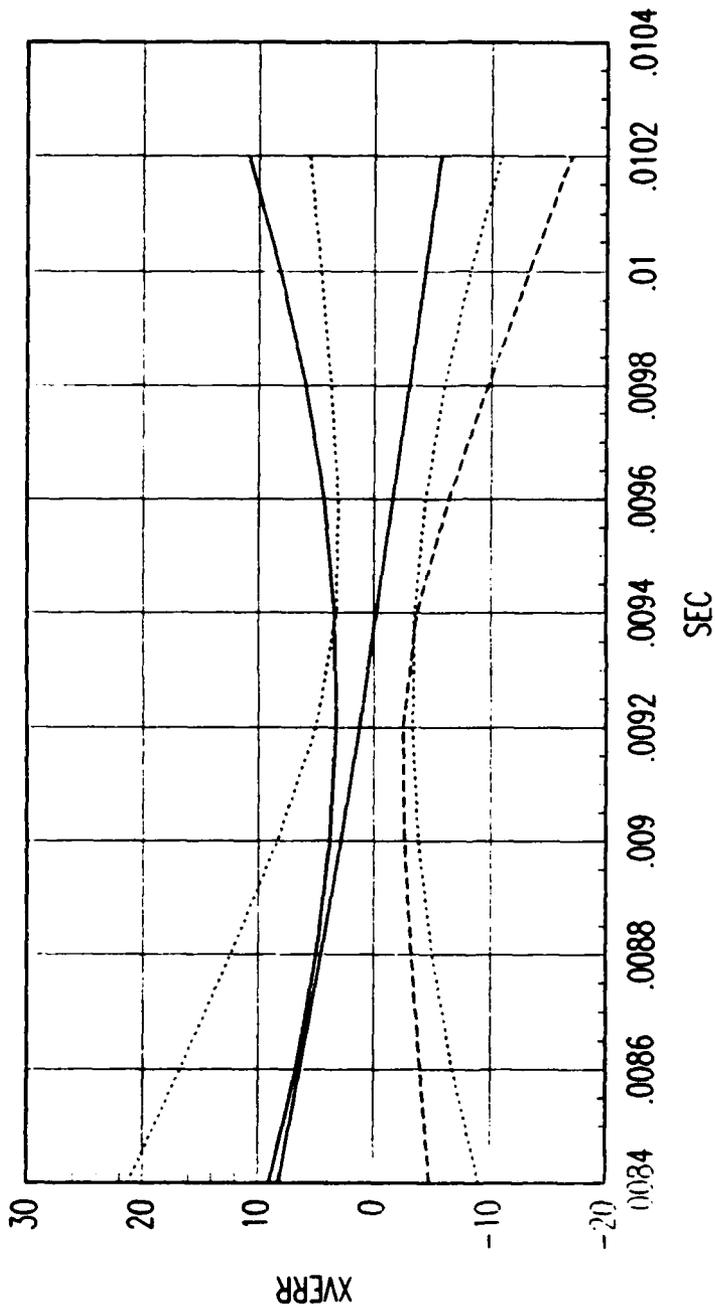


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.256. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

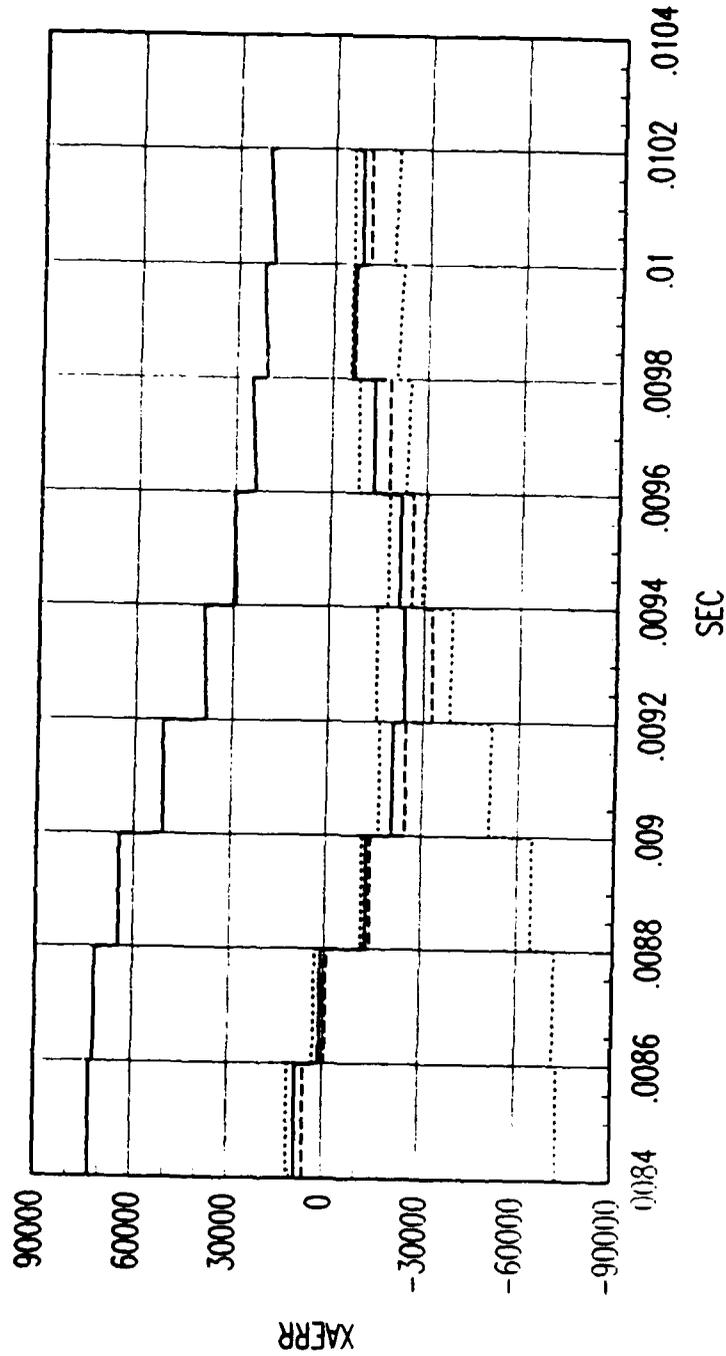


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.257. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

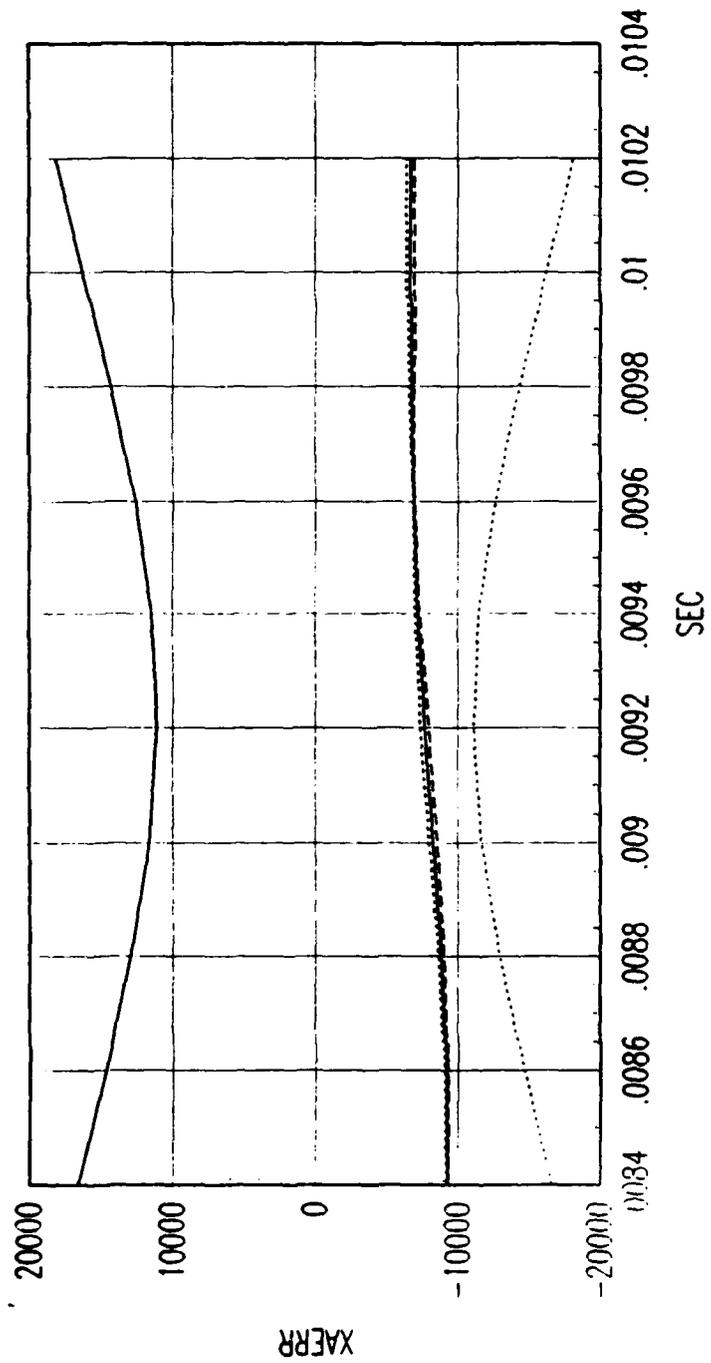


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.258. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

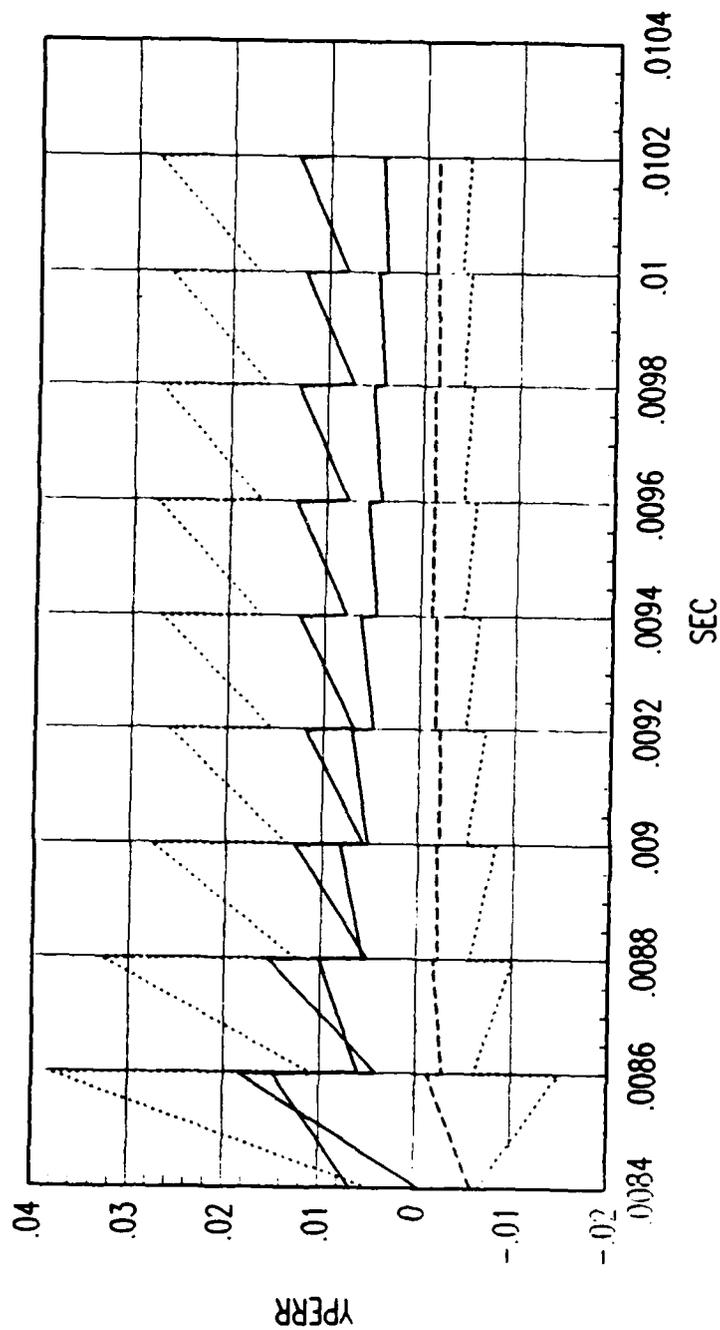


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.259. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

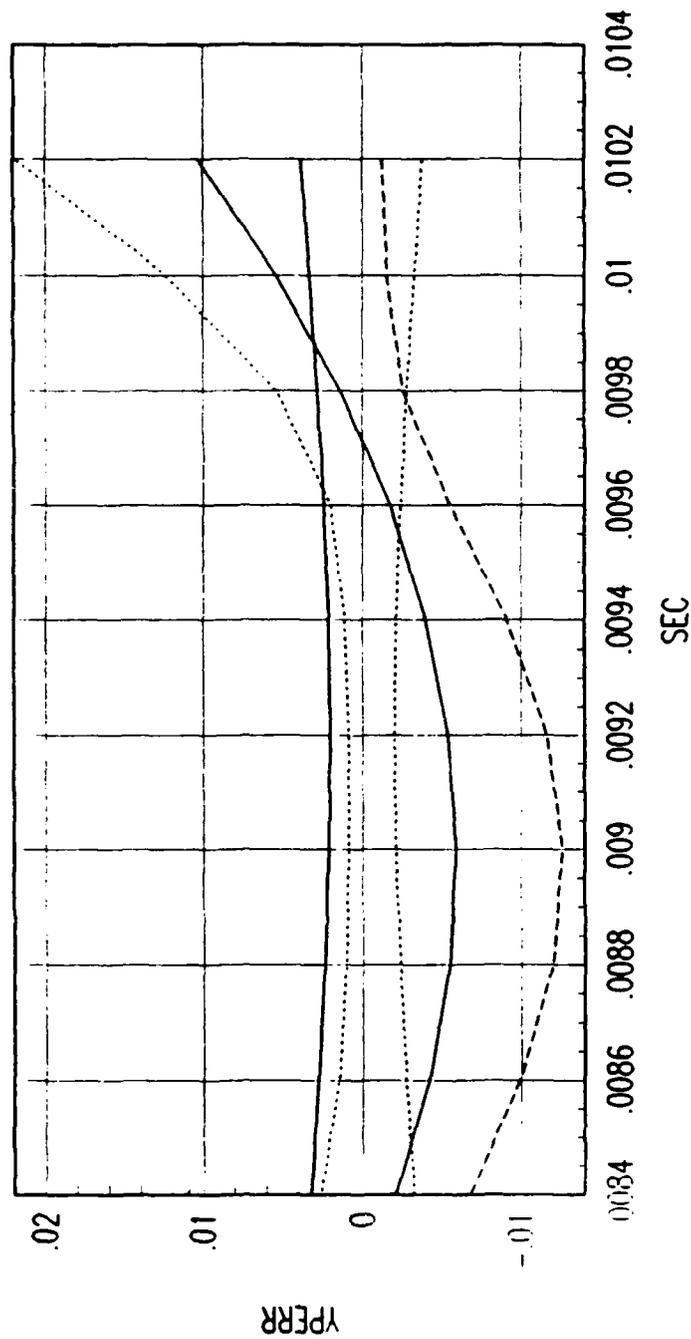


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.260. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

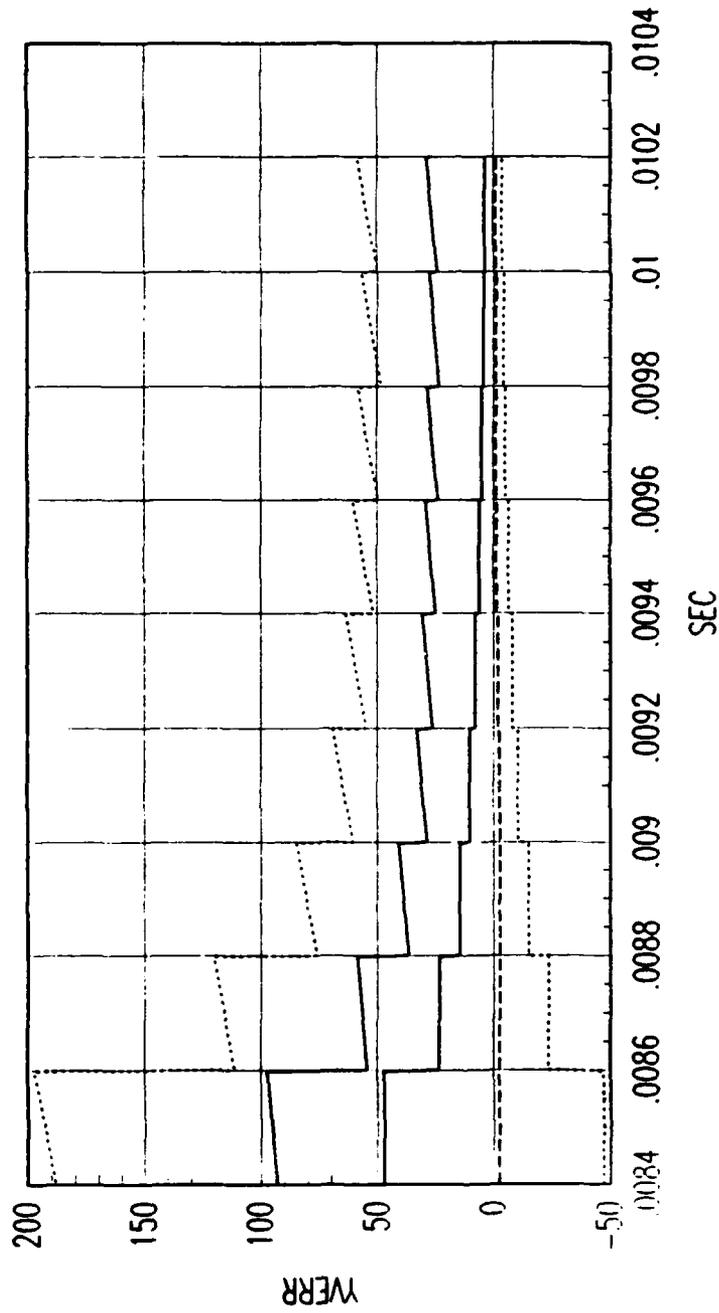


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.261. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

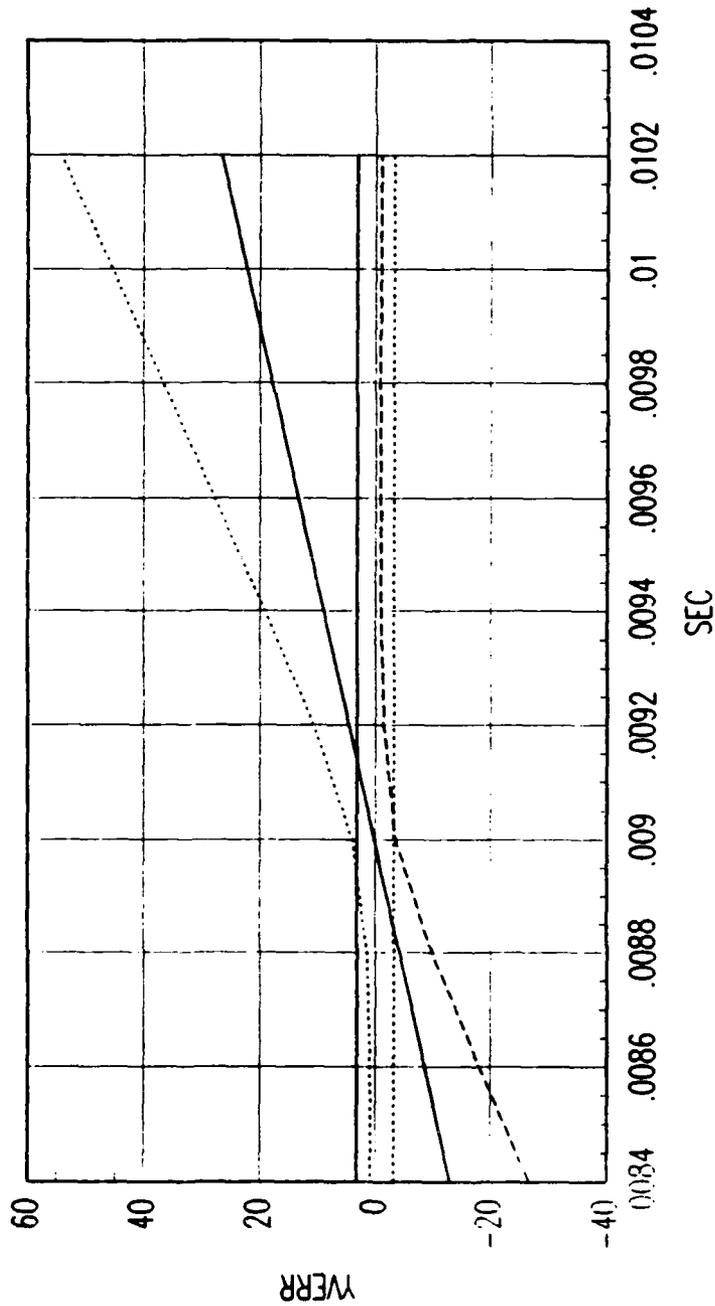
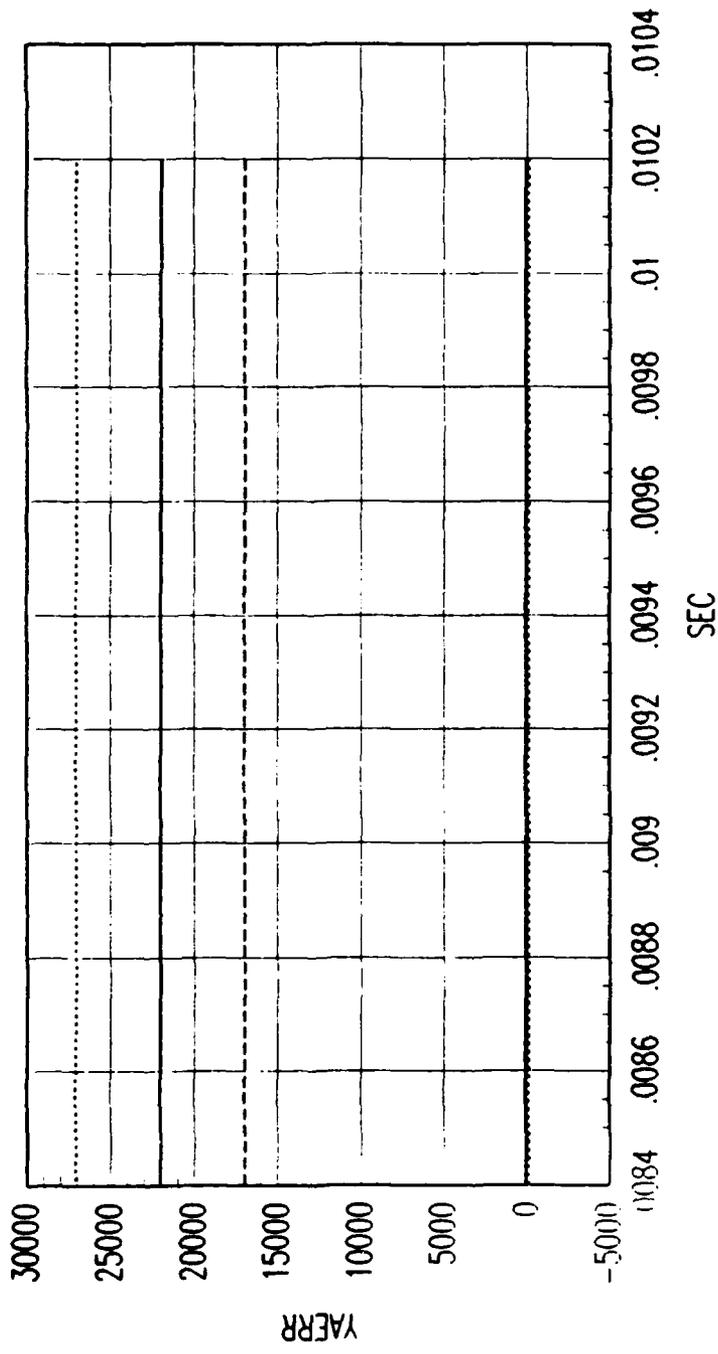


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.262. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.263. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

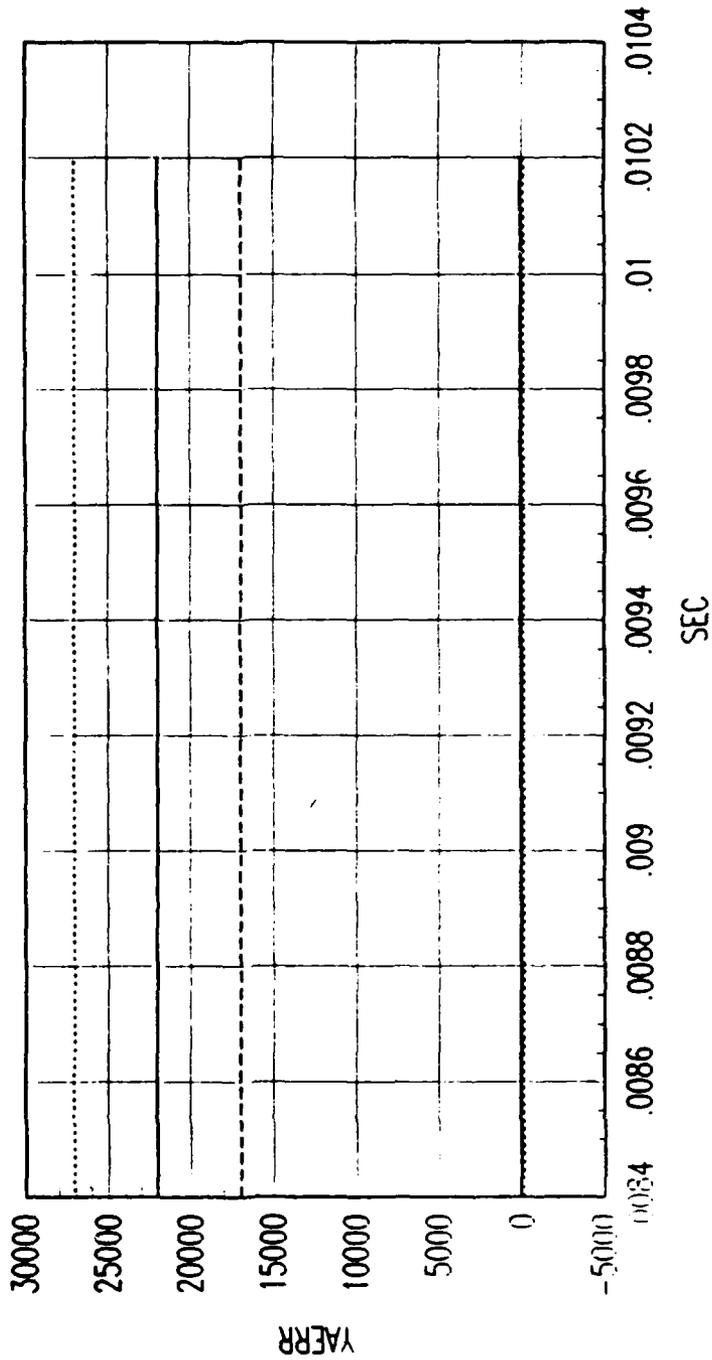


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.264. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

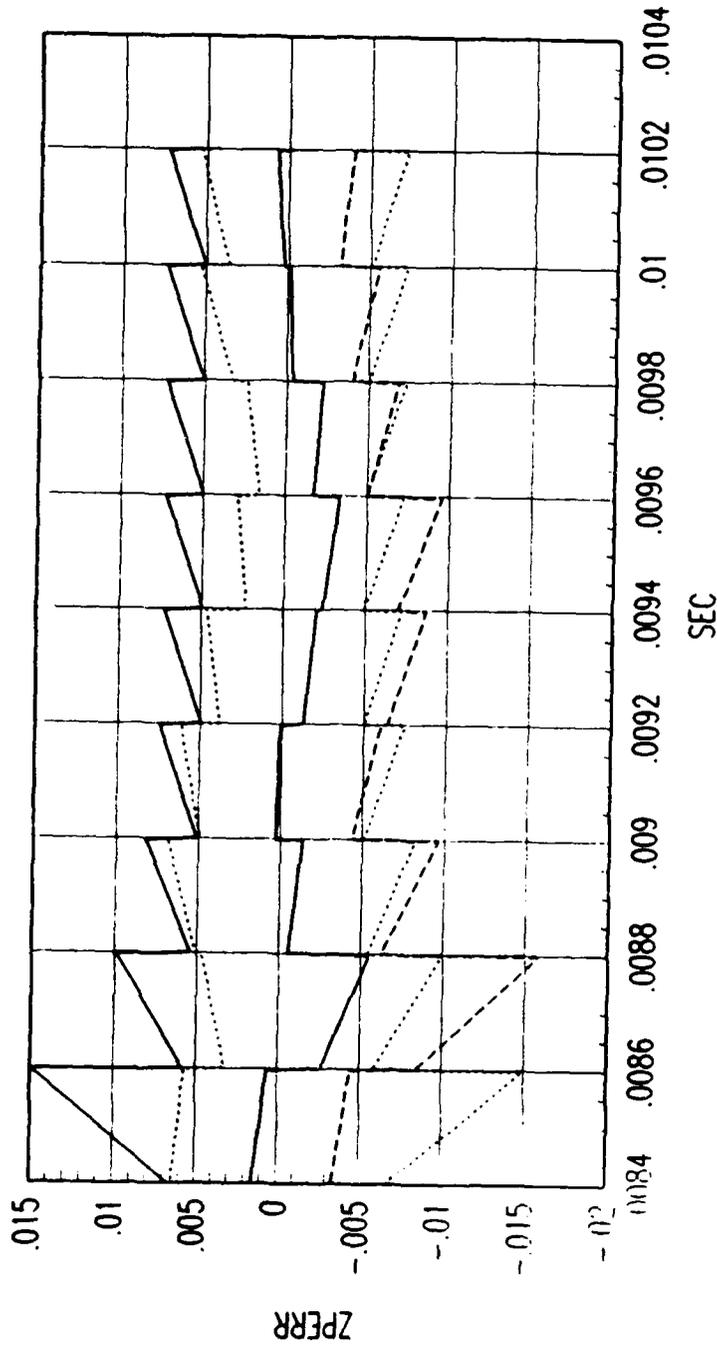
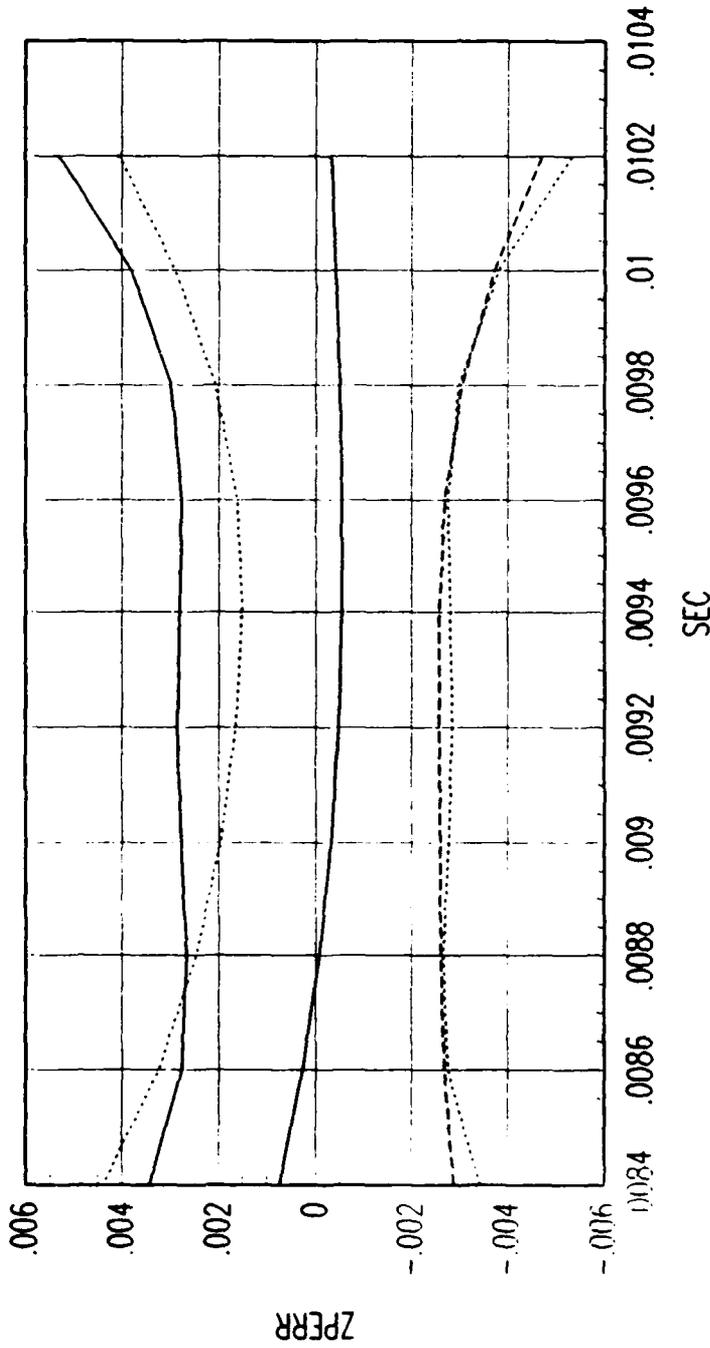


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.265. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.266. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

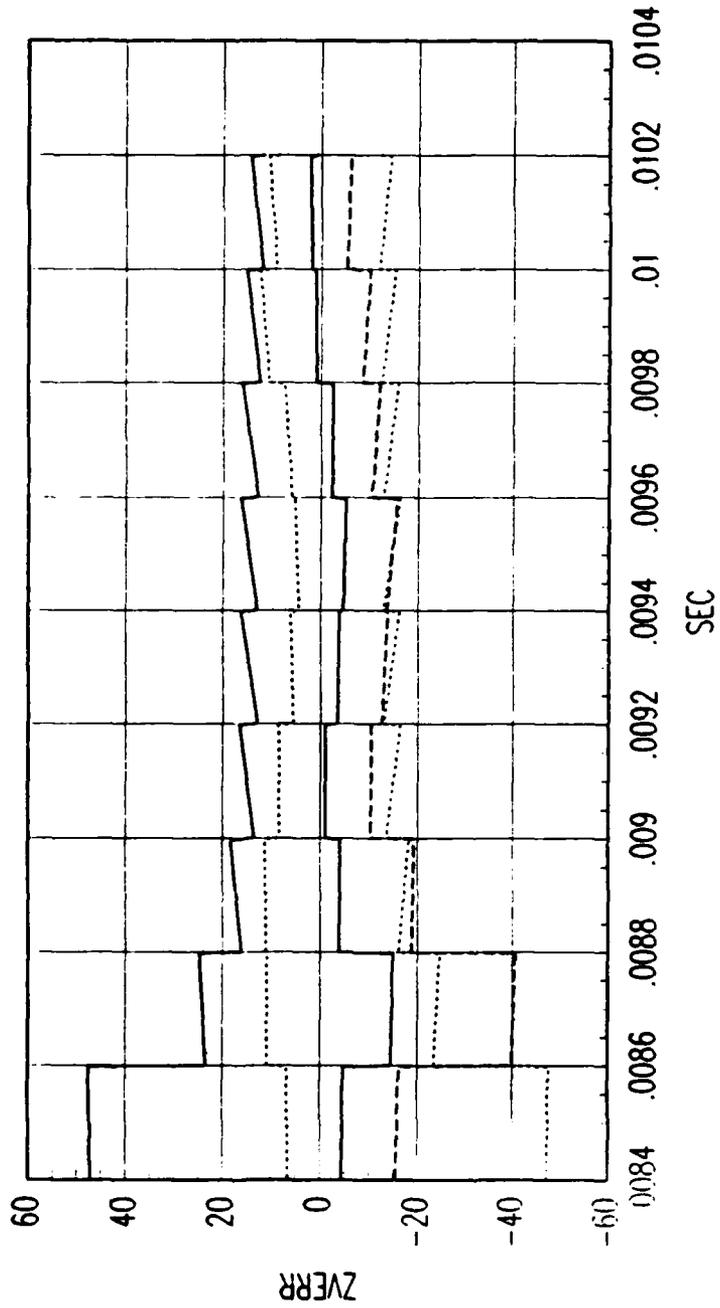
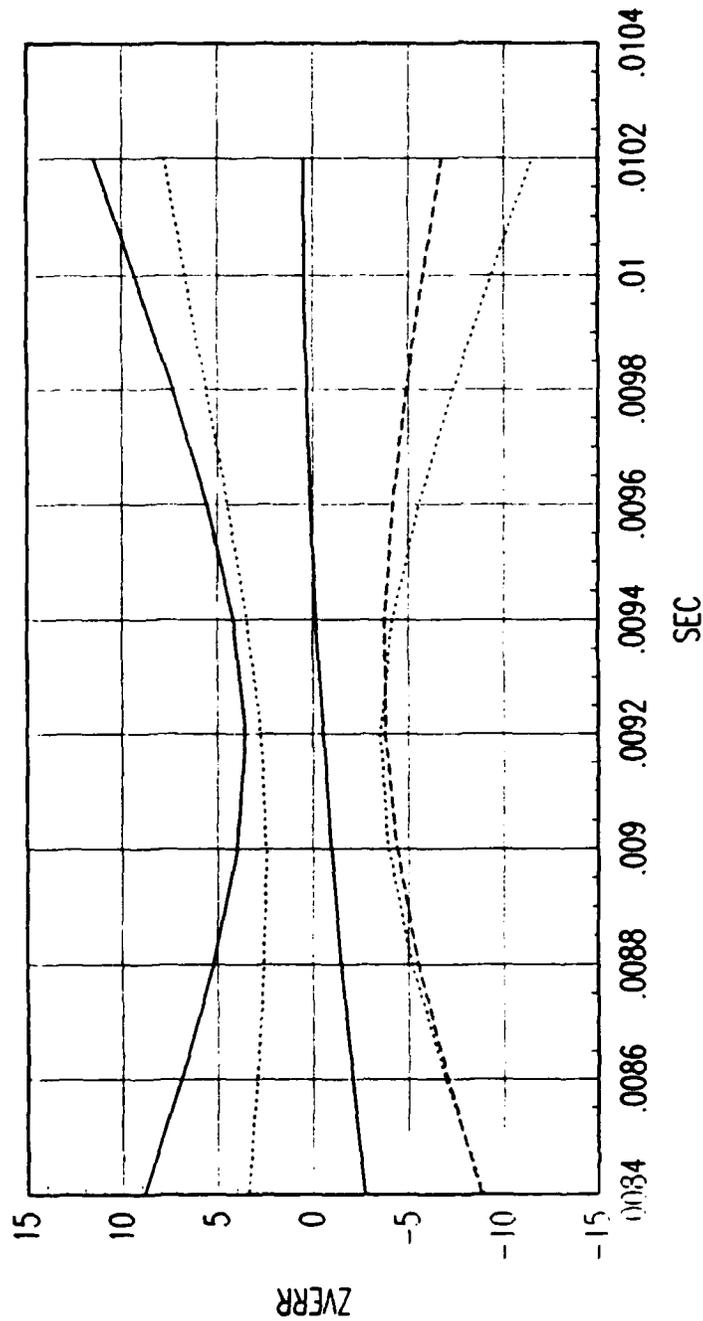


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.267. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.268. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

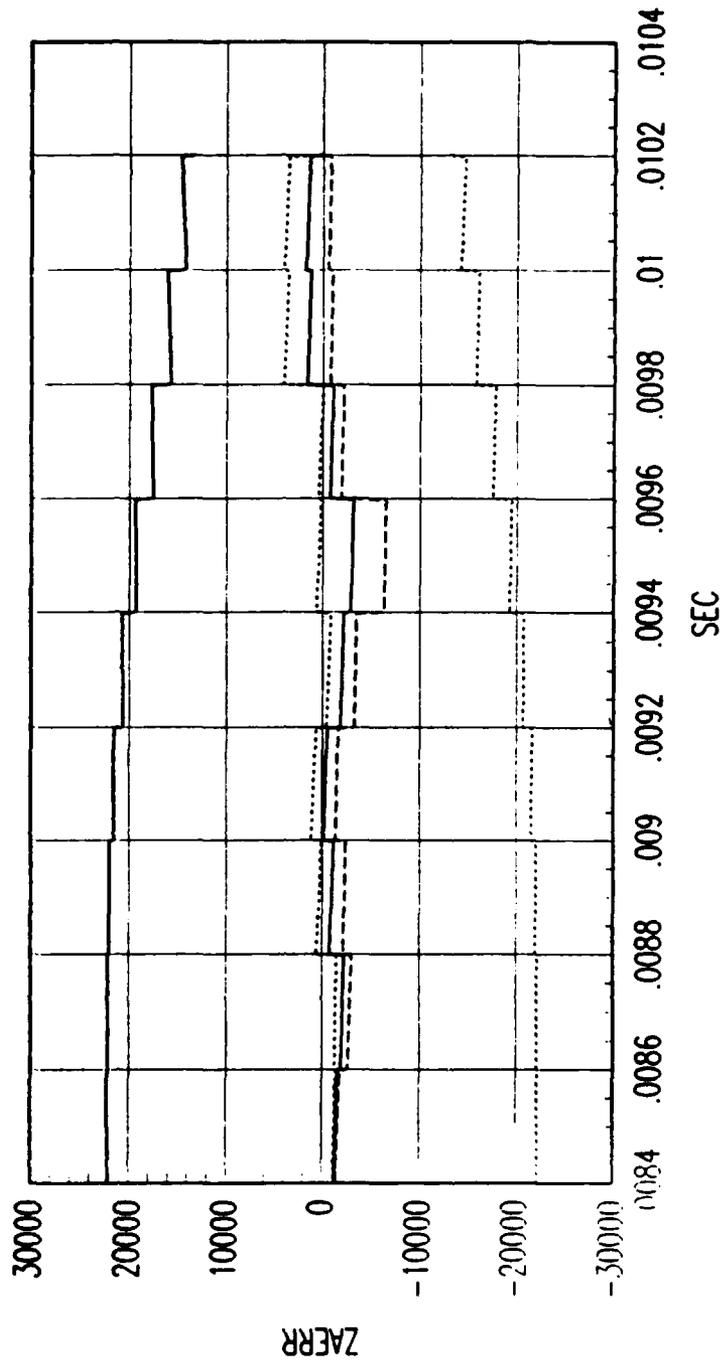


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.269. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

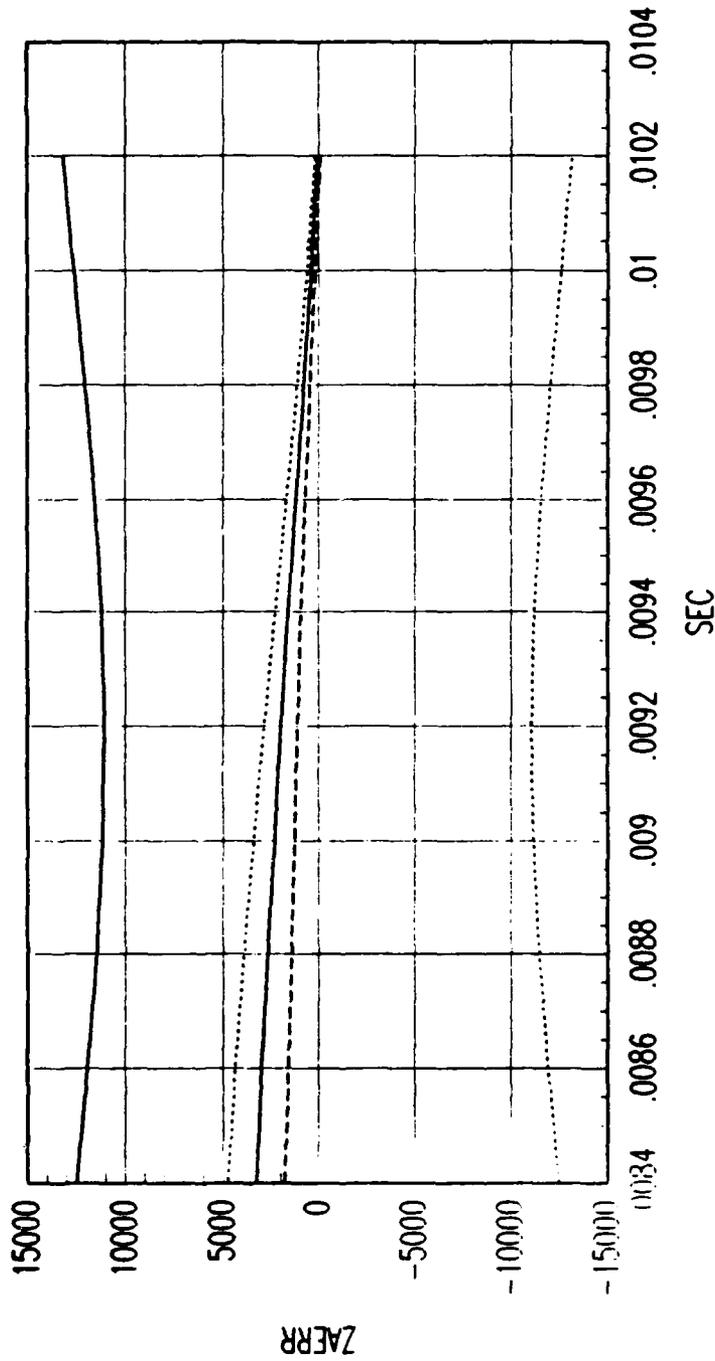
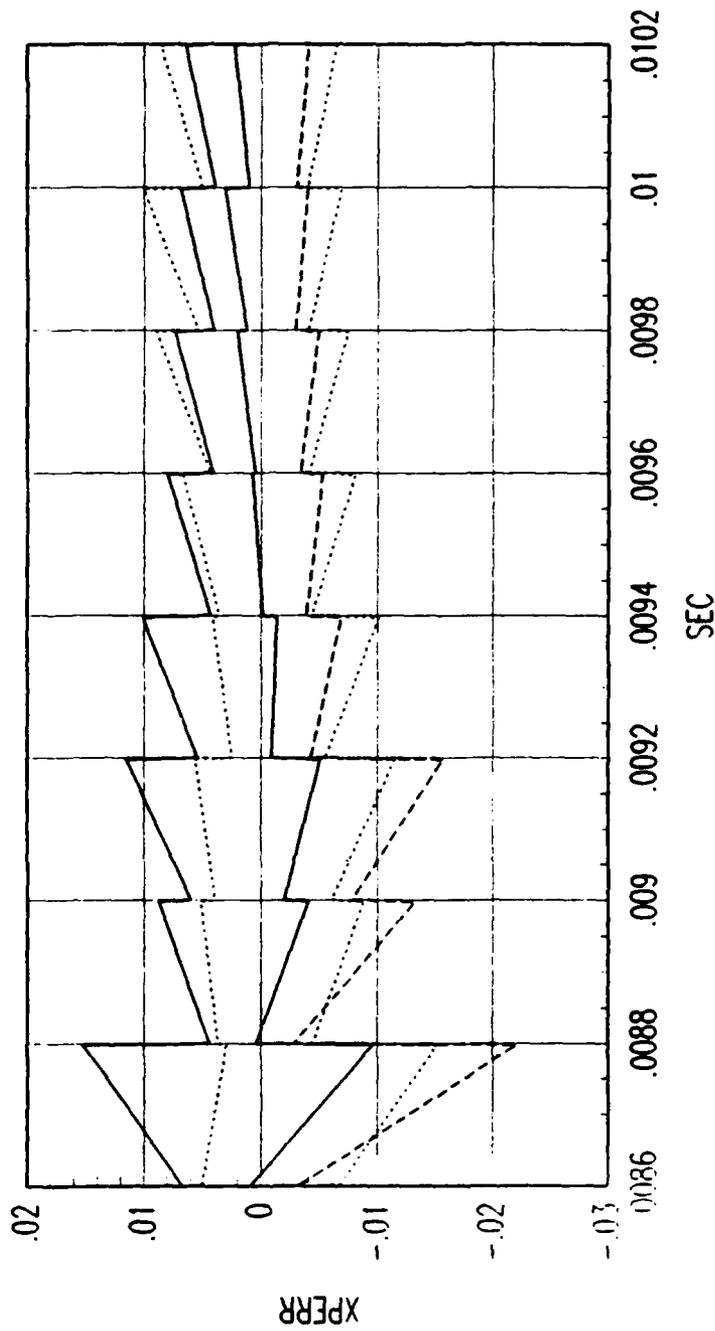


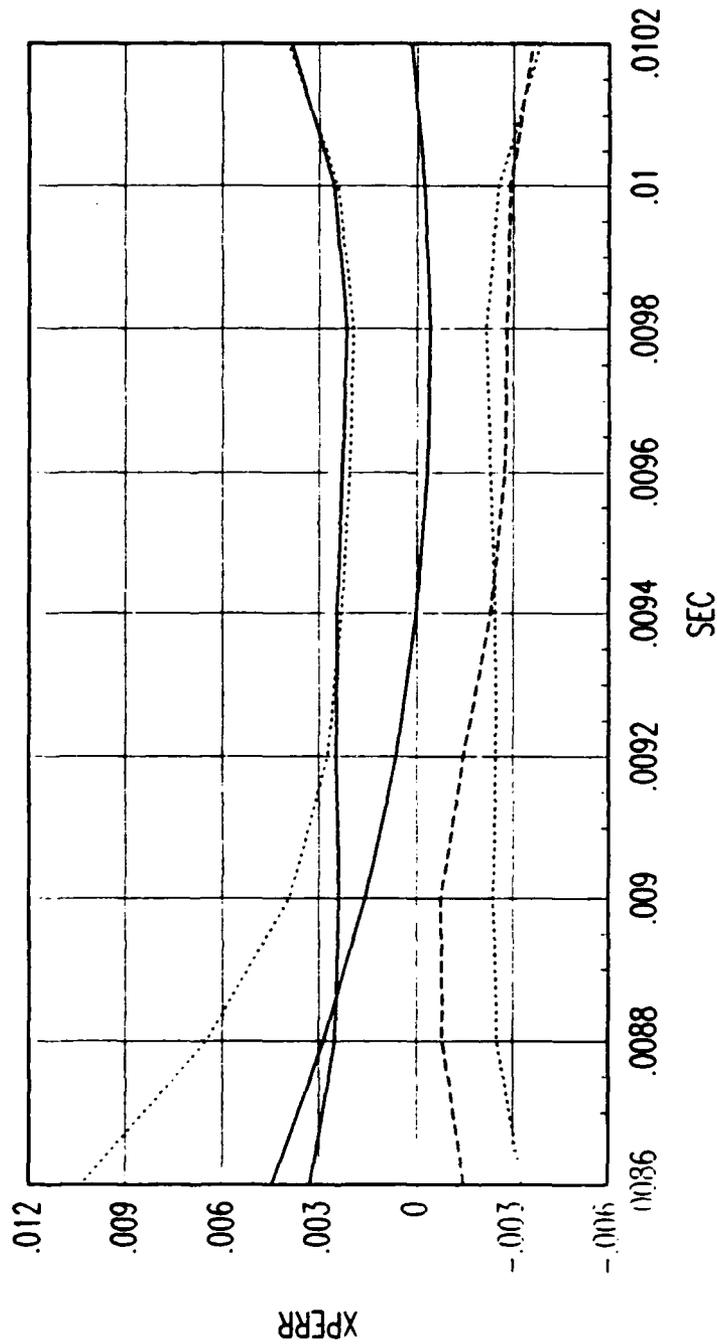
FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.270. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



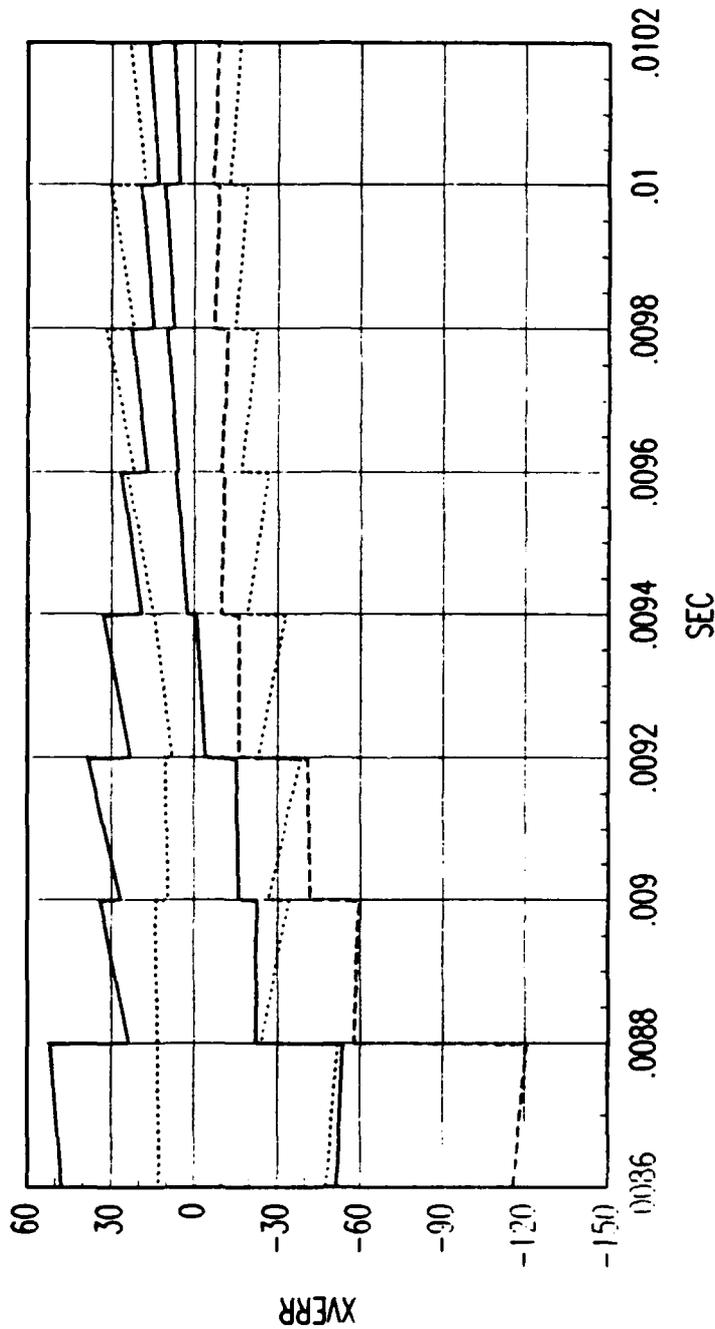
FRG. B FORWARD TIME EKF POSITION ERROR

Figure C.271. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



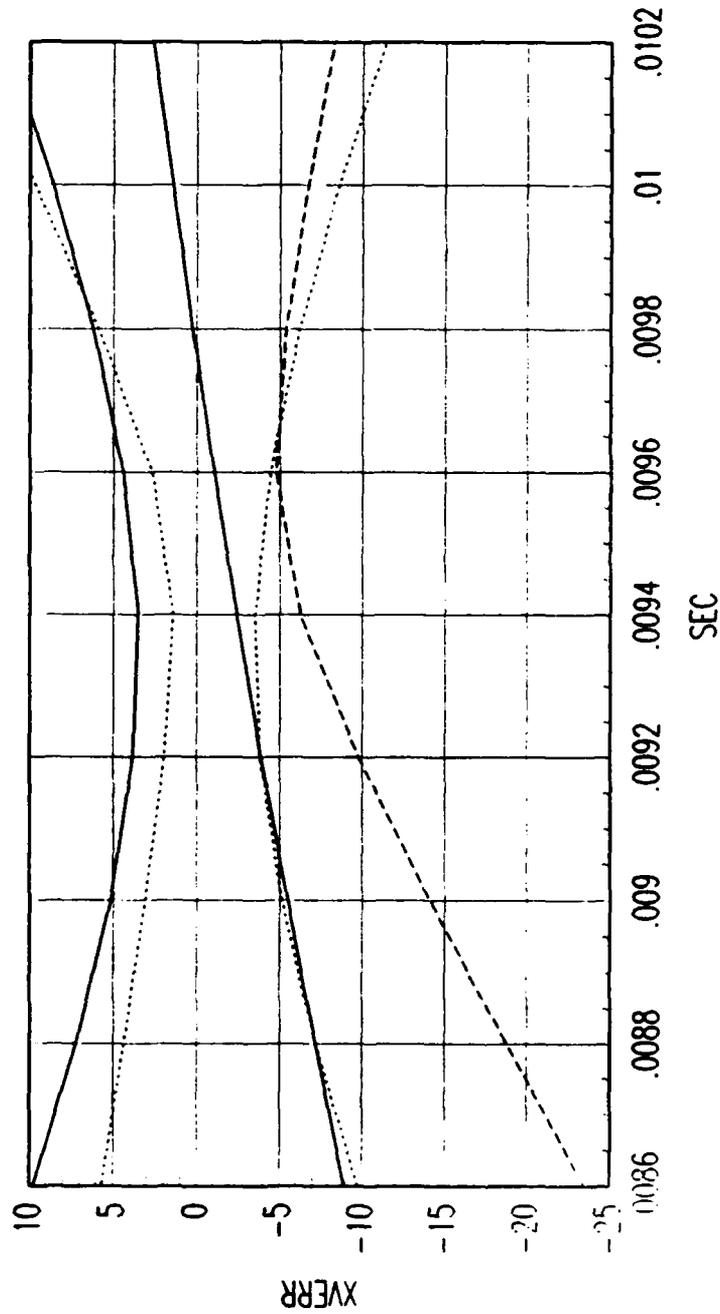
FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.272. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



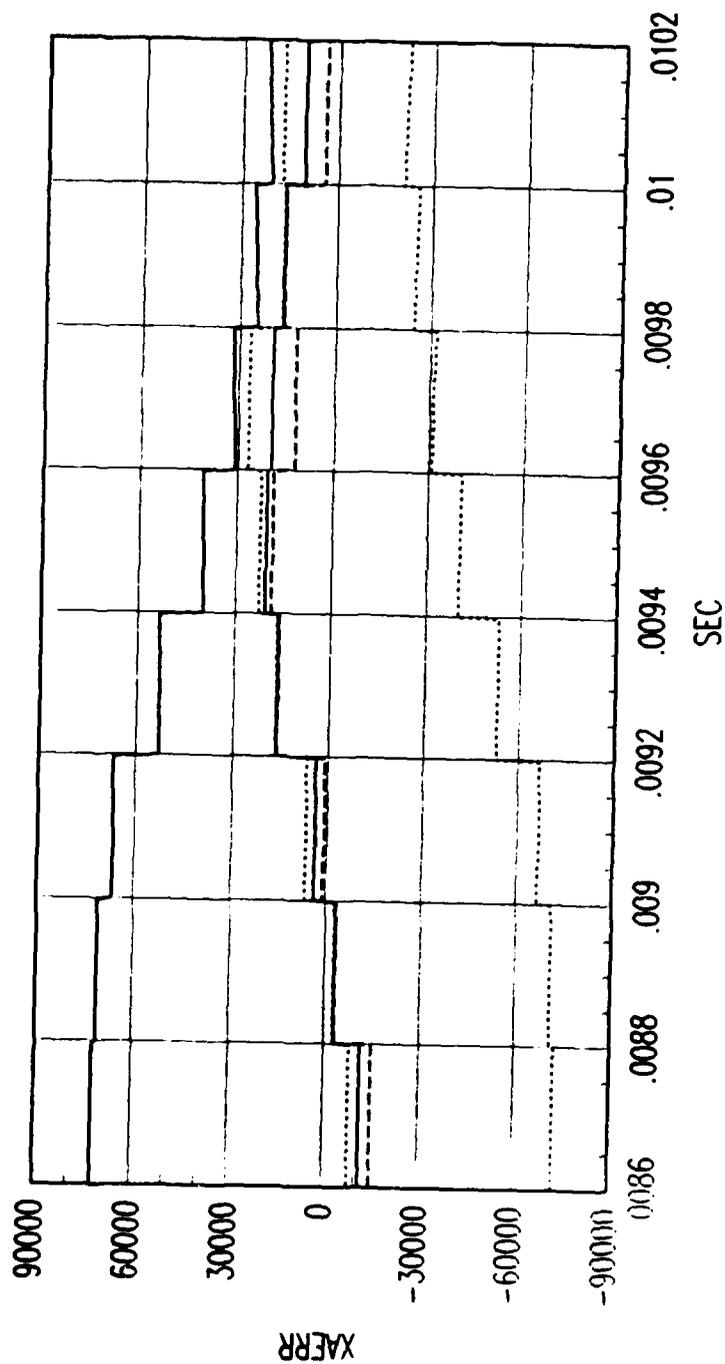
FRG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.273. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



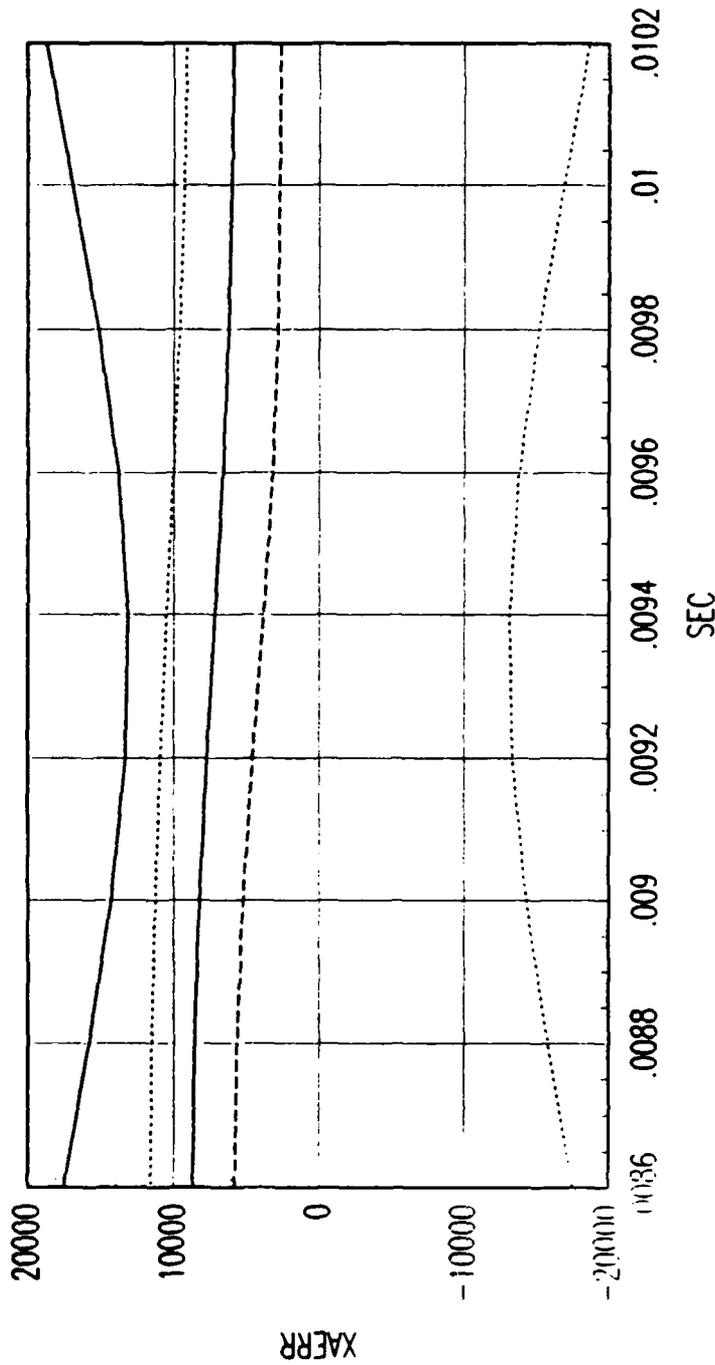
FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.274. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



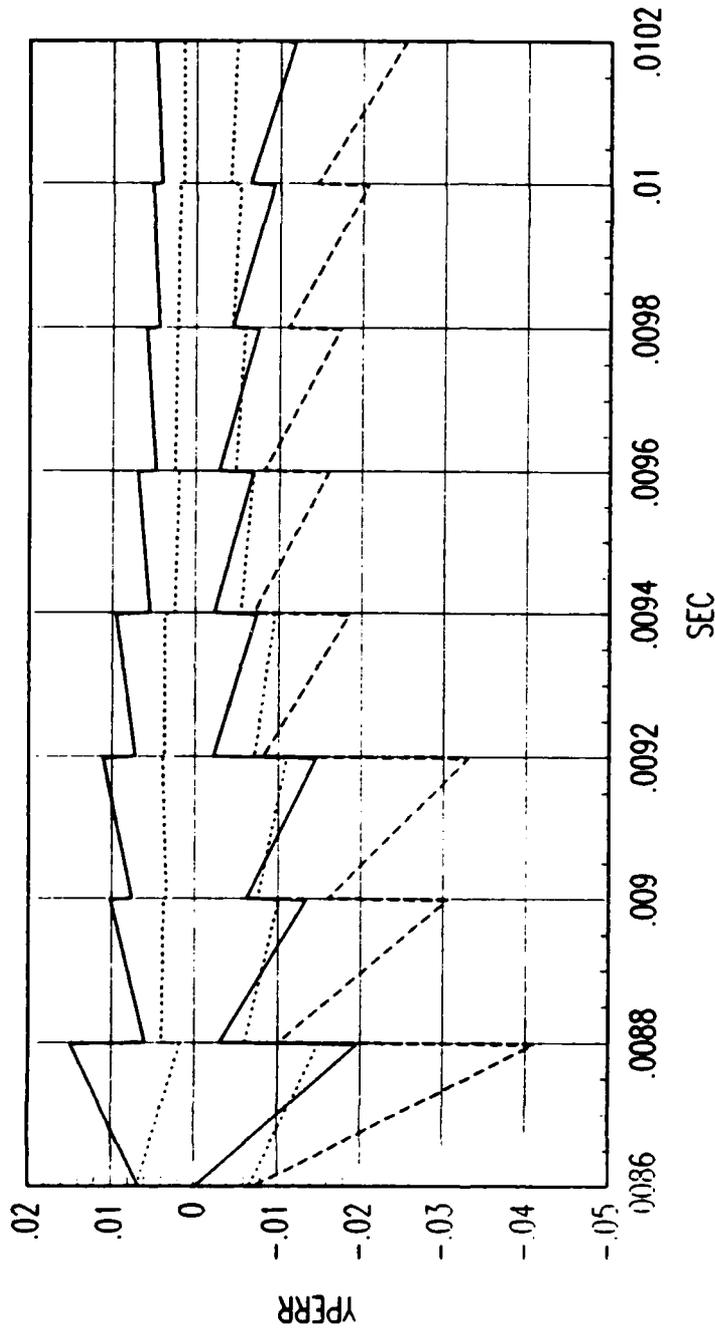
FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.275. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



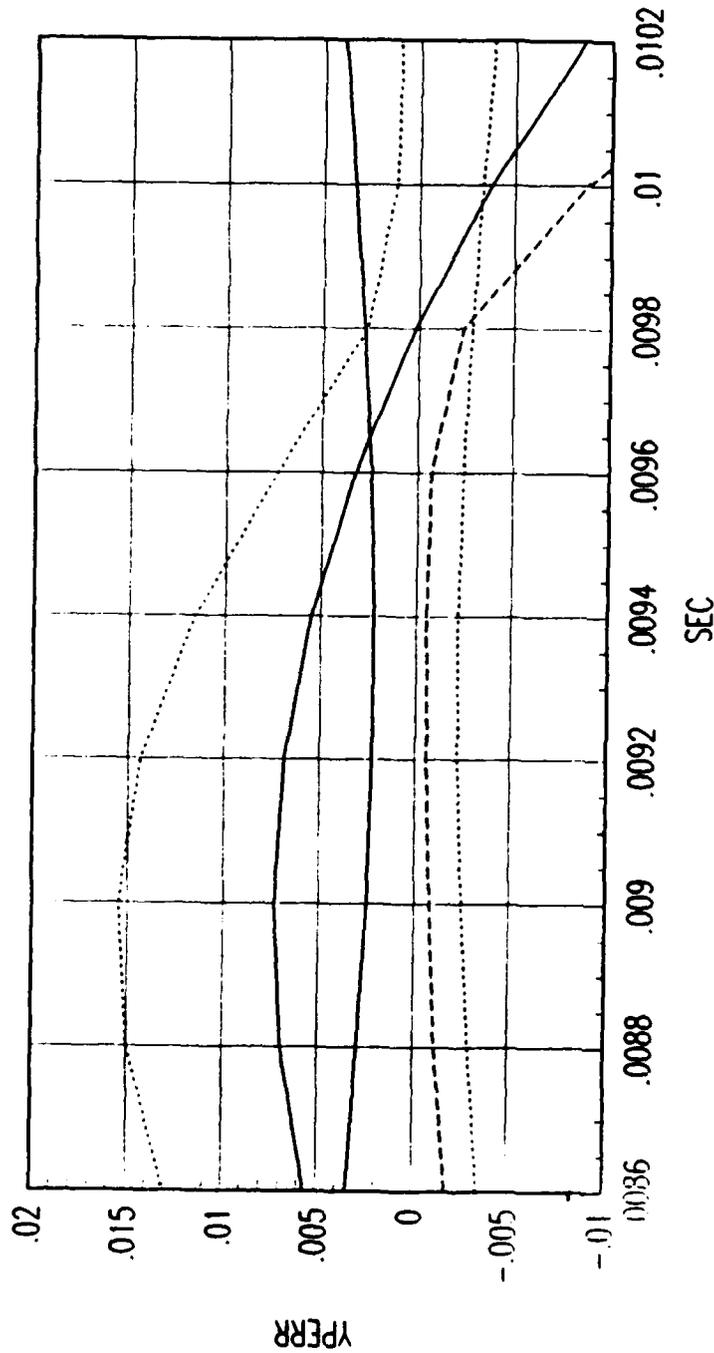
FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.276. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. B FORWARD TIME EKF POSITION ERROR

Figure C.277. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.278. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

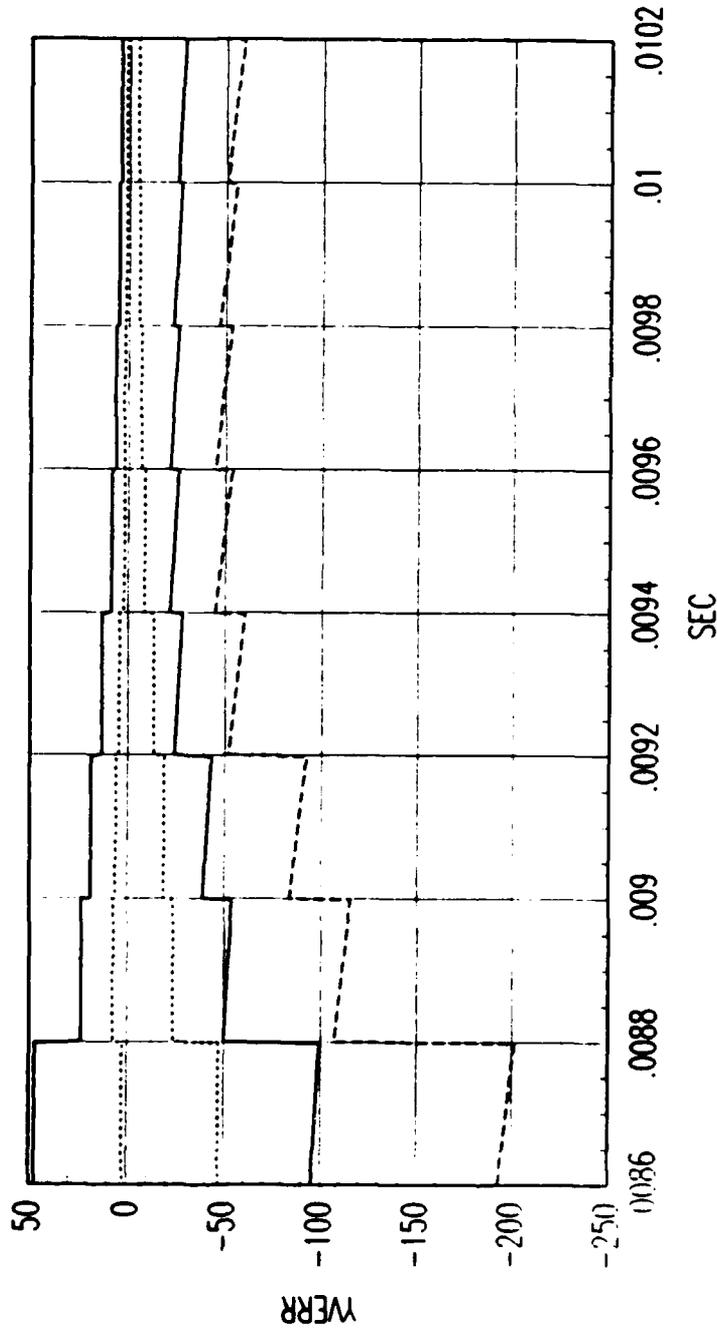


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.279. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

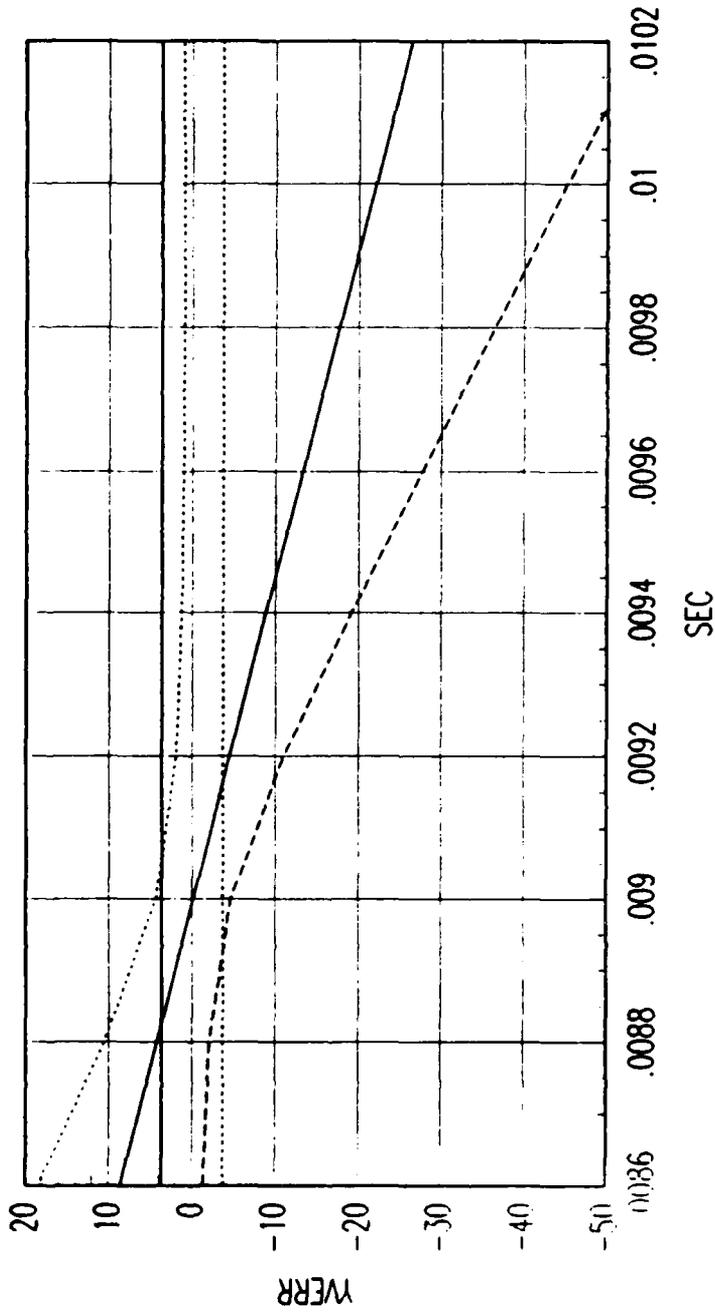


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.280. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

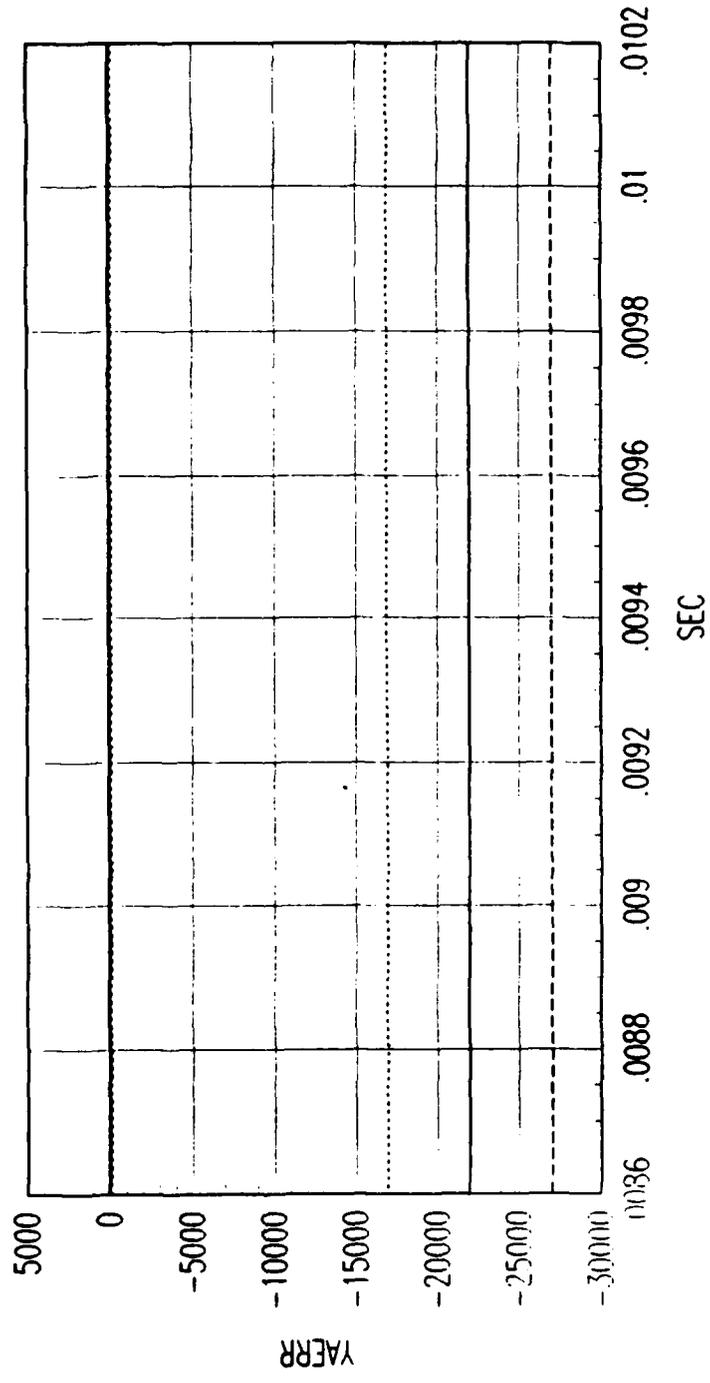
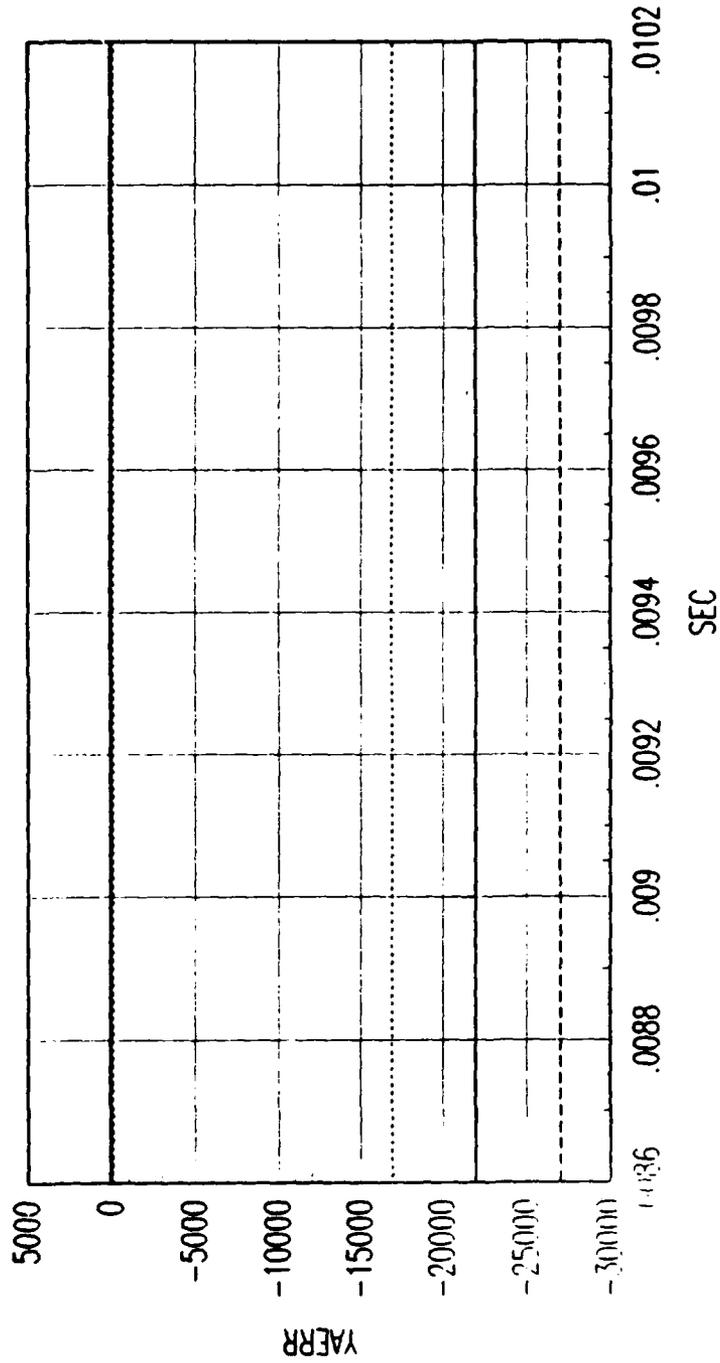


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.281. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.282. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

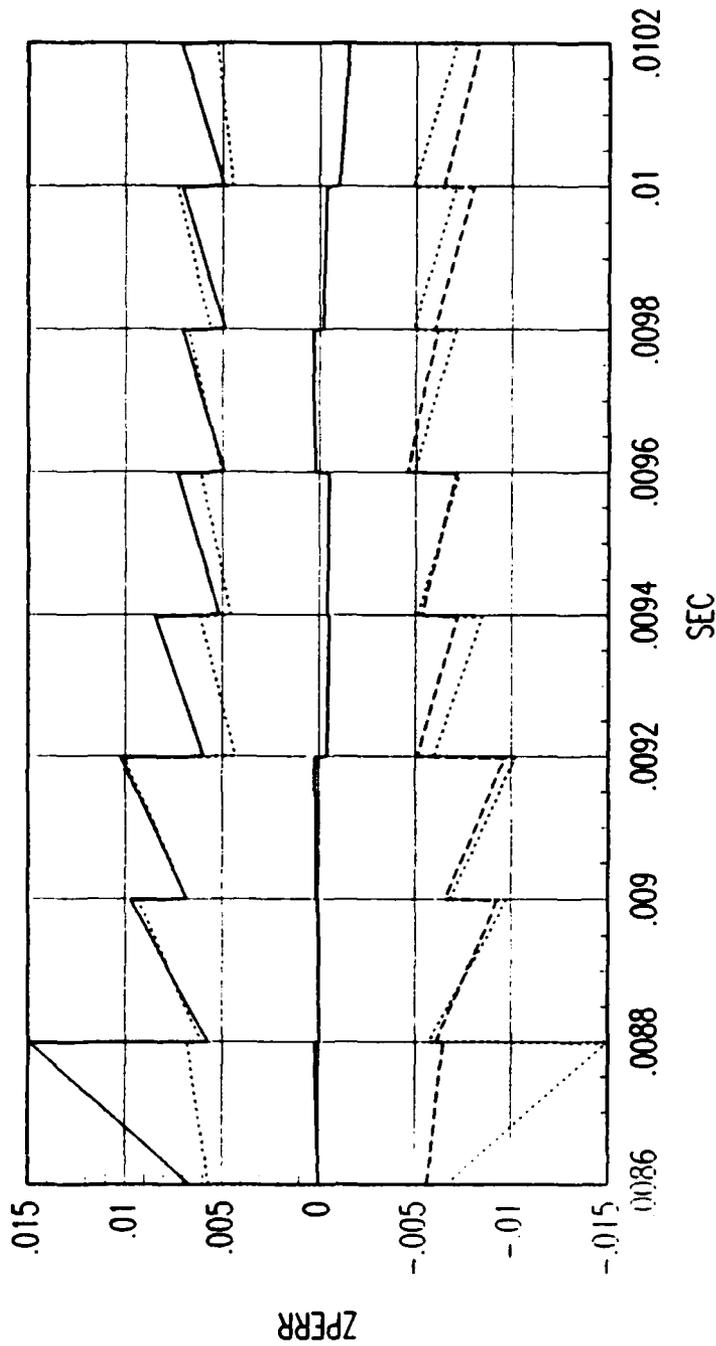


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.283. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

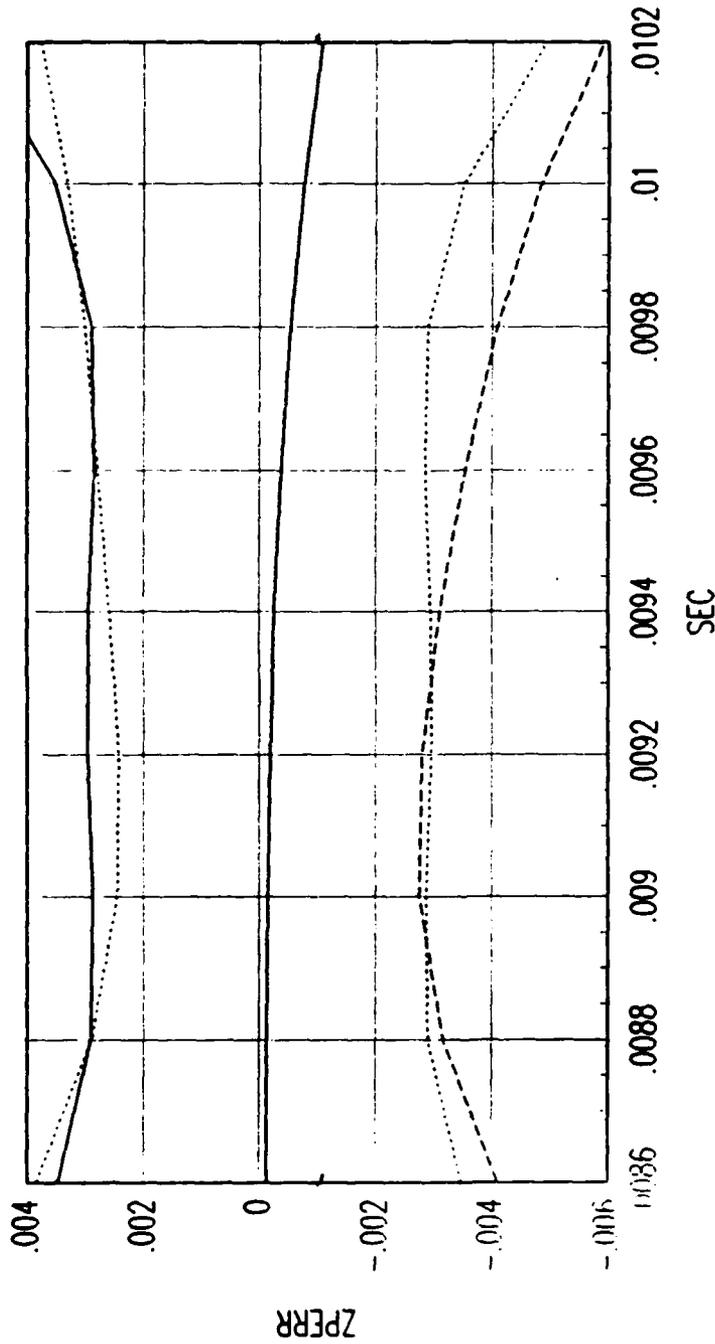


FIG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.284. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

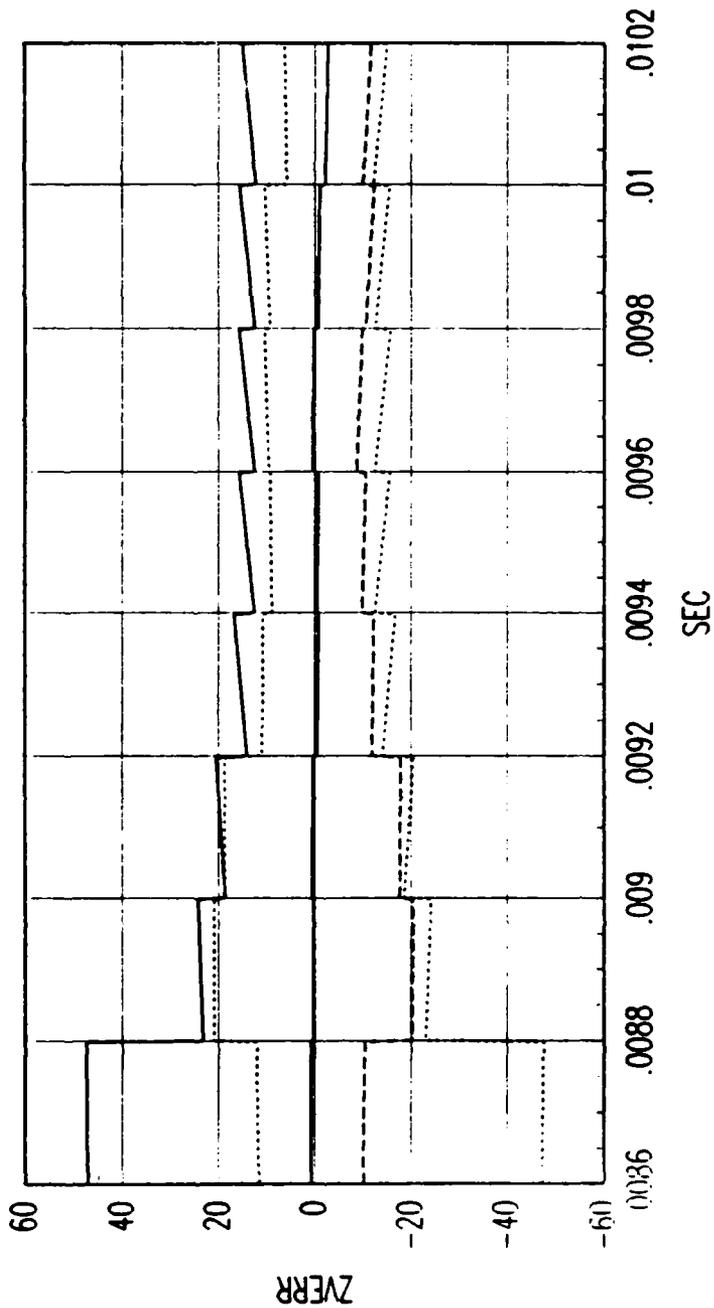


FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.285. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

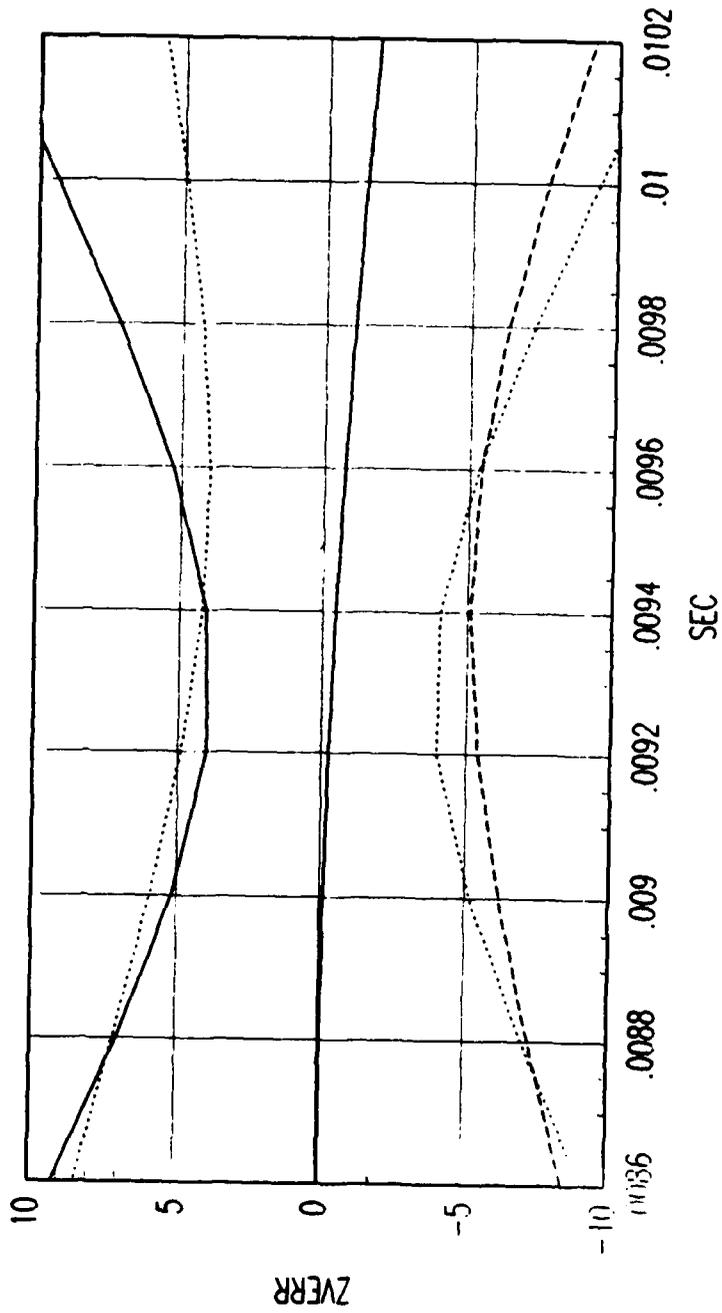


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.286. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

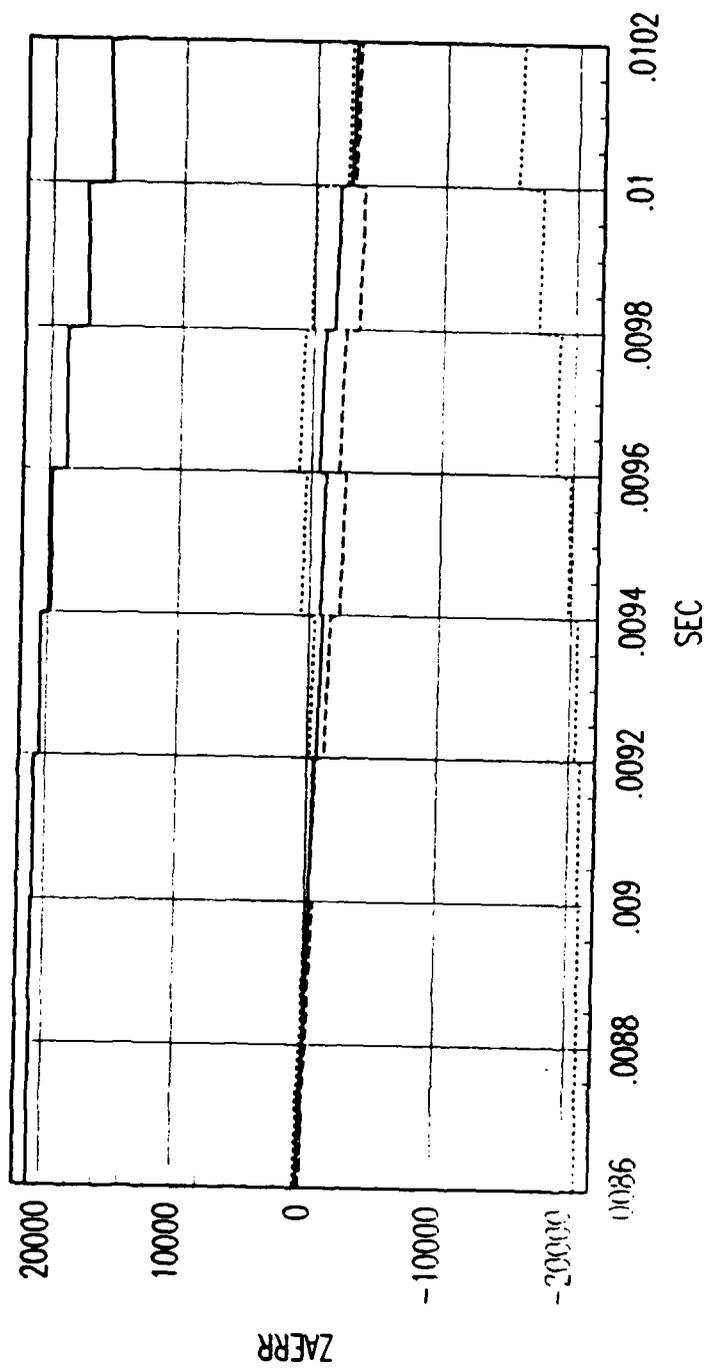


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.287. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

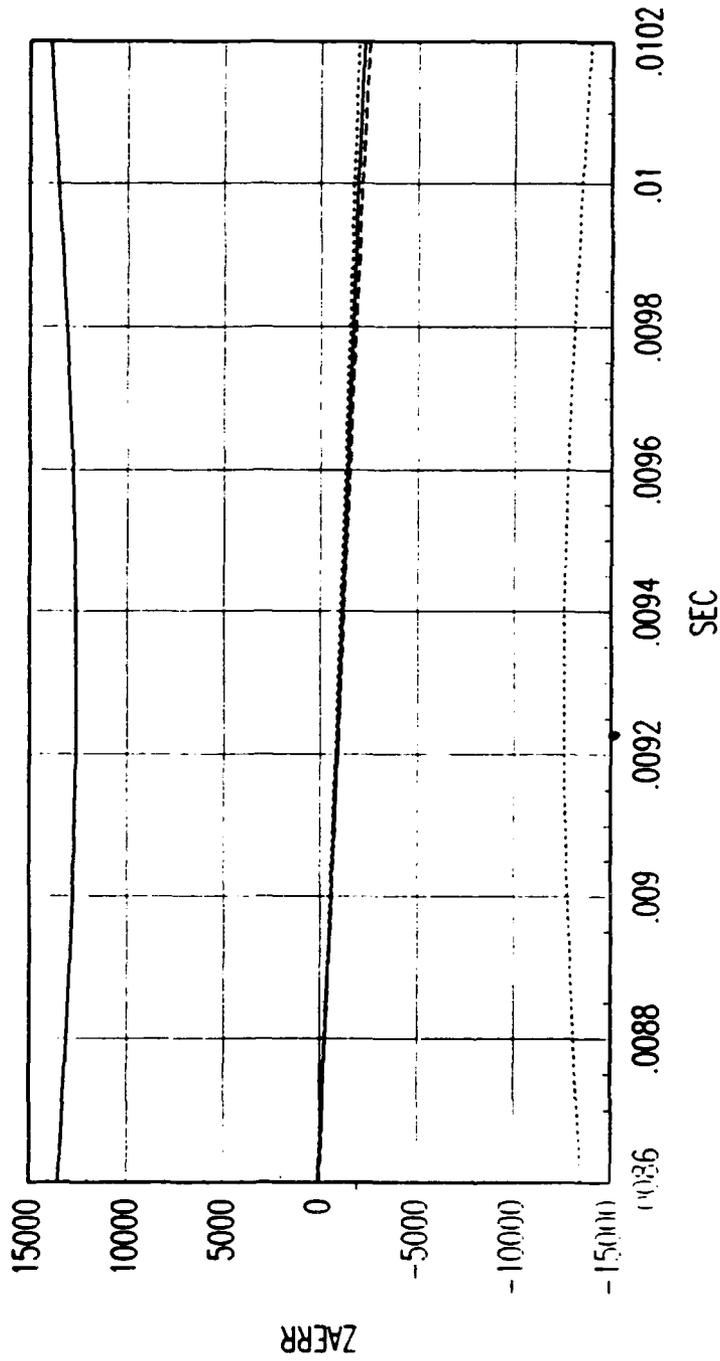


FIG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.288. Tracking Error Plot, Category 3, $v(t_0) = 6000$ ft/sec.

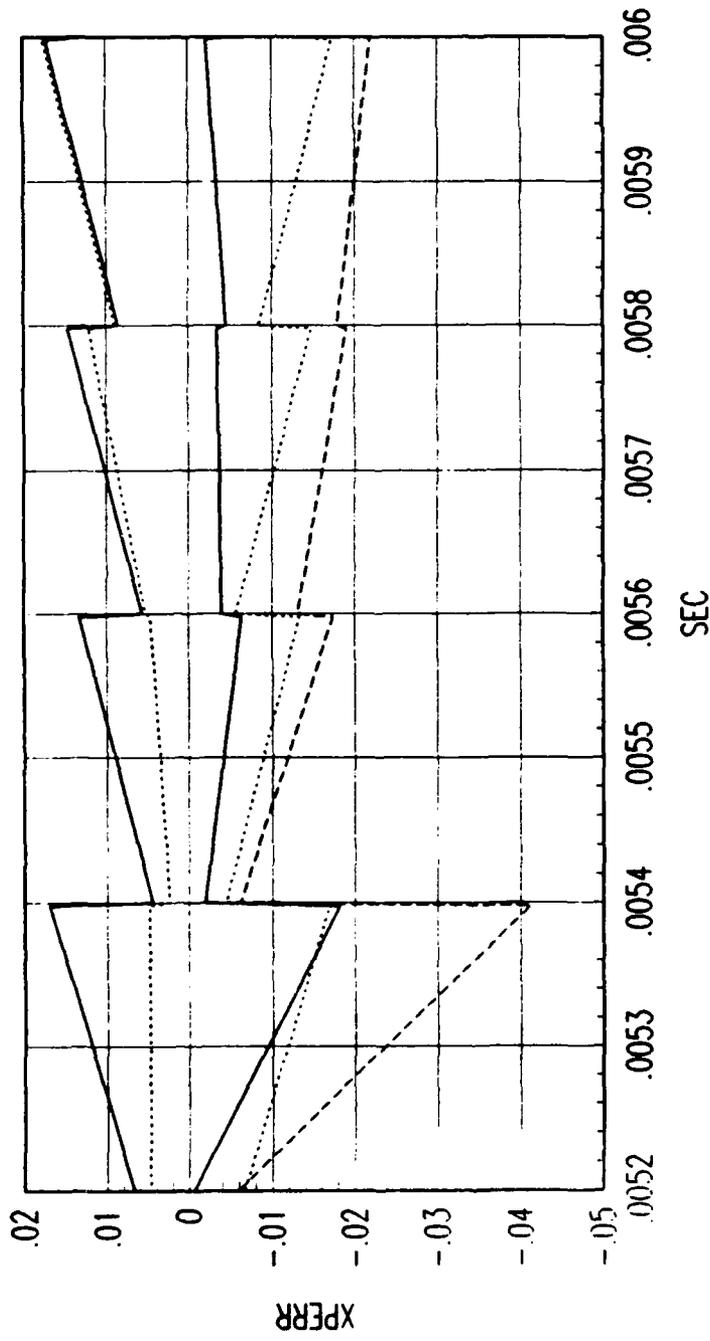


FIG. A FORWARD TIME EXF POSITION ERROR

Figure C.289. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

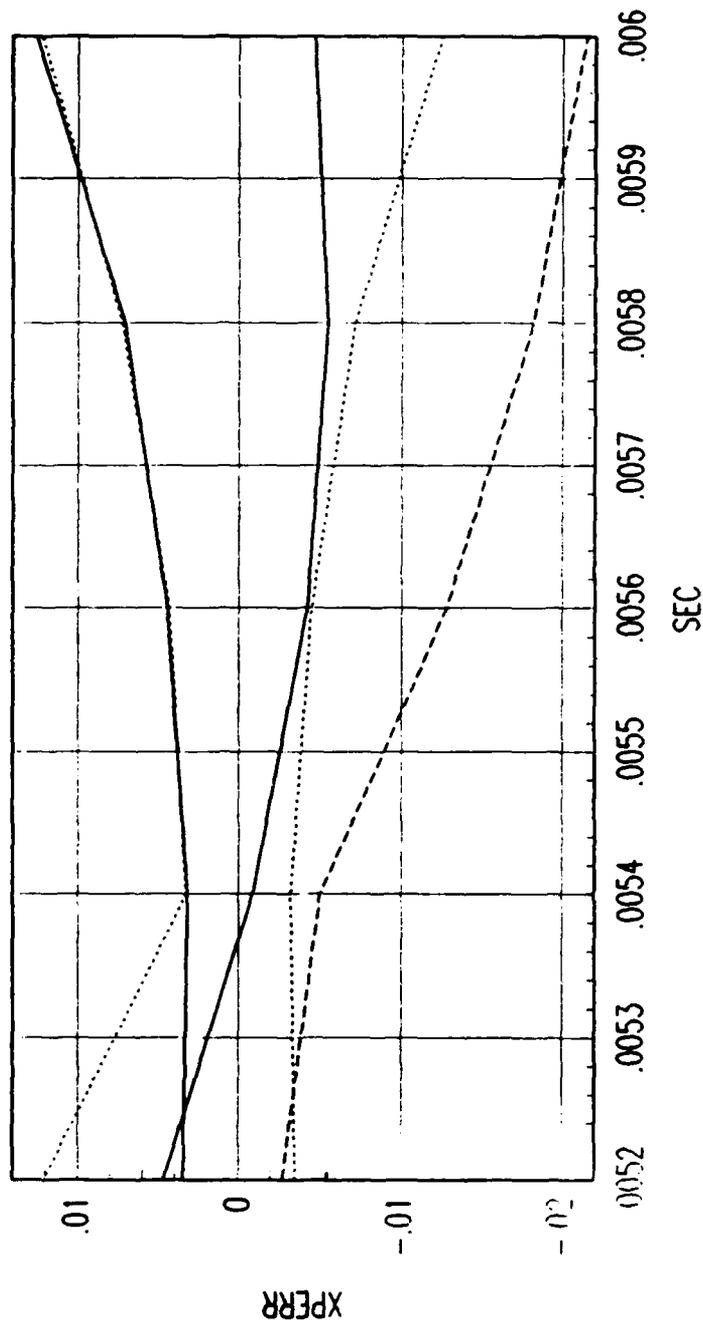


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.290. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

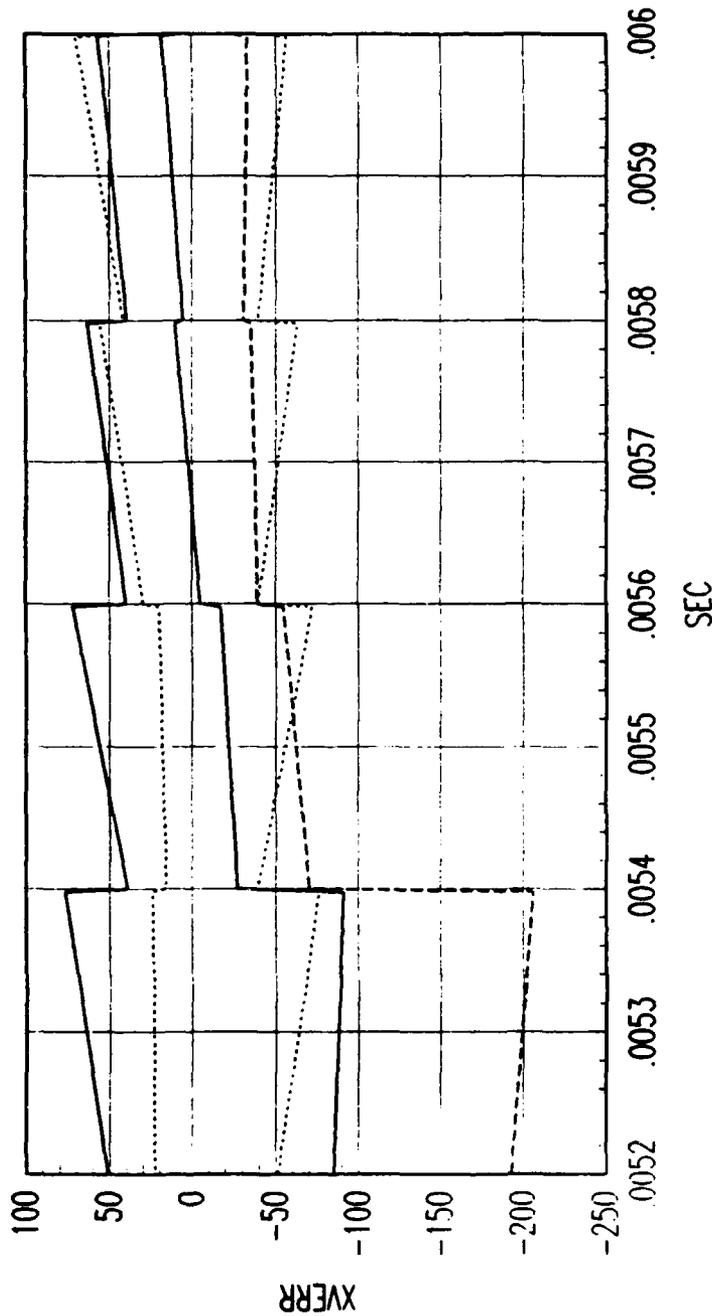
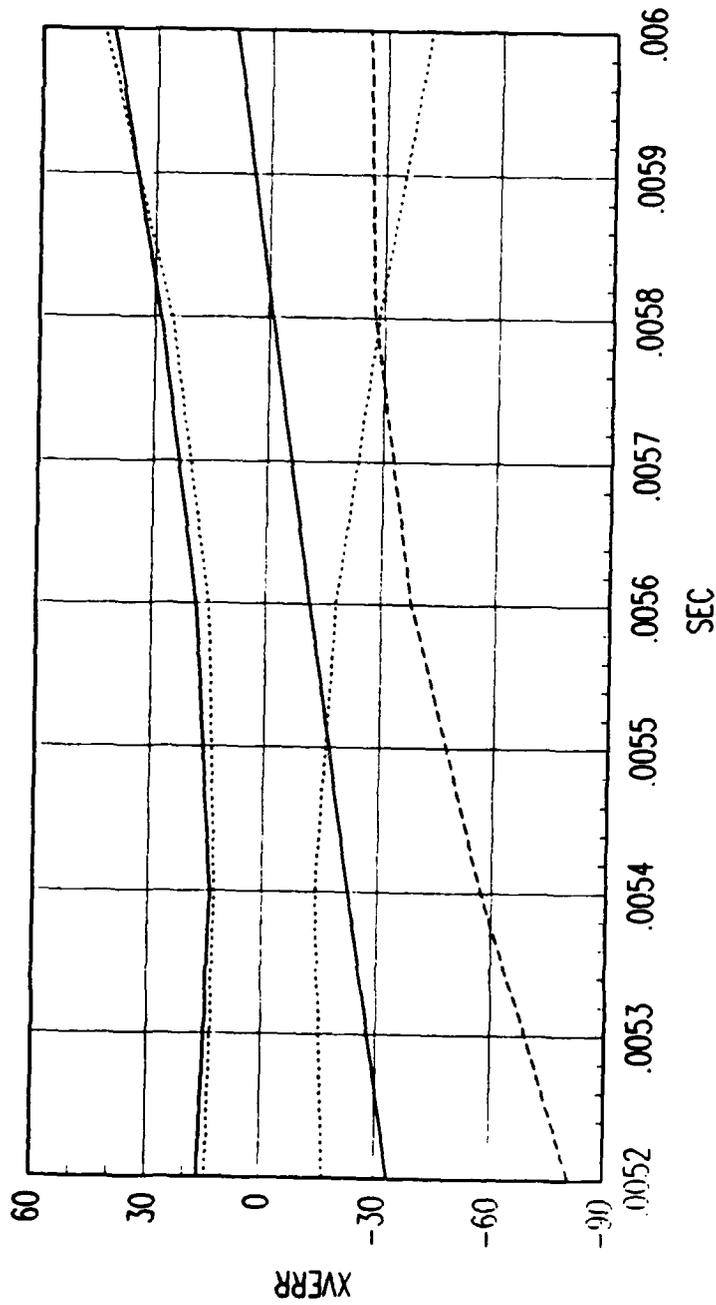


FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.291. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.292. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

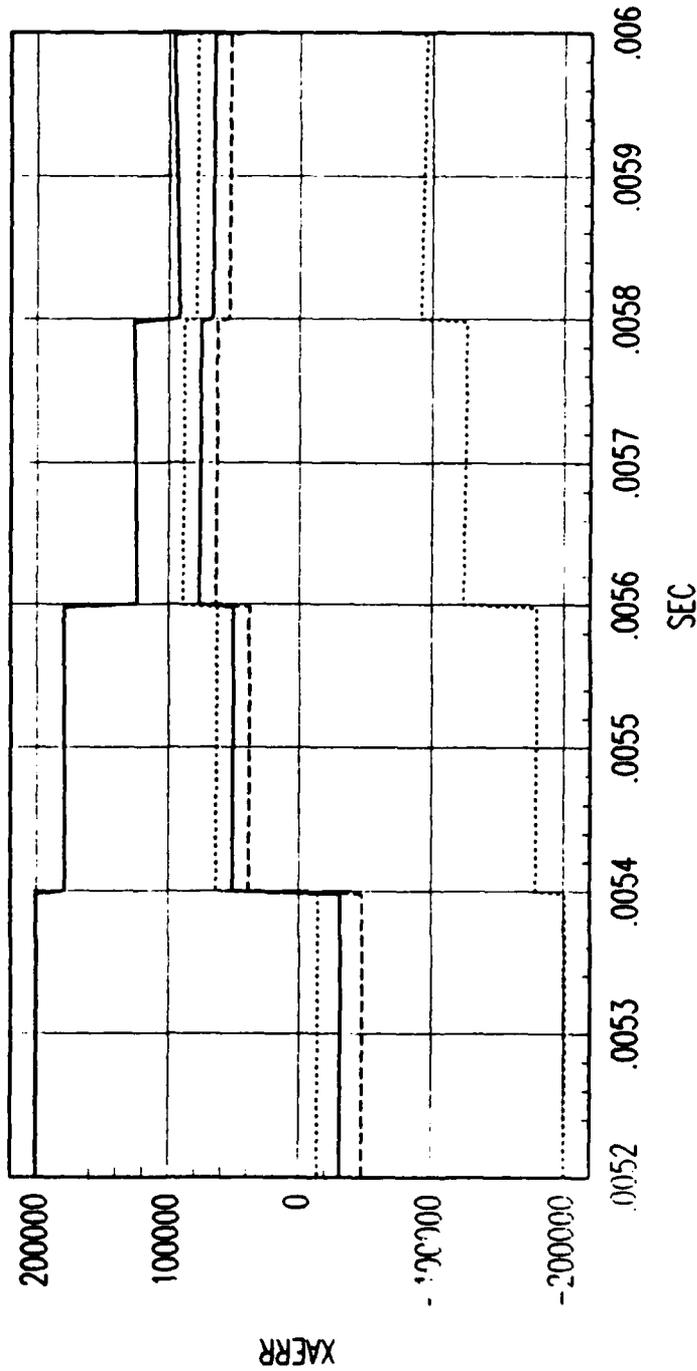


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.293. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

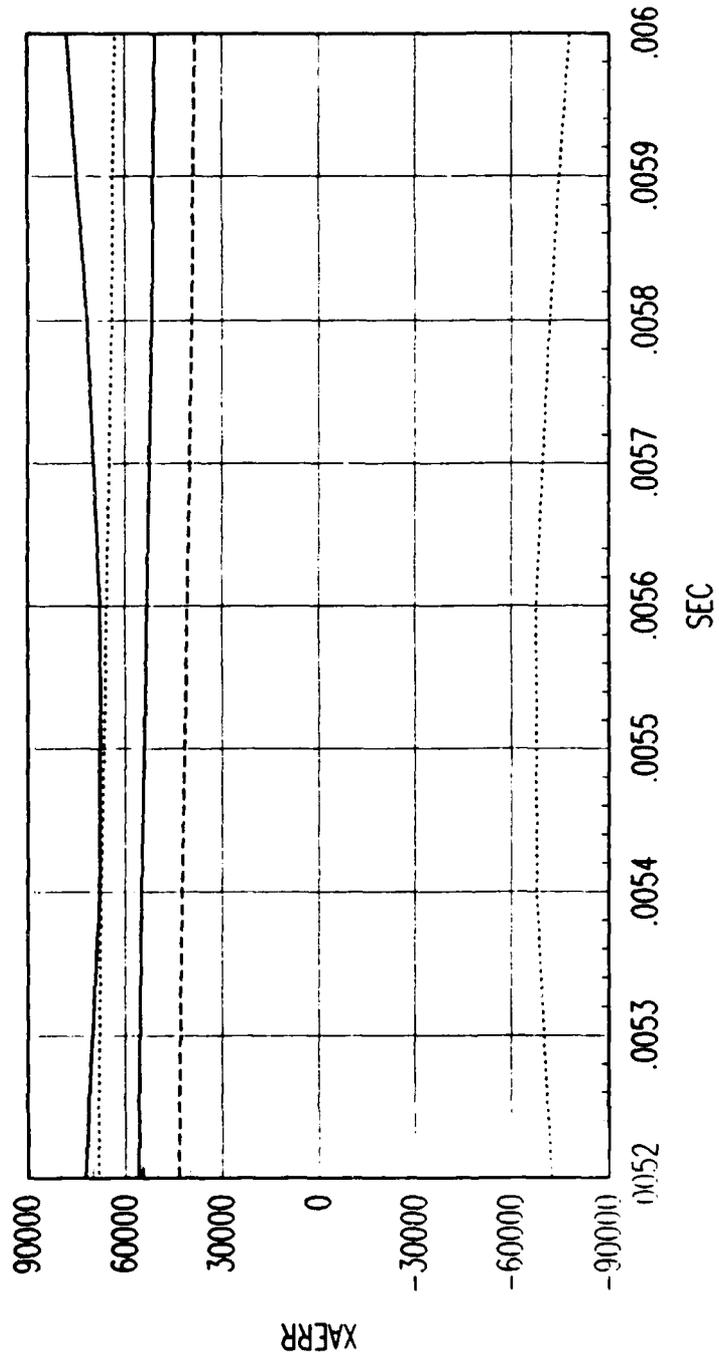
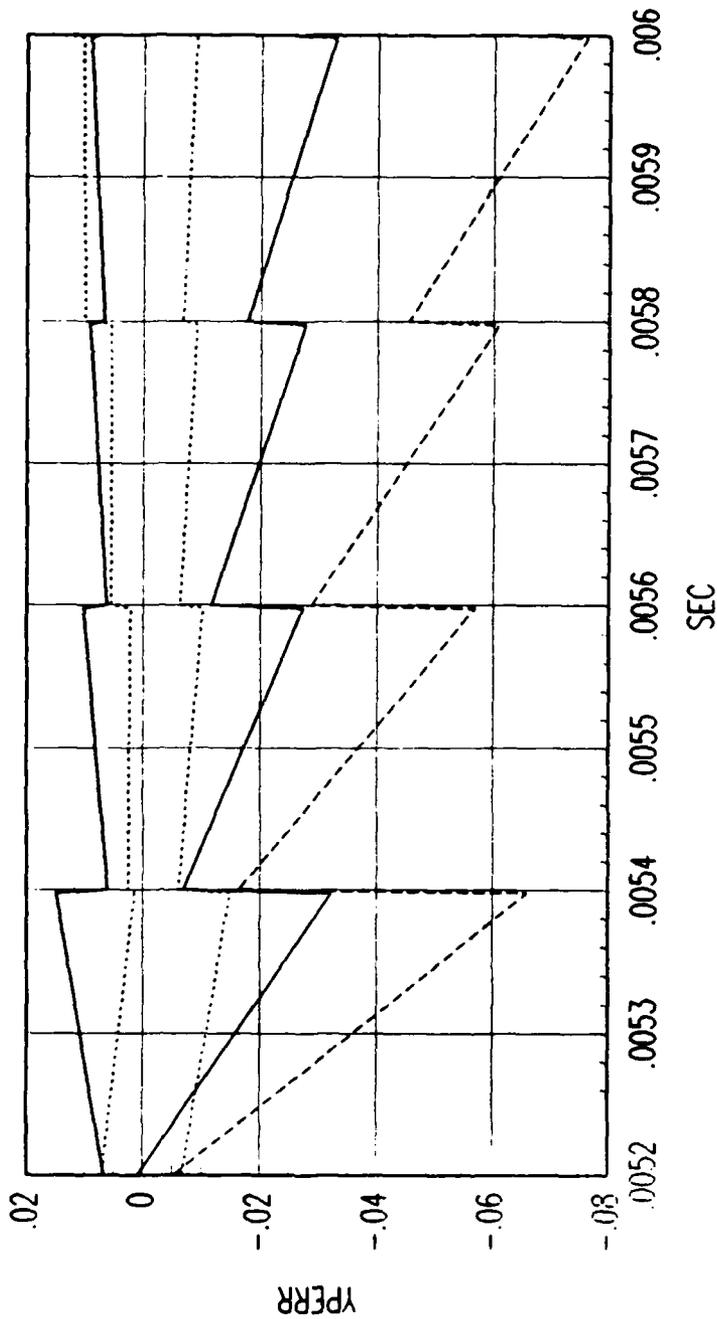


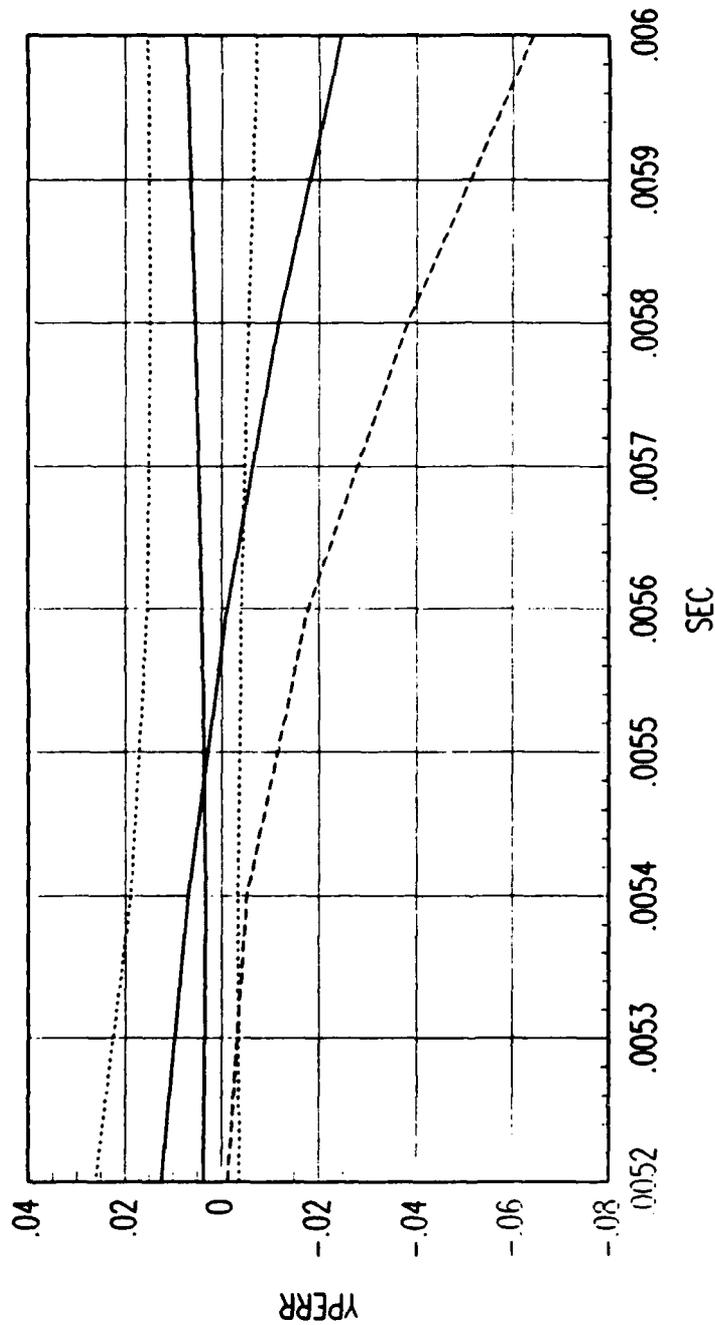
FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.294. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. A FORWARD TIME POSITION ERROR

Figure C.295. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.296. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

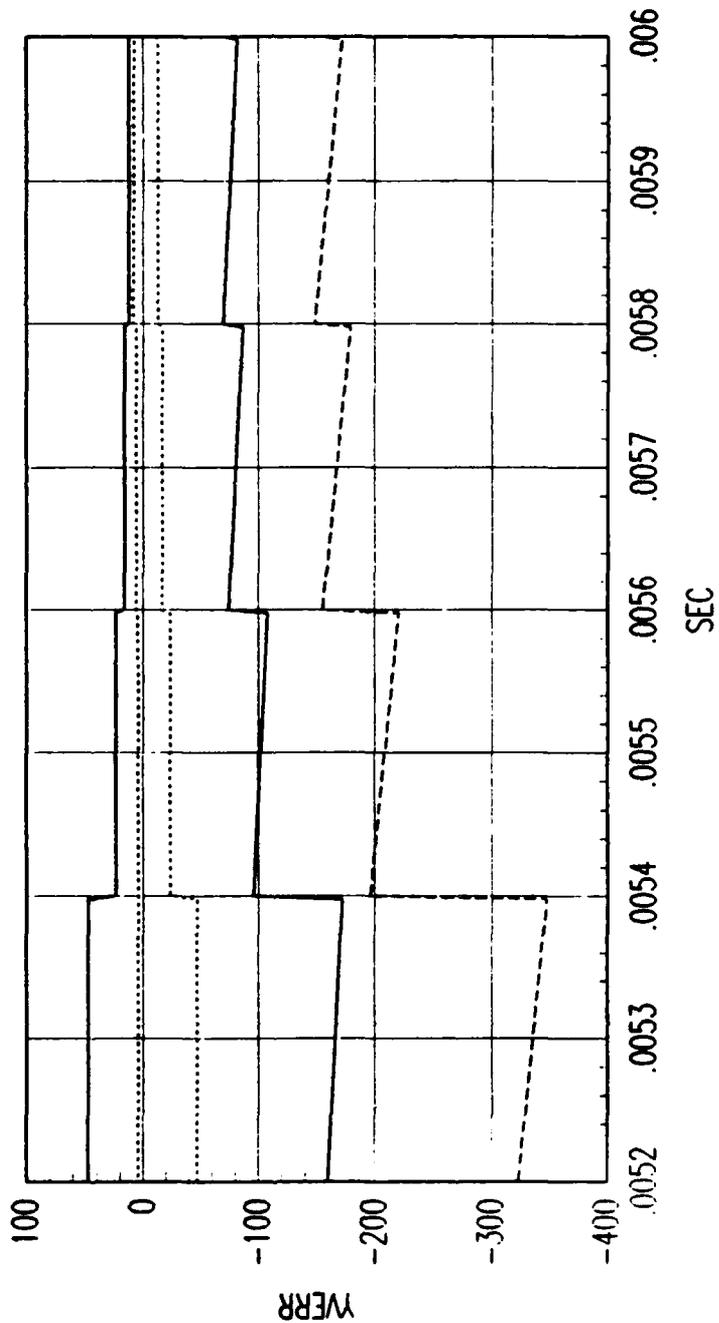
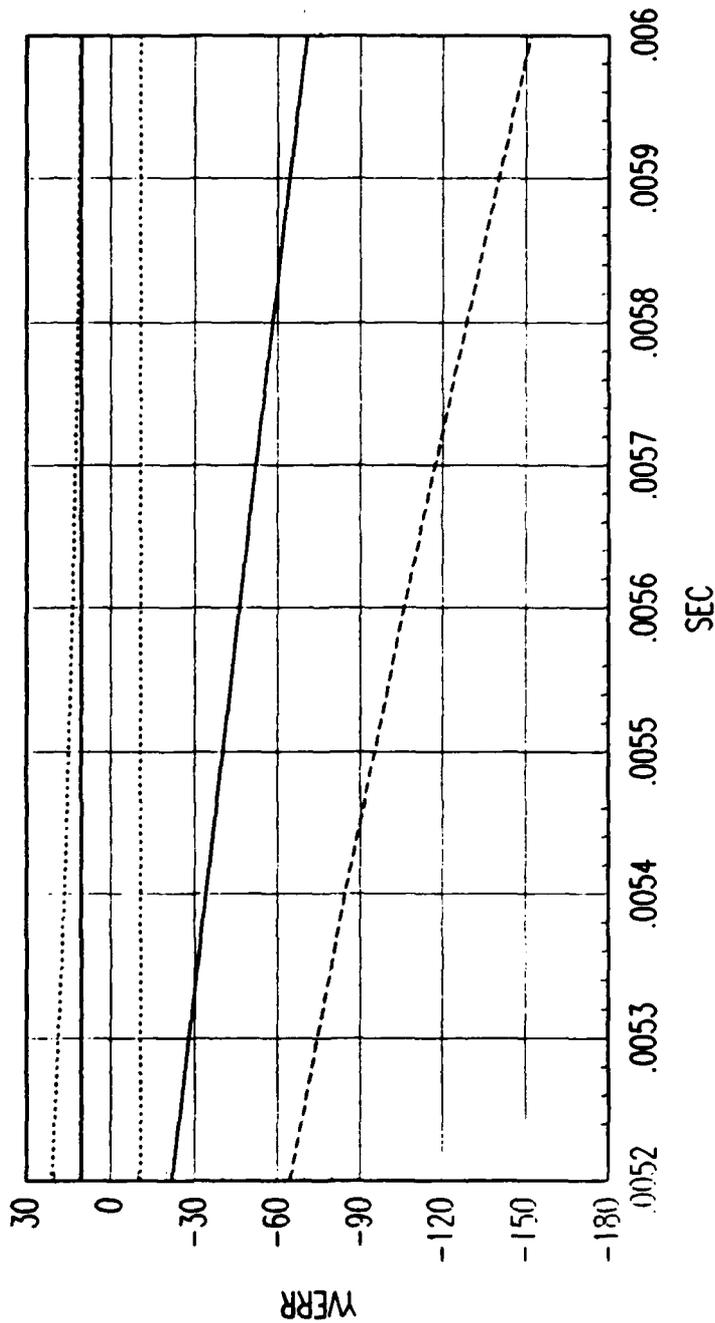


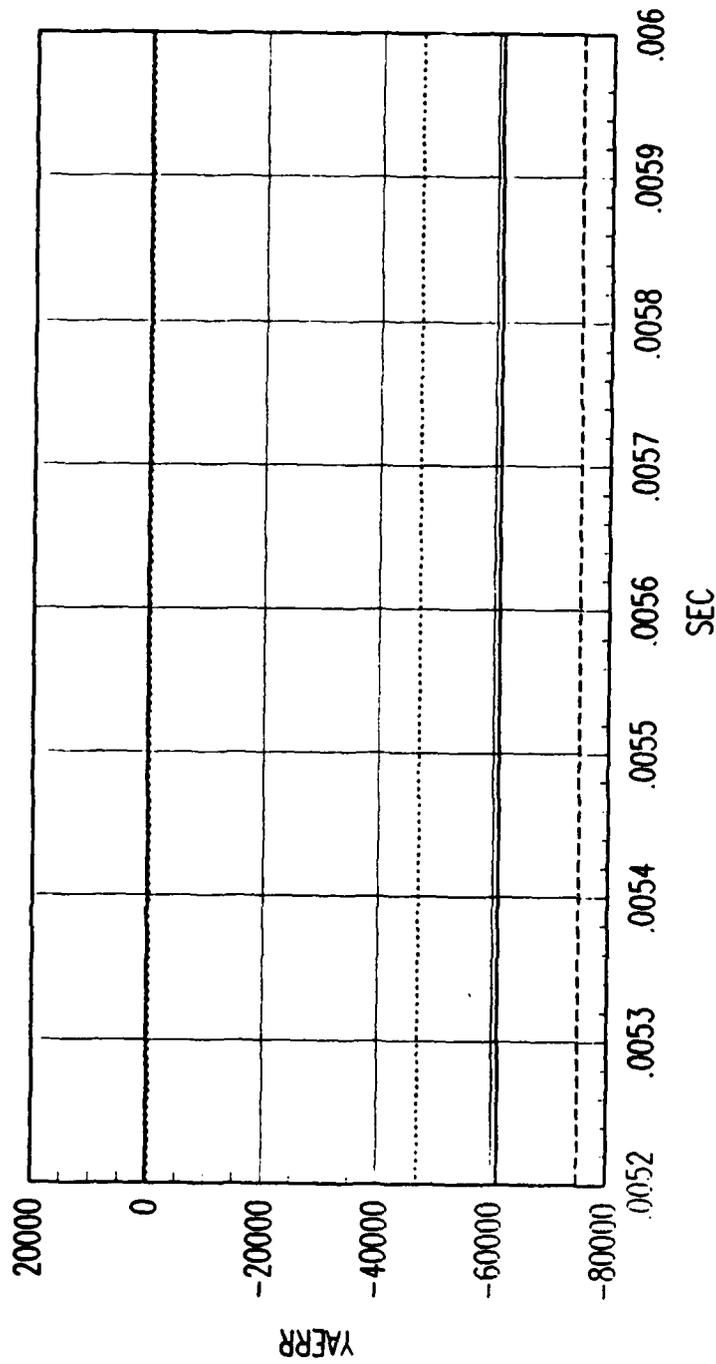
FIG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.297. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.298. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.299. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

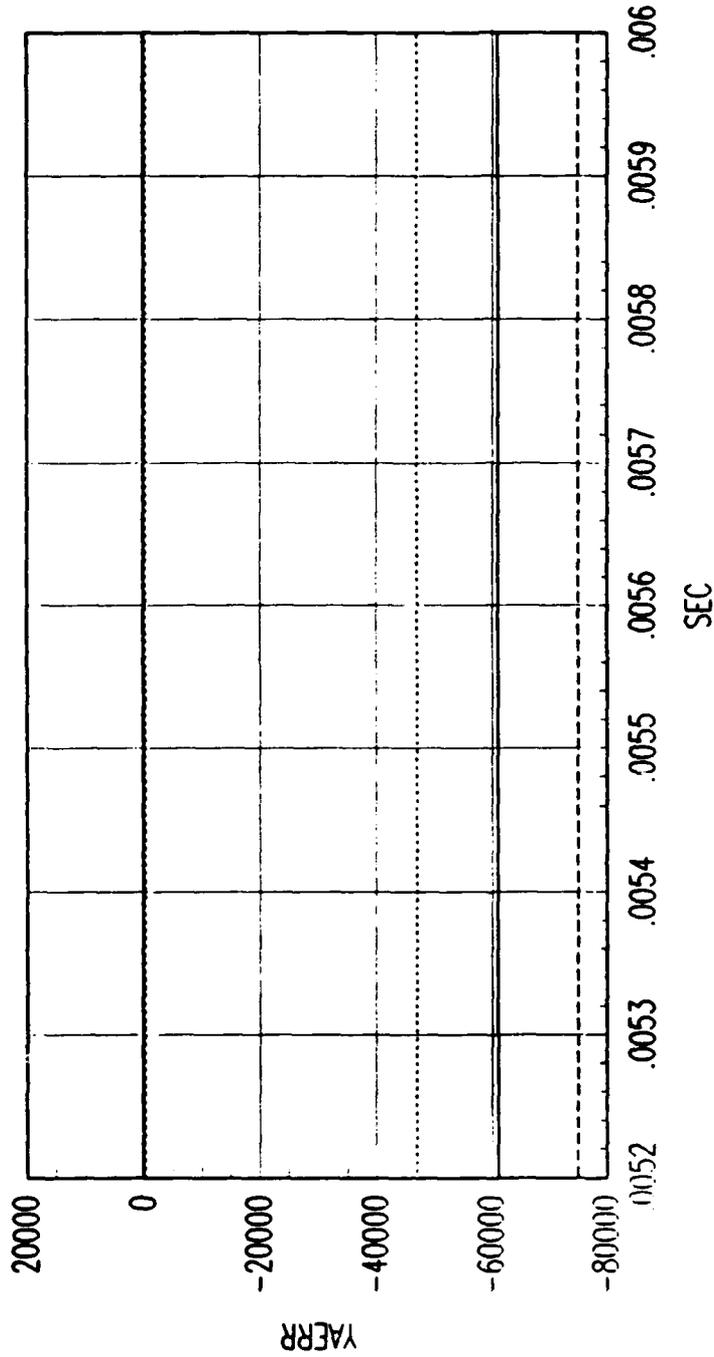


FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.300. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

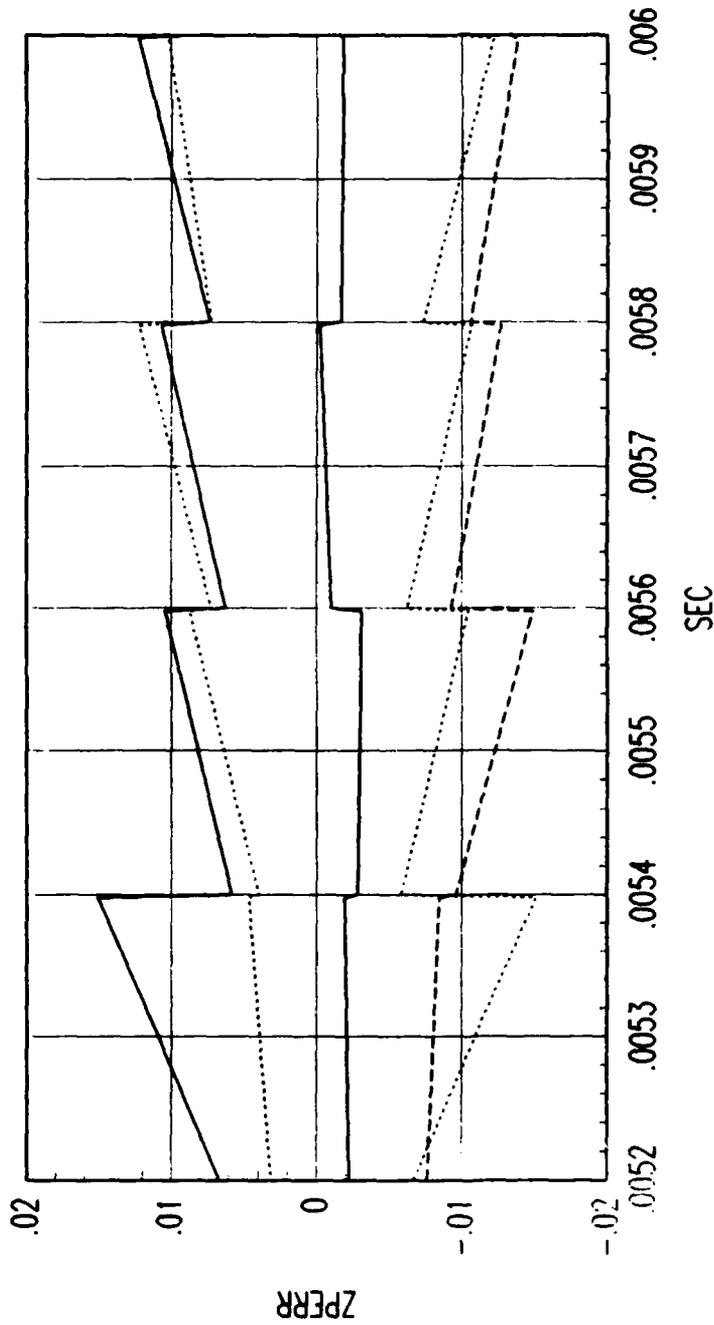


FIG. A FORWARD TIME EKF POSITION ERROR

Figure C.301. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

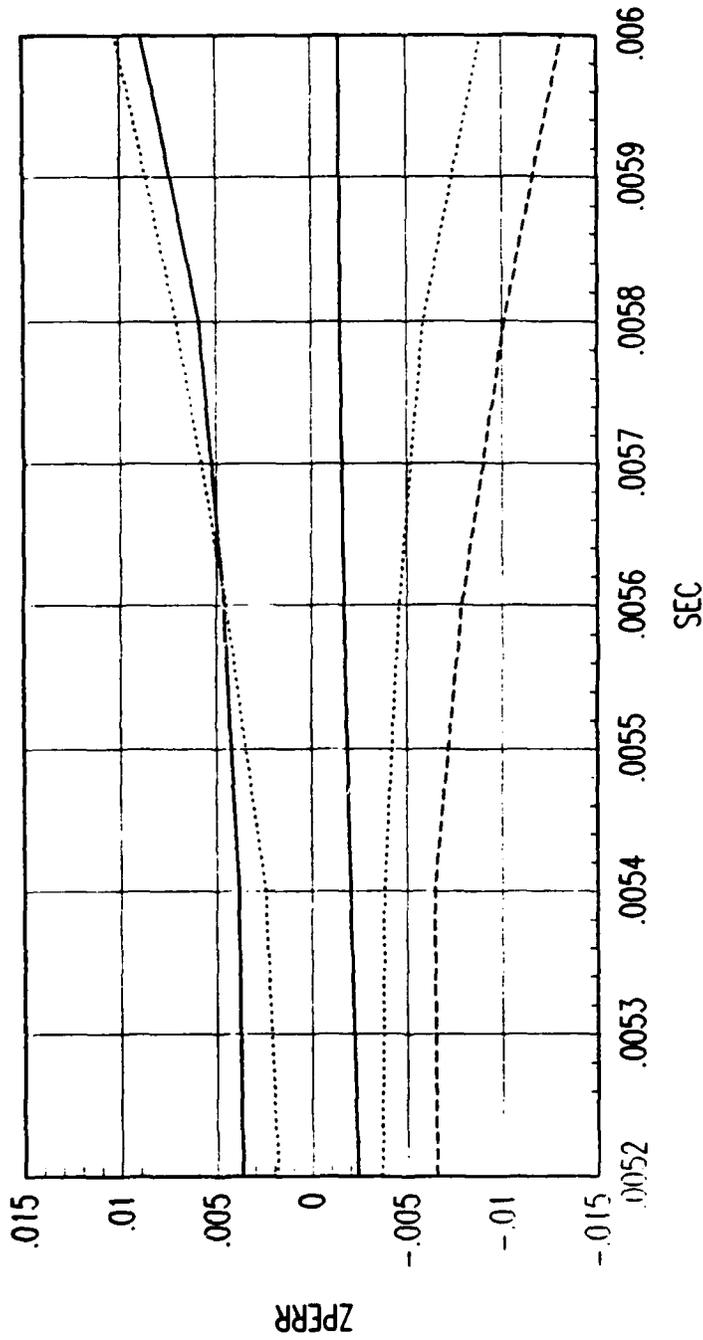
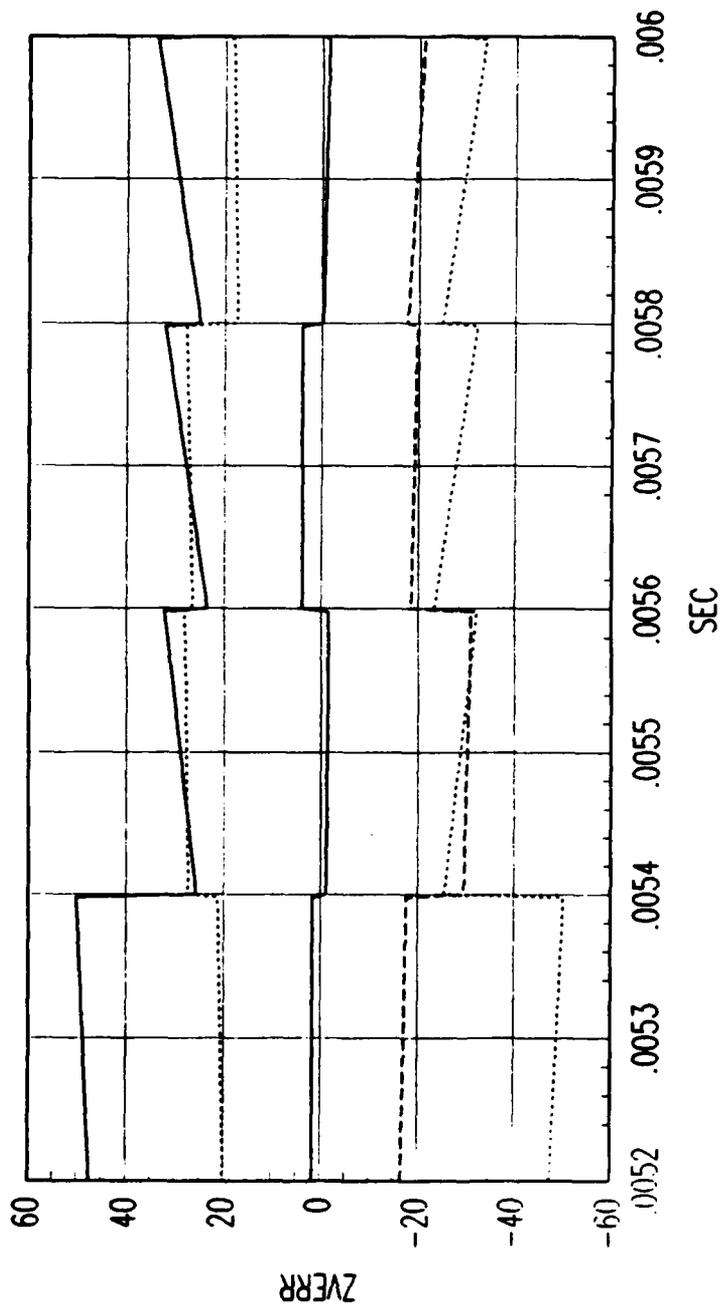


FIG. A REVERSE TIME SMOOTHER POSITION ERROR

Figure C.302. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. A FORWARD TIME EKF VELOCITY ERROR

Figure C.303. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

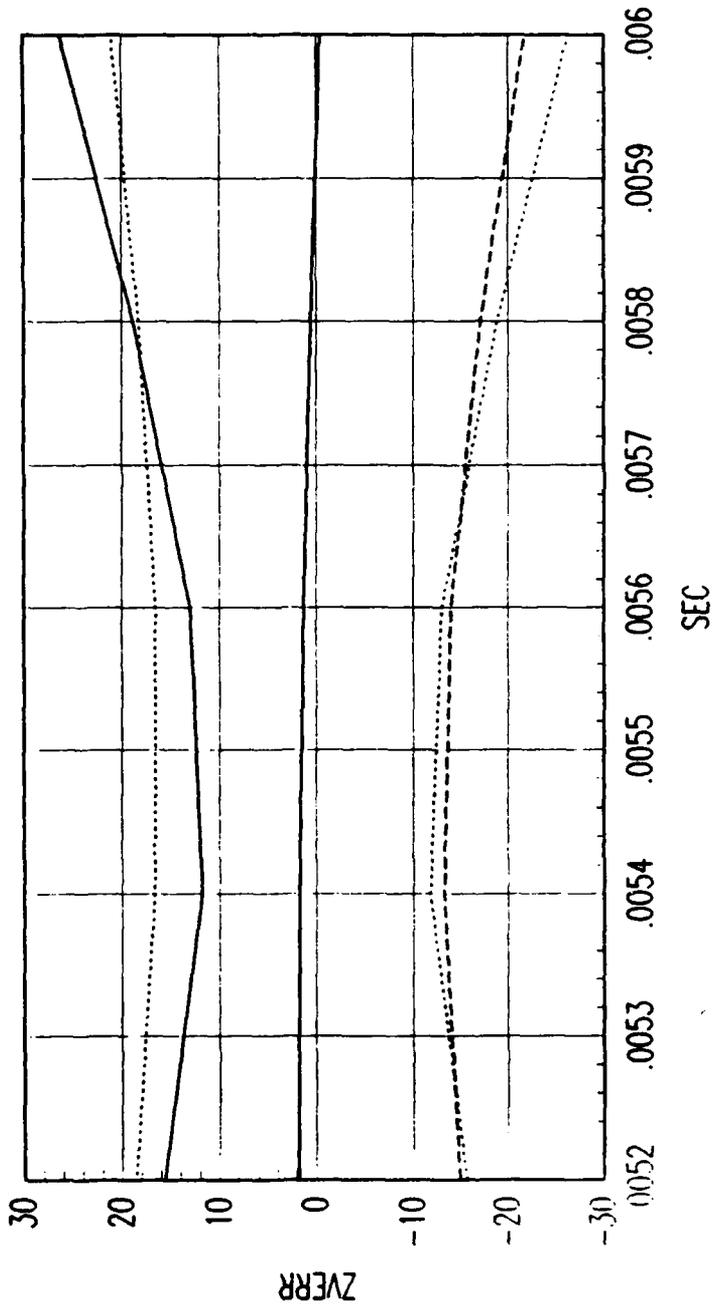


FIG. A REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.304. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

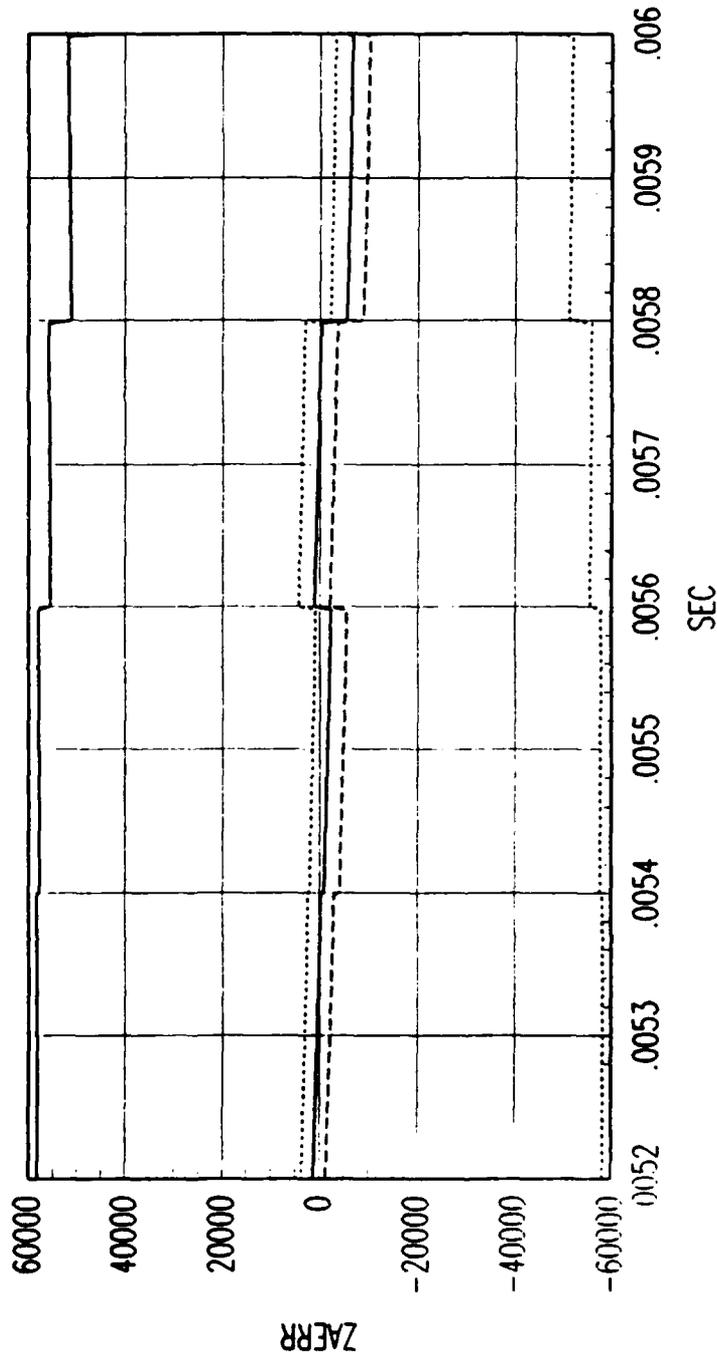


FIG. A FORWARD TIME EKF ACCELERATION ERROR

Figure C.305. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

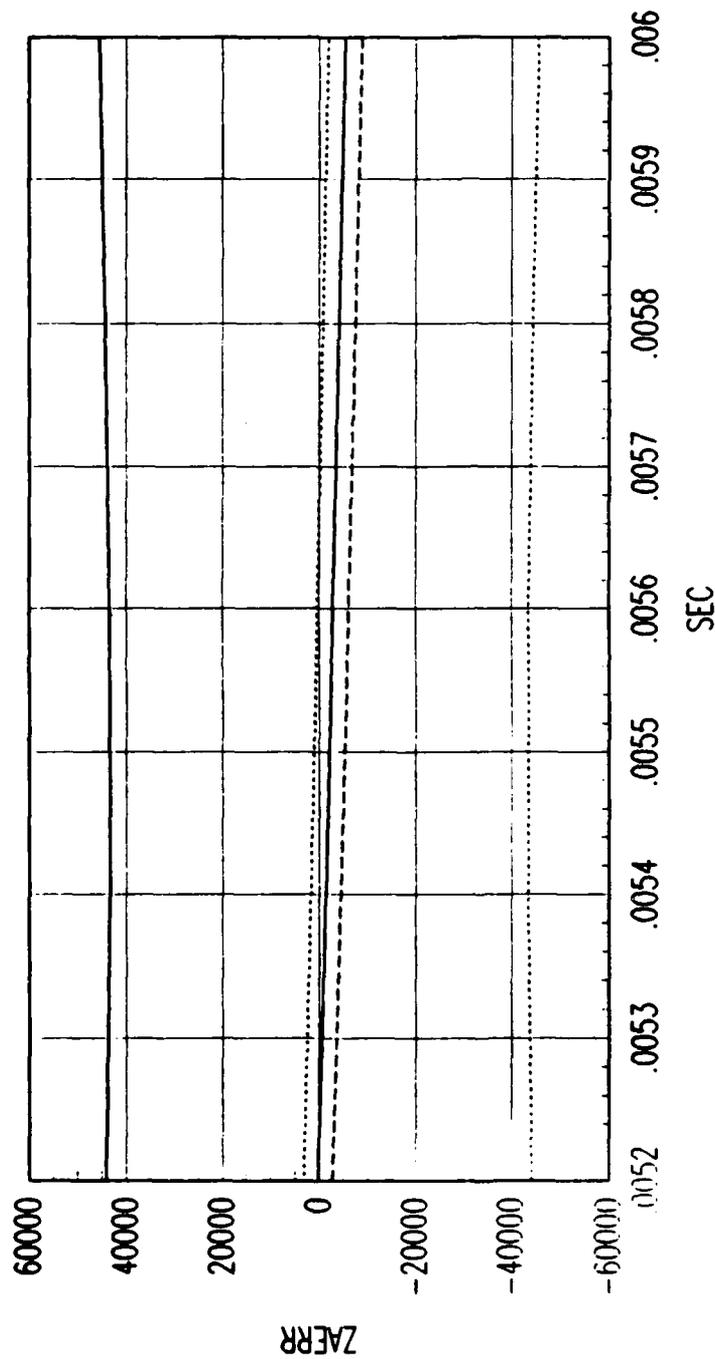
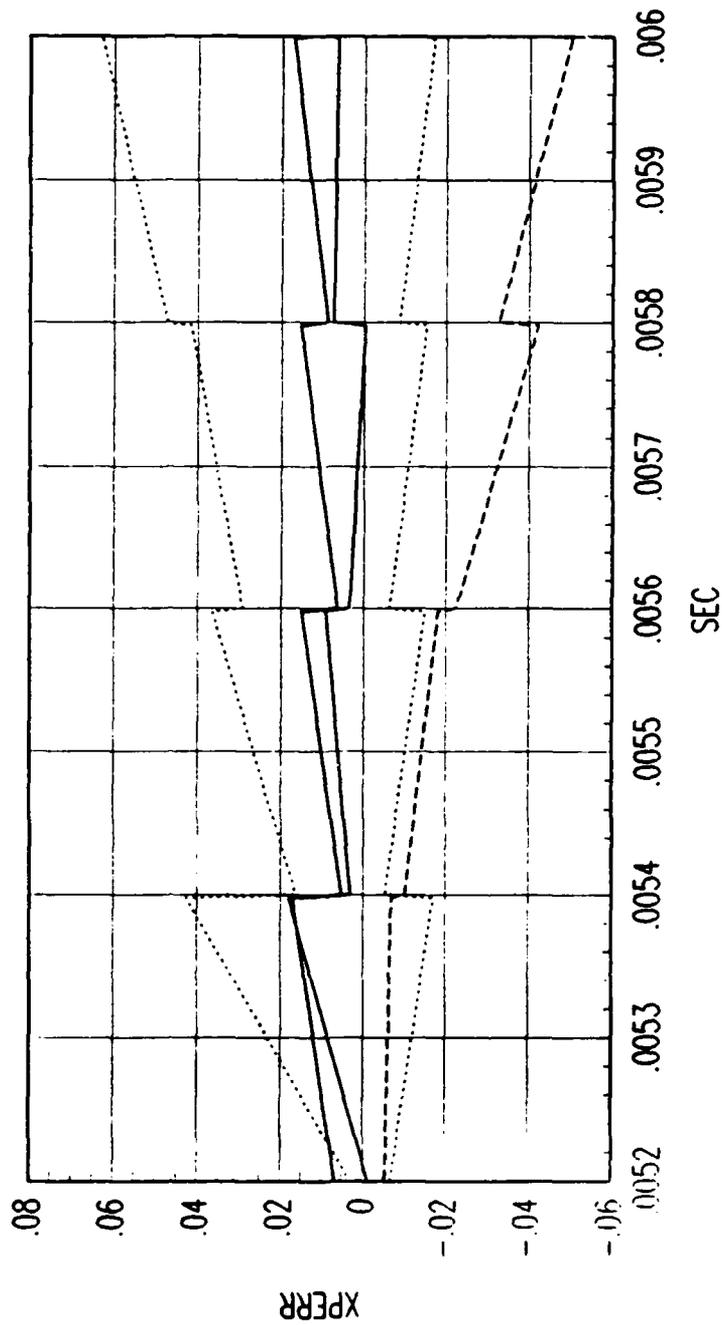


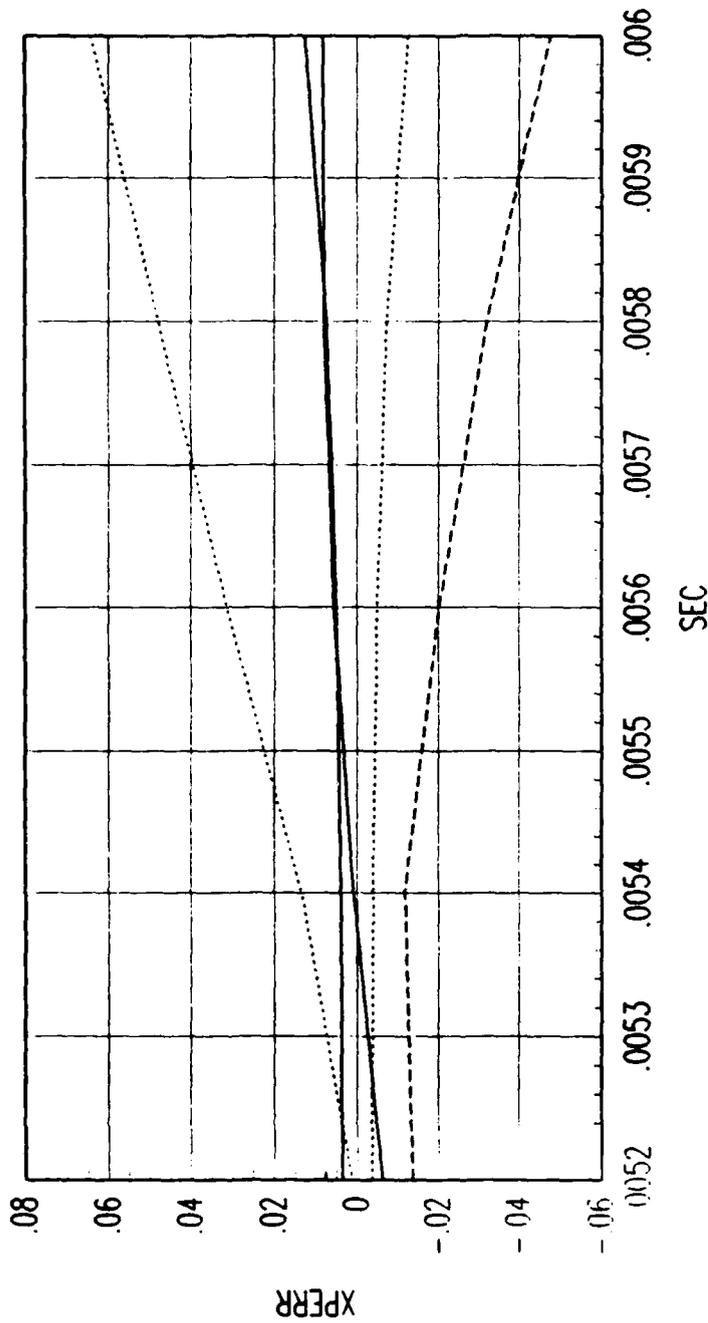
FIG. A REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.306. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



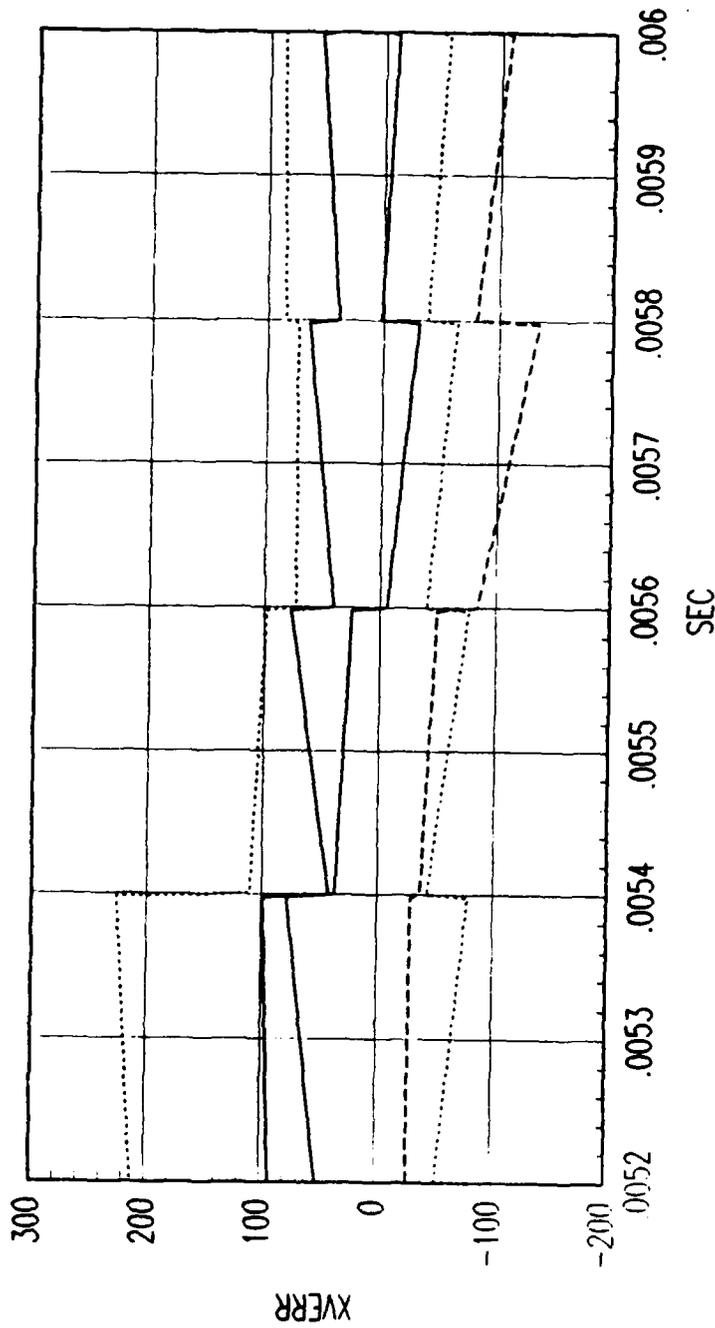
FRG. B FORWARD TIME EKF POSITION ERROR

Figure C.307. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.308. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.309. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

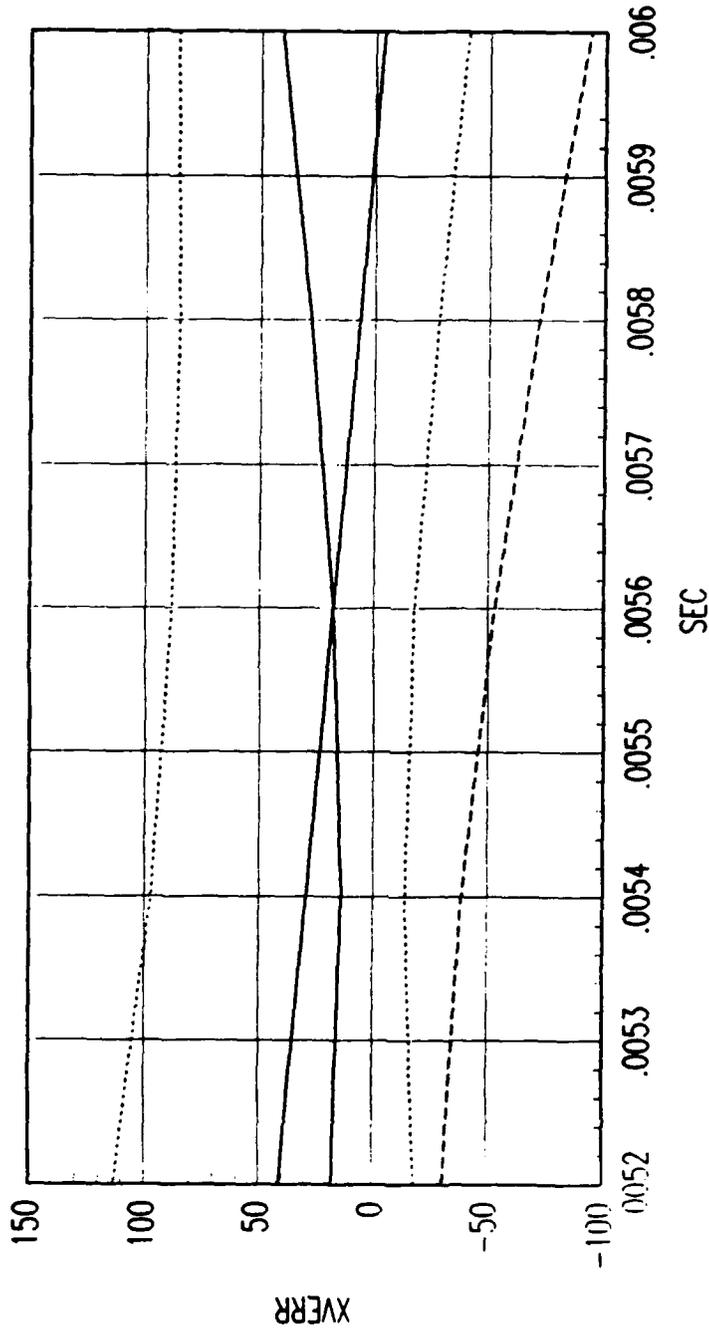
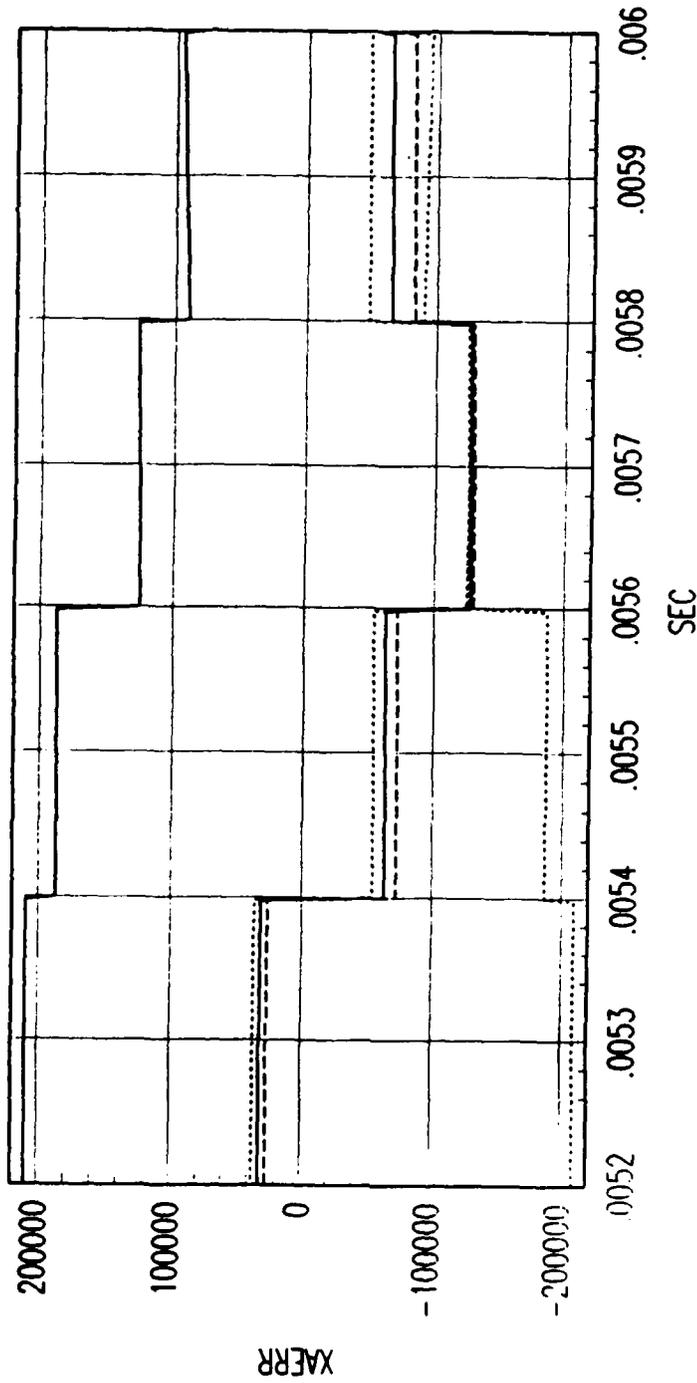


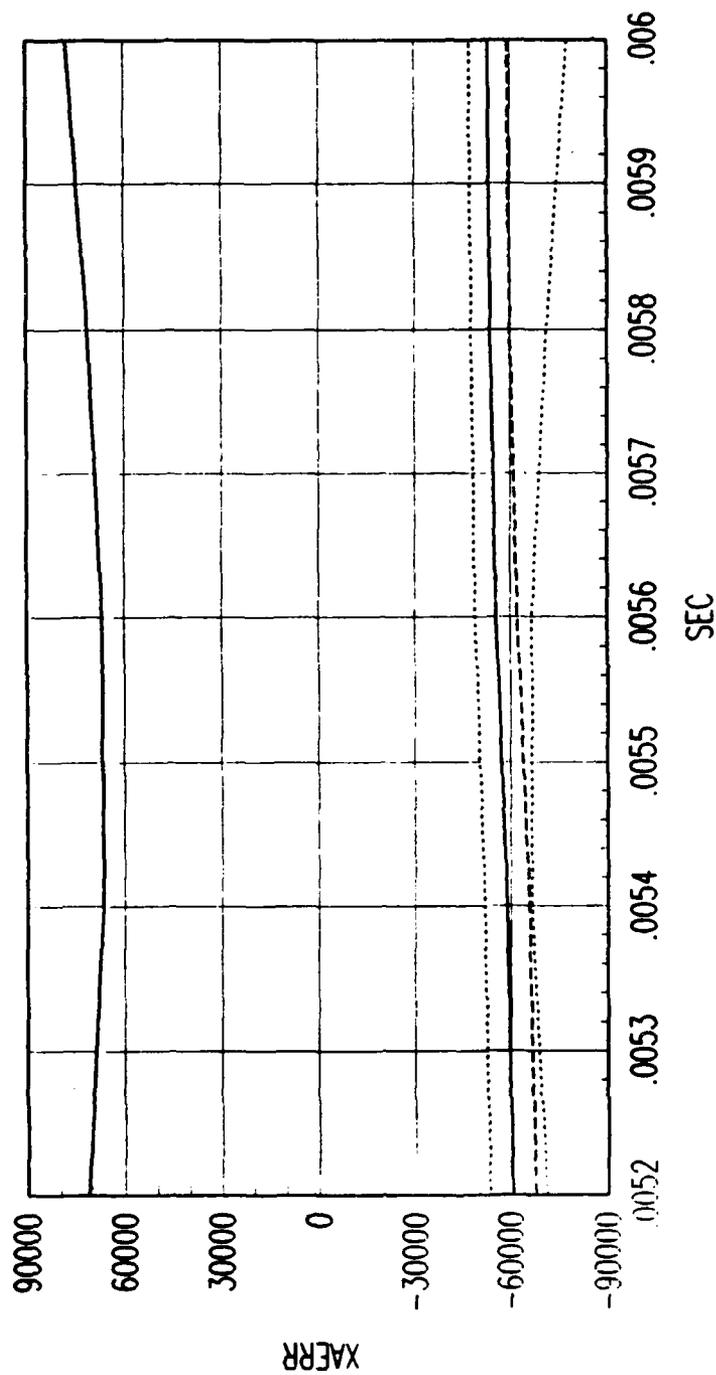
FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.310. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.311. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.312. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

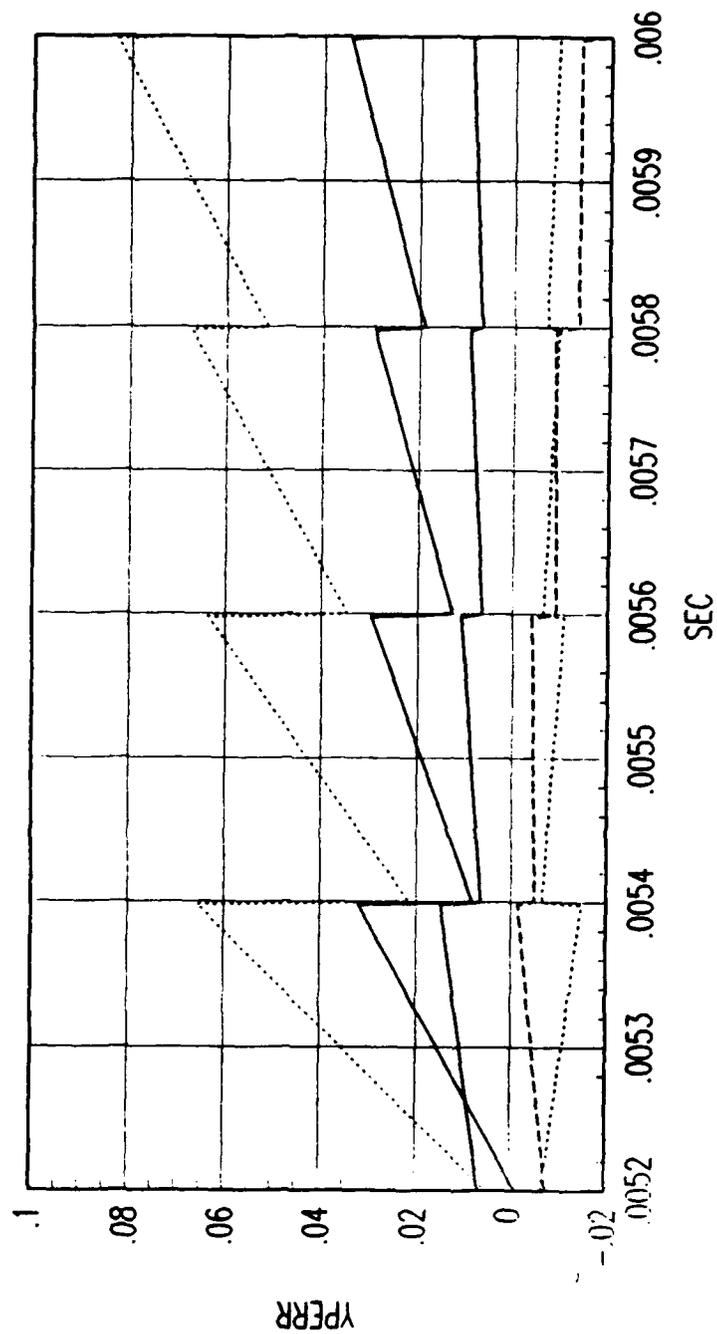
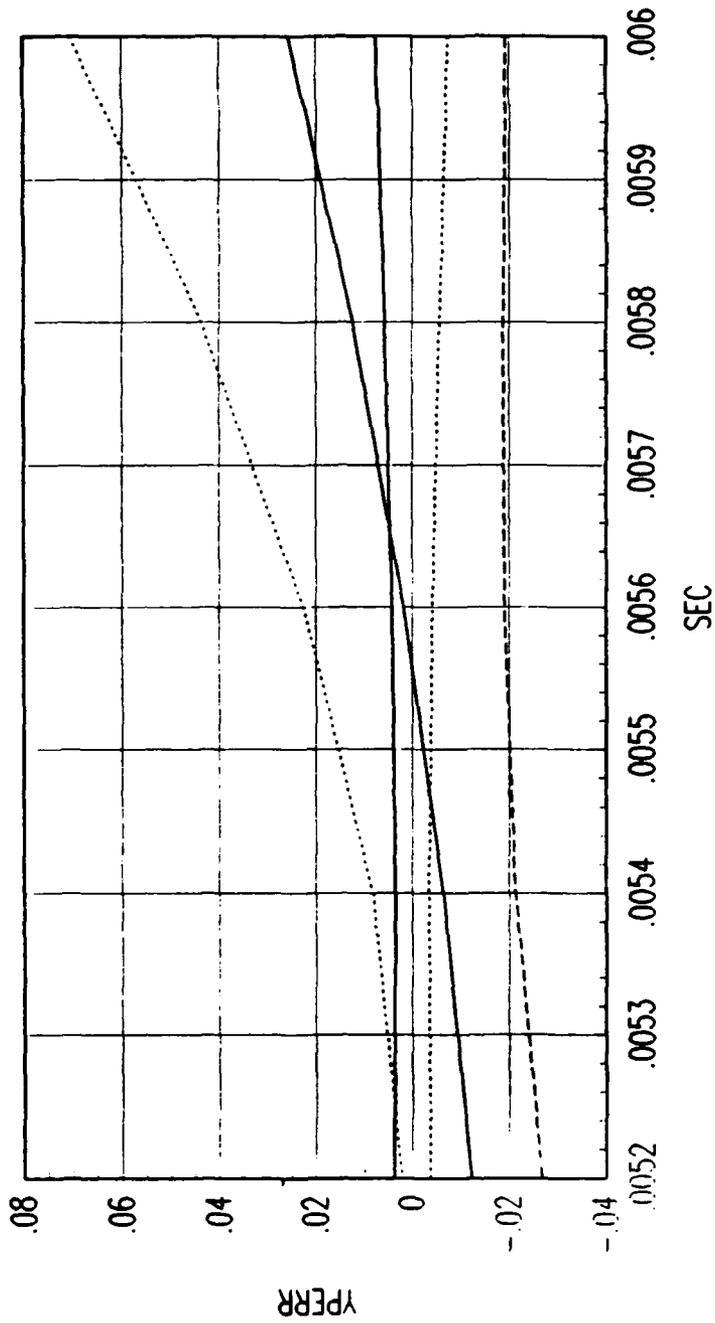


FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.313. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.314. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

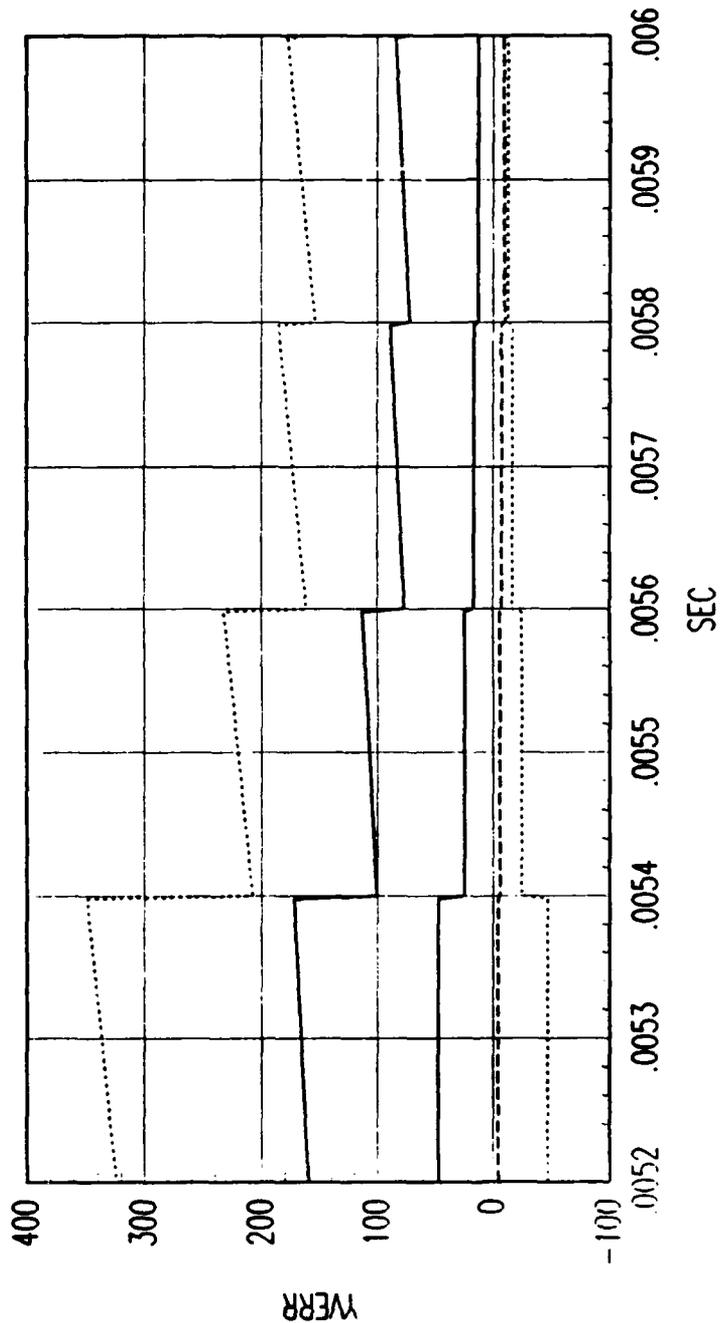
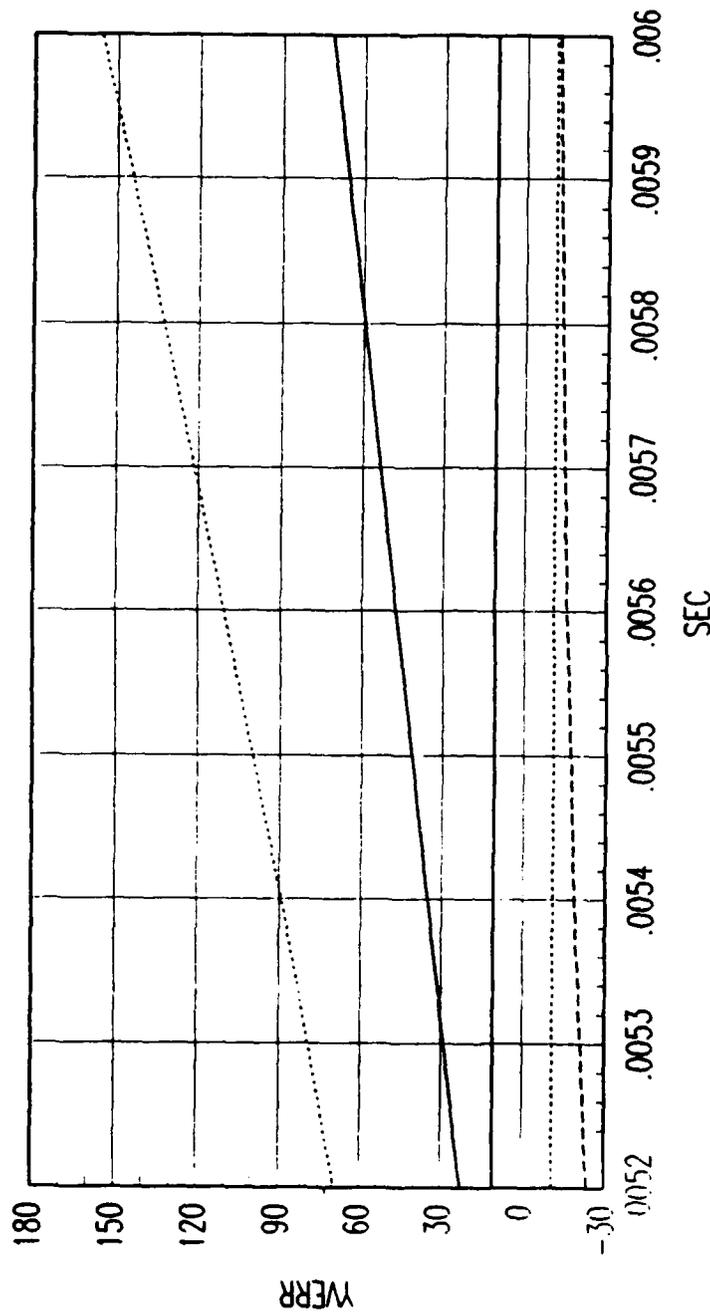


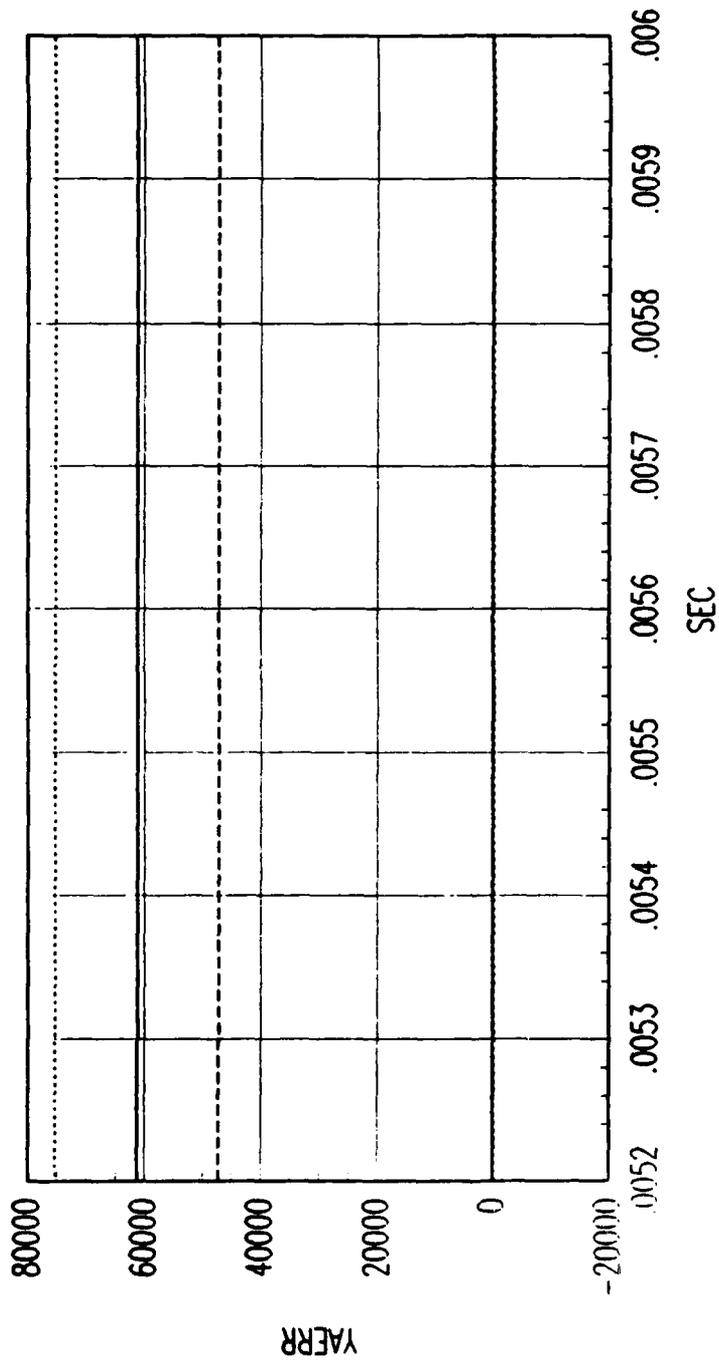
FIG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.315. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



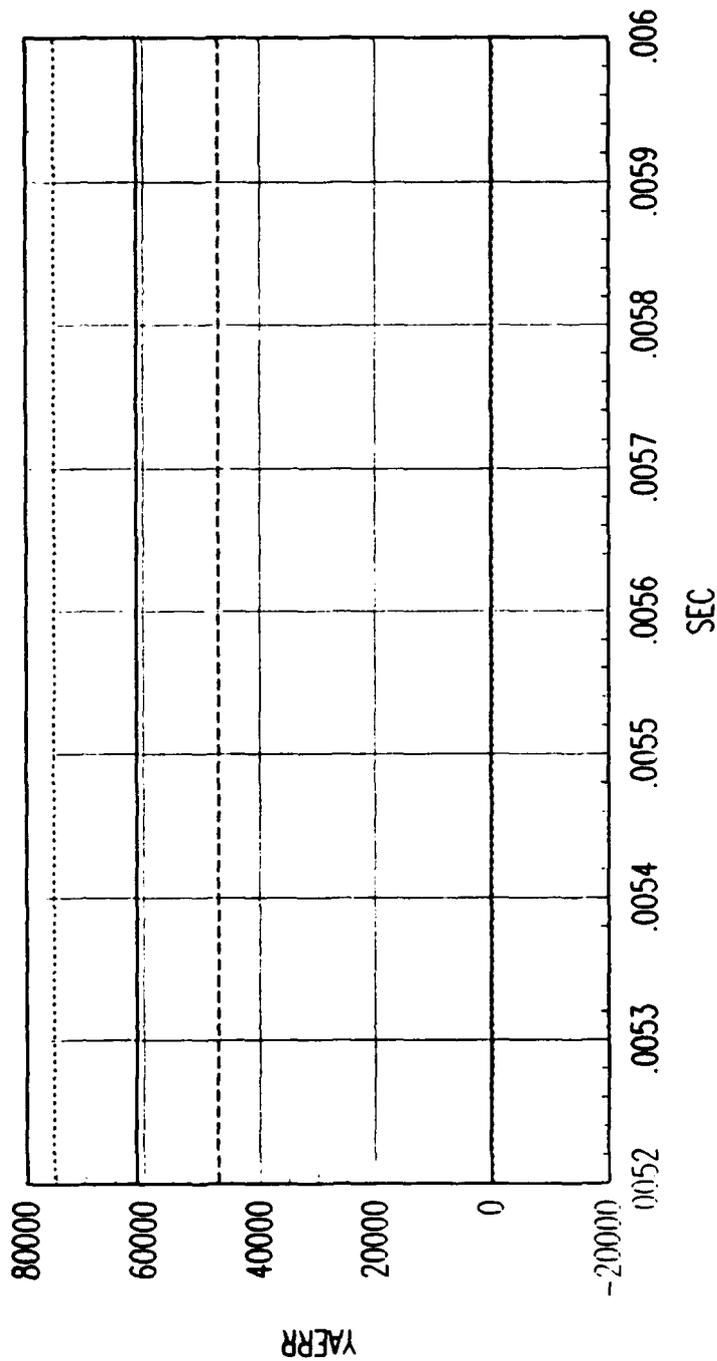
FRG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.316. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.317. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.318. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

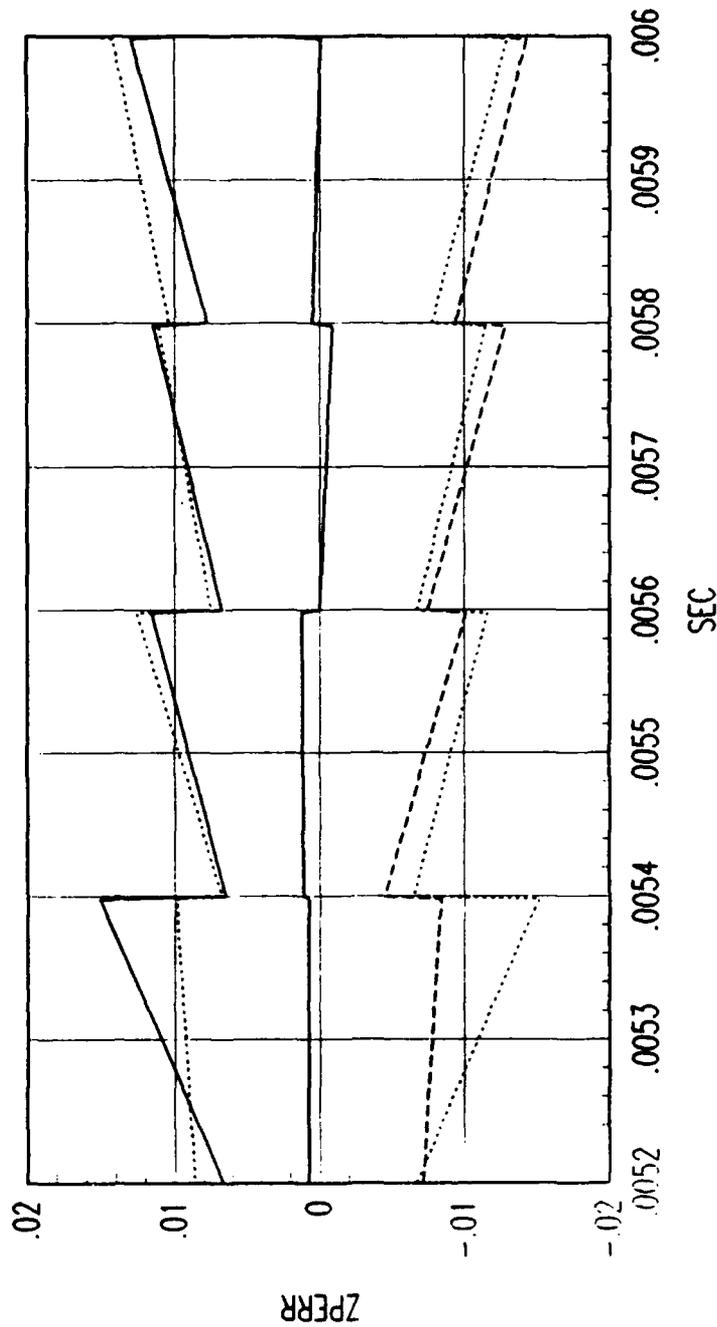
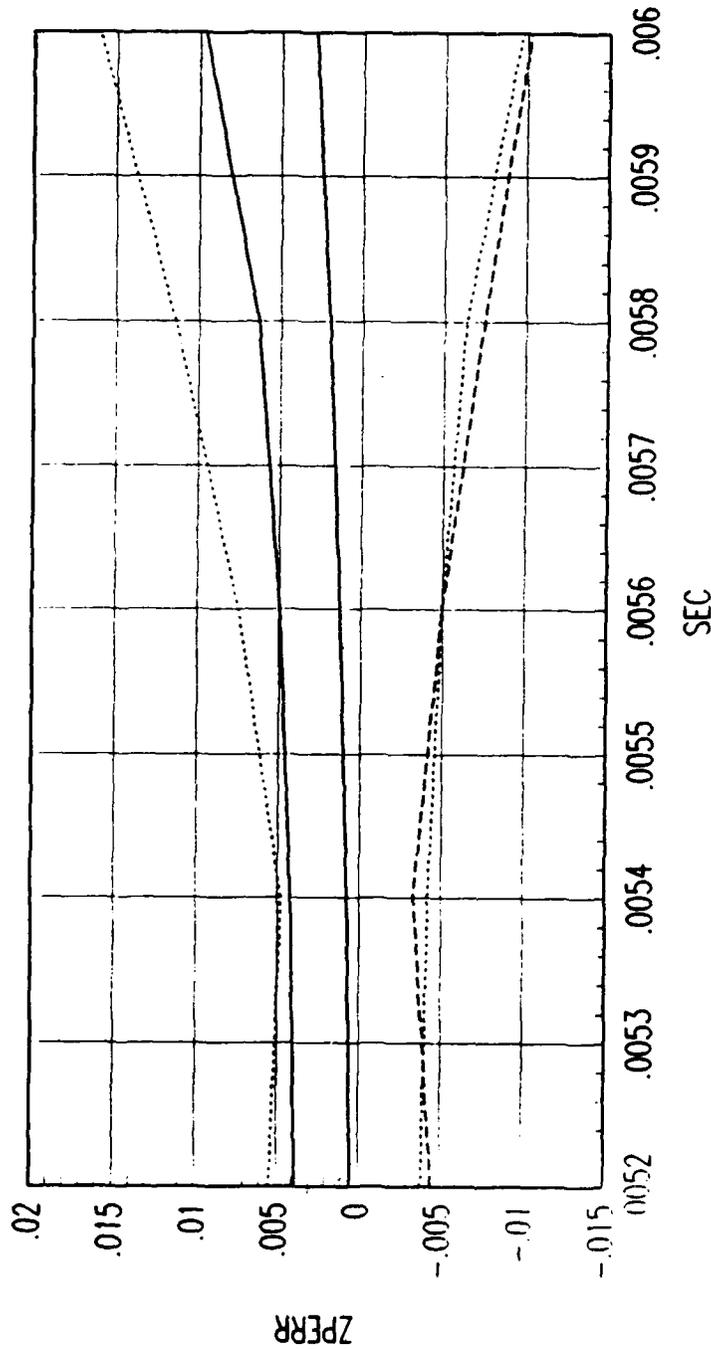


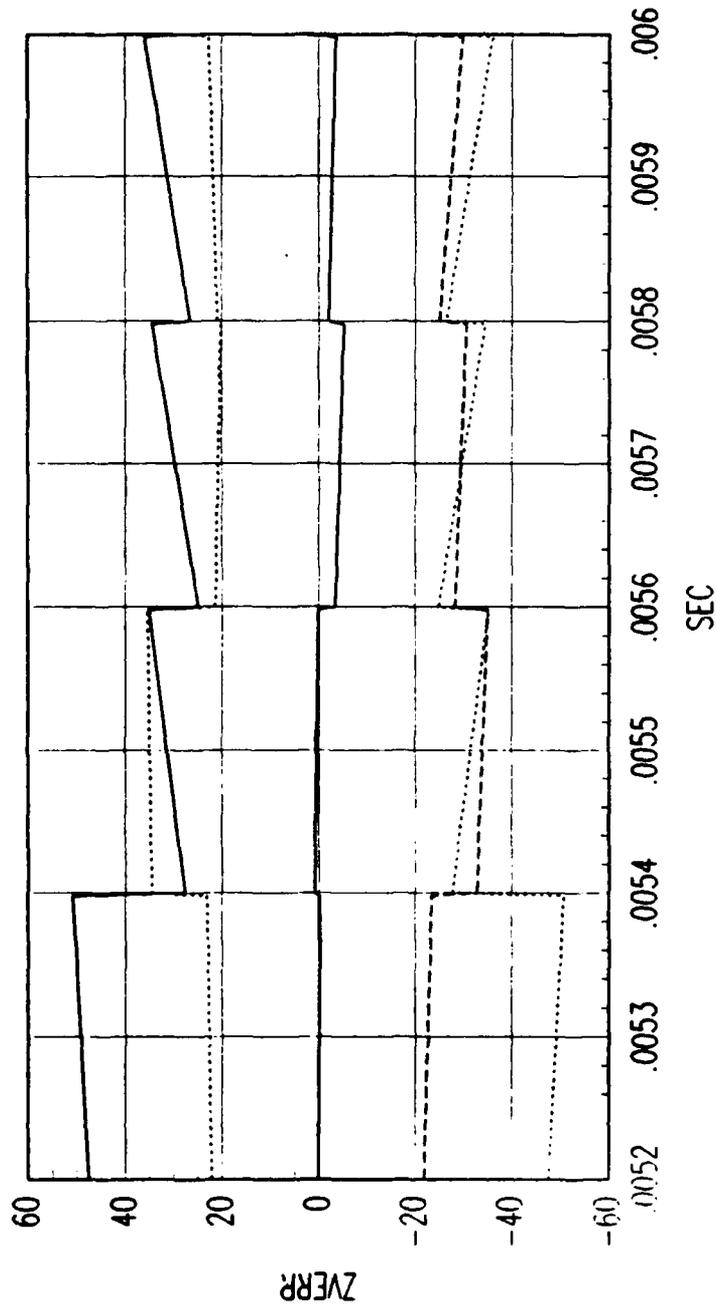
FIG. B FORWARD TIME EKF POSITION ERROR

Figure C.319. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER POSITION ERROR

Figure C.320. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B FORWARD TIME EKF VELOCITY ERROR

Figure C.321. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

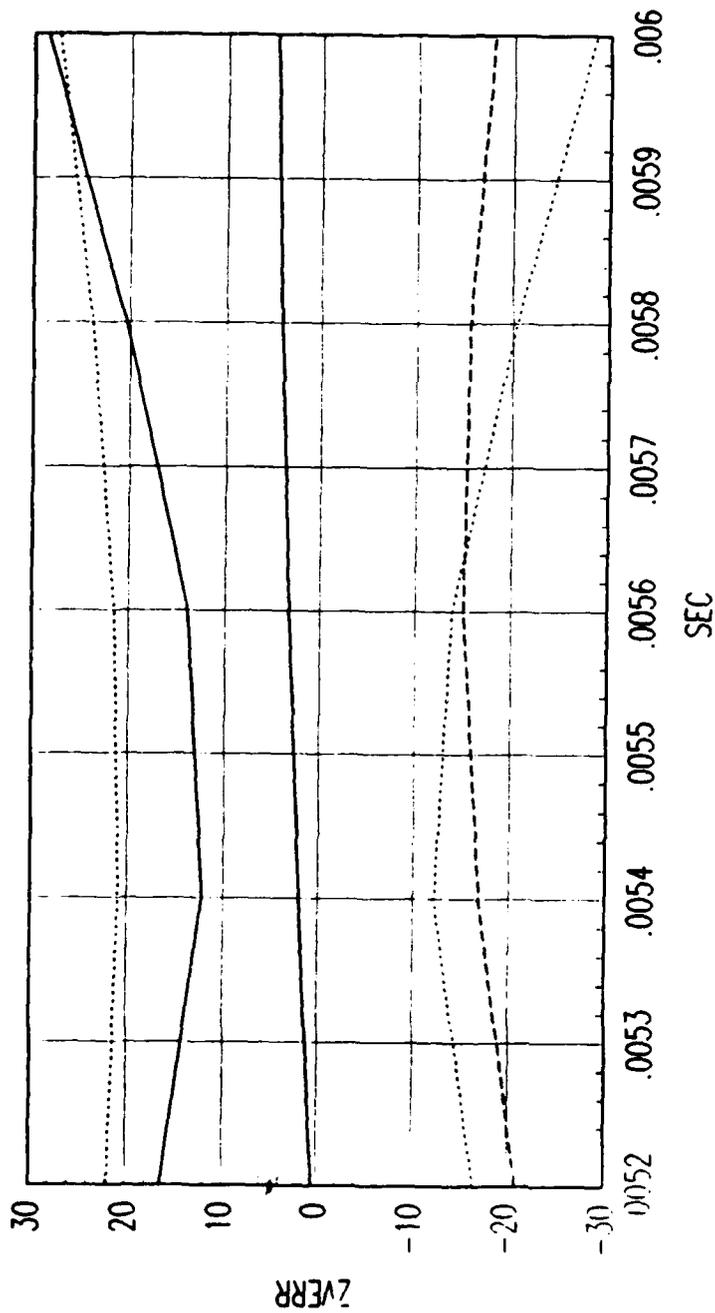


FIG. B REVERSE TIME SMOOTHER VELOCITY ERROR

Figure C.322. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

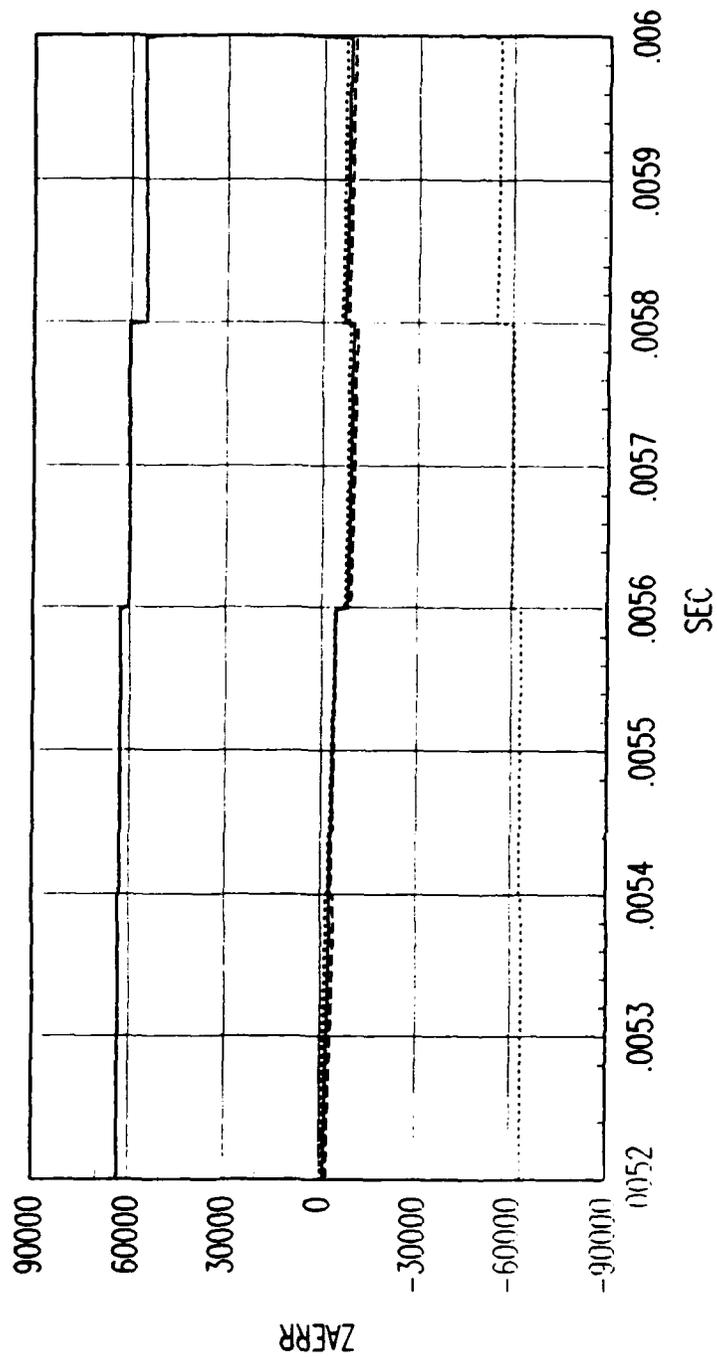
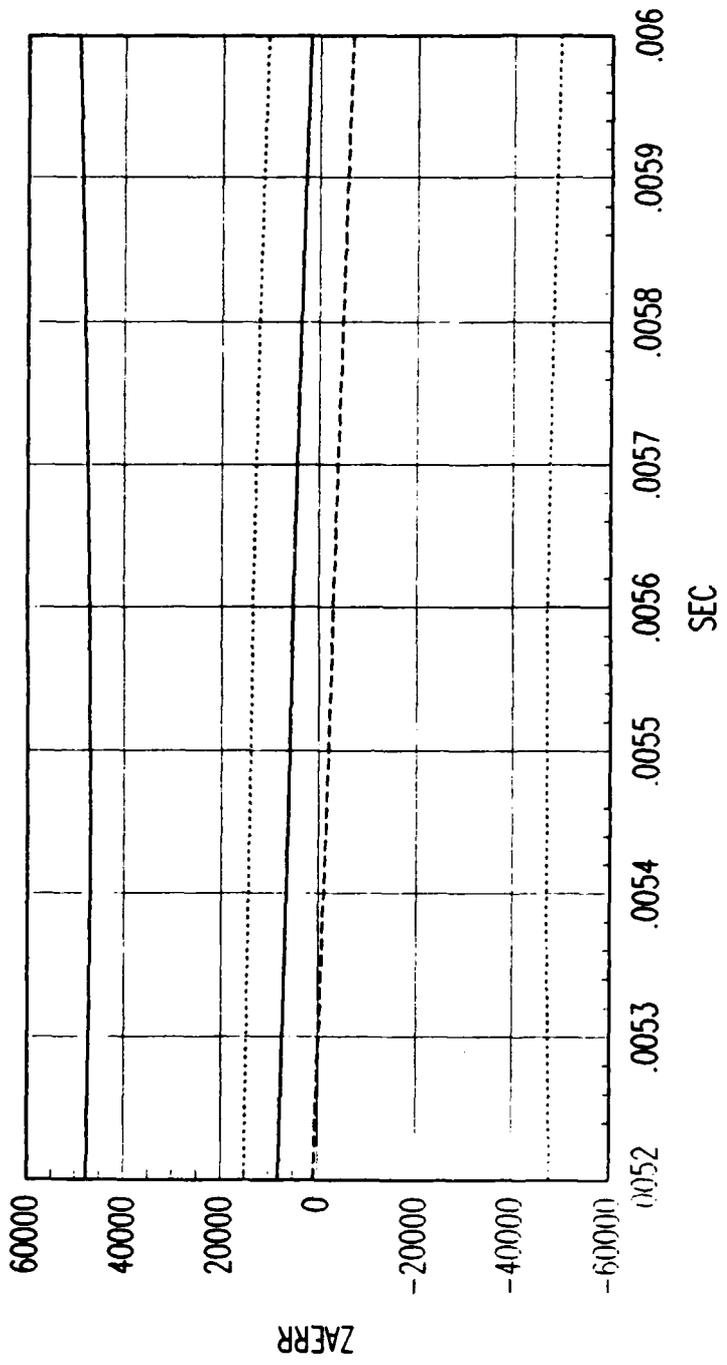


FIG. B FORWARD TIME EKF ACCELERATION ERROR

Figure C.323. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.



FRG. B REVERSE TIME SMOOTHER ACCELERATION ERROR

Figure C.324. Tracking Error Plot, Category 3, $v(t_0) = 10000$ ft/sec.

Bibliography

1. Aikens, Richard S., and Patrick M. Epperson. Techniques for Operating Charge Coupled Devices (CCD's) in Very High Speed Framing Mode, 19 March 1985. Contract N00014-83-K-0268. Tucson AZ: Department of Chemistry, University of Arizona (AD-A152 878).
2. Allan, Roger. "ISSCC: Special-purpose chips," Electronic Design, 34: 119-123 (20 February 1986).
3. Anderton, H., and R. R. Beyer. "Dynamic Imaging with Television Cameras," Advances in Electronics and Electron Physics: Photo-Electronic Image Devices, Vol 28A, 229- 236, edited by J. D. McGee et al. New York: Academic Press, 1969.
4. Barbe, David F. "Image Devices Using the Charge-Coupled Concept," Charge-Coupled Devices: Technology and Applications, 130-159, edited by Roger Melen and Dennis Buss. New York: IEEE Press, 1977.
5. Bar-Shalom, Yaakov, and Edison Tse. "Tracking in a Cluttered Environment with Probabilistic Data Association," Automatica, Vol 11, 451-460, 1975. Elmsford, New York: Pergamon Press.
6. Bluzer, N. et al. Silicon Peristaltic CCD Array Development, 20 November 1986. Contract F33615-81-C-1425. Baltimore MD: Westinghouse Defense and Electronic Systems Center (AD-B110 440L).
7. Brehm, W. F., and J. L. Buckley. Design Study of a Laser Radar System for Spaceflight Application, December 1979. Contract F19628-78-C-0204. Philadelphia PA: General Electric Space Division (AD-A082 332).
8. Brogan, William L. Modern Control Theory, 276-277. Englewood Cliffs, New Jersey, 1985.
9. Bunnik, N. J. J., et al. Development of an Airborne CCD Scanner for Land and Sea Applications, 11 September 1984. Amsterdam, The Netherlands: National Aerospace Laboratory NLR (AD-B100 207).
10. Burke, Hubert K., and Gerald J. Michon. "Charge-Injection Imaging: Operating Techniques and Performances Characteristics," Charge-Coupled Devices: Technology and Applications, 186-195, edited by Roger Melen and Dennis Buss. New York: IEEE Press, 1977.

11. Chang, Kuo-Chu, et al. "Joint Probabilistic Data Association in Distributed Sensor Networks," IEEE Transactions on Automatic Control, Vol AC-31, No 10, 889-897, October 1986. New York: Institute of Electrical and Electronic Engineers, Inc.
12. Corbin, R. W. A Survey of Shuttering Techniques for the RAE High-speed Television Recording System, 15 November 1977. London, England: Royal Aircraft Establishment (AD-A060 105).
13. D'Azzo, John J. and Constantine H. Houppis. Linear Control System Analysis and Design, 136-138. New York: McGraw-Hill Book Company, 1981.
14. Department of the Air Force. AD Technical Facilities: Vol 1 - Range Instrumentation Systems and Technical Support Facilities. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, January 1985 (AD-B097 027L).
15. Department of the Air Force. AD Technical Facilities: Vol II - Land Test Areas. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, June 1984 (AD-B088 184L).
16. Department of the Air Force. AIM-9L Debris Model Arena Test. AD-TR-86-75. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, November 1986 (AD-B107 902).
17. Department of the Air Force. Gator Mine System Test. AD-TR-80-15. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, March 1980 (AD-B045 560).
18. Department of the Air Force. Generic Antiaircraft Missile Warhead Test. AD-TR-86-68. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, September 1986.
19. Department of the Air Force. HAVE BLOCK Sympathetic Detonation Suppression Tests. AD-TR-87-41. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, August 1987.
20. Department of the Air Force. Limited Safety Tests of CBU-89/B Gator Mine System. AD-TR-86-25. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, June 1986.

21. Department of the Air Force. Vulnerability and Lethality Testing System (VALTS). ADTC-TR-72-127. Eglin AFB FL: 3246 Test Wing, Armament Division, Air Force Systems Command, December 1972 (AD-909 145).
22. Eshbach, Ovid W., and Mott Souders. Handbook of Engineering Fundamentals. p742 New York: John Wiley and Sons, 1975.
23. Farina, A., and F.A. Studer. Radar Data Processing, Vol 1-Introduction and Tracking, 222-228, Letchworth, Herts, England: Research Studies Press Ltd, 1985.
24. Fortmann, Thomas E., Yaakov Bar-Shalom, and Molly Scheffe. "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," IEEE Journal of Oceanic Engineering, Vol OE-8, No 3, 173-184, July 1983. New York: Institute of Electrical and Electronics Engineers, Inc.
25. Fu, King-Sun, et al. Robotics: Control, Sensing, Vision, and Intelligence. New York: McGraw-Hill Inc, 1987.
26. Goetze, G. W. and A. H. Boerio. "SEC Camera-tube Performance Characteristics and Applications," Advances in Electronics and Electron Physics: Photo-Electronic Image Devices, Vol 28A, 159-171, edited by J. D. McGee et al. New York: Academic Press, 1969.
27. Groves, P. R., and R. J. Corps. "Applications of the Image Isocon Tube," Advances in Electronics and Electron Physics: Photo-Electronic Image Devices, Vol 28B, 827-836, edited by J. D. McGee et al. New York: Academic Press, 1969.
28. Harrison, David C., and Barry E. Burke. Large Area Focal Plane Comprising Charge-Coupled Devices and Fiber Optics, September 1987. Lexington, MA: Lincon Laboratory, Massachusetts Institute of Technology (AD-A187 549).
29. Kim, Lt Young Soo. Analysis of Contrast Sensitivity and Spectral Response of Imaging Systems. MS Thesis, Naval Postgraduate School, Monterey CA, December 1985 (AD-A164 228).
30. Kosonocky, Walter F., and Donald J. Sauer. "The ABCs of CCDs," Charge-Coupled Devices: Technology and Applications, 2-7, edited by Roger Melen and Dennis Buss. New York: IEEE Press, 1977.

31. Kosonocky, Walter F., et al. 160 x 244 Element Ptsi Schottky Barrier IR-CCD Image Sensor, September 1985. Contract F19628-81-C-0108. Princeton NJ: David Sarnoff Research Center, RCA Laboratories (AD-B096 504).
32. MATRIX-X Reference Manual. Integrated Systems Inc. Palo Alto, California, 1986.
33. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Vol 2, 5-15, 39-45. New York: Academic Press, 1982.
34. Miller, Charlie E. "Some Modern Application Techniques of Stroboscopic High-speed Photography," Proceedings of the Ninth International Congress on High-Speed Photography, 175- 181, edited by William G. Hyzer and William G. Chace. New York: Society of Motion Picture and Television Engineers, August 1970.
35. Purvis, 2LT Jay T. Tracking Multiple Ballistic Targets with Infrared and Radar Measurements. MS Thesis, AFIT/GE/ENG/86D-57. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 1986.
36. Polder, L. D. "Beam-discharge Lag in a Television Pick-up Tube," Advances in Electronics and Electron Physics: Photo-Electronic Image Devices, Vol 28A, 237-245, edited by J. D. McGee et al. New York: Academic Press, 1969.
37. Reid, Donald B. "An Algorithm for Tracking Multiple Targets," IEEE Transactions on Automatic Control, Vol AC-24, No 6, 843-854, December 1979. New York: Institute of Electrical and Electronic Engineers, Inc.
38. Séquin, Carlo H., et al. "All-Solid-State Camera for the 525-Line Television Format," Charge-Coupled Devices: Technology and Applications, 179-185, edited by Roger Melen and Dennis Buss. New York: IEEE Press, 1977.
39. Singer, R. "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, Vol AES-6, 473-483, July 1970. New York: Institute of Electrical and Electronics Engineers, Inc.
40. Skolnik, Merrill I. Introduction to Radar Systems, New York: McGraw-Hill Inc, 1980.

41. Solop, N. A., and O. V. Gofayzen. "Conductivity of a Target and Resolution of a Vidicon," Motion Picture and Television Engineering (Selected Articles), FTD-HT-23-1348-72. Translated by Robert D. Hill from Tekhnika Kino i Televideniya, 12: 50-51. Wright-Patterson AFB OH: Foreign Technology Division, Air Force Systems Command, 1971 (AD-752 874).
42. Trenholme, John B., and John L. Emmett. "Xenon Flashlamp Model for Performance Prediction," Proceedings of the Ninth International Congress on High-Speed Photography, 299-302, edited by William G. Hyzer and William G. Chace. New York: Society of Motion Picture and Television Engineers, August 1970.

Vita

Captain Ronald J. Beyers was born on [REDACTED]
[REDACTED] He graduated from high school [REDACTED]
[REDACTED] in June 1977 and enlisted in the
United States Air Force the same year. In 1980, he received
an appointment to the United States Air Force Academy in
Colorado Springs, Colorado, from which he received a regular
commission and the degree of Bachelor of Science in
Electrical Engineering in May 1984. Upon graduation, he was
assigned to the Air Defense Weapons Center, Tyndall AFB,
Florida, where he served as the chief of telemetry support
for the Tactical Air Command's Weapons Systems Evaluation
Program until entering the School of Engineering, Air Force
Institute of Technology, in June 1987.

[REDACTED]

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS			
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited			
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/88D-2		5. MONITORING ORGANIZATION REPORT NUMBER(S)			
6a. NAME OF PERFORMING ORGANIZATION School of Engineering	6b. OFFICE SYMBOL (if applicable) AFIT/ENA	7a. NAME OF MONITORING ORGANIZATION			
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433		7b. ADDRESS (City, State, and ZIP Code)			
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 3246 Test Wing	8b. OFFICE SYMBOL (if applicable) TZW	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
8c. ADDRESS (City, State, and ZIP Code) Eglin AFB, FL 32548		10. SOURCE OF FUNDING NUMBERS			
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) JOINT PROBABILITY DATA ASSOCIATION ON TRACKING MULTIPLE MUNITIONS FRAGMENTS					UNCLASSIFIED
12. PERSONAL AUTHOR(S) Ronald J. Beyers, Capt USAF					
13a. TYPE OF REPORT MS Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 1988 December		15. PAGE COUNT 501	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Kalman Filtering, Stochastic Estimation, JPDA, PDA, Multi-Target Tracking		
FIELD	GROUP	SUB-GROUP			
20	03				
12	03				
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Thesis Chairman: Rob Williams, Capt, USAF Assistant Professor of Electrical Engineering					
<p><i>Approved for release in accordance with AFR 199-1 10 Jun 89</i></p>					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Capt Rob Williams		22b. TELEPHONE (Include Area Code) 513-255-6027		22c. OFFICE SYMBOL AFIT/ENG	

UNCLASSIFIED

The purpose of this study was to research and apply current technology electronic data acquisition and tracking techniques to conventional munitions live-fire testing. Previously applied technologies for munitions testing have proven themselves expensive in materials, labor, and time. The specific scope of this study was to research methods to electronically acquire and track the position, velocity, and acceleration of multiple warhead fragments as they dispersed from the test-blast center. A design specification for a maximum trackable fragment speed was set at 10,000 ft/sec. The theoretical application of xenon strobe illuminated (2.0 microsecond flash duration), orthogonally oriented Charge-Coupled Device (CCD) cameras (in sets of two) provides three dimensional image measurements at a 2.0 microsecond exposure, 5000 frame/sec rate. The acquired and assumed noisy fragment position measurements (recorded digitally) are post-mission processed through an Extended Kalman Filter (EKF) based Joint Probability Data Association (JPDA) multiple target/tracker state estimator followed by a backward time Rauch-Tung-Striebel (RTS) fixed-interval optimal smoother. Strong emphasis was placed on Monte-Carlo computer simulation testing of this EKF/JPDA/RTS tracker-smoother algorithm. Representative trajectories of straight, curving, and crossing spherical fragments at 3000, 6000 and 10000 ft/sec were modeled and tracked with promising accuracies in position and velocity. The presented fragment data acquisition system was deemed realizable and practical with existing technologies, although the CCD camera 5000 frame/sec requirement was found difficult to obtain reliably. The initial proposed system hardware cost will be high; however, critical system components (such as cameras) survive the test blast and are continuously reusable to keep overall long-term costs down. In addition, the entire data reduction process is reduced from days or weeks to several hours (overnight) on an autonomous EKF/JPDA/RTS computer program.

UNCLASSIFIED